

One Identity Safeguard for Privileged Sessions 6.0.13

Administration Guide

Copyright 2022 One Identity LLC.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of One Identity LLC .

The information in this document is provided in connection with One Identity products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. One Identity does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (http://www.OneIdentity.com) for regional and international office information.

Patents

One Identity is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at http://www.OneIdentity.com/legal/patents.aspx.

Trademarks

One Identity and the One Identity logo are trademarks and registered trademarks of One Identity LLC. in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit our website at www.OneIdentity.com/legal. All other trademarks are the property of their respective owners.

Legend

WARNING: A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

SPS Administration Guide
Updated - 11 July 2022, 08:40
Version - 6.0.13

Contents

Preface **18**

Summary of changes 18

Introduction **26**

The major benefits of One Identity Safeguard for Privileged Sessions (SPS) 26

Application areas 27

The concepts of One Identity Safeguard for Privileged Sessions (SPS) **29**

The philosophy of One Identity Safeguard for Privileged Sessions (SPS) 30

Policies 31

Credential Stores 34

Plugin framework 35

Indexing 37

Supported protocols and client applications 38

- HTTP 38
- Secure Shell Protocol 39
- Remote Desktop Protocol 39
- ICA 40
- Telnet 40
- Remote Desktop Gateway Server Protocol (RDGSP) 41
- Virtual Network Computing 41
- VMware Horizon View 41

Modes of operation 41

- Transparent mode 42
- Single-interface transparent mode 42
- Non-transparent mode 44
- Inband destination selection 45

Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) 46

- Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using SSH 46
- Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using RDP 49
- Connecting to a server through One Identity Safeguard for Privileged Sessions 52

(SPS) using an RD Gateway
Archive and backup concepts 53
Configuration export 54
System backup 55
Connection backup 56
Connection archive 57
Support bundle 58
Debug logs 59
Connection logs 59
Core dump files 60
Maximizing the scope of auditing 60
IPv6 in One Identity Safeguard for Privileged Sessions (SPS) 63
SSH hostkeys 63
Authenticating clients using public-key authentication in SSH 64
The gateway authentication process 64
Four-eyes authorization 65
Network interfaces 66
High Availability support in One Identity Safeguard for Privileged Sessions (SPS) 67
Firmware and high availability 67
Versions and releases of One Identity Safeguard for Privileged Sessions (SPS) 68
Accessing and configuring One Identity Safeguard for Privileged Sessions (SPS) 68
Cloud deployment considerations 70
Before you start: platforms and resources 70
AWS deployment 70
Azure deployment 72
Limitations 74
Prerequisites 75
High Availability and redundancy in Microsoft Azure 76
Redundancy 76
High Availability 76
The Welcome Wizard and the first login 77
The initial connection to One Identity Safeguard for Privileged Sessions (SPS) 77
Creating an alias IP address (Microsoft Windows) 78
Creating an alias IP address (Linux) 84

Modifying the IP address of One Identity Safeguard for Privileged Sessions (SPS) ... 85
Accessing the Welcome Wizard from a non-standard interface ... 85
Configuring One Identity Safeguard for Privileged Sessions (SPS) with the Welcome Wizard ... 86
Logging in to One Identity Safeguard for Privileged Sessions (SPS) and configuring the first connection ... 97
Basic settings ... 105
Supported web browsers and operating systems ... 105
The structure of the web interface ... 106
Elements of the main workspace ... 109
Multiple users and locking ... 111
Web interface timeout ... 112
Preferences ... 112
Network settings ... 114
Configuring user and administrator login addresses ... 118
Managing logical interfaces ... 120
Routing uncontrolled traffic between logical interfaces ... 123
Configuring the routing table ... 123
Configuring date and time ... 124
System logging, SNMP and e-mail alerts ... 126
Configuring system logging ... 126
Configuring e-mail alerts ... 129
Configuring SNMP alerts ... 131
Querying SPS status information using agents ... 133
Customize system logging in One Identity Safeguard for Privileged Sessions (SPS) 134
Configuring system monitoring on SPS ... 137
Configuring monitoring ... 138
Health monitoring ... 138
Preventing disk space fill-up ... 139
System related traps ... 140
Traffic related traps ... 143
Data and configuration backups ... 146
Creating a backup policy using Rsync over SSH ... 147
Creating a backup policy using SMB/CIFS ... 150
Creating a backup policy using NFS ... 154

Creating configuration backups ... 157
Creating data backups ... 158
Encrypting configuration backups with GPG ... 158
Archiving and cleanup ... 159
Creating a cleanup policy ... 160
Creating an archive policy using SMB/CIFS ... 161
Creating an archive policy using NFS ... 166
Archiving or cleaning up the collected data ... 169
Forwarding data to third-party systems ... 170
Using the Splunk forwarder ... 171
Using the universal SIEM forwarder ... 173
Message types forwarded to SIEMs ... 175
Message format forwarded to SIEMs ... 177
Joining to One Identity Starling ... 300
Joining SPS to One Identity Starling with Credential String ... 301
Unjoining SPS from One Identity Starling ... 302
User management and access control ... 303
Managing One Identity Safeguard for Privileged Sessions (SPS) users locally ... 303
Creating local users in One Identity Safeguard for Privileged Sessions (SPS) ... 304
Deleting a local user from One Identity Safeguard for Privileged Sessions (SPS) ... 305
Setting password policies for local users ... 306
Managing local usergroups ... 308
Managing One Identity Safeguard for Privileged Sessions (SPS) users from an LDAP database ... 310
Authenticating users to a RADIUS server ... 316
Authenticating users with X.509 certificates ... 318
Managing user rights and usergroups ... 320
Assigning privileges to usergroups for the One Identity Safeguard for Privileged Sessions (SPS) web interface ... 323
Modifying group privileges ... 324
Finding specific usergroups ... 325
Using usergroups ... 327
Built-in usergroups of One Identity Safeguard for Privileged Sessions (SPS) ... 328
Listing and searching configuration changes ... 332
Using the internal search interface ... 334

Filtering ... 336
Exporting the results ... 336
Customizing columns of the internal search interface ... 336
Displaying the privileges of users and user groups ... 338
Managing One Identity Safeguard for Privileged Sessions (SPS) ... 342
Controlling One Identity Safeguard for Privileged Sessions (SPS): reboot, shutdown 343
Disabling controlled traffic ... 344
Disabling controlled traffic permanently ... 345
Managing Safeguard for Privileged Sessions (SPS) clusters ... 346
Cluster roles ... 346
Enabling cluster management ... 348
Building a cluster ... 349
Assigning roles to nodes in your cluster ... 353
Configuration synchronization across nodes in a cluster ... 355
Configuration synchronization and SSH keys ... 356
Using a configuration synchronization plugin ... 356
Monitoring the status of nodes in your cluster ... 360
Updating the IP address of a node in a cluster ... 361
Managing a cluster with configuration synchronization without central search ... 363
Managing a cluster with central search configuration and configuration synchronization ... 366
Managing a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster ... 368
HA cluster configuration and management options ... 372
Adjusting the synchronization speed ... 373
Redundant heartbeat interfaces ... 374
Next-hop router monitoring ... 376
Upgrading One Identity Safeguard for Privileged Sessions (SPS) ... 377
Upgrade checklist ... 378
Upgrading One Identity Safeguard for Privileged Sessions (SPS) (single node) ... 380
Upgrading a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster ... 381
Troubleshooting ... 384
Exporting the configuration of One Identity Safeguard for Privileged Sessions (SPS) ... 384

Importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) ... 386
Managing the One Identity Safeguard for Privileged Sessions (SPS) license ... 387
Updating the SPS license ... 388
Accessing the One Identity Safeguard for Privileged Sessions (SPS) console ... 389
Using the console menu of One Identity Safeguard for Privileged Sessions (SPS) ... 389
Enabling SSH access to the One Identity Safeguard for Privileged Sessions (SPS) host ... 391
Changing the root password of One Identity Safeguard for Privileged Sessions (SPS) ... 394
Firmware update using SSH ... 395
Exporting and importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) using the console ... 396
Sealed mode ... 397
Disabling sealed mode ... 398
Out-of-band management of One Identity Safeguard for Privileged Sessions (SPS) ... 398
Configuring the IPMI from the console ... 399
Configuring the IPMI from the BIOS ... 402
Managing the certificates used on One Identity Safeguard for Privileged Sessions (SPS) ... 407
Generating certificates for One Identity Safeguard for Privileged Sessions (SPS) ... 409
Uploading external certificates to One Identity Safeguard for Privileged Sessions (SPS) ... 410
Generating TSA certificate with Windows Certificate Authority on Windows Server 2008 ... 414
Generating TSA certificate with Windows Certificate Authority on Windows Server 2012 ... 419

General connection settings ... 434
Configuring connections ... 434
Modifying the destination address ... 441
Configuring inband destination selection ... 443
Modifying the source address ... 446
Creating and editing channel policies ... 448
Real-time content monitoring with Content Policies ... 451
Creating a new content policy ... 452
Configuring time policies ... 457
Creating and editing user lists ... 457

Authenticating users to an LDAP server ... 459
Audit policies ... 465
Encrypting audit trails ... 465
Timestamping audit trails with built-in timestamping service ... 469
Timestamping audit trails with external timestamping service ... 472
Digitally signing audit trails ... 474
Verifying certificates with Certificate Authorities ... 476
Signing certificates on-the-fly ... 478
Creating an external Signing CA ... 481
Creating a Local User Database ... 486
Configuring cleanup for the One Identity Safeguard for Privileged Sessions (SPS) connection database ... 488
HTTP-specific settings ... 490
Limitations in handling HTTP connections ... 491
Authentication in HTTP and HTTPS ... 491
Creating a new HTTP authentication policy ... 492
Setting up HTTP connections ... 493
Setting up a transparent HTTP connection ... 493
Enabling One Identity Safeguard for Privileged Sessions (SPS) to act as a HTTP proxy ... 495
Enabling TLS encryption in HTTP ... 498
Configuring half-sided SSL encryption in HTTP ... 500
Session-handling in HTTP ... 501
Creating and editing protocol-level HTTP settings ... 502
ICA-specific settings ... 506
Setting up ICA connections ... 506
Supported ICA channel types ... 507
Creating and editing protocol-level ICA settings ... 509
One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a Citrix environment ... 510
Troubleshooting Citrix-related problems ... 514
RDP-specific settings ... 515
Supported RDP channel types ... 516
Creating and editing protocol-level RDP settings ... 519

Network Level Authentication (NLA) with One Identity Safeguard for Privileged Sessions (SPS) 524
Network Level Authentication (NLA) with domain membership 524
Using One Identity Safeguard for Privileged Sessions (SPS) across multiple domains 526
Network Level Authentication without domain membership 527
Verifying the certificate of the RDP server in encrypted connections 527
Enabling TLS-encryption for RDP connections 529
Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway 531
Configuring Remote Desktop clients for gateway authentication 534
Inband destination selection in RDP connections 538
Usernames in RDP connections 539
Saving login credentials for RDP on Windows 541
Configuring RemoteApps 542
Configuring SPS to enable exporting files from audit trails after RDP file transfer via clipboard 544

SSH-specific settings 545
Setting the SSH host keys of the connection 546
Supported SSH channel types 547
Authentication Policies 552
Creating a new authentication policy 552
Client-side authentication settings 554
Local client-side authentication 556
Relayed authentication methods 557
Configuring your Kerberos environment 558
Kerberos authentication settings 559
Server host keys 560
Automatically adding the host keys of a server to One Identity Safeguard for Privileged Sessions (SPS) 561
Manually adding the host key of a server 561
Creating and editing protocol-level SSH settings 564
Supported encryption algorithms 567

Telnet-specific settings 570
Enabling TLS-encryption for Telnet connections 571
Creating a new Telnet authentication policy 574

Extracting username from Telnet connections ... 577
Creating and editing protocol-level Telnet settings ... 578
Inband destination selection in Telnet connections ... 580
Limitations of using TN5250 protocol with IBM iSeries Access for Windows ... 580

VMware Horizon View connections ... 582

One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a VMware environment ... 582

VNC-specific settings ... 584

Enabling TLS-encryption for VNC connections ... 584
Creating and editing protocol-level VNC settings ... 589

Indexing audit trails ... 592

Configuring the internal indexer ... 593
Configuring external indexers ... 599
Prerequisites and limitations ... 600
Hardware requirements for the external indexer host ... 600
Configuring One Identity Safeguard for Privileged Sessions (SPS) to use external indexers ... 601
Installing the external indexer ... 602
Configuring the external indexer ... 603
Uploading decryption keys to the external indexer ... 605
Configuring a hardware security module (HSM) or smart card to integrate with external indexer ... 606
Setting up and testing the environment ... 607
Encrypting a PKCS#11 PIN ... 608
Starting and restarting the external-indexer service when using a custom password for PKCS#11 PIN encryption ... 609
Configuring SoftHSM ... 609
Configuring AWS CloudHSM ... 611
Configuring a smart card ... 612
Customizing the indexing of HTTP traffic ... 613
Starting the external indexer ... 614
Disabling indexing on One Identity Safeguard for Privileged Sessions (SPS) ... 615
Managing the indexers ... 615
Upgrading the external indexer ... 616
Troubleshooting external indexers ... 617

Monitoring the status of the indexer services ... 617
HTTP indexer configuration format ... 618
HTTP indexer configuration options ... 619

Using the Search interface ... 622
Assigning search privileges ... 629
Specifying time ranges ... 632
Using search filters ... 635
List of available search filters ... 636
Searching in the contents of audit trails ... 694
Displaying statistics on search results ... 702
Analyzing data using One Identity Safeguard for Privileged Analytics ... 704
The search and filter process ... 711
Viewing session details ... 716
Replaying audit trails in your browser ... 721
Replaying encrypted audit trails in your browser ... 724
Creating report subchapters from search queries ... 729
Creating search-based report subchapters from search results ... 729
Creating search-based report subchapters from scratch ... 731
Search interface changes between version 5.0 and 6.0 ... 733

Searching session data on a central node in a cluster ... 738

Advanced authentication and authorization techniques ... 740
Configuring usermapping policies ... 740
Configuring gateway authentication ... 742
Configuring out-of-band gateway authentication ... 744
Performing out-of-band gateway authentication on One Identity Safeguard for Privileged Sessions (SPS) ... 748
Performing inband gateway authentication in SSH and Telnet connections ... 749
Performing inband gateway authentication in RDP connections ... 750
Troubleshooting gateway authentication ... 751
Configuring four-eyes authorization ... 752
Configuring four-eyes authorization ... 752
Performing four-eyes authorization on One Identity Safeguard for Privileged Sessions (SPS) ... 755
Using credential stores for server-side authentication ... 757
Configuring local Credential Stores ... 758

Performing gateway authentication to RDP servers using local Credential Store and NLA ... 761
Configuring password-protected Credential Stores ... 762
Unlocking Credential Stores ... 766
Using a custom Credential Store plugin to authenticate on the target hosts ... 767
Integrating external authentication and authorization systems ... 768
How Authentication and Authorization plugins work ... 769
Using a custom Authentication and Authorization plugin to authenticate on the target hosts ... 771
Performing authentication with AA plugin in terminal connections ... 773
Performing authentication with AA plugin in Remote Desktop connections ... 773
Integrating ticketing systems ... 774
Performing authentication with ticketing integration in terminal connections ... 774
Performing authentication with ticketing integration in Remote Desktop connections ... 775
Creating a custom plugin ... 776
Plugin troubleshooting ... 776
Reports ... 777
Contents of the operational reports ... 778
Configuring custom reports ... 778
Creating reports from audit trail content ... 782
Creating report subchapters from search queries ... 785
Creating search-based report subchapters from search results ... 786
Creating search-based report subchapters from scratch ... 788
Creating statistics from custom database queries ... 790
Database tables available for custom queries ... 794
The alerting table ... 796
The aps table ... 796
The archives table ... 797
The audit_trail_downloads table ... 797
The channels table ... 798
The closed_connection_audit_channels view ... 803
The closed_not_indexed_audit_channels view ... 803
The connection_events view ... 804
The connection_occurrences view ... 805
The connections view ... 805

The events table ... 808
The file_xfer table ... 809
The http_req_resp_pair table ... 809
The indexer_jobs table ... 810
The occurrences table ... 811
The progresses table ... 811
The results table ... 812
The skipped_connections table ... 812
The usermapped_channels view ... 812
Querying trail content with the lucene-search function ... 817
Generating partial reports ... 818
Creating PCI DSS reports ... 820
Contents of PCI DSS reports ... 822

The One Identity Safeguard for Privileged Sessions (SPS) RPC API ... 826
Requirements for using the RPC API ... 826
RPC client requirements ... 827
Locking One Identity Safeguard for Privileged Sessions (SPS) configuration from the RPC API ... 828
Documentation of the RPC API ... 828
Enabling RPC API access to One Identity Safeguard for Privileged Sessions (SPS) ... 829

The One Identity Safeguard for Privileged Sessions (SPS) REST API ... 830

One Identity Safeguard for Privileged Sessions (SPS) scenarios ... 831
Configuring public-key authentication on One Identity Safeguard for Privileged Sessions (SPS) ... 831
Configuring public-key authentication using local keys ... 832
Configuring public-key authentication using an LDAP server and a fixed key ... 833
Configuring public-key authentication using an LDAP server and generated keys ... 834
Organizing connections in non-transparent mode ... 835
Organizing connections based on port numbers ... 835
Organizing connections based on alias IP addresses ... 836
Using inband destination selection in SSH connections ... 837
Using inband destination selection with PuTTY ... 837
Using inband destination selection with OpenSSH ... 839
Using inband selection and nonstandard ports with PuTTY ... 840
Using inband selection and nonstandard ports with OpenSSH ... 841

Using inband destination selection and gateway authentication with PuTTY 842
Using inband destination selection and gateway authentication with OpenSSH 844
SSH usermapping and keymapping in AD with public key 845

Troubleshooting One Identity Safeguard for Privileged Sessions (SPS) 854
Network troubleshooting 854
Gathering data about system problems 856
Viewing logs on One Identity Safeguard for Privileged Sessions (SPS) 856
Changing log verbosity level of One Identity Safeguard for Privileged Sessions (SPS) 858
Collecting logs and system information for error reporting 860
Support hotfixes 861
Status history and statistics 863
Connection statistics 865
Memory 866
Disk 867
CPU 868
Network connections 869
Interface 870
Load average 871
Number of processes 872
Displaying custom connection statistics 872
Troubleshooting a One Identity Safeguard for Privileged Sessions (SPS) cluster 873
Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses 873
Recovering One Identity Safeguard for Privileged Sessions (SPS) if both nodes broke down 876
Recovering from a split brain situation 876
Replacing a HA node in a One Identity Safeguard for Privileged Sessions (SPS) cluster 879
Resolving an IP conflict between cluster nodes 880
Understanding One Identity Safeguard for Privileged Sessions (SPS) RAID status 882
Restoring One Identity Safeguard for Privileged Sessions (SPS) configuration and data 883
VNC is not working with TLS 884
Configuring the IPMI from the BIOS after losing IPMI password 884
Incomplete TSA response received 889
Using UPN usernames in audited SSH connections 890

Using SPS with SPP **891**
Configuring the Passwords-initiated workflow 892
Configuring SPP for Passwords-initiated workflow 893
Configuring the Sessions-initiated workflow 894
Configuring SPP for Sessions-initiated workflow 894
Configuring SPS for Sessions-initiated workflow 895
Joining SPS to SPP 897
Troubleshooting the SPS to SPP join 900
Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join error resolution 900
Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues 903
Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join error resolution 904
Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues 907

Configuring external devices **909**
Configuring advanced routing on Linux 909
Configuring advanced routing on Cisco routers 911
Configuring advanced routing on Sophos UTM (formerly Astaro Security Gateway) firewalls 916

Using SCP with agent-forwarding **919**

Security checklist for configuring One Identity Safeguard for Privileged Sessions (SPS) **921**
Encryption-related settings 921
Connection policies 922
Appliance access 922
Networking considerations 923

Jumplists for in-product help **924**
Basic Settings > Management 924
Basic Settings > Local Services 925
Basic Settings > System 926
<Protocol name> Control > Global Options 926
Overview 927
Common to all backends 928

POSIX LDAP backend 928
Active Directory LDAP backend 930

Appendix: Deprecated features 933
Deprecated: Using the Search (classic) interface 934
Searching audit trails: the One Identity Safeguard for Privileged Sessions (SPS) connection database 934
Connection details 937
Replaying audit trails in your browser in Search (classic) 942
Replaying encrypted audit trails in your browser 946
Using the content search 950
Connection metadata 958
Using and managing search filters 964
The search and filter process 965
Displaying statistics on search results 970

About us 974
Contacting us 974
Technical support resources 974

Glossary 975

1

Preface

Welcome to the One Identity Safeguard for Privileged Sessions 6.0.13 Administrator Guide.

This document describes how to configure and manage the One Identity Safeguard for Privileged Sessions (SPS). Background information for the technology and concepts used by the product is also discussed.

Summary of changes

Version 5 F11 - 6.0

Changes in product:

- Support for the Search (classic) interface is deprecated. One Identity recommends using the Search interface instead. For more information, see Using the Search interface.
- In the Search interface, it is now possible to use the **Alerts** tab to view content policy alerts triggered in the session. For more information, see Viewing session details on page 716.
- LDAP authentication settings have been enhanced and simplified. For more information, see Managing One Identity Safeguard for Privileged Sessions (SPS) users from an LDAP database.
- It is now possible to search in the contents of the audit trails for trails of graphical sessions created and indexed with SPS 6.0. Also SPS does not store data for in-session search and you need to download the audit trail to search in the contents of the trail. For more information, see Searching in the contents of audit trails.

Changes in documentation:

- The documentation of the obsolete Search (classic) interface has been moved to an appendix. For the documentation of the Search interface, see Using the Search interface.

- X.509 host certificates and DSA host keys are not supported in SSH and have been removed from the document.

Version 5 F10 - 5 F11

Changes in product:

- It is now possible to run backup policies more than once a day. For more information, see Creating a backup policy using Rsync over SSH, Creating a backup policy using SMB/CIFS, and Creating a backup policy using NFS.
- You can now export the search results into a comma-separated values (CSV) file from the **Search** page. For more information, see Using the Search interface.
- You can now uniformly set the TLS security settings of HTTP, RDP, Telnet, and VNC connections, including the permitted ciphers and TLS versions on the **<Protocol> Control > Settings** pages.

  To ensure the security of your sessions, SSL encryption is not supported anymore, only TLS 1.0 and later.

  For more information, see Creating and editing protocol-level VNC settings, Creating and editing protocol-level RDP settings, Creating and editing protocol-level Telnet settings, and Creating and editing protocol-level HTTP settings.
- The process of using core and boot firmware options when upgrading a high availability SPS cluster to a newer firmware version has been simplified. For more information, see Upgrading a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster.
- When using X.509 certificates to authenticate on the SPS web interface, SPS can now extract the name of the user from the `UserPrincipalName` field of the certificate. For more information, see Authenticating users with X.509 certificates.

Changes in documentation:

- Added information about Windows-side settings, which cause RDP connection failures. For more information, see RDP-specific settings on page 515.
- Updated document with reference about creating a custom plugin. For more information, see Creating a custom plugin.
- Updated information about verbosity levels 8-10, which contain highly sensitive data and must be handled with caution. For more information, see Changing log verbosity level of One Identity Safeguard for Privileged Sessions (SPS) on page 858.
- To manually reindex audit trails, you must contact our Support Team. This information has been added to the document. For more information, see Indexing audit trails.
- Added a summary of events by message type for the universal SIEM forwarder. For more information, see Message format forwarded to SIEMs.
- You must have the Remote Desktop (RD) Licensing role installed when configuring

RemoteApps. For more information, see Configuring RemoteApps.

- A number of editorial changes have been made.

Version 5 F9 - 5 F10

Changes in product:

- It is now possible to assign users to access sessions only for connections for which they are granted permission. For more information, see Assigning search privileges.
- It is now possible to use an external Signing CA plugin. For more information, see Signing certificates on-the-fly on page 478.
- Session tags allow you to get basic information about the session and its contents at a glance. For more information, see Viewing session details on page 716.
- Multiple administrators can access the SPS web interface simultaneously, but only one of them can modify the configuration. It is now possible for other administrators to continue as read-only. For more information, see Multiple users and locking.
- It is now possible to add additional group-membership attributes using the **Check the user DN in these groups** options. For more information, see Authenticating users to an LDAP server.
- SPS can now distinguish the audited HTTP requests and responses based on the session cookies of web applications. For details, see Creating and editing protocol-level HTTP settings.

Version 5 F8 - 5 F9

Changes in product:

- SPS has been extended with the Splunk forwarder, which allows you to automatically send file-based data to Splunk.

  Use the Splunk forwarder if you need to analyze or make changes to the data before you forward it, or you need to control where the data goes based on its contents. For more information, see Using the Splunk forwarder .
- SPS has been extended with the universal SIEM forwarder, which allows you to automatically send file-based data to Splunk, ArcSight, or other third-party systems, in a format that your SIEM can understand.

  Use the universal SIEM forwarder if you need a less resource-heavy solution. For more information, see Using the universal SIEM forwarder .
- Debug bundle has been renamed to support bundle. For more information, see Support bundle .
- SPS now provides a way to authenticate non-transparent HTTP/HTTPS connections on SPS to local and external backends (LDAP, Microsoft Active Directory, RADIUS). The client must support proxy authentication. For more information, see Creating a new HTTP authentication policy .

Version 5 F7 - 5 F8

Changes in product:

- It is now possible to search for scripted sessions. For more information, see Analyzing data using One Identity Safeguard for Privileged Analytics on page 704 .
- The **Indexing history** section on the **Indexer > Indexer status** page has been removed and it is now possible to search for indexing details. For more information about the indexing search filters that you can use, see List of available search filters on page 636 .
- SPS can now be configured to check out passwords from the built-in or external credential stores, such as One Identity Safeguard for Privileged Passwords, and play them in during a connection using the TN3270 protocol.
- When using a hardware security module (HSM) or smart card to integrate with an external indexer, the chroot is not used anymore, the solutions provided by RedHat/CentOS can be used. Configuring a hardware security module (HSM) or smart card to integrate with external indexer on page 606 has been updated to reflect the simplification of configuration steps.
- The **Basic Settings > Local Services > Required minimum version of encryption protocol** option is removed as of One Identity Safeguard for Privileged Sessions (SPS) version 6.0.13 .

  Regardless of the TLS version you configured previously, SPS will uniformly use TLS version 1.2. This change might have the effect that using old (likely unsupported) browsers, it will not be possible to access the web interface of SPS .
- Command detection and window title detection in content policies have changed and they are case-insensitive as of SPS version 5.8.0. In earlier versions, both used to be case-sensitive. For more information, see Creating a new content policy on page 452 .
- Searching for group memberships is now case insensitive.

Version 5 F6 - 5 F7

Changes in product:

- The **System Monitor** now displays statistics about the amount of logs ingested. For details, see The structure of the web interface on page 106 .
- You can now choose to upload a certificate chain when configuring a remote syslog server to send system log messages to. For details, see Configuring system logging on page 126 .
- When you want to create a backup or archive policy on SPS instances that are nodes in a cluster, you can choose to include the node ID in the path to the relevant directory name to prevent cluster nodes from backing up data to the same location, and so overwriting each other's data. For details, see Data and configuration backups on page 146 and Archiving and cleanup on page 159 .

- It is now possible to promote a node to become the Central Management node of a cluster and to add nodes to a cluster using the web interface of One Identity Safeguard for Privileged Sessions . For details, see Building a cluster on page 349 .
- When you have uploaded a configuration synchronization plugin, it is now possible to enable the plugin through the web interface of One Identity Safeguard for Privileged Sessions . For details, see Using a configuration synchronization plugin on page 356.
- SPS now provides information about the status of configuration synchronization. For details, see Monitoring the status of nodes in your cluster on page 360.
- The script used for exporting and importing the configuration of One Identity Safeguard for Privileged Sessions through the console has been updated. For details, see Exporting and importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) using the console on page 396.
- It is now possible to turn any search query or statistics into a subchapter that can be included in reports. You can define reports about the monitored traffic in a more flexible and easy-to-use way than was possible before. For details, see Creating report subchapters from search queries on page 785 .
- When you have a cluster of nodes set up, you can now search all session data recorded by all nodes in the cluster on a single node. For details, see Searching session data on a central node in a cluster on page 738 .
- The RPC API is deprecated as of version 5 F7 of SPS and will be removed in an upcoming feature release. For detail, see The One Identity Safeguard for Privileged Sessions (SPS) RPC API on page 826 .

Changes in documentation:

- A number of editorial changes have been made.
- Added information about configuration synchronization across cluster nodes and its impact on SSH keys. For details, see Configuration synchronization and SSH keys and Setting the SSH host keys of the connection on page 546 .
- Added a section about how to update the IP address of a Managed Host node in a cluster. For details, see Updating the IP address of a node in a cluster on page 361 .

Version 5 F5 - 5 F6

Changes in product:

- When you have a set of two or more One Identity Safeguard for Privileged Sessions instances in your deployment, you now have the possibility to join them into a cluster, and manage them from one central location. You can monitor their status and update their configuration centrally. For details, see Managing Safeguard for Privileged Sessions (SPS) clusters on page 346 .
- In the Search interface, it is now possible to use the flow view for a quick visualization of the session activities. For details, see Using the Search interface on page 622 .
- It is now possible to specify an accuracy level for Optical Character Recognition (OCR). For details, see Configuring the internal indexer on page 593 .

Version 5 F4 - 5 F5

Changes in product:

- It is now possible to specify the base DN of LDAP subtrees for users and for groups separately. Specifying a sufficiently narrow base for the LDAP subtrees can speed up LDAP operations. For details, see Managing One Identity Safeguard for Privileged Sessions (SPS) users from an LDAP database on page 310 and Authenticating users to an LDAP server on page 459 .
- You now have the option to configure connection policies with near real-time indexing priority, meaning that you can start indexing sessions while they are still ongoing. This requires that you configure your indexers with the appropriate settings and capabilities. For details, see Configuring the internal indexer on page 593 and Configuring the external indexer on page 603 .
- It is now possible to use a hardware security module (HSM) or a smart card to store the decryption keys required for decrypting audit trails when using an external indexer. For details, see Configuring a hardware security module (HSM) or smart card to integrate with external indexer on page 606 .
- In the Search interface, it is now possible to display statistics, analyze data using Privileged Account Analytics, and use the timeline for a quick time range selection. For details, see Using the Search interface on page 622 .
- The documentation of the obsolete Audit Player application has been removed from the document. For the documentation of the Safeguard Desktop Player application, see Safeguard Desktop Player User Guide .

Version 5 F3 - 5 F4

Changes in product:

- Using the Search interface on page 622 has been added to the document.
- Configuring RDP banners has been documented in Creating and editing protocol-level RDP settings on page 519 .

Changes in documentation:

- The steps describing how to recover from a split brain situation have been clarified. For more information, see Recovering from a split brain situation on page 876 .
- The screenshots and descriptions in Status history and statistics on page 863 have been updated.
- The open source licenses that apply to certain components of SPS have been consolidated into Open source licenses .
- The documentation of the obsolete Audit Player application has been moved to an appendix. For the documentation of the Safeguard Desktop Player application, see Safeguard Desktop Player User Guide .
- Uploading decryption keys to the external indexer on page 605 has been updated.

Version 5 F2 - 5 F3

Changes in product:

- SPS 's RESTful API has been enhanced with the following new functionalities:
  - *New content endpoint:* /api/audit/sessions/<session-id>/content. It enables you to search in the contents of individual connections. For details, see "Searching in connection content" in the REST API Reference Guide
  - *Filter events:* The filtering functionality is now added to the api/audit/sessions/<session-id>/events endpoint, too. You can now search in the events of individual connections. For more information, see "Session events" in the REST API Reference Guide.
- In order to better integrate SPS with Privileged Account Analytics, some architectural changes have been introduced. For more information, see REST API Reference Guide .
- Enabling TLS-encryption in an RDP connection policy has been simplified. When the connection is encrypted, SPS has to show a certificate to the peer. You can define the type of certificate to show to the peers. For details, see Enabling TLS-encryption for RDP connections on page 529 .
- You can now configure the required minimum version of the default web listener. The default setting is TLS 1.2. For details, see Configuring user and administrator login addresses on page 118 .
- You can now select the depth of indexing: lightweight and full indexing. Lightweight indexing is now enabled by default, you only have to configure it if you want full indexing. Lightweight indexing is faster than full indexing, and indexes only Command and Window title events. It does not index any other screen content (for example, text that is displayed in a terminal or that appears in an RDP window). For details, see Configuring the internal indexer on page 593 .
- RDP 4 and RDP 5 have been removed from Creating and editing protocol-level RDP settings on page 519 .
- The Audit Player application can now replay audit trails that contain graphical X11 sessions (the contents of the X11 Forward channel of the SSH protocol). For further details, see "Replay X11 sessions" in the Safeguard Desktop Player User Guide .
- Plugin configuration files in support bundle: When creating support bundles for troubleshooting purposes, SPS now includes the configuration files of any plugins installed. For details, see "Collecting logs and system information for error reporting" in the Administration Guide .

Changes in documentation:

- Added description of `scbAMQPError` to Traffic related traps on page 143 .

Version 5 F1 - 5 F2

Changes in product:

- It is now possible to set the Maximum Transmission Unit (MTU) per VLAN interface. For more information, see Network settings on page 114 and Managing logical interfaces on page 120 .
- In addition to displaying upgrade logs and boot messages on the local console, SPS now shows information about the upgrade and reboot processes on the web interface, too. For details, see Controlling One Identity Safeguard for Privileged Sessions (SPS): reboot, shutdown on page 343 and Upgrade checklist on page 378 .
- You can now configure Certificate Revocation Lists (CRLs) to be included in certificates. For further details, see Signing certificates on-the-fly on page 478 .

Changes in documentation:

- Added description of how to configure the IPMI from the BIOS. For details, see Configuring the IPMI from the BIOS on page 402 and Configuring the IPMI from the BIOS after losing IPMI password on page 884 .
- Added section explaining limitation when using the TN5250 protocol with IBM iSeries Access for Windows. For detailed information, see Limitations of using TN5250 protocol with IBM iSeries Access for Windows on page 580 .
- Added explanation of why audit trails could have the **Indexing (queued / all)** status in the **Waiting for processing** section. For more information, see Monitoring the status of the indexer services on page 617 .

Version 5 LTS - 5 F1

Changes in product:

- It is now possible to create customized configuration instances of Credential Store and Authentication and Authorization (AA) plugins if the plugin `.zip` file includes an optional sample configuration file. For more information, see Using a custom Credential Store plugin to authenticate on the target hosts on page 767 and Using a custom Authentication and Authorization plugin to authenticate on the target hosts on page 771 .
- You can now customize the configuration of the syslog-ng application that is running on SPS . For details, see Customize system logging in One Identity Safeguard for Privileged Sessions (SPS) on page 134 .

2

Introduction

This section introduces One Identity Safeguard for Privileged Sessions (SPS) in a non-technical manner, discussing how and why is it useful, and what additional security it offers to an existing IT infrastructure.

The major benefits of One Identity Safeguard for Privileged Sessions (SPS)

Application areas

The major benefits of One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) is part of the One Identity Safeguard solution, which in turn is part of One Identity's Privileged Access Management portfolio. Addressing large enterprise needs, SPS is a privileged session management solution which provides industry-leading access control, session recording and auditing to prevent privileged account misuse and accelerate forensics investigations.

SPS is a quickly deployable enterprise device, completely independent from clients and servers - integrating seamlessly into existing networks. It captures the activity data necessary for user profiling and enables full user session drill down for forensic investigations.

SPS has full control over the SSH, RDP, Telnet, TN3270, TN5250, Citrix ICA, and VNC connections, giving a framework (with solid boundaries) for the work of the administrators. The most notable features of SPS are the following:

Central policy enforcement

SPS acts as a centralized authentication and access-control point in your IT environment which protects against privileged identity theft and malicious insiders. The granular access management helps you to control who can access what and when on your critical IT assets.

Prevention of malicious activities

SPS monitors privileged user sessions in real-time and detects policy violations as they occur. In case of detecting a suspicious user activity (for example entering a destructive command, such as the "rm"), SPS can send you an alert or immediately terminate the connection.

Greater accountability (deterrance)

SPS audits "who did what", for example on your database- or SAP servers. Aware of this, your employees will do their work with a greater sense of responsibility leading to a reduction in human errors. By having an easily interpreted, tamper-proof record in encrypted, timestamped, and digitally signed audit trails, finger-pointing issues can be eliminated.

Faster, cost-effective compliance audits

SPS makes all user activity traceable by recording them in high quality, tamper-proof and easily searchable audit trails. All data is stored in encrypted, timestamped and signed files, preventing any modification or manipulation. The movie-like audit trails ensure that all the necessary information is accessible for ad-hoc analyses or audit reports.

Lower troubleshooting and forensics costs

When something wrong happens, everybody wants to know the real story. Analyzing thousands of text-based logs can be a nightmare and may require the participation of external experts. The ability to easily reconstruct user sessions allows you to shorten investigation time and avoid unexpected cost.

Application areas

Fastest return to value and extremely low TCO

One Identity Safeguard for Privileged Sessions (SPS) is a turnkey network appliance - its implementation and configuration is fast and simple. Compared to competitors, there is no need to purchase and install any additional software (for example, Windows or MS SQL servers) or hardware to have SPS fully functioning. Full implementation typically takes only 3-5 days! No need for long and costly professional services for implementation and customization. After deployment, SPS operates in the background like a black box of an airplane - there is no need for any extra workload to operate it.

Independent, agentless device

Compared to agent-based solutions, there is no need for installing and updating agents on clients or servers, eliminating unnecessary maintenance and potential security issues. As a host independent gateway, SPS can control and monitor access to any type of systems incl. all Windows/UNIX/Linux servers, mainframes, network devices, security devices, web-

based applications or thin client environments, such as VMware Horizon View (formerly known as VMware View), Citrix Virtual Apps (formerly known as Citrix XenApp) or Citrix Virtual Desktops (formerly known as Citrix XenDesktop).

Transparent, "router-like" operation

As a proxy gateway, SPS can operate as a router in the network – invisible to the user and to the server. As a transparent solution, SPS requires minimal changes to the existing network. Also, since it operates on the network level, users can keep using the client applications they are familiar with, and do not have to change their work processes, unlike jump host solutions.

Granular access control

Since SPS has full access to the inspected traffic, security managers can granularly control who can access what and when on the servers. For example, they can selectively permit or deny access to protocol channels: enable terminal sessions in SSH, but disable port-forwarding and file transfers, or enable desktop access for RDP, but disable file sharing. In addition, SPS supports real-time shadowing allowing an authorizer to follow the administrator's session in real-time and terminate his/her connection in case of detecting a policy violation.

Real-time prevention of malicious activities

SPS can monitor transferred content in real time and can send alerts or even block connections if a certain pattern is detected in the traffic. Predefined patterns can be a risky command in a text-oriented protocol or a suspicious application in a graphical connection. This command and application level policy can prevent malicious user activities as they happen instead of just recording or reporting them.

Industry-leading session recording and auditing

SPS is the leading session auditing solution on the market offering Optical Character Recognition (OCR) capabilities to log ALL data about privileged actions in graphical user interfaces as well as text-based protocols. SPS can support and audit file transfers, as well. All data is recorded into searchable movie-like audit trails, making it easy to find relevant information in forensics or troubleshooting situations. In case of any problems (server misconfiguration, database manipulation, unexpected shutdown), the circumstances of the event are readily available in the audit trails, thus the cause of the incident can be easily identified. Auditors can do free-text searches in the content of text-based and graphical sessions. They can search for EVERY events (for example, mouse clicks, pressing Enter) and texts seen by the user.

To protect the sensitive information included in the communication, the two directions of the traffic (client-server and server-client) can be separated and encrypted with different keys, thus sensitive information like passwords are displayed only when necessary.

3

The concepts of One Identity Safeguard for Privileged Sessions (SPS)

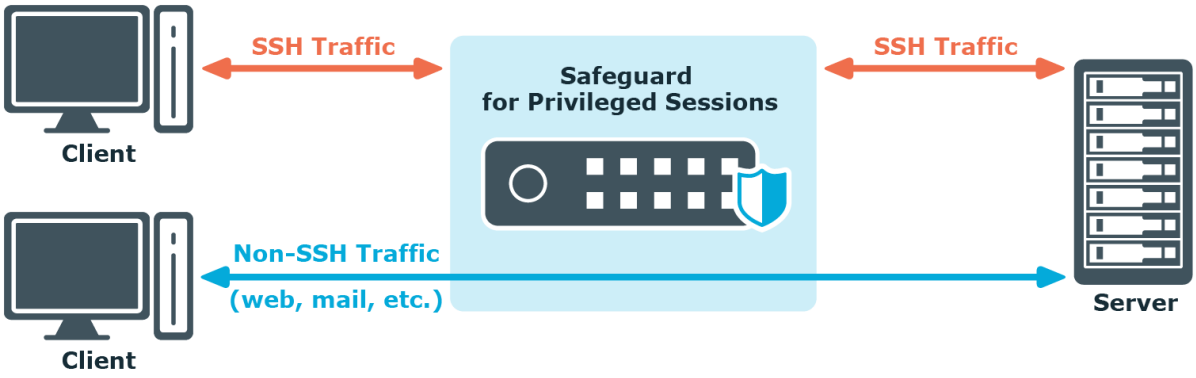
This section discusses the technical concepts of One Identity Safeguard for Privileged Sessions (SPS).

- The philosophy of One Identity Safeguard for Privileged Sessions (SPS)
- Policies
- Credential Stores
- Plugin framework
- Indexing
- Supported protocols and client applications
- Modes of operation
- Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS)
- Archive and backup concepts
- Maximizing the scope of auditing
- IPv6 in One Identity Safeguard for Privileged Sessions (SPS)
- SSH hostkeys
- Authenticating clients using public-key authentication in SSH
- The gateway authentication process
- Four-eyes authorization
- Network interfaces
- High Availability support in One Identity Safeguard for Privileged Sessions (SPS)
- Versions and releases of One Identity Safeguard for Privileged Sessions (SPS)
- Accessing and configuring One Identity Safeguard for Privileged Sessions (SPS)

The philosophy of One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) is a device that examines network traffic at the application level, that is, Layer 7 or the application layer of the OSI model. All communication must conform to the standards of the respective protocol. SPS examines Secure Shell (SSH, including forwarded X11 traffic), Secure Copy (SCP), SSH File Transfer Protocol (SFTP), Remote Desktop (RDP), HTTP, Independent Computing Architecture (Citrix ICA), Telnet, VMware Horizon View, and VNC connections, ignoring and simply forwarding all other types of traffic. SPS uses man-in-the-middle techniques to decrypt and terminate (when necessary) the inspected connections. It separates the connections into two parts (client — SPS, SPS — server) and inspects all traffic, so that no data can be directly transferred between the server and the client.

Figure 1: Inspecting SSH traffic with SPS



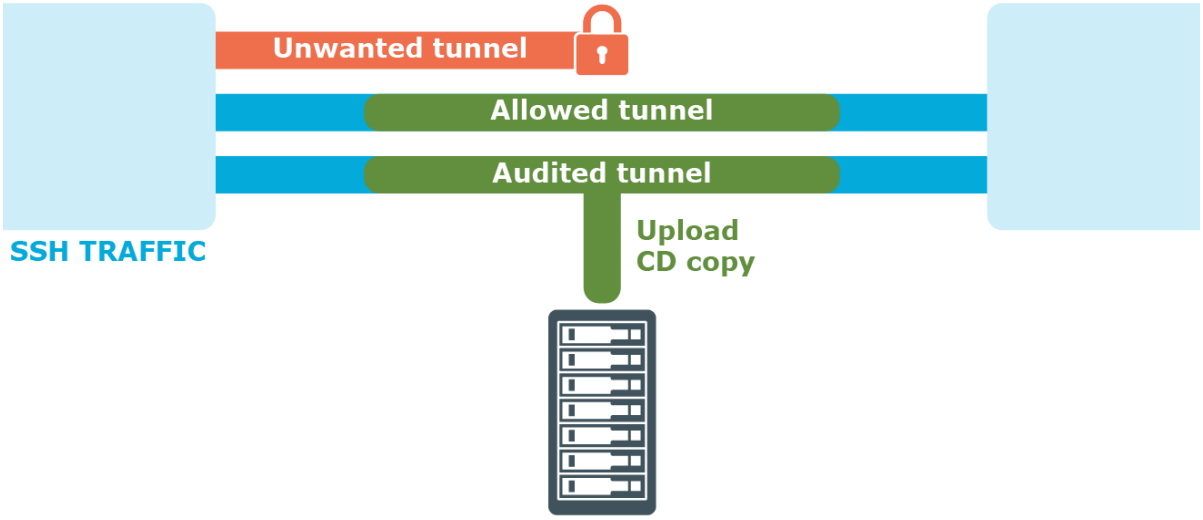
SPS has full control over the initial negotiation phase of the connection, when the client and the server decide the parameters of the encryption to be used in the communication. SPS can restrict the use of the various algorithms, forbidding the use of weak ones — an effective shield against downgrade attacks.

Since SPS isolates the client-server connection into two separate connections, the permitted algorithms can be different on the client and the server side.

SPS controls the connections right from the beginning — including user authentication. That way it is easy to mandate strong authentication for protocols where user information is available (for example, SSH), because SPS can limit the allowed authentication methods and also the users permitted to access the servers.

SPS uses various policies to restrict who, when, and how can access a connection or a specific channel of the protocol. These policies (based on username, authentication method used, and so on) can be applied to connections between particular clients and servers, or also to specific channels of a connection (for example, only to terminal-sessions in SSH, or desktop-sharing in RDP).

Figure 2: Controlling protocol channels

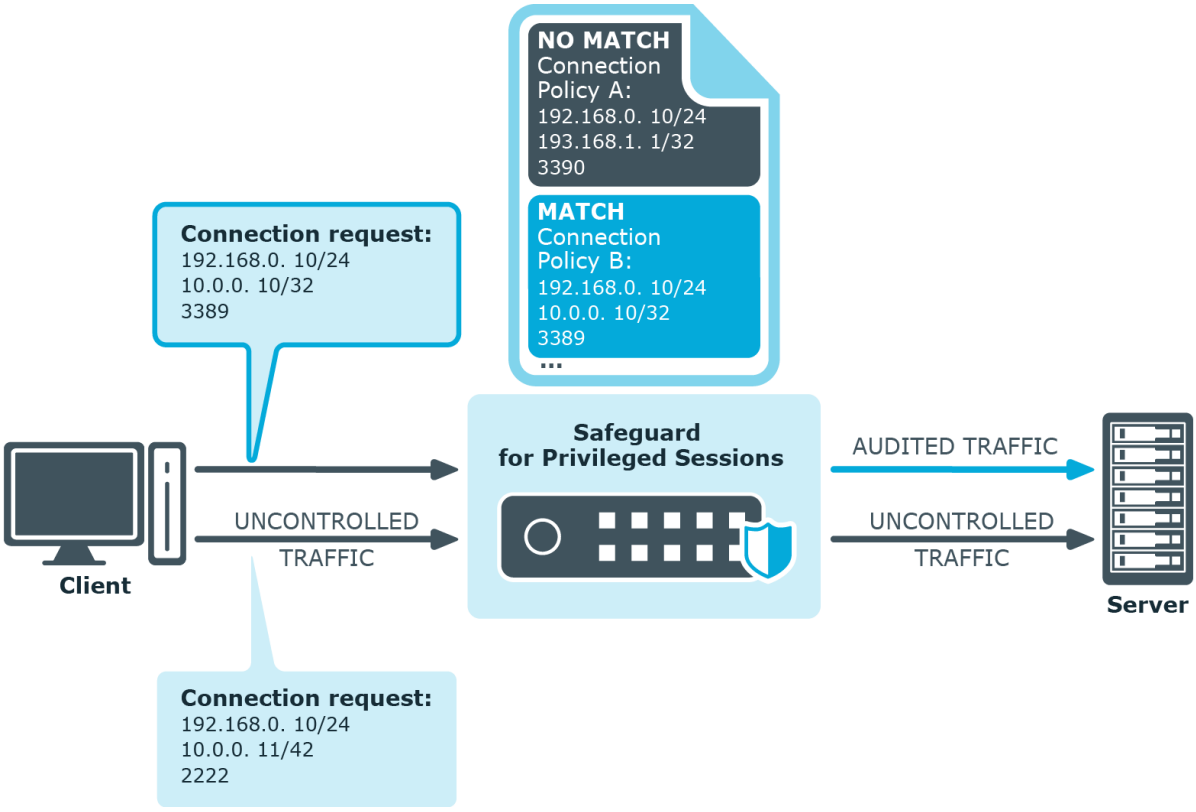


SPS is configured by an administrator or auditor using a web browser.

Policies

One Identity Safeguard for Privileged Sessions (SPS) controls access to connections through a set of policies. Policies let you specify various parameters of a connection, and so define the types of connections that SPS should monitor and restrict access to. When a connection request reaches SPS, SPS compares the connection policies to the parameters of the connection request one-by-one, starting with the first policy in the policy list. The first connection policy completely matching the connection request is applied to the connection.

Figure 3: Processing policies



This section provides a brief definition of each policy type and also explains the hierarchy between them.

A *connection policy* allows you to specify details of the connection between a particular client and server that you want to restrict in any way. In addition to setting basic details (such as the source and destination addresses), or more advanced ones (such as authentication to SPS or the server), the connection policy also references other policies that allow you to define further specifics of the connections you wish to control.

Depending on the protocol, the connection policy may also allow you to configure:

- A Credential Store that allows users auto logon to the target server.

  For information on Credential Stores, see Credential Stores on page 34.
- A plugin that allows integration with external systems, which users can be optionally authenticated to (before authenticating to the target server).

  For information on plugins, see Plugin framework on page 35.

For details on configuring connection policies, see Configuring connections on page 434.

A *channel policy* serves to control channel usage (for example, terminal session and Secure Copy in SSH, or drawing and clipboard in RDP) within a given connection. It lists channels that are allowed within a connection, and it also lets you specify restriction rules based on user lists, user groups, or the IP address of the client or server. You can also reference a content policy and a time policy within the channel policy, and it is also within the channel policy that you enable auditing for a specific channel.

For details on configuring channel policies, see Creating and editing channel policies on page 448.

A *content policy* lets you log an event, send an alert, or terminate a connection if a particular command or text (that you specify in the policy) appears in the command line or on the screen.

For details on creating a content policy, see Creating a new content policy on page 452.

A *time policy* specifies the timeframe when users are permitted to access a particular channel and so restricts the availability of that channel.

For details on configuring time policies, see Configuring time policies on page 457.

An *audit policy* enables you to prevent the manipulation of audit trails files that store the recorded activities of privileged users by providing you with options to encrypt, timestamp, and sign these files.

For details on creating audit policies, see Audit policies on page 465.

An *authentication policy* defines those client-side and server-side authentication methods that can be used in a connection.

For details on creating authentication policies, see Authentication Policies on page 552.

An *LDAP policy* lets you set details of the LDAP server to which you wish to authenticate users of the connections you are controlling.

For details on creating an LDAP policy, see Authenticating users to an LDAP server on page 459.

A *usermapping policy* specifies the usernames that are allowed access to the remote server and the user groups that are allowed to use the specified username.

For details on configuring usermapping policies, see Configuring usermapping policies on page 740.

An *archiving policy* lets you configure details of the archiving process that enables you to archive connection-related data and audit trails. You can configure, for example, the target server where archived files are to be stored, or the directory structure in which to organize your archived files.

For details on creating archiving policies, see Archiving and cleanup on page 159.

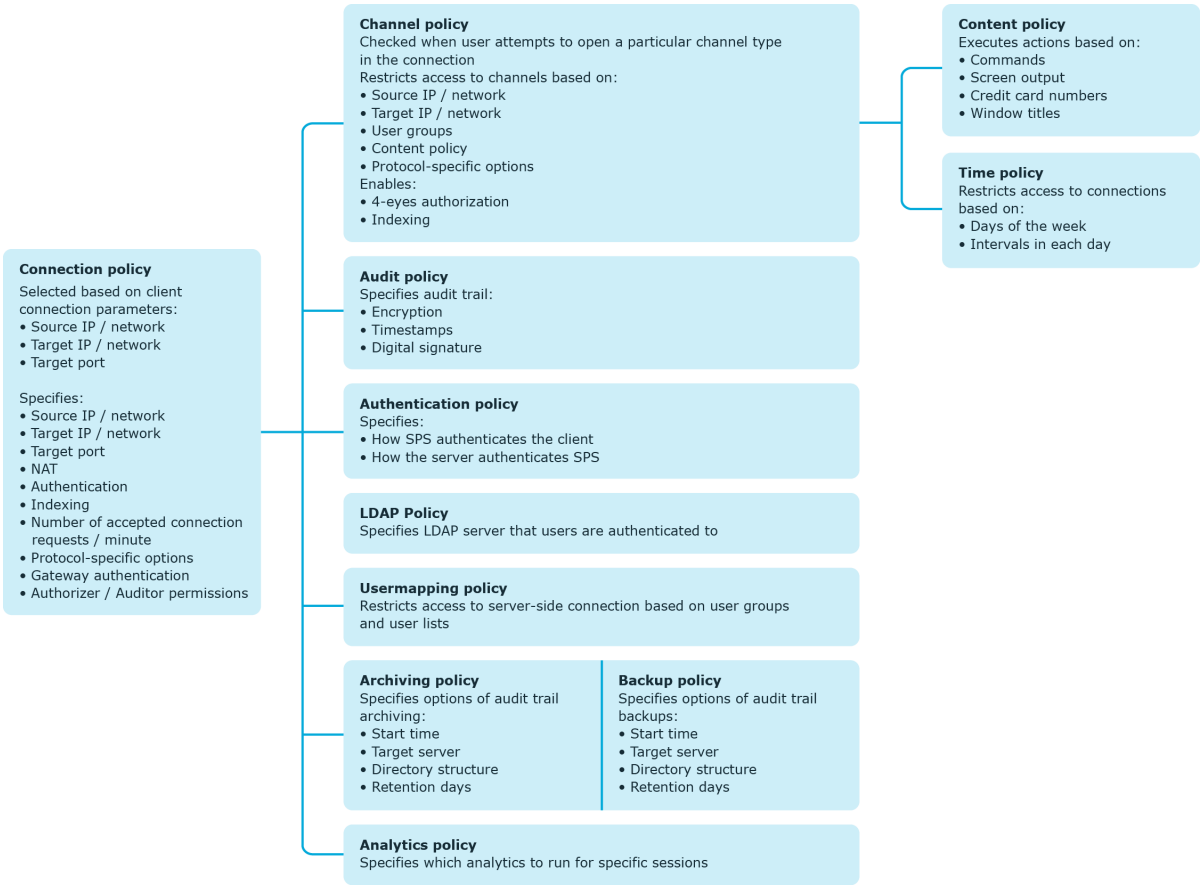
A *backup policy* defines the address of the backup server where you can back up connection data, the protocol to use to access it, details of authenticating to the backup server, and so on.

For details on creating backup policies, see Data and configuration backups on page 146.

An *analytics policy* lets you specify the analytics that you wish to run for specific sessions, and also determine the weight that scores given by the selected analytics should have in the final aggregated score.

For details on configuring analytics policies, see "Configure analytics" in the Safeguard for Privileged Analytics Configuration Guide.

Figure 4: Policies of SPS

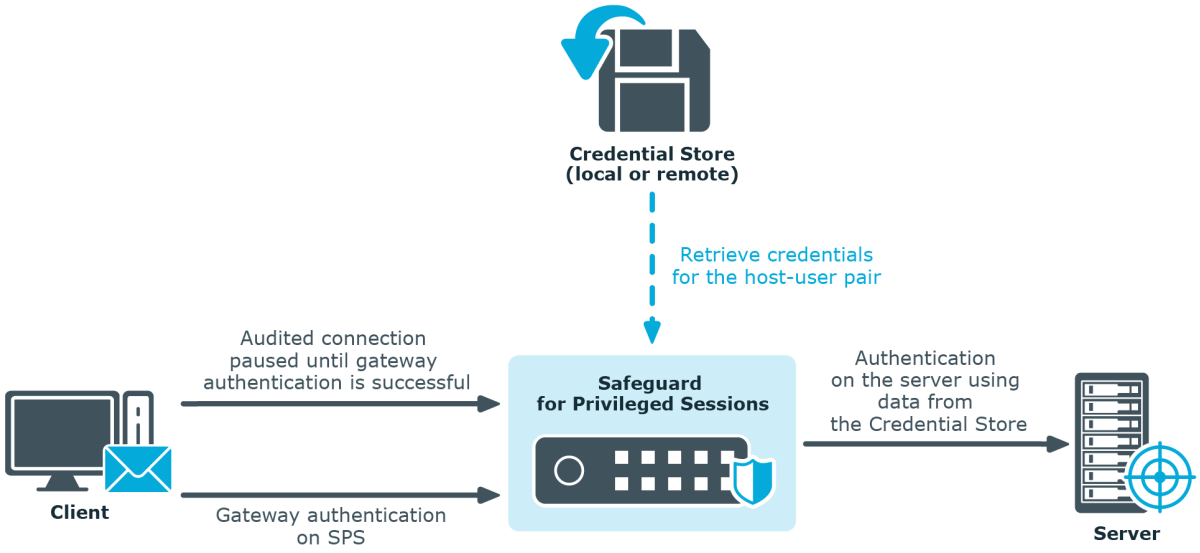


Credential Stores

Credential Stores are repositories of user credentials (for example, passwords, private keys, certificates). They are used for authenticating a user to the target server that the user wishes to access, without the user actually having access to those credentials. Credentials are retrieved transparently from One Identity Safeguard for Privileged Sessions's (SPS's) local Credential Store or an external, third-party password management system by SPS impersonating the authenticated user. This automatic password retrieval is crucial, as this method protects the confidentiality of passwords since users can never access them.

Users accessing connections that use Credential Stores must authenticate on SPS using gateway authentication. They only have to use their gateway password to log in to SPS, and if they are allowed to access the target server, SPS automatically logs in using the Credential Store. For details on gateway authentication, see The gateway authentication process on page 64.

Figure 5: Authenticating using Credential Stores



Credential Stores can be stored locally on SPS, or on a remote device. For remote Credential Stores, SPS integrates with external authentication and authorization systems using plugins.

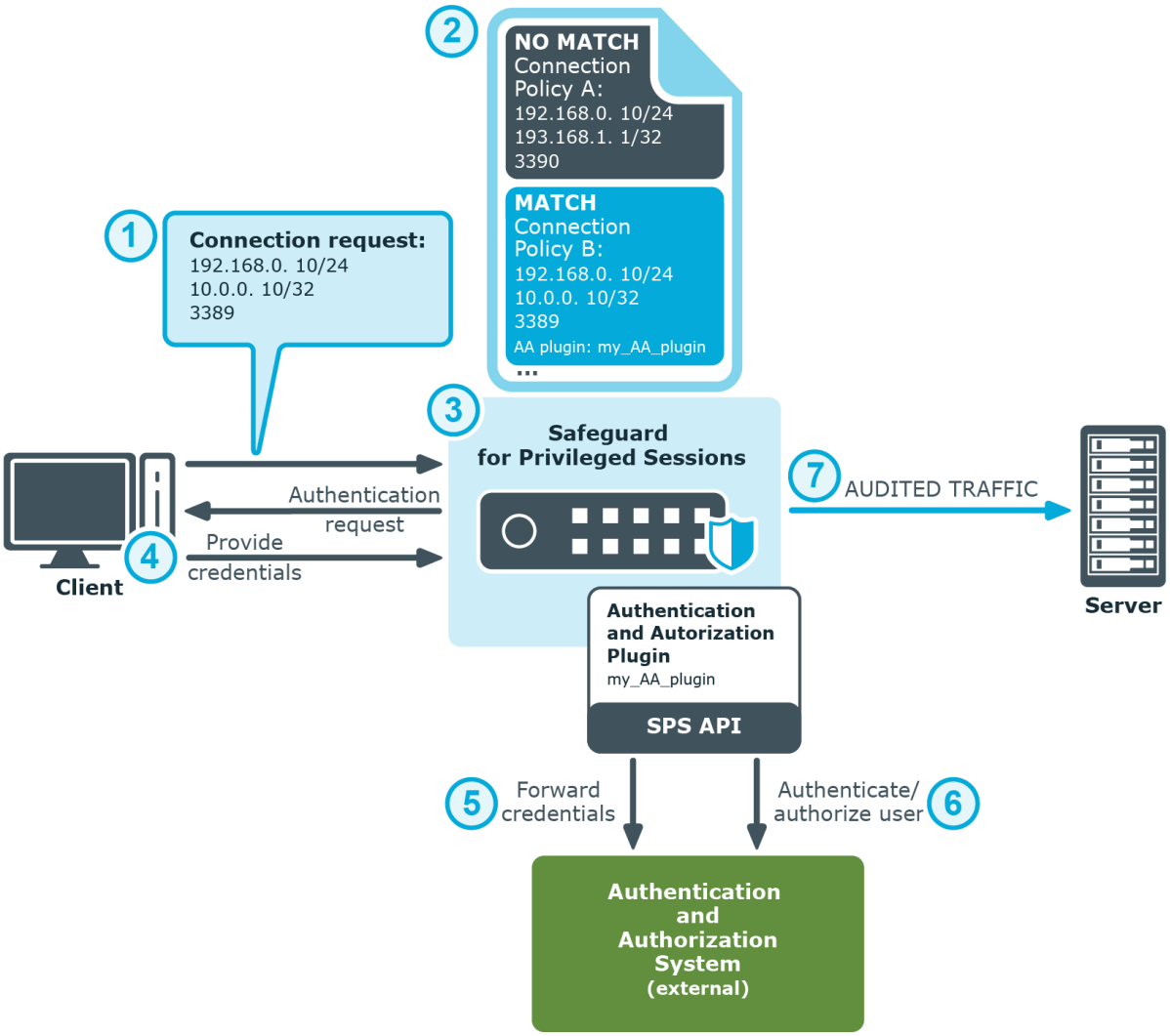
For further information on Credential Stores including configuration details, see Using credential stores for server-side authentication on page 757.

Plugin framework

One Identity Safeguard for Privileged Sessions (SPS) provides a plugin framework to integrate SPS with external authentication and authorization systems, such as an external Credential Store, a ticketing system, or any third-party authentication or authorization solution.

Authenticating users to an external authentication and authorization system on page 36 and the process overview that follows describe how user authentication works at a high level when there is an external authentication and authorization system involved:

Figure 6: Authenticating users to an external authentication and authorization system



1. The client tries to establish a connection to the target server.
2. SPS notices that an AA plugin is configured in the connection policy matching the connection. This is treated as gateway authentication. For details on gateway authentication, see The gateway authentication process on page 64.
3. SPS prompts the client for credentials.
4. The client provides authentication details to SPS when prompted.
5. SPS forwards the client's details to the external authentication and authorization system using the SPS API.
6. The external authentication and authorization system verifies the data received and provides feedback to SPS about the result.
7. If the client is granted access by the external authentication and authorization

system, SPS authenticates the client to the target server, and establishes the connection.

For further information on plugins including configuration details, see Integrating ticketing systems on page 774 and Integrating external authentication and authorization systems on page 768.

Indexing

One Identity Safeguard for Privileged Sessions (SPS) can index the contents of audit trails, making the records of privileged users' activities easily searchable.

Audit trails contain user activity data recorded from terminal sessions (such as SSH and Telnet) and graphical protocols (such as RDP, Citrix ICA, and VNC). Examples of data recorded in audit trails are: mouse activity, keystrokes, and so on. Using its own indexer service or one or more external indexers, SPS determines elements of the content visible on the user's screen at a given point in time. Screen content elements include commands, window titles, IP addresses, user names, and so on.

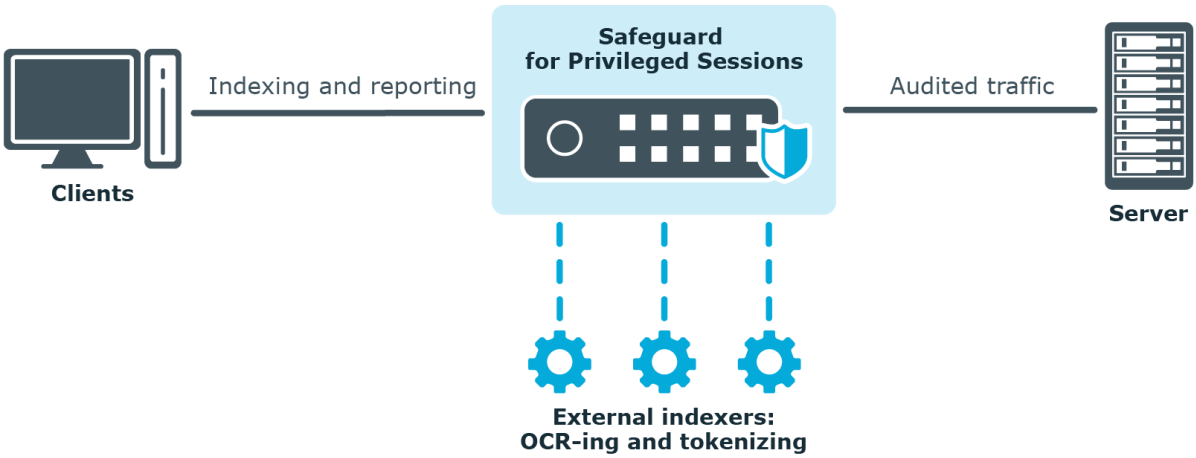
The indexer generates the following types of output as a result of processing the audit trail files:

- text
- screenshot files
- replayable video files

SPS then takes the output of indexing and breaks that down into searchable units.

Indexing and the process overview that follows describe how indexing works at a high level:

Figure 7: Indexing audit trail files



1. SPS monitors and records the protocol traffic in the audited connections passing through SPS. Protocol traffic data is recorded in audit trail files.
2. Once a connection has been closed, SPS sends the audit trail files to the indexer.
3. The indexer parses the contents of the audit trail files, and builds an "inventory" of the privileged user's activity data based on what appeared on their screen.

   In the case of a terminal session, screen content corresponds to the activity data that is captured in a terminal window. In the case of graphical protocols, screen content is whatever is visible in the graphical user interface of the applications the user is interacting with. In the latter case, the indexer's Optical Character Recognition (OCR) engine extracts text that appeared on the screen (for example, window titles).
4. The indexer returns the information extracted from the parsed audit trail files to SPS.
5. SPS processes the outcome of parsing and OCR-ing done in the previous phase and makes the data searchable.
6. Once indexed, the contents of the audit trails can be searched from SPS's web interface.

For details on how to configure SPS's internal indexer or one or more external indexers, see Indexing audit trails on page 592.

Supported protocols and client applications

One Identity Safeguard for Privileged Sessions (SPS) supports the following protocols and clients. As a general rule, client applications not specifically tested, but conforming to the relevant protocol standards, should work with SPS. One Identity supports the listed client and server applications only on a best-effort basis after their vendor or manufacturer declares end-of-support or extended (or any other non-standard support) period for them. Best-effort basis means that without the vendor support we only can fix issues with our existing knowledge in the problematic area, and can implement straightforward fixes only.

Example:

Microsoft provides mainstream and extended support periods for Windows Server 2012 Standard as described here. One Identity follows these periods and our best-effort support period starts at the same time when the mainstream period ends at Microsoft.

HTTP

One Identity Safeguard for Privileged Sessions (SPS) supports the HTTP 1.0 and 1.1 standards.

Secure Shell Protocol

One Identity Safeguard for Privileged Sessions (SPS) supports only the SSHv2 protocol. The older and insecure v1 version is not supported.

Supported client and server applications:

- OpenSSH (client and server)

  Client and server tested with a weekly build of the latest available version.
- OpenSSH (client and server)

  Client and server tested with version OpenSSH_7.1p2 and OpenSSL 1.0.2f-fips 28 Jan 2016.
- Dropbear (client and server)

  Tested with version 2015.67.
- SecureCRT (Windows, client)

  Tested with version 8.5
- PUTTY (client)

  Tested with version 0.65.

Remote Desktop Protocol

Supported Windows client applications:

The built-in applications of the Windows 7 SP1, Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019, and Windows 10 platforms.

Supported Mac OS X client applications

The Royal TSX client application, tested with Royal TSX 4.2.1 on Mac OS X Mojave.

Supported server (target) applications

The built-in applications of Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019, and Windows 10 platforms.

Accessing Remote Desktop Services (RemoteApp programs) is also supported.

NOTE:

Other Remote Desktop clients are not explicitly supported, but may be compatible with One Identity Safeguard for Privileged Sessions (SPS). When using an alternative client application, note the following limitations:

- The rdesktop application and other client applications (for example, JAVA clients) that build on the rdesktop codebase do not support RDP shadowing and Remote Desktop Gateway connections.
- The Remote Desktop Connection Client for Mac application does not support RDP shadowing.

ICA

One Identity Safeguard for Privileged Sessions (SPS) is certified for the following server versions:

- Citrix Virtual Apps (formerly known as Citrix XenApp) 6.5
- Citrix Virtual Apps 7.6
- Citrix Virtual Apps 7.15
- Citrix Virtual Desktops (formerly known as Citrix XenDesktop) 6.5
- Citrix Virtual Desktops 7.6
- Citrix Virtual Desktops 7.15

For details on the deployment scenarios that support Citrix Virtual Desktops (formerly known as Citrix XenDesktop), see One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a Citrix environment on page 510.

The latest version of the Citrix Workspace app (formerly known as Citrix Receiver) for Windows, Linux and MacOS is supported.

Telnet

Telnet traffic must conform to RFC 854 and to various extensions described in the following RFCs: 856-861, 652-658, 698, 726-27, 732-736, 749, 779, 885, 927, 933, 1041, 1043, 1053, 1073, 1079, 1091, 1096-97, 1184, 1372, 1408, 1572, 2066, 2217, 2840, 2941, 2946.

TN3270:

Telnet 3270 terminal protocol.

TN5250:

Telnet 5250 terminal protocol, as described in RFC2877.

Remote Desktop Gateway Server Protocol (RDGSP)

One Identity Safeguard for Privileged Sessions (SPS) can act as a Remote Desktop Gateway (also called RD Gateway) and transfer the incoming connections to RDP connections.

Virtual Network Computing

Safeguard for Privileged Sessions (SPS) supports the Remote Framebuffer (RFB) protocol which is used in various open source VNC implementations. RFB versions 3.3-3.8 are supported.

The following client and server applications are supported if they are built on the open source RFB protocol:

- RealVNC
- UltraVNC
- TightVNC
- KVM
- Vino

VMware Horizon View

VMware Horizon View Clients using the Remote Desktop (RDP) display protocol to access remote servers are supported. For details, see VMware Horizon View connections on page 582.

Modes of operation

One Identity Safeguard for Privileged Sessions (SPS) can be configured to monitor both *transparent* and *non-transparent* connections.

- In *transparent* mode, SPS acts as a transparent router between two network segments. For details, see Transparent mode on page 42.
- You can also use policy-based routing to forward connections within the same network segment to SPS, in which case it acts like a *single interface transparent router*. For details, see Single-interface transparent mode on page 42.
- In *non-transparent* mode, users have to address SPS to initiate connections to protected servers. For details, see Non-transparent mode on page 44.

- When addressing SPS, you can also use *inband destination selection* to choose the server to connect to. For details, see Inband destination selection on page 45.

One Identity recommends that you design the network topology so that only management and server administration traffic passes SPS. This ensures that the services and applications running on the servers are accessible even in case SPS breaks down, so SPS cannot become a single point of failure.

Transparent mode

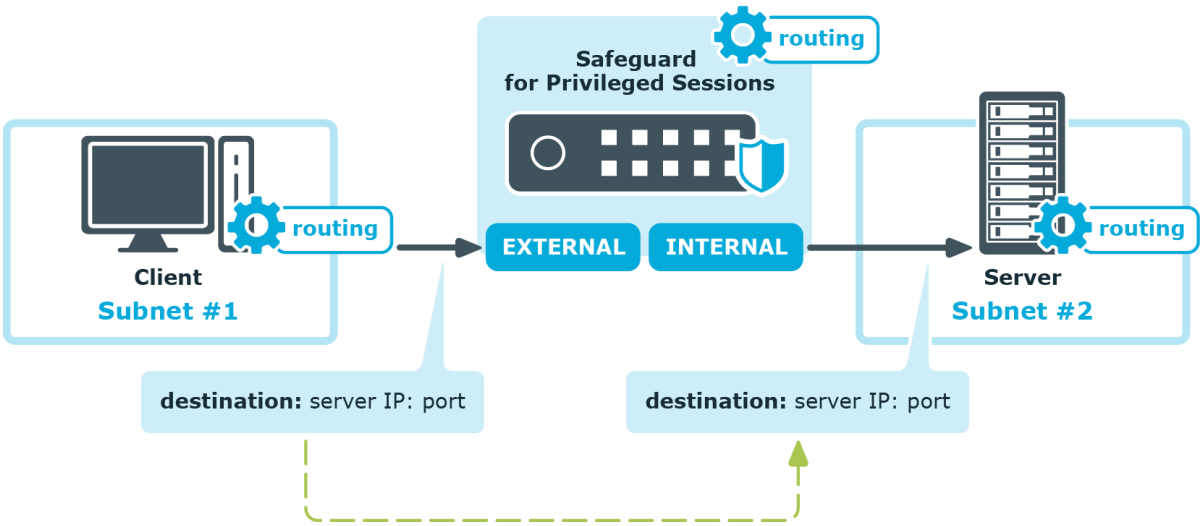
In *transparent* mode, One Identity Safeguard for Privileged Sessions (SPS) acts as a transparent router connecting the network segment of the administrators to the segment of the protected servers at the network layer (Layer 3 in the OSI model). All connections must pass through SPS to reach the servers — SPS is a proxy gateway, completely separating the protected servers from the rest of the network. Controlled connections and traffic are inspected on the application level, while other types of connections are simply forwarded on the packet level.

SPS can also be configured to act as a *single-interface transparent router*. For details, see Single-interface transparent mode on page 42.

CAUTION:

Transparent mode does not support multicast traffic.

Figure 8: SPS in transparent mode

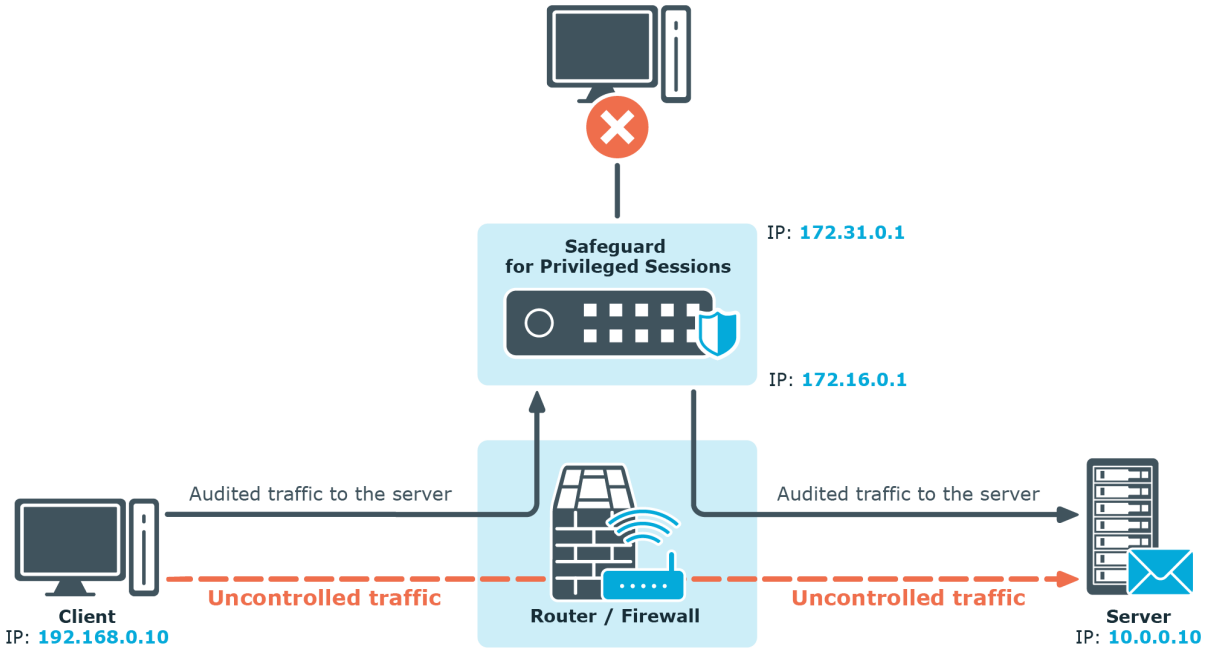


Single-interface transparent mode

Single-interface transparent mode is similar to transparent mode, but both client-side and server-side traffic use the same interface. An external device that actively redirects the audited traffic to One Identity Safeguard for Privileged Sessions (SPS) (typically, a firewall,

a router, or a layer3 switch) is required . To accomplish this, the external device must support advanced routing (also called policy-based routing or PBR). For details on configuring an external device to work with SPS in single-interface transparent mode, see Configuring external devices on page 909.

Figure 9: SPS in single-interface transparent mode



Advantages

The advantages of using the single-interface transparent mode are:

- Totally transparent for the clients, no need to modify their configuration.
- The network topology is not changed.
- Only the audited traffic is routed to SPS, production traffic is not.

Disadvantages

The disadvantages of using the single-interface transparent mode are:

- SPS acts as a man-in-the-middle regarding the connection between the client and the target server. Instead of a single client-server connection, there are two separate connections: the first between the client and SPS, and a second between SPS and the server. Depending on how you configure SPS, the source IP in the SPS-server connection can be the IP address of SPS, or the IP address of the client. In the latter case — when operating in transparent mode (including single-interface transparent mode) — SPS performs IP spoofing. Consult the security policy of your organization to see if it permits IP spoofing on your network.

- Traffic must be actively routed to SPS using an external device. Consequently, a network administrator can disable SPS by changing routing rules.
- When adding a new port or subnet to the list of audited connections, the configuration of the external device must be modified as well.
- A network administrator can (intentionally or unintentionally) easily disable monitoring of the servers, therefore additional measures have to be applied to detect such activities.

Non-transparent mode

In *non-transparent* mode, One Identity Safeguard for Privileged Sessions (SPS) acts as a bastion host (that is, administrators can address only SPS, the administered servers cannot be targeted directly). The firewall of the network has to be configured to ensure that only connections originating from SPS can access the servers. SPS determines which server to connect to based on the parameters of the incoming connection (the IP address of the administrator and the target IP and port).

Non-transparent mode inherently ensures that only the controlled (management and server administration) traffic reaches SPS. Services and applications running on the servers are accessible even in case SPS breaks down, so SPS cannot become a single point of failure.

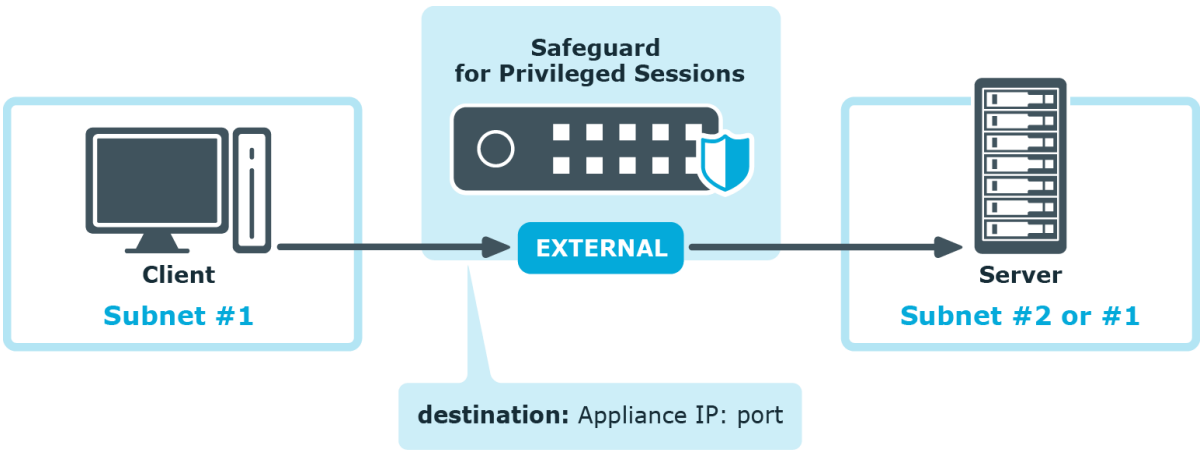
TIP:

Non-transparent mode is useful if the general (that is, not inspected) traffic is very high and could not be forwarded by SPS.

NOTE:

In case there is a high number of target devices, do not use fixed address rules in non-transparent mode, as configuration validation might fail. Consider using one of the dynamic configuration options, such as inband destination selection or transparent mode.

Figure 10: SPS in non-transparent mode

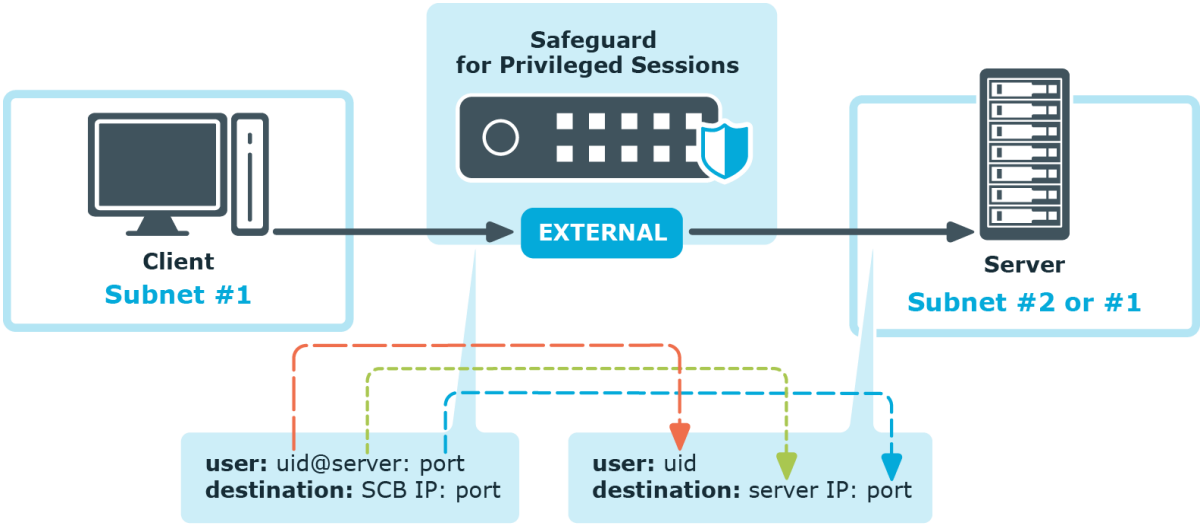


Non-transparent mode is often used together with inband destination selection. For details, see Inband destination selection on page 45).

Inband destination selection

Inband destination selection allows you to create a single connection policy and allow users to access any server by including the name of the target server in their username (for example, `ssh username@targetserver:port@scb_address`). One Identity Safeguard for Privileged Sessions (SPS) can extract the address from the username and direct the connection to the target server.

Figure 11: Inband destination selection



Since some client applications do not permit the `@` and `:` characters in the username, alternative characters can be used as well:

- To separate the username and the target server, use the `@` or `%` characters, for example: `username%targetserver@scb_address`
- To separate the target server and the port number, use the `:`, `+`, or `/` characters, for example: `username%targetserver+port@scb_address`

You can use both IPv4 and IPv6 addresses with inband destination selection. For IPv6 addresses, add square brackets to separate the address and the port number:

```
username@[targetserver_ipv6]:port@[scb_address_ipv6]:port
```

When Network Level Authentication (NLA) is disabled, you can omit the username when starting an RDP connection (for example, use only `%targetserver`). The user can type the username later in the graphical login screen. However, the username must be specified if Network Level Authentication (NLA) is used in the connection.

For other details on inband destination selection in RDP connections, see Inband destination selection in RDP connections on page 538.

You can find examples of using inband destination selection in Using inband destination selection in SSH connections on page 837.

Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS)

When a client initiates a connection to a server, One Identity Safeguard for Privileged Sessions (SPS) performs a procedure similar to the ones detailed below. The exact procedure depends on the protocol used in the connection.

- For SSH connections, see Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using SSH on page 46.
- For RDP and other connections, see Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using RDP on page 49.

Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using SSH

The following describes what happens when a client connects to a server through One Identity Safeguard for Privileged Sessions (SPS) and how the different configuration options and policies of SPS affect this process. Note that this procedure does not cover the scenarios when inband destination selection is used.

1. **Client-side connection**

   The client tries to connect to the server. SPS receives the connection request and establishes the TCP connection with the client.
2. SPS examines the connection request: it checks the IP address of the client and the IP address and port number of the intended destination server. If these parameters of the request match a connection policy configured on SPS, SPS inspects the connection in detail. Other connections are ignored by SPS, and simply forwarded on the packet level.

   The selected connection policy determines all settings and parameters of the connection.

NOTE:

One Identity Safeguard for Privileged Sessions (SPS) compares the connection policies to the parameters of the connection request one-by-one, starting with the first policy in the policy list. The first connection policy completely matching the connection request is applied to the connection.

For details, see Configuring connections on page 434.

3. SPS selects the destination server based on the **Target** parameter of the connection policy. Network address translation of the target address can be performed at this step. For details, see Modifying the destination address on page 441.
4. SPS selects the source address used in the server-side connection based on the **SNAT** parameter of the connection policy. For details, see Modifying the source address on page 446.
5. The client authenticates itself using an authentication method permitted by the **Authentication policy** set in the Connection policy. Different connections can use different authentication policies, thus allow different authentication methods. The Authentication policy also restricts which users can connect to the server if public-key authentication is used. SPS can authenticate the user to a **Local User Database**, or to a remote LDAP (for example, Microsoft Active Directory) or RADIUS server. This is inband authentication, since it is performed in the same connection that the client originally established to communicate with the server.

   The username used in this authentication step is referred to as the **Gateway username** and is used to determine the **Gateway group** memberships of the user. For details, see Authentication Policies on page 552.

   If an **AA plugin** is configured in SPS, the client may be prompted to provide additional information when authenticating to the server. For details on the **AA plugin**, see Integrating external authentication and authorization systems on page 768. Note that if the plugin sets or overrides the username of the connection, a Usermapping policy needs to be configured and set in the Connection policy. For further information, see Configuring usermapping policies on page 740.
6. If the **Gateway authentication** option is set in the Connection policy, SPS pauses the connection until the user completes a gateway authentication on the SPS web interface. This is out-of-band authentication, since it is performed in an independent connection. For details, see The gateway authentication process on page 64.
7. If the **Usermapping policy** option is set in the Connection policy, SPS checks if the Usermapping policy permits the users of the gateway group to access the username used in the server-side connection (the remote username, for example, `root`). For details, see Configuring usermapping policies on page 740.
8. Before establishing the server-side connection, SPS can evaluate the channel policy to determine if the connection might be permitted at all, for example, it is not denied by a Time policy. SPS performs this check if the **SSH Control > Settings > Enable pre channel check** option is enabled. For details, see Creating and editing protocol-level SSH settings on page 564.

For the SSH protocol, SPS checks the **From** (client address), **Gateway group**, and **Time policy** restrictions set in the Channel policy of the Connection policy. For details, see Creating and editing channel policies on page 448.

9. **Server-side connection**

   SPS sets up the server-side connection and does the following:

   a. SPS establishes the TCP connection to the server.

   b. SPS negotiates the protocol parameters of the connection (for example, SSH encryption parameters) according to the **SSH Control > Settings** of the connection policy.

   c. SPS displays an SSH hostkey to the client. This hostkey is either generated on SPS, or it is the hostkey of the server (if it is available on SPS). The connection policy determines the hostkey shown to the client.

      IMPORTANT:

      If the SSH Settings of the Connection Policy enable only RSA keys, set the RSA key shown to the client in the Connection Policy.

   d. SPS verifies the hostkey of the server according to the **Server side hostkey setting** option of the Connection policy (in general, you can manage the server hostkeys on the **SSH Control > Server Host Keys** page). If the server has not been contacted before, SPS can accept and store the hostkey of the server. Alternatively, the hostkey of the server can be manually uploaded to SPS. For details, see Server host keys on page 560.

10. SPS performs the authentication on the server, using the data received from:

    - the client during the client-side authentication, or
    - a local or external Credential Store (for details, see Using credential stores for server-side authentication on page 757).

11. SPS authorizes the connection based on the Channel policy. It checks:

    - If the Channel policy includes a **User List** restriction for the **Gateway group** or **Remote group**, SPS checks if the user can access the server. If needed, SPS connects to the LDAP servers set in the **LDAP Servers** policy to resolve the group memberships of the user. For details, see Creating and editing user lists on page 457.
    - SPS consults the **Time policy** assigned to the channel policy. Channels may be opened only within the allowed period.

      TIP:

      Time policies are a good way to ensure that the server can be accessed only within the specified timeframe.

12. Both the server- and the client-side connections have been established. From this step, the client can try to open any type and any number of channels in the connection.

13. If 4-eyes authorization is set in the Channel policy, the SSH session of the client is paused until the authorizer permits the client to connect to the server. Who can authorize the session depends on the **Access Control** settings of the Connection policy. For details, see Four-eyes authorization on page 65.

14. The client starts to work on the server. Information about the connection is now available on the **Search** page.

    - SPS records the entire communication into digitally encrypted audit trails if auditing is enabled in the Channel policy, and encryption is configured in the Audit policy used in the Connection policy. For details, see Creating and editing channel policies on page 448 and Audit policies on page 465.
    - If a Content policy is configured in the Channel policy, SPS monitors the connection in real time, and raises an alert or terminates the connection if the user performs an undesired action. For details, see Real-time content monitoring with Content Policies on page 451.

    If the user opens another channel within the same connection, SPS consults the Channel policy of the connection to see if the channel is permitted, and processes it accordingly.

15. **Post-processing the connection**

    Once the connection has been closed, the following post-processing steps take place:

    a. After the client closes the connection, or it is terminated for some reason (for example, it times out, or a Content policy or a 4-eyes auditor terminates it), SPS indexes the contents of the audit trail (if the **Record audit trail** option of the Channel policy, and the **Enable indexing** option of the Connection policy are enabled).

    b. SPS creates a backup of the data and the audit trail of the connection, and archives it to a remote server, if a Backup policy and an Archive policy is set in the Connection policy. For more information, see Data and configuration backups on page 146 and Archiving and cleanup on page 159.

    c. When the **Delete search metadata from SPS after** period expires, SPS deletes all data about the connection from its database.

Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using RDP

The following describes what happens when a client connects to a server through One Identity Safeguard for Privileged Sessions (SPS) and how the different configuration options and policies of SPS affect this process.

1. **Client-side connection**

   The client tries to connect to the server. SPS receives the connection request and establishes the TCP connection with the client.

2. SPS examines the connection request: it checks the IP address of the client and the IP address and port number of the intended destination server. If these parameters of the request match a connection policy configured on SPS, SPS inspects the connection in detail. Other connections are ignored by SPS, and simply forwarded on the packet level.

   The selected connection policy determines all settings and parameters of the connection.

   > NOTE:
   >
   > One Identity Safeguard for Privileged Sessions (SPS) compares the connection policies to the parameters of the connection request one-by-one, starting with the first policy in the policy list. The first connection policy completely matching the connection request is applied to the connection.
   >
   > For details, see Configuring connections on page 434.

3. SPS selects the destination server based on the **Target** parameter of the connection policy. Network address translation of the target address can be performed at this step. For details, see Modifying the destination address on page 441.

4. SPS selects the source address used in the server-side connection based on the **SNAT** parameter of the connection policy. For details, see Modifying the source address on page 446.

5. If an **AA plugin** is configured in SPS, the client may be prompted to provide additional information when authenticating to the server. For details on the **AA plugin**, see Integrating external authentication and authorization systems on page 768. Note that if the plugin sets or overrides the username of the connection, a Usermapping policy needs to be configured and set in the Connection policy. For further information, see Configuring usermapping policies on page 740.

6. SPS checks if the client uses a version of the RDP protocol that is enabled in the **Protocol settings** of the Connection policy. Depending on the protocol version, different encryption is used in the connection, and different parameters are required in the Connection policy.

7. Before establishing the server-side connection, SPS can evaluate the channel policy to determine if the connection might be permitted at all, for example, it is not denied by a Time policy. SPS performs this check if the **RDP Control > Settings > Enable pre channel check** option is enabled. For details, see Creating and editing protocol-level RDP settings on page 519.

8. **Server-side connection**

    a. SPS establishes the TCP connection to the server.

    b. SPS checks the protocol parameters of the connection (for example, the version of the RDP protocol used) according to the **Protocol settings** of the Connection policy. The RDP handshake is performed simultaneously on the server- and the client-side.

9. The server opens a Drawing channel for the user to perform authentication.

10. SPS authorizes the connection based on the Channel policy. It checks:

    - If the Channel policy includes a **User List** restriction for the **Gateway group** or **Remote group**, SPS checks if the user can access the server. If needed, SPS connects to the LDAP servers set in the **LDAP Servers** policy to resolve the group memberships of the user. For details, see Creating and editing user lists on page 457.
    - SPS consults the **Time policy** assigned to the channel policy. Channels may be opened only within the allowed period.

      TIP:

      Time policies are a good way to ensure that the server can be accessed only within the specified timeframe.

11. If the **Gateway authentication** option is set in the Connection policy, SPS pauses the connection until the user completes a gateway authentication on the SPS web interface. This is out-of-band authentication, since it is performed in an independent connection. For details, see The gateway authentication process on page 64.

    It is also possible to perform gateway authentication inband, without having to access SPS's web interface. For details, see Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using an RD Gateway on page 52.

12. SPS performs the authentication on the server, using the data received from:

    - the client during the client-side authentication, or
    - a local or external Credential Store (for details, see Using credential stores for server-side authentication on page 757).

13. If the authentication fails for any reason, SPS terminates the client-side connection as well. This is required to verify the username of the client when it attempts to access the server again.

14. If 4-eyes authorization is set in the Channel policy, the RDP session of the client is paused until the authorizer permits the client to connect to the server. Who can authorize the session depends on the **Access Control** settings of the Connection policy. For details, see Four-eyes authorization on page 65.

15. The client starts to work on the server. Information about the connection is now available on the **Search** page.

- SPS records the entire communication into digitally encrypted audit trails if auditing is enabled in the Channel policy, and encryption is configured in the Audit policy used in the Connection policy. For details, see Creating and editing channel policies on page 448 and Audit policies on page 465.
- If a Content policy is configured in the Channel policy, SPS monitors the connection in real time, and raises an alert or terminates the connection if the user performs an undesired action. For details, see Real-time content monitoring with Content Policies on page 451.

If the user opens another channel within the same connection, SPS consults the Channel policy of the connection to see if the channel is permitted, and processes it accordingly.

16. **Post-processing the connection**

    Once the connection has been closed, the following post-processing steps take place:

    a. After the client closes the connection, or it is terminated for some reason (for example, it times out, or a Content policy or a 4-eyes auditor terminates it), SPS indexes the contents of the audit trail (if the **Record audit trail** option of the Channel policy, and the **Enable indexing** option of the Connection policy are enabled).

    b. SPS creates a backup of the data and the audit trail of the connection, and archives it to a remote server, if a Backup policy and an Archive policy is set in the Connection policy. For more information, see Data and configuration backups on page 146 and Archiving and cleanup on page 159.

    c. When the **Delete search metadata from SPS after** period expires, SPS deletes all data about the connection from its database.

Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using an RD Gateway

The following describes what happens when a client connects a server through One Identity Safeguard for Privileged Sessions (SPS) using a Remote Desktop Gateway (or RD Gateway), and how the different configuration options and policies of SPS affect this process. For details on the configuration process, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on page 531.

1. The client connects to port 443 of the Remote Desktop Gateway configured in the Remote Desktop software. The address of the Remote Desktop Gateway is an alias IP address of SPS. To process the connection request, SPS must have a Connection policy that is configured to handle RDP connection requests on the alias IP, and that has the **Act as a Remote Desktop Gateway** option enabled.
2. The client authenticates on Remote Desktop Gateway (that is, on SPS). Technically, this is an inband gateway authentication on the Domain Controller of SPS's domain

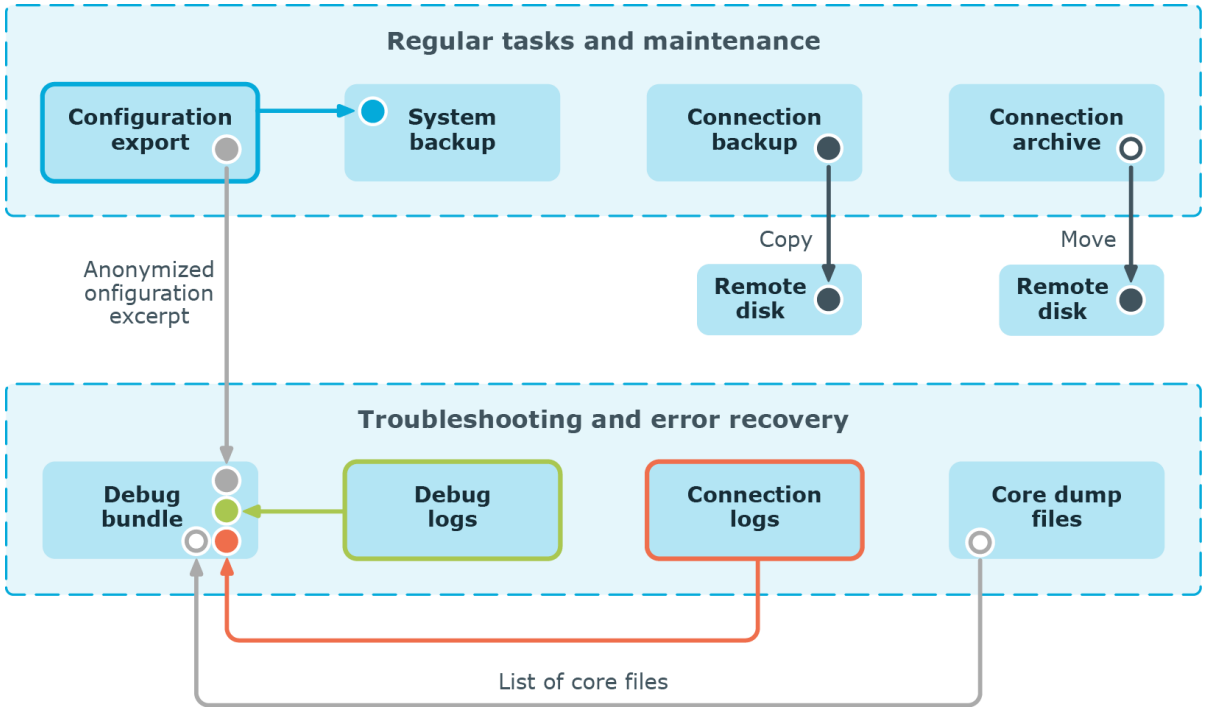
(SPS must be the member of a domain, for details, see Network Level Authentication (NLA) with domain membership on page 524). The username used in this authentication step is referred to as the **Gateway username** and is used to determine the **Gateway group** memberships of the user.

3. The client tries to connect to the server. From this point on, this connection is processed as described in Connecting to a server through One Identity Safeguard for Privileged Sessions (SPS) using RDP on page 49.

Archive and backup concepts

You can export, backup and save various types of data from One Identity Safeguard for Privileged Sessions (SPS), and it also creates log files, dumps and bundles to help the Support Team in troubleshooting errors.

Figure 12: Archive and backup concepts



The following sections describe these in detail:

- Configuration export
- System backup
- Connection backup
- Connection archive
- Support bundle

- Debug logs
- Connection logs
- Core dump files

Configuration export

The configuration of One Identity Safeguard for Privileged Sessions (SPS) can be exported to your local machine from the **Basic Settings > System > Export configuration** page. The configuration export in itself is always a one-time action that cannot be configured in policies. However, the system backup (System backup on page 55), that contains the configuration export in addition to other items, can be configured as a scheduled policy and is saved to a backup server.

The exported file is a gzip-compressed archive. On Windows platforms, it can be decompressed with common archive managers such as the free 7-Zip tool.

The name of the exported file is `<hostname_of_SPS>-YYYYMMDDTHHMM.config`, the `-encrypted` or `-gpg` suffix is added for password-encrypted and GPG-encrypted files, respectively. Because the configuration export contains highly sensitive information, it is strongly suggested that you use encryption when generating the export.

For details on how to export the configuration of SPS, see: Exporting the configuration of SPS.

The configuration export is used for

- Manually archiving the configuration.
- Reinstalling a SPS machine and restoring its configuration.
- Migrating the configuration of an already installed SPS to a freshly installed SPS of the same version and therefore creating a machine with an identical configuration.

The configuration export contains the following

- Configuration XML file
- Every change of the configuration of SPS. You can also access these changes at **AAA > Accounting** in a search interface.
- Certificates, for example:
  - CA certificates
  - TSA certificates
  - Signing CA
- Stored key files, for example:
  - Trusted keys
  - User keys

    - RDP5 RSA key
- User Preferences that are configured at **User Menu > Preferences**.
- Certificates and corresponding private keys in your private keystore that are configured at **User Menu > Private Keystore** . Only the content of the **Permanent** keystore is exported.
- Custom Report Logo configured at **Reporting > Configuration**.
- Plugins and any data persisted by plugins.
- Local Credentials Store (the SQLite database) configured at **Policies > Credential Stores**.

System backup

The system backup contains the configuration export in addition to other items. It can be configured as a scheduled policy and is saved to a backup server.

Because the configuration export, which is part of the system backup contains highly sensitive information, it is strongly suggested that you use encryption when generating the export. For details on encrypting the configuration export part, see: Encrypting configuration backups with GPG.

For details on how to perform a system backup of One Identity Safeguard for Privileged Sessions (SPS), see: Creating configuration backups. It is a two-step process:

1. Create a backup policy at **Policies > Backup & Archive/Cleanup > Backup policies**.
2. Assign that policy to the system backup at **Basic Settings > Management > System backup > System backup policy**.

   Select **Encrypt configuration**.

For details on how to restore the configuration and data of SPS from a complete backup, for example, after a hardware replacement, see: Restoring SPS configuration and data.

The system backup is used for

- Recovery in case of errors.

The system backup contains the following

- `config` directory:

  One configuration export file per scheduled backup.
- `db` directory:

  A database dump from SPS's connection metadata database, one `.sql` file overwritten with the actual dump on a daily basis.
- `reports` directory:

The scheduled daily, weekly, monthly system reports that are accessible at **Reporting > Reports** are saved in `.pdf` files.

- `rrd` directory:

  The output files of the internal system monitoring tool (Munin). These are the files that are used in generating graphs/charts on the **Basic Settings > Dashboard** page.

- `sql` directory:

  The internal SQLite databases, for example metadata about the reports.

Connection backup

The connection backup, also known as data backup contains the audit files and connection metadata of a connection. It can be configured as a scheduled policy and is saved to a backup server.

For details on how to perform a connection backup of a connection, see: Creating data backups. It is a three-step process:

1. Configure a system backup. Restoring a data backup works only if a matching system configuration and metadata is available, that is, if a system backup is restored first.
2. Create a backup policy at **Policies > Backup & Archive/Cleanup > Backup policies**.
3. Navigate to **<Protocol name> Control > Connections**. Select the connection you want to back up. Select the previously created backup policy in the **Backup policy** field.

For details on how to restore the configuration and data of One Identity Safeguard for Privileged Sessions (SPS) from a complete backup, for example, after a hardware replacement, see: Restoring SPS configuration and data.

The connection backup is used for

- Saving the created audit trail files and indexing metadata of a connection to a remote share. This is a copy operation in terms of data files.
- Recovery: In case of a hardware replacement, creating configuration export, system backup and connection backups is essential.
- Migration: Creating a machine identical to another SPS machine.

The connection backup contains the following

- The audit trails of the connection, that is, the `.zat` files storing the recorded activities of the administrators. For details on audit trails, see Audit Policies.
- The index of the audit trail that makes the content of the audit trail searchable. For details on indexing audit trails, see Indexing audit trails.

NOTE:

Audit trails and index files are large. This means that backing up a connection requires a significant amount of free hardware space. Make sure you have enough free hardware space for those connections that you want to back up.

Connection archive

The connection archive, also known as data archive contains the audit files and connection metadata of a connection. In terms of contents, it is similar to a connection backup. It can be configured as a scheduled policy and is saved to an archive server. Archiving transfers data from One Identity Safeguard for Privileged Sessions (SPS) to an external storage solution, cleanup removes (deletes) old files. Archived data can be accessed and searched, but cannot be restored (moved back) to the SPS appliance.

For details on how to perform a connection archive of a connection, see: Archiving or cleaning up the collected data. It is a two-step process:

1. Create an archive policy at **Policies > Backup & Archive/Cleanup > Archive/Cleanup policies**.
2. Navigate to **<Protocol name> Control > Connections**. Select the connection you want to archive. Select the previously created archive policy in the **Archive/Cleanup policy** field.

CAUTION:

Hazard of data loss Never delete an Archive Policy if data has been archived with it. This will make the already archived data inaccessible.

Do not "remake" an Archive Policy (that is, deleting an Archive Policy and then creating another one with the same name but different parameters). This will make data inaccessible, and identifying the root cause of the issue complicated.

If you want to change the connection parameters (that is when you perform a storage server migration), you must make sure that the share contents and file permissions are kept unmodified and there are no archiving or backup tasks running.

On the other hand, if you want to add a new network share to your archives, proceed with the following steps:

1. **Create a new empty SMB/NFS network share.**
2. **Create a new Archive Policy that points to this network share.**
3. **Modify your Connection Policy(es) to archive using the newly defined Archive Policy.**
4. **Make sure to leave the existing Archive Policy unmodified.**

It is also safe to extend the size of the network share on the server side.

The connection archive is used for

- Moving the created audit trail files and indexing metadata of a connection to a remote share. This is a move operation in terms of data files. Archived data can be accessed and searched, but cannot be restored (moved back) to the SPS appliance.
- Freeing up hardware space on SPS.

The connection archive contains the following

- The audit trails of the connection, that is, the `.zat` files storing the recorded activities of the administrators. For details on audit trails, see Audit Policies.
- The index of the audit trail that makes the content of the audit trail searchable. For details on indexing audit trails, see Indexing audit trails.

Support bundle

To track down support requests, the One Identity Support Team might request you to collect system-state and debugging information. This information is collected automatically, and contains log files, the anonymized excerpt of the configuration export file of One Identity Safeguard for Privileged Sessions (SPS), and various system-statistics. To generate a support bundle, navigate to **Basic Settings > Troubleshooting > Create support bundle**.

The exported file is a zip-compressed archive.

The name of the exported file is `debug_info-<hostname>YYYYMMDDHHMM`. Sensitive data like key files and passwords are automatically removed from the configuration files.

For details on how to create a support bundle, see: Collecting logs and system information for error reporting.

The support bundle is used for

- Collecting a snapshot of the past week's system-state information for the One Identity Support Team for troubleshooting and debugging purposes.
- Collecting information about a specific error by generating data for a defined time interval where the event that causes the error is reproduced. This is also used by the One Identity Support Team for troubleshooting and debugging purposes.

The support bundle contains the following

- Debug logs, Connection logs and OS logs of the past week, one file per day. If there are too many events in a day, the log file in the support bundle only contains a truncated version of the connection logs. In this case, the complete log file is only accessible at `/var/log/messages-<day>`.
- An excerpt of the configuration export file:

    - The anonymized version of the configuration XML file
    - Plugins
- System-state information (for example, version details, statistics, memory usage, system warnings, and so on).
- List of core files. This list might indicate previous system crashes.
- RAID controller information.
- Upgrade logs
- Dashboard data

Debug logs

To increase the log level of the non-connection-related events, for example, to add the commands executed by the One Identity Safeguard for Privileged Sessions (SPS) web interface to the logs, enable debug level logging at **Basic Settings > Management > Verbose system logs > Enable**.

These logs are accessible at `/var/log/scb-<day>`.

The debug logs are used for

- Our Support Team uses this to investigate the reasons behind a web user interface-related issue.

The debug logs contain the following

- Logs generated by the SPS web interface.
- System daemon logs.
- Logs of periodic cron jobs.

Connection logs

The connection logs contain all connection-related information of the past week, one file per day. A file contains all logs for all connections for a single day.

The logging level of One Identity Safeguard for Privileged Sessions (SPS) can be set separately for every protocol. To change the verbosity level of SPS, navigate to **<Protocol name> Control > Global Options**.

These logs are accessible at `/var/log/zorp-<protocol-name>-<day>`.

NOTE:

The verbosity level ranges from 1 (no logging) to 10 (extremely detailed), with level 4 being the default normal level. To debug complex problems, you might have to increase the verbosity level to 7. Higher level is needed only in extreme cases.

CAUTION:

High verbosity levels generate very large amount of log messages and might result in a very high load on the machine.

For log levels 8-10, the logs contain highly sensitive data for all connections, as well as passwords and private keys in plain text format.

The connection logs are used for

- Our Support Team uses this to investigate the reasons behind a failed connection.

The connection logs contain the following

- Connection success/failure events
- Other connection-related events

Core dump files

One Identity Safeguard for Privileged Sessions (SPS) automatically generates core dump files if an important software component (for example, Zorp) of the system crashes for some reason. These core dump files can be of great help to the One Identity Support Team to identify problems. When a core dump file is generated, the SPS administrator receives an alerting e-mail, and an SNMP trap is generated if alerting is properly configured (for details, see Configuring system monitoring on SPS on page 137 and System logging, SNMP and e-mail alerts on page 126).

To list and download the generated core dump files, navigate to **Basic Settings > Troubleshooting > Core files**.

For details on core dump files, see: Gathering data about system problems.

The core dump files are used for

- The One Identity Support Team uses this to investigate the reasons behind a system crash.

The core dump files contain the following

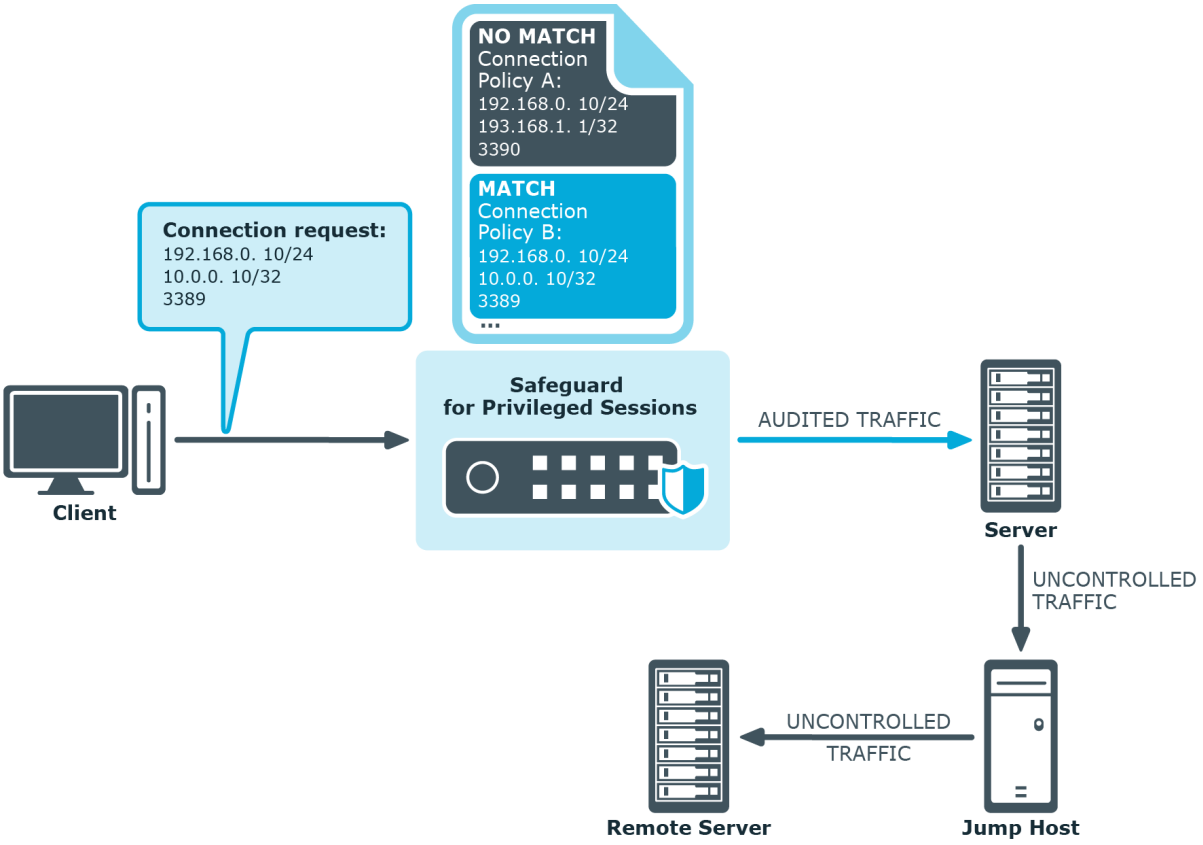
- The recorded state of the working memory of a computer program at a specific time, generally when the program has crashed or otherwise terminated abnormally.

Maximizing the scope of auditing

In certain special scenarios, One Identity Safeguard for Privileged Sessions (SPS) may examine and audit network traffic with some limitations, depending on the configuration.

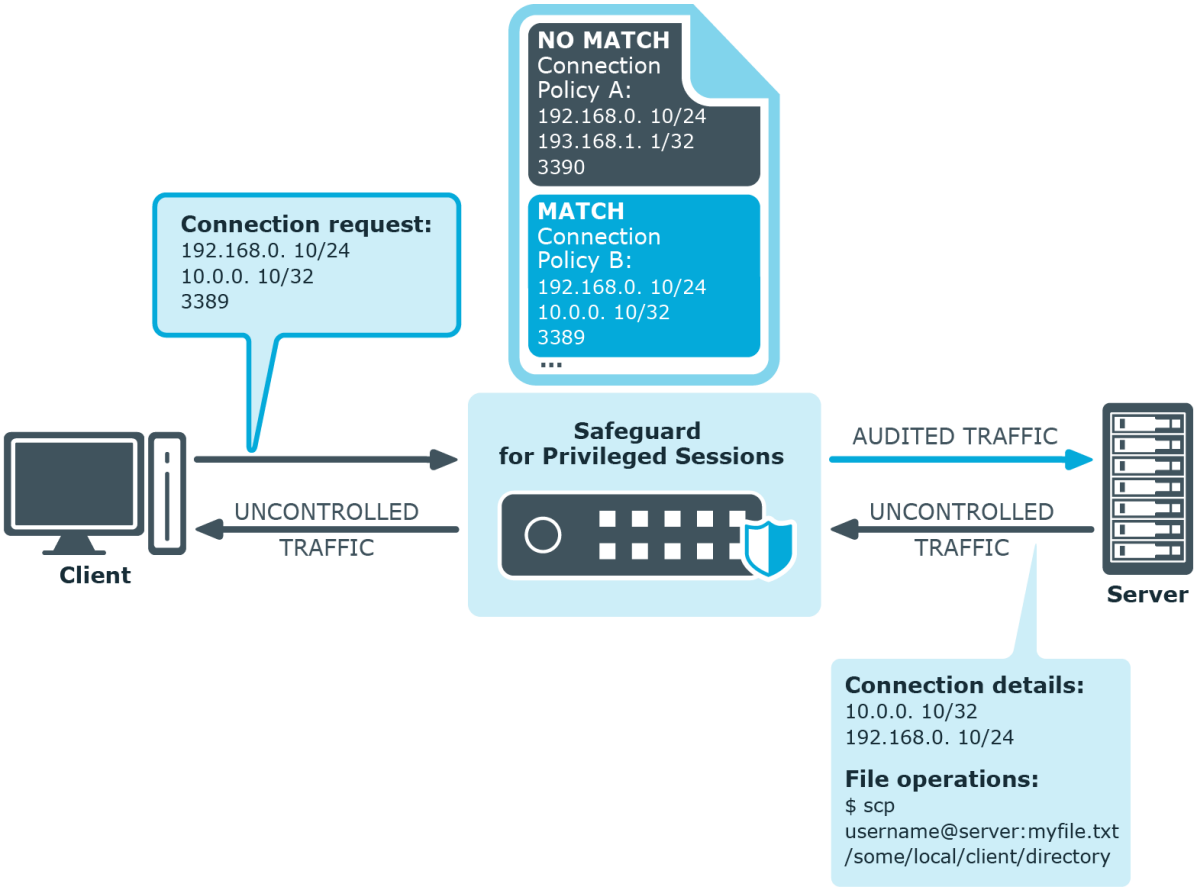
In the first scenario, your organization uses jump hosts to access remote servers or services. In this case, SPS ignores the connection between the target server and the remote server, as it does not go through SPS.

Figure 13: Connection to a remote server through a jump host



In the next scenario, a file operation is performed going from the target server to the client (for example, copying a file using SCP). In this case, the direction of the connection is switched, as compared to the initial client-to-server direction.

Figure 14: File operation in the "reverse" direction



In these scenarios, SPS may not:

- Restrict channels allowed in the connection.
- Audit file operations.

  When you wish to search for the audit files of these connections, there will be no results returned on the **Search** page.
- Allow authentication on the remote server if the user authenticates to the target server using a Credential Store.

If you want all connections in these scenarios to be audited, make sure that you add a connection policy for:

- The connection between the target server and any remote servers.
- The connection going from the target server to the client.

IPv6 in One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) supports IPv6 for monitoring connections only. You can define both IPv4 and IPv6 addresses for its logical network interfaces, and configure connections between IPv4 and IPv6 networks (for example, from a client with an IPv4 address to a target with an IPv6 address). You can also use IPv6 addresses with inband destination selection.

> NOTE:
>
> IPv6 support in ICA connections is currently experimental only.

When configuring IPv6 addresses, SPS shortens the address to its canonical form (omitting leading zeroes, and replacing consecutive sections of zeroes with a double colon). Take the following address as an example:

`2001:0db8:0000:0000:0000:ff00:0042:8329`

SPS shortens the address to its canonical form:

`[2001:db8::ff00:42:8329]`

Additionally, where the IP address and the port is displayed together, IPv6 addresses are shown between brackets. For example, the same address with a port number of 443 is displayed as:

`[2001:db8::ff00:42:8329]:443`

You can search for both the initial (full) and the canonical form on the SPS Search page. For searches performed using the RPC API, you have to use the canonical form.

To provide the network range for IPv6 addresses, use network prefixes. Pay attention to the differences between IPv4 and IPv6 network ranges: for IPv4, you can limit the address range to a single address with a prefix of `/32`, but to achieve the same on an IPv6 network, you have to use set the prefix to `/128`.

SSH hostkeys

SSH communication authenticates the remote SSH server using public-key cryptography, either using plain hostkeys, or X.509 certificates. Client authentication can also use public-key cryptography. The identity of the remote server can be verified by inspecting its hostkey or certificate. When trying to connect to a server via One Identity Safeguard for Privileged Sessions (SPS), the client sees a hostkey (or certificate) shown by SPS. This key is either the hostkey of SPS, or the original hostkey of the server, provided that the private key of the server has been uploaded to SPS. In the latter case, the client will not notice any difference and have no knowledge that it is not communicating directly with the server, but with SPS.

Authenticating clients using public-key authentication in SSH

Public-key authentication requires a private and a public key (or an X.509 certificate) to be available on One Identity Safeguard for Privileged Sessions (SPS). First, the public key of the user is needed to verify the user's identity in the client-side SSH connection: the key presented by the client is compared to the one stored on SPS. SPS uses a private key to authenticate itself to the sever in the server-side connection. SPS can use the private key of the user if it is uploaded to SPS. Alternatively, SPS can generate a new keypair, and use its private key for the server-side authentication, or use agent-forwarding, and authenticate the client with its own key.

CAUTION:

If SPS generates the private key for the server-side authentication, then the public part of the keypair must be imported to the server, otherwise the authentication will fail. Alternatively, SPS can upload the public key (or a generated X.509 certificate) into an LDAP database.

The gateway authentication process

When gateway authentication is required for a connection, the user must authenticate on One Identity Safeguard for Privileged Sessions (SPS) as well.

This additional authentication can be performed:

- *Out-of-band*: in a protocol-independent way, on the web interface of SPS.

  That way the connections can be authenticated to the central authentication database (for example, LDAP or RADIUS), even if the protocol itself does not support authentication databases. Also, connections using general usernames (for example, `root`, `Administrator`, and so on) can be connected to real user accounts.

- *Inband*: when the protocol allows it, using the incoming connection itself for communication with the authentication database.

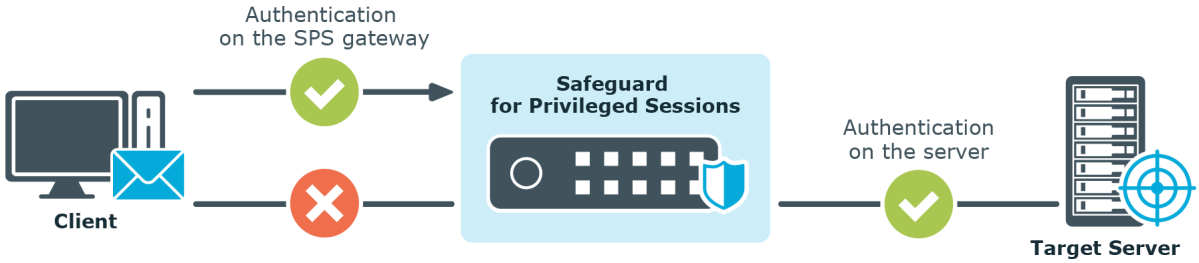
  It is the SSH, RDP, and Telnet protocols that allow gateway authentication to be performed also inband, without having to access the SPS web interface.

  For SSH and Telnet connections, inband gateway authentication must be performed when client-side authentication is configured. For details on configuring client-side authentication, see Client-side authentication settings on page 554.

  For RDP connections, inband gateway authentication must be performed when SPS is acting as a Remote Desktop Gateway (or RD Gateway). In this case, the client authenticates to the Domain Controller or a local user database. For details, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on page 531.

  In the case of RDP connections, inband gateway authentication can also be performed if an AA plugin is configured.

Figure 15: Gateway authentication



Technically, the process of gateway authentication is the following:

1. The user initiates a connection from a client.
2. If gateway authentication is required for the connection, SPS pauses the connection.
3. *Out-of-band authentication:*

   The user logs in to the SPS web interface, selects the connection from the list of paused connections, and enables it. It is possible to require that the authenticated session and the web session originate from the same client IP address.

   Inband authentication:

   SPS requests the username and optionally the credentials for gateway authentication. The user logs in to the SPS gateway.
4. The user performs the authentication on the server.

   NOTE:

   Gateway authentication can be used together with other advanced authentication and authorization techniques like four-eyes authorization, client- and server-side authentication, and so on.

Four-eyes authorization

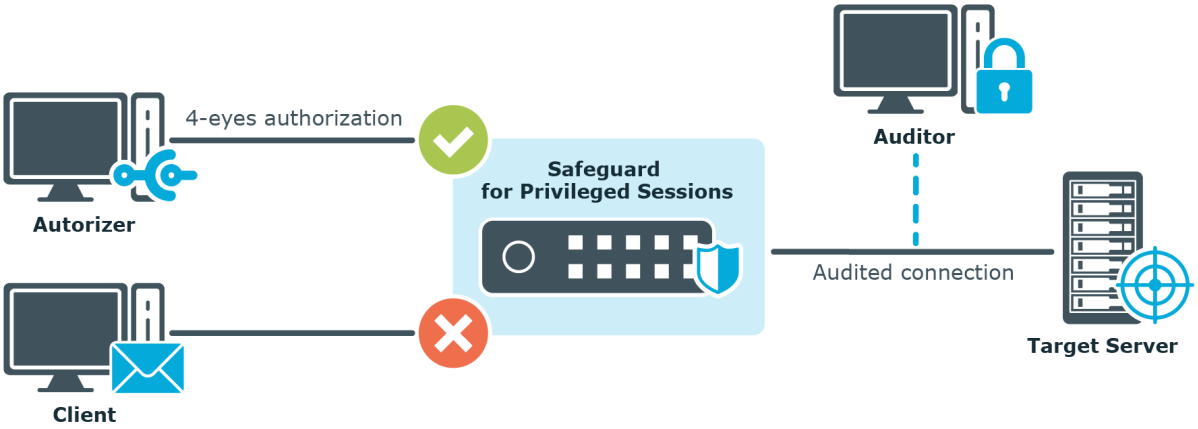
When four-eyes authorization is required for a connection, a user (called authorizer) must authorize the connection on One Identity Safeguard for Privileged Sessions (SPS) as well. This authorization is in addition to any authentication or group membership requirements needed for the user to access the remote server. Any connection can use four-eyes authorization, so it provides a protocol-independent, out-of-band authorization and monitoring method.

The authorizer has the possibility to terminate the connection any time, and also to monitor real-time the events of the authorized connections: SPS can stream the traffic to the Safeguard Desktop Player application, where the authorizer (or a separate auditor) can watch exactly what the user does on the server, just like watching a movie.

NOTE:

The auditor can only see the events if the required decryption keys are available on the host running the Safeguard Desktop Player application.

Figure 16: Four-eyes authorization



Technically, the process of four-eyes authorization is the following:

NOTE:

Four-eyes authorization can be used together with other advanced authentication and authorization techniques like gateway authentication , client- and server-side authentication, and so on.

1. The user initiates a connection from a client.
2. If four-eyes authorization is required for the connection, SPS pauses the connection.
3. The authorizer logs in to the SPS web interface, selects the connection from the list of paused connections, and enables it.
4. The user performs the authentication on the server.
5. The auditor (who can be the authorizer, but it is possible to separate the roles) watches the actions of the user real-time.

Network interfaces

The One Identity Safeguard for Privileged Sessions (SPS) hardware has five network interfaces: three physical interfaces for handling traffic, the HA interface for communicating with other nodes in a High Availability cluster, and the IPMI. The T10 hardware has two additional network interfaces available: the SFP+ interfaces labeled A and B. For details on hardware installation, see "One Identity Safeguard for Privileged Sessions Hardware Installation Guide" in the Installation Guide.

You can assign any number of logical interfaces (alias IP addresses and netmasks) to a physical interface, and each logical interface can have its own VLAN ID. For more information on managing logical interfaces, see Managing logical interfaces on page 120.

The routing rules determine which interface is used for transferring remote backups and syslog messages of SPS.

TIP:

One Identity recommends that you direct backups, syslog and SNMP messages, and e-mail alerts to a dedicated interface. For details, see Configuring the routing table on page 123.

The HA interface is an interface reserved for communication between the nodes of SPS clusters. The HA interface uses the Ethernet connector labeled as 4 (or HA). For details on high availability, see High Availability support in One Identity Safeguard for Privileged Sessions (SPS) on page 67.

In case of T10 hardware, the SFP+ interfaces are available for both proxy traffic and for local services. This means that these interfaces can be used for the same purposes as the other 3 physical interfaces.

The Intelligent Platform Management Interface (IPMI) interface allows system administrators to monitor system health and to manage SPS events remotely. IPMI operates independently of the operating system of SPS.

High Availability support in One Identity Safeguard for Privileged Sessions (SPS)

High Availability clusters (also called HA clusters) can stretch across long distances, such as nodes across buildings, cities or even continents. The goal of HA clusters is to support enterprise business continuity by providing location-independent load balancing and failover.

In High Availability (HA) mode, two One Identity Safeguard for Privileged Sessions (SPS) units with identical configurations are operating simultaneously. These two units are the primary node and the secondary node (previously also referred to as the master node and the slave node). The primary node shares all data with the secondary node, and if the primary node stops functioning, the other one becomes immediately active, so the servers are continuously accessible.

You can find more information on managing a high availability SPS cluster in Managing a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster on page 368.

One Identity recommends using a High Availability SPS cluster instead of a standalone SPS appliance. A standalone SPS appliance can become a single point of failure (SPOF), and its failure can severely impact your business.

Firmware and high availability

When powering on the One Identity Safeguard for Privileged Sessions (SPS) nodes in High Availability mode, both nodes boot and start the firmware. There is a difference, however, between the two nodes in the services that they start on booting. The secondary node will launch only a few services, those that are required for High Availability support (that is, for

awareness of the primary node and data synchronization). The rest of the services (for example, managing connections) start only on the primary node.

Upgrading the SPS firmware via the web interface automatically upgrades the firmware on both nodes.

Versions and releases of One Identity Safeguard for Privileged Sessions (SPS)

The following release policy applies to One Identity Safeguard for Privileged Sessions (SPS):

- *Long Term Supported or LTS releases* (for example, SPS 6.0) are supported for 3 years after their original publication date and for 1 year after the next LTS release is published (whichever date is later). The second digit of the revisions of such releases is 0 (for example, SPS 6.0.1). Maintenance releases to LTS releases contain only bugfixes and security updates.
- *Feature releases* (for example, SPS 6.1) are supported for 6 months after their original publication date and for 2 months after a succeeding Feature or LTS release is published (whichever date is later). Feature releases contain enhancements and new features, presumably 1-3 new features per release. Only the last feature release is supported (for example, when a new feature release comes out, the last one becomes unsupported in 2 months).

For a full description of long-term-supported and feature releases, open the SPS product page on the Support Portal and navigate to **Product Life Cycle & Policies > Product Support Policies > Software Product Support Lifecycle Policy**.

CAUTION:

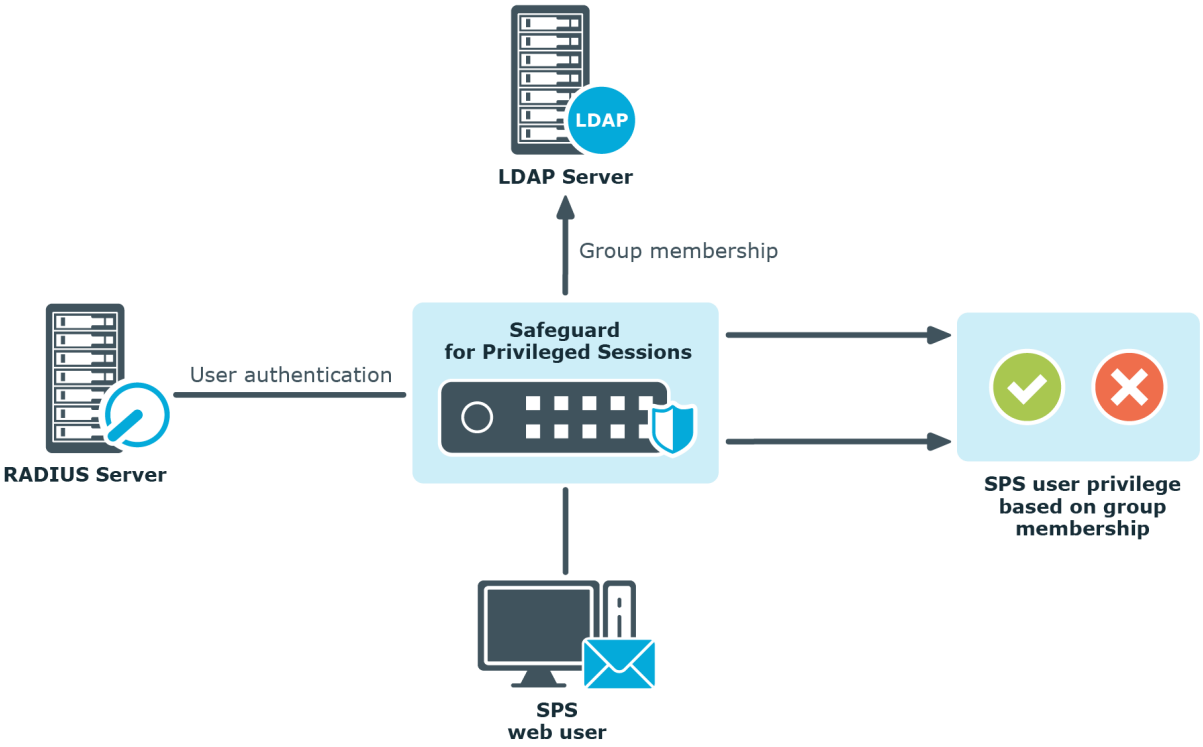
Downgrading from a feature release is not supported. If you upgrade from an LTS release (for example, 6.0) to a feature release (6.1), you have to keep upgrading with each new feature release until the next LTS version (in this case, 7.0) is published.

Accessing and configuring One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) has a web interface and is configured from a browser. The users of SPS can be authenticated using local, LDAP, or RADIUS databases. The privileges of users are determined by group memberships that can be managed either locally on SPS, or centrally in an LDAP database. Assigning privileges to groups is based on Access Control Lists (ACLs). It is also possible to match groups existing in the LDAP database to a set of SPS privileges. Access control in SPS is very detailed, it is

possible to define exactly who can access which parts of the interface and of the stored data.

Figure 17: Authenticating the users of SPS



Cloud deployment considerations

Safeguard for Privileged Sessions (SPS) can be run from the cloud.

Before you start: platforms and resources

When setting up a virtual environment, carefully consider the configuration aspects such as CPU, memory availability, I/O subsystem, and network infrastructure to ensure the virtual layer has the necessary resources available. For more information on environment virtualization, see One Identity's Product Support Policies.

Platforms that have been tested with the cloud deployments are:

- AWS Virtual Machine (VM): For more information, see For more information, see AWS deployment on page 70..
- Azure Virtual Machine (VM): For more information, see Using Azure.

AWS deployment

Safeguard for Privileged Sessions (SPS) can be run in the cloud using Amazon Web Services (AWS).

To deploy the Amazon Machine Image (AMI) of SPS from AWS, visit the AWS marketplace listing for Safeguard for Privileged Sessions (here) and follow the Deployment steps .

Limitations

- If High Availability (HA) operation mode is required in a virtual environment, use the HA function provided by the virtual environment.
- When running SPS in a virtual environment, use a single network interface.
- During AWS installation, connecting directly to the Internet using a public IP address is not supported. Instead, you must access the Internet via a Virtual Private Network or a jump host.
- Factory reset is not an option for virtual appliances. To factory reset a virtual appliance, just redeploy the appliance.

Disk size considerations

CAUTION: Before making any changes to the disk size, shut down the VM (stopped and deallocated).

SPS deploys with a minimal OS disk size. You should increase the size of the OS disk based on your estimated usage and budget.

Disk size can be handled through Amazon Elastic Compute Cloud (Amazon EC2). For more information, see Getting Started with Amazon EC2. When you start up the VM, SPS automatically resizes the OS disk volume to use the available space.

AWS security considerations

Running SPS in AWS comes with some security considerations that do not apply to the hardware appliance. One Identity recommends:

- Do not give Safeguard a public IP address.
- Use the AWS key vault to encrypt the disk.
- Limit access within AWS to the Safeguard virtual machine. SPS in AWS cannot protect against rogue Administrators in the same way the hardware appliance can.

Static IP address required

Configure the SPS VM with a static IP address in AWS. In AWS, the IP address must not change after the VM is deployed. If you need to change the IP address, take a backup, deploy again, and restore the backup. You can script the VM deploy to pick up an existing virtual NIC with the IP address configuration. For details, see the Amazon Virtual Private Cloud (VPC) documentation.

Deployment steps

AWS automatically licenses the operating system during the deployment with an AWS KMS.

AWS Marketplace steps

1. Go to the AWS marketplace listing for SPS (here).
2. On the One Identity Safeguard for Privileged Sessions page, click **Continue to Subscribe**.
3. Advance through the resource creation screens to configure your instance. In addition to the Disk size considerations, AWS security considerations, and Static IP address required, One Identity recommends you select the **m4.2xlarge** instance type.
4. Once you have finished configuring the instance, select to launch the instance.

   NOTE: The instance launch process may take a while to complete.
5. Once the instance has finished launching, log into the web client using your static IP address. Change the admin password immediately.

   NOTE: Ensure that you upgrade SPS to the latest available release.

Azure deployment

Purpose:

The following describes how to have a One Identity Safeguard for Privileged Sessions running in Microsoft Azure.

To have a One Identity Safeguard for Privileged Sessions running in Microsoft Azure

1. **Deploy One Identity Safeguard for Privileged Sessions from the Microsoft Azure Marketplace**

   Create and configure a One Identity Safeguard for Privileged Sessions virtual machine (VM) in the Azure portal. For details, see the Microsoft Azure documentation, here we just describe the SPS-specific settings.

   a. Login to the Azure portal, select **One Identity Safeguard for Privileged Sessions** from the Azure Marketplace, then click **Create**.

   b. Fill the required fields of the **Basics** blade. Note that you must fill the **User name** and **Authentication Password/SSH public key** fields, but SPS will not actually use these settings (SPS will use the parameters you configure in the SPS Welcome Wizard).

   c. Choose a size for the VM. If you want to use this machine in production and need help about sizing or architecture design, contact your One Identity sales representative.

      The number of interfaces you can use depends on the size of your Azure VM. If your VM allows you to use multiple interfaces, you can configure multiple interfaces in SPS. For details, see VM with multiple NICs.

   d. On the **Settings** blade, disable monitoring.

   e. When the deployment is finished, navigate to the network settings of the new VM in the Azure portal. Change the IP address of the SPS network interface to Static, and note down the IP address and the hostname (you will need it in the SPS Welcome Wizard).

   f. If you want to backup or archive data from SPS into Azure, create an Azure File Share. Note down the following information of the file share, because you will need it to configure SPS backups and archiving: `URL`, `Username`, `Password`.

      CAUTION:

      If you have multiple SPS VMs, make sure to use a separate file share for each SPS.

2. **Complete the SPS Welcome Wizard**

   Complete the SPS Welcome Wizard (for details, see "Configuring One Identity Safeguard for Privileged Sessions (SPS) with the Welcome Wizard" in the Administration Guide). Note the following points specific for Azure deployments. When configuring the network settings of SPS note the following points.

   CAUTION:

   Do not export or import configuration between a physical SPS deployment and a virtual one. Because of the differences and limitations between physical and virtual appliances, configure the virtual appliance from scratch to ensure proper functionality. When you migrate a virtual SPS to another one, you can export and import the configuration.

   a. Into the **Physical interface EXT or 1 — IP address** field, enter the static IP address of the SPS VM that you set on the Azure portal.

   b. **Default GW**: The default gateway is usually the first address in a subnet (for example, if your subnet is `10.7.0.0/24`, then the gateway will be `10.7.0.1`).

   c. **Hostname**: Use the hostname you have configured for the SPS VM on the Azure portal.

   d. **DNS server**: You can use any DNS server that the SPS VM can access, even public ones.

3. **Configure SPS**

   Login to SPS and configure it.

   a. Configure backups for SPS. For backup and archiving purposes One Identity recommends the built-in file shares of Azure. For details on configuring backups, see "Data and configuration backups" in the Administration Guide.

   b. Configure archiving for SPS. For backup and archiving purposes One Identity recommends the built-in file shares of Azure. For details on configuring backups, see "Archiving and cleanup" in the Administration Guide. Configuring Archiving policy is highly recommended: because if the disk of the VM fills up, SPS stops working.

   c. Configure a server: set up a host that is on the same subnet as SPS, and enable Remote Desktop (RDP) or Secure Shell (SSH) access to it.

   d. Configure a connection on SPS to forward the incoming RDP or Secure Shell (SSH) connection to the host and establish a connection to the host. See "Logging in to One Identity Safeguard for Privileged Sessions (SPS) and configuring the first connection" in the Administration Guide for details.

   e. Replay your session in the browser. See "Replaying audit trails in your browser in Search (classic)" in the Administration Guide for details.

In case you have questions about SPS, or need assistance, contact your One Identity representative.

Limitations

The following limitations apply to SPS when you deploy it from the Microsoft Azure Marketplace.

CAUTION:

Do not export or import configuration between a physical SPS deployment and a virtual one. Because of the differences and limitations between physical and virtual appliances, configure the virtual appliance from scratch to ensure proper functionality. When you migrate a virtual SPS to another one, you can export and import the configuration.

- Root login is not available on the console.
- SSH access is only available after you have completed the Welcome Wizard.
- Currently, the data that is entered during the provisioning phase (for example, the username and the IP address) of creating the virtual machine in Azure is not transferred to SPS. Therefore, only the data entered in the Welcome Wizard will be used.
- By default, you can only use Physical interface 1 (eth0) of SPS, with a single IP address. Aside from changing the IP address of SPS, do not modify other interface-related settings (additional logical interfaces, IP forwarding, and so on) on the **Basic Settings > Network** page of SPS.

  The number of interfaces you can use depends on the size of your Azure VM. If your VM allows you to use multiple interfaces, you can configure multiple interfaces in SPS. For details, see VM with multiple NICs.
- The **Seal the box** functionality is not available.
- The High Availability support of SPS was designed to work between two physical SPS appliances. This feature is not available in Azure environments. For further details, see the High Availability and redundancy in Microsoft Azure.
- Due to Azure requirements, an additional 5-minute delay has been added to the boot process. This ensures that the root device appears in the system.
- The size of the hard disk in Azure is 100 Gb. You cannot extend this virtual disk size later, nor can you write to Samba or other disks. In case you run out of disk space, either configure a **Backup policy** and an **Archive policy** if you have a server for this purpose, or configure a **Cleanup policy** that deletes the audit trails at certain time intervals. For details, see "Data and configuration backups" in the Administration Guide and "Archiving and cleanup" in the Administration Guide.
- SPS currently cannot receive its IP address using DHCP. Make sure that:

- The IP address you have configured in Azure and the IP address that you configure for SPS for the **Physical interface 1** on the Networking settings part of the Welcome Wizard are the same. Otherwise, you will not be able to access SPS.
- You set the internal IP static on the Network Interfaces tab of the Virtual Machine.
- Do not assign a public IP address to SPS, use SPS as a component of your internal infrastructure. If you absolutely must configure Welcome Wizard from a publicly accessible IP address, note that SPS will be publicly accessible. If you assign a public IP to the web management interface, consider the following:
  - Select a complex passphrase.
  - Limit access to the management interface based on the source IP address, and make sure that brute-force protection for the administrator web login is enabled (they are enabled by default). For details, see "Configuring user and administrator login addresses" in the Administration Guide.
  - Configure an email alert or SNMP trap for administrator logon events. For details, see "Configuring e-mail alerts" in the Administration Guide and "Configuring SNMP alerts" in the Administration Guide.
  - Forward the logs of SPS to a log server (for example, to a syslog-ng server, or an syslog-ng Store Box appliance) so that if the local logs are compromised, you still have an authentic copy of the original logs.
  - For security reasons, disable SSH access to SPS when it is not needed. Accessing the SPS host directly using SSH is not recommended or supported, except for troubleshooting purposes. If you enable SSH access, restrict the clients that can access SPS based on their source IP address, and make sure that brute-force protection is enabled (they are enabled by default). For details, see "Enabling SSH access to the One Identity Safeguard for Privileged Sessions (SPS) host" in the Administration Guide.
  - To prevent unauthorized access to the audit trail files recorded on SPS, configure proper access control rules for the user groups and encrypt every audit trail. If you use encryption, store your keys in the personal or in the temporary key store. For details, see "Encrypting audit trails" in the Administration Guide,
- Upgrading SPS in Azure is the same as upgrading a physical appliance: you have to upload the firmware on the SPS web interface. For detailed instructions, see Upgrade Guide.

Prerequisites

The following prerequisites must be met to deploy SPS in Microsoft Azure:

- You have a valid One Identity Safeguard for Privileged Sessions license. When deployed from the Microsoft Azure Marketplace, the One Identity Safeguard for Privileged Sessions uses the "Bring your own license" model. Note that to deploy two active SPS nodes as an availability set, you must purchase two standalone SPS licenses. To purchase a license, contact our Sales Team.
- Microsoft recommends to use the Azure Resource Manager (ARM) deployment model. When you install SPS from the Azure Marketplace, SPS supports only this deployment method. If you need to deploy SPS into and infrastructure that uses the Classic deployment model, contact your One Identity sales representative.
- You have a Microsoft Azure account.

High Availability and redundancy in Microsoft Azure

In a Microsoft Azure deployment, the high-availability and redundancy of the SPS appliance is provided by the Microsoft Azure infrastructure, according to the Azure Storage SLA.

Redundancy

The data in your Microsoft Azure storage account is always replicated to ensure durability and high availability, meeting the Azure Storage SLA. The exact type of replication depends on your storage account settings, but every disk is stored in 3 copies.

For details, see Locally redundant storage in the *Azure Storage replication* document, and Service Healing - Auto-recovery of Virtual Machines.

High Availability

If a hardware failure occurs, Azure moves the Virtual Machine to another location and restarts it in 5-15 minutes. In case you require higher SLA, you are recommended to deploy two standalone SPS nodes into an availability set. Note that to deploy two active SPS nodes as an availability set, you must purchase two standalone SPS licenses.

For details, see Locally redundant storage in the *Azure Storage replication* document, and Service Healing - Auto-recovery of Virtual Machines.

4

The Welcome Wizard and the first login

This section describes the initial steps of configuring One Identity Safeguard for Privileged Sessions (SPS). Before completing the steps in The initial connection to One Identity Safeguard for Privileged Sessions (SPS), unpack, assemble, and power on the hardware. Connect interface 1 (labelled 1 or EXT) to the local network, or directly to the computer from which SPS will be configured.

NOTE:

For details on unpacking and assembling the hardware, see "One Identity Safeguard for Privileged Sessions Hardware Installation Guide" in the Installation Guide. For details on how to create a high availability SPS cluster, see "Installing two SPS units in HA mode" in the Installation Guide. For more information about the supported browsers, see Supported web browsers and operating systems.

The initial connection to One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) can be connected from a client machine using web browsers and accessed from the local network.

NOTE:

For details on supported browsers, see Supported web browsers and operating systems.

Starting with version 3.1, SPS attempts to receive an IP address automatically via DHCP. If it fails to obtain an automatic IP address, it starts listening for HTTPS connections on the `192.168.1.1` IP address. Note that certain switch configurations and security settings can interfere with SPS receiving an IP address via DHCP. SPS accepts connections via its interface 1 (labelled *1* or *EXT*). For details on the network interfaces, see Network interfaces on page 66).

TIP:

The SPS console displays the IP address on which interface 1 is listening.

If SPS is listening on the `192.168.1.1` address, note that the `192.168.1.0/24` subnet must be accessible from the client. If the client machine is in a different subnet (for example, its IP address is `192.168.10.X`), but in the same network segment, the easiest way is to assign an alias IP address to the client machine. Creating an alias IP on the client machine virtually puts both the client and SPS into the same subnet, so that they can communicate. To create an alias IP complete the following steps.

- For details on creating an alias IP on Microsoft Windows, see Creating an alias IP address (Microsoft Windows) on page 78.
- For details on creating an alias IP on Linux, see Creating an alias IP address (Linux) on page 84.
- If configuring an alias interface is not an option for some reason, you can modify the IP address of SPS. For details, see Modifying the IP address of One Identity Safeguard for Privileged Sessions (SPS) on page 85.

CAUTION:

The Welcome Wizard can be accessed only using interface 1, as the other network interfaces are not configured yet.

Creating an alias IP address (Microsoft Windows)

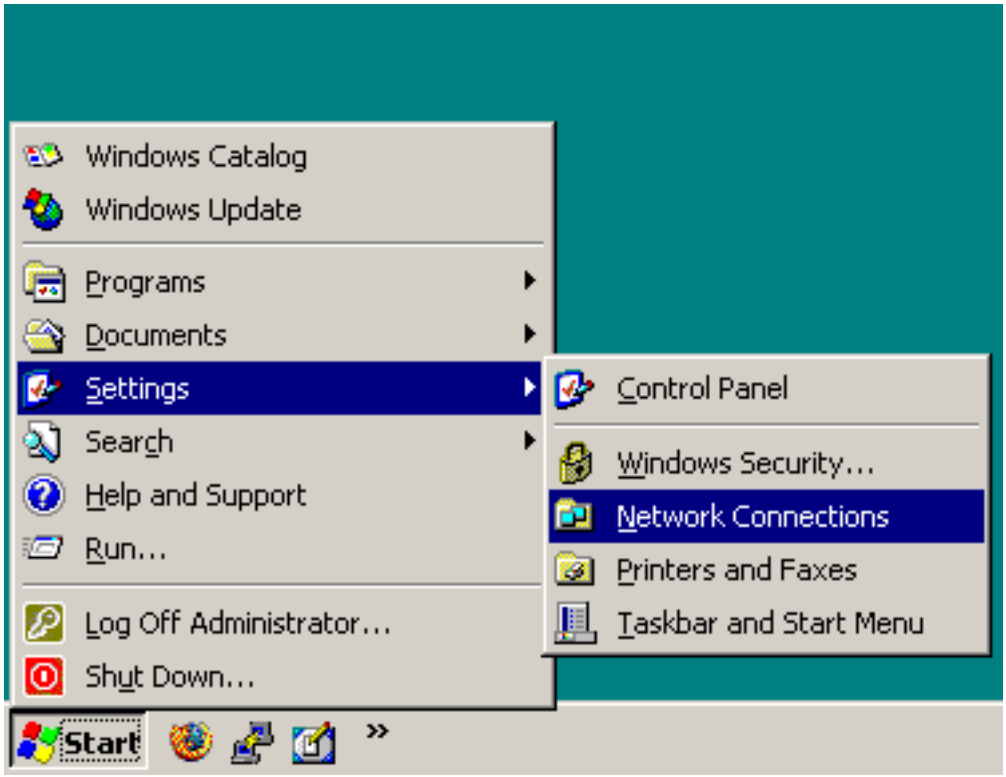
Purpose

This procedure describes how to assign an alias IP address to a network interface on Microsoft Windows platforms.

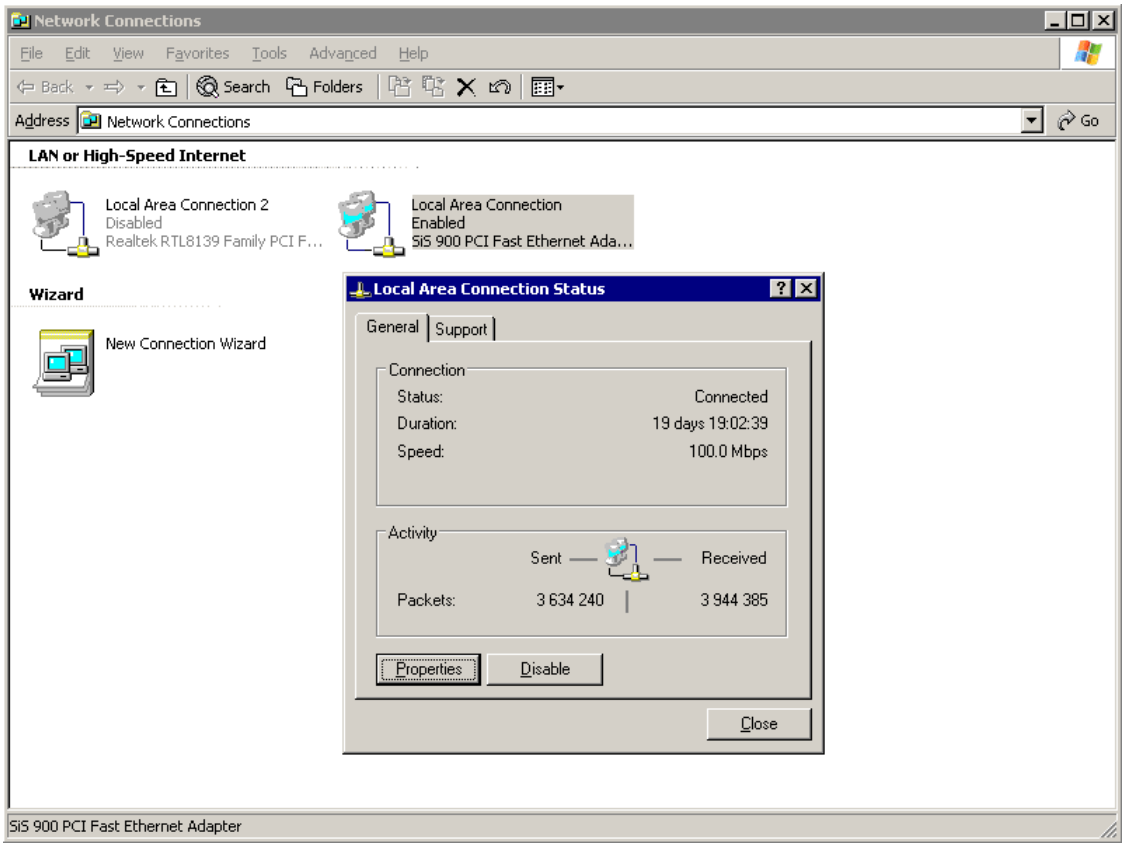
Steps

To assign an alias IP address to a network interface on Microsoft Windows platforms

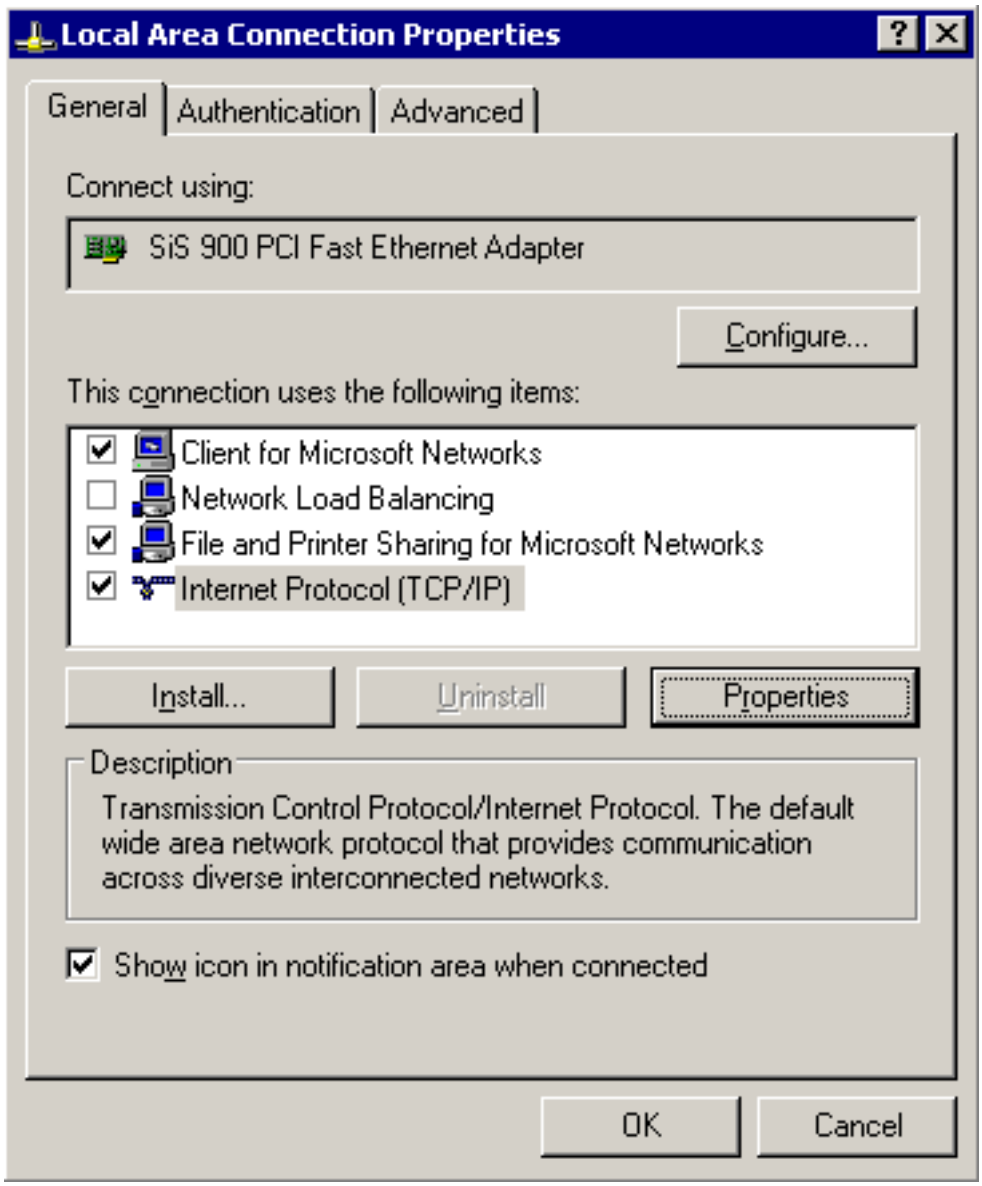
1. Navigate to **Start menu > Settings > Network Connections**.



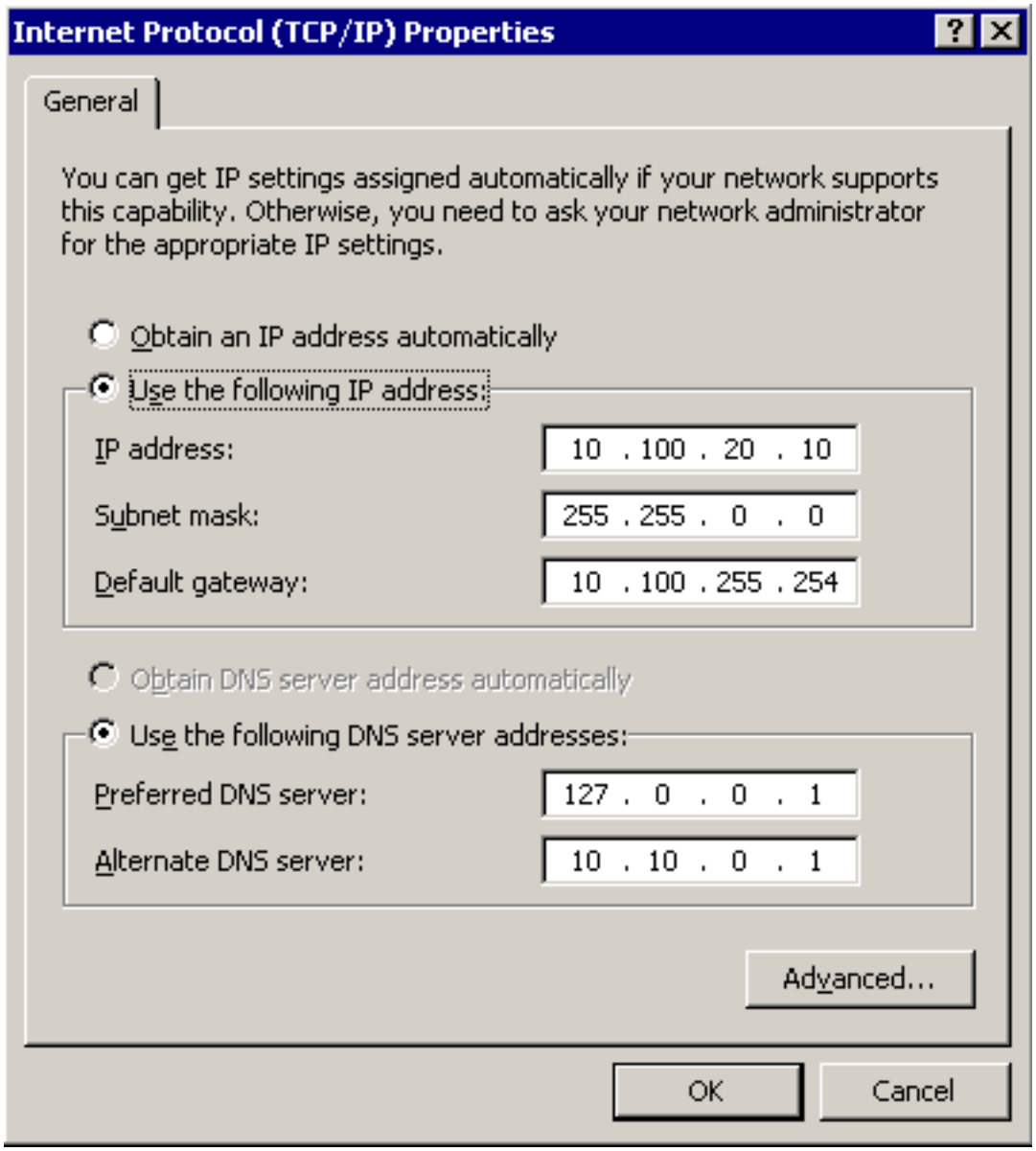
2. Double-click **Local Area Connection** and then click **Properties**.



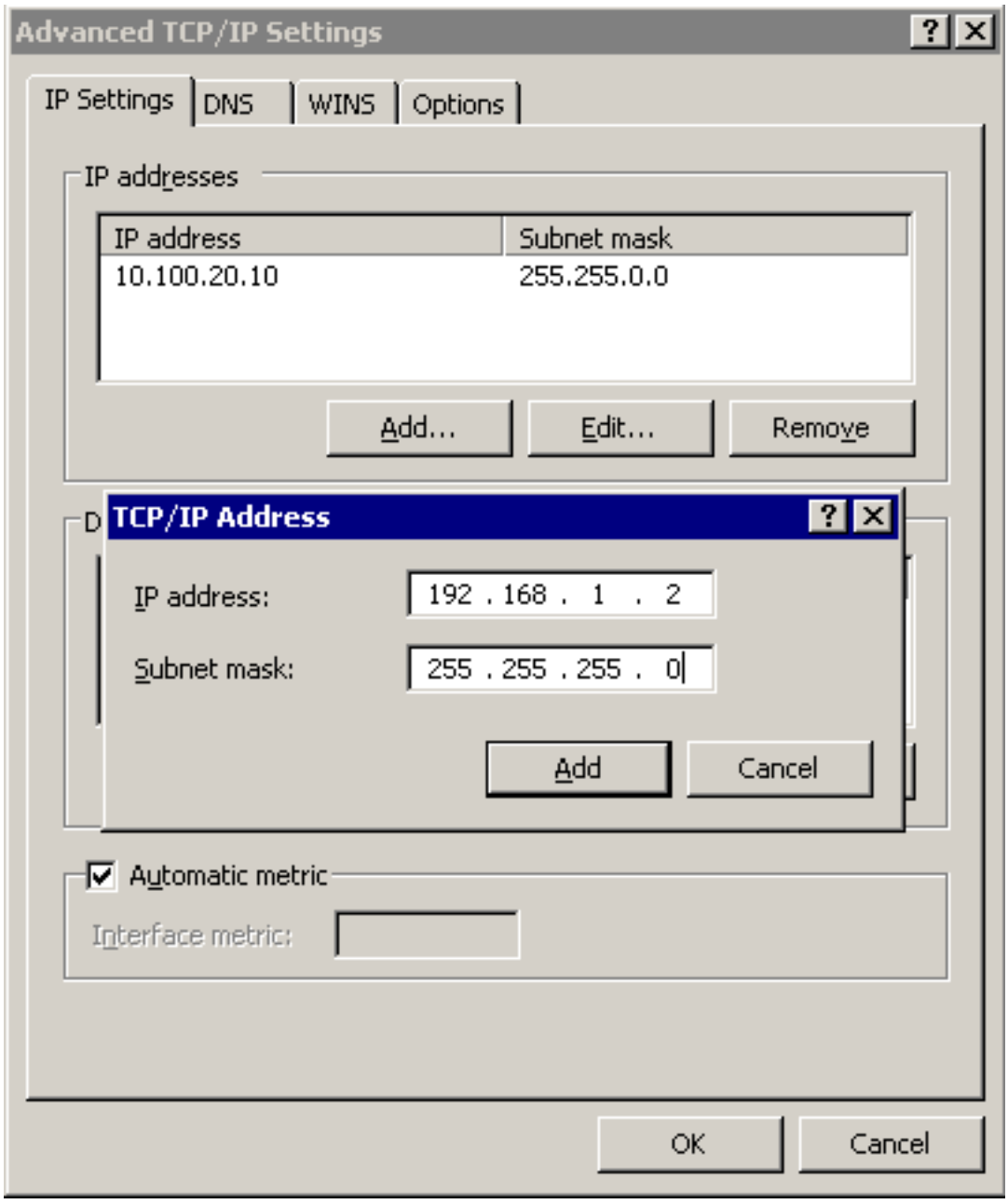
3. Select the **Internet Protocol (TCP/IP)** component in the list and click **Properties**.



4. To display the **Advanced TCP/IP Settings** window, click **Advanced**.



5. Select the **IP Settings** tab, and click **Add** in the **IP Addresses** section.



6. Enter `192.168.1.2` into the **IP Address** field, and enter `255.255.255.0`. into the **Netmask** field.

> **CAUTION:**
>
> **If your internal network uses the `192.168.1.0/24` IP range, it is possible that the `192.168.1.1` and `192.168.1.2` addresses are already in use. In this case, disconnect One Identity Safeguard for Privileged Sessions (SPS) from the network, and connect a computer directly to interface 1 (labelled 1 or EXT) using a standard cross-link cable.**

7. To complete the procedure, click **Add**.

Creating an alias IP address (Linux)

Purpose

The following describes how to assign an alias IP address to a network interface on Linux platforms.

Steps

To assign an alias IP address to a network interface on Linux platforms

1. Start a terminal console (for example, **gnome-terminal**, **konsole**, **xterm**, and so on).
2. Issue the following command as root:

   ```
   ifconfig <ethX>:0 192.168.1.2
   ```

   where `<ethX>` is the ID of the network interface of the client, usually `eth0` or `eth1`.
3. Issue the **ifconfig** command. The `<ethX>:0` interface appears in the output, having `inet addr:192.168.1.2`.
4. Issue the **ping -c 3 192.168.1.1** command to verify that One Identity Safeguard for Privileged Sessions (SPS) is accessible. A similar result is displayed:

   ```
   user@computer:~$ ping -c 3 192.168.1.1
   PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
   64 bytes from 192.168.1.1: icmp-seq=1 ttl=63 time=0.357 ms
   64 bytes from 192.168.1.1: icmp-seq=2 ttl=63 time=0.306 ms
   64 bytes from 192.168.1.1: icmp-seq=3 ttl=63 time=0.314 ms

   --- 192.168.1.1 ping statistics ---
   3 packets transmitted, 3 received, 0% packet loss, time 2013ms
   rtt min/avg/max/mdev = 0.306/0.325/0.357/0.030 ms
   ```
5. Open the page `https://192.168.1.1` from your browser and accept the certificate shown. The Welcome Wizard of SPS appears.

Modifying the IP address of One Identity Safeguard for Privileged Sessions (SPS)

Purpose

The following describes how to configure One Identity Safeguard for Privileged Sessions (SPS) to listen for connections on a custom IP address.

CAUTION:

Use this procedure only before the initial configuration of SPS, that is, before completing the Welcome Wizard. For details on changing the IP address or other network settings of a configured SPS system, see Network settings on page 114.

If you change the IP address of SPS, make sure that you use this address as the Physical interface 1 — IP address in the Networking settings section of the Welcome Wizard (see Configuring interface 1).

Steps

To configure SPS to listen for connections on a custom IP address

1. Access SPS from the local console, and log in with username `root` and password `default`.
2. Select **Shells > Core shell** in the Console Menu.
3. Change the IP address of SPS:

   ifconfig eth0 <IP-address> netmask 255.255.255.0

   Replace `<IP-address>` with an IPv4 address suitable for your environment.
4. Set the default gateway using the following command:

   route add default gw <IP-of-default-gateway>

   Replace `<IP-of-default-gateway>` with the IP address of the default gateway.
5. Type **exit**, then select **Logout** from the Console Menu.
6. Open the page *https://<IP-address-you-set-for-SPS>* from your browser and accept the certificate shown. The Welcome Wizard of SPS appears.

Accessing the Welcome Wizard from a non-standard interface

The following describes how to complete the Welcome Wizard on an interface other than **Physical interface 1**.

CAUTION:

Use this procedure only before the initial configuration of One Identity Safeguard for Privileged Sessions (SPS), that is, before completing the Welcome Wizard. For details on changing the IP address or other network settings of a configured SPS system, see Network settings on page 114.

If you change the network configuration of SPS to complete the Welcome Wizard on a non-standard interface, DO NOT use the IP address you use to access the Welcome Wizard as the Physical interface 1 — IP address in the Networking settings section of the Welcome Wizard (see Configuring interface 1). Otherwise, you will not be able to access SPS after the Welcome Wizard is completed.

To complete the Welcome Wizard on an interface other than Physical interface 1

1. Access SPS from the local console, and log in with username `root` and password `default`.
2. Select **Shells > Core shell** in the Console Menu.
3. Change the IP address of SPS:

   ifconfig eth0 <IP-address> netmask 255.255.255.0

   Replace `<IP-address>` with an IPv4 address suitable for your environment.
4. Set the default gateway using the following command:

   route add default gw <IP-of-default-gateway>

   Replace `<IP-of-default-gateway>` with the IP address of the default gateway.
5. Type **exit**, then select **Logout** from the Console Menu.
6. Open the page *https://<IP-address-you-set-for-SPS>* from your browser and accept the certificate shown. The Welcome Wizard of SPS appears.

Configuring One Identity Safeguard for Privileged Sessions (SPS) with the Welcome Wizard

Purpose

The Welcome Wizard guides you through the basic configuration steps of One Identity Safeguard for Privileged Sessions (SPS). All parameters can be modified before the last step by using the **Back** button of the wizard, or later via the web interface of SPS.

Steps

To configure SPS with the Welcome Wizard

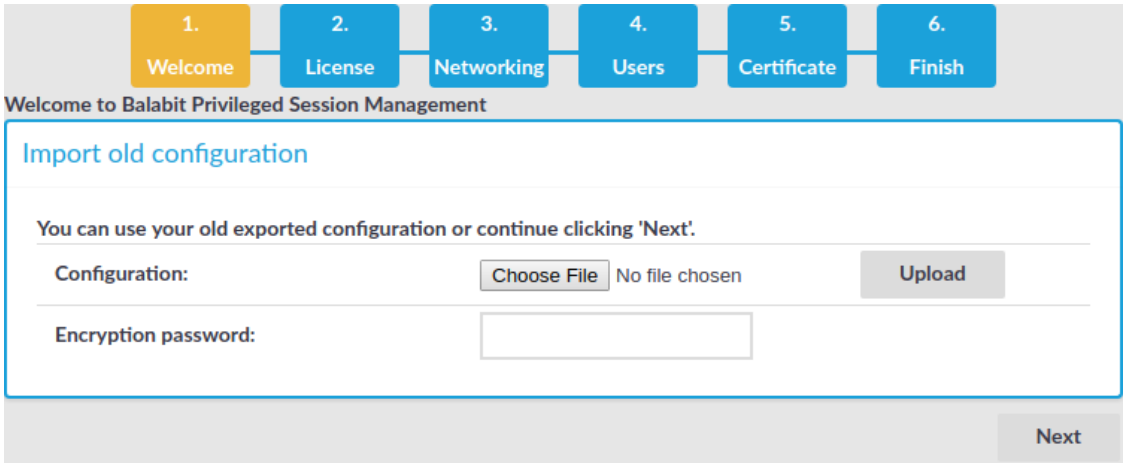
1. Open the *https://<IP-address-of-SPS-interface>* page in your browser and accept the displayed certificate. The Welcome Wizard of SPS appears.

   TIP:

   The SPS console displays the IP address the interface is listening on. SPS either receives an IP address automatically via DHCP, or if a DHCP server is not available, listens on the `192.168.1.1` IP address.

2. When configuring SPS for the first time, click **Next**.

Figure 18: The Welcome Wizard



You can import an existing configuration from a backup file. Use this feature to restore a backup configuration after a recovery, or to migrate an existing SPS configuration to a new device.

CAUTION:

Do not export or import configuration between a physical SPS deployment and a virtual one. Because of the differences and limitations between physical and virtual appliances, configure the virtual appliance from scratch to ensure proper functionality. When you migrate a virtual SPS to another one, you can export and import the configuration.

a. Click **Browse** and select the configuration file to import.

> NOTE:
>
> It is not possible to directly import a GPG-encrypted configuration into SPS, it has to be decrypted locally first.

b. Enter the passphrase used when the configuration was exported into the **Encryption passphrase** field.

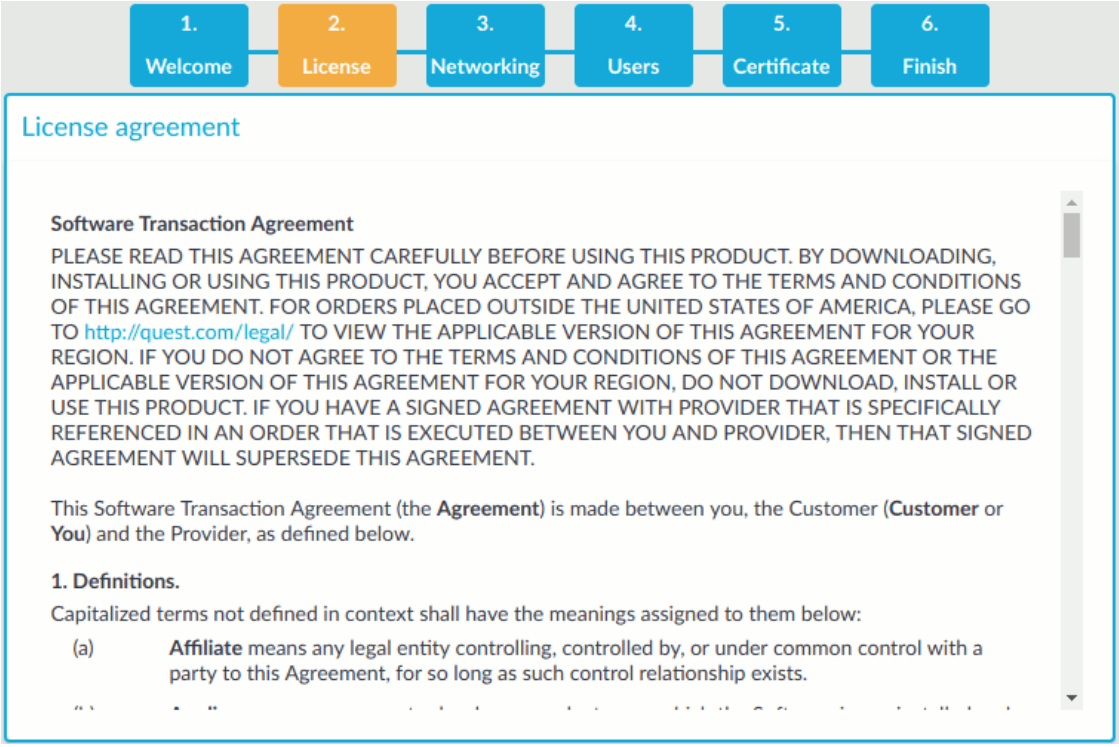
   For details on restoring configuration from a configuration backup, see Restoring One Identity Safeguard for Privileged Sessions (SPS) configuration and data on page 883

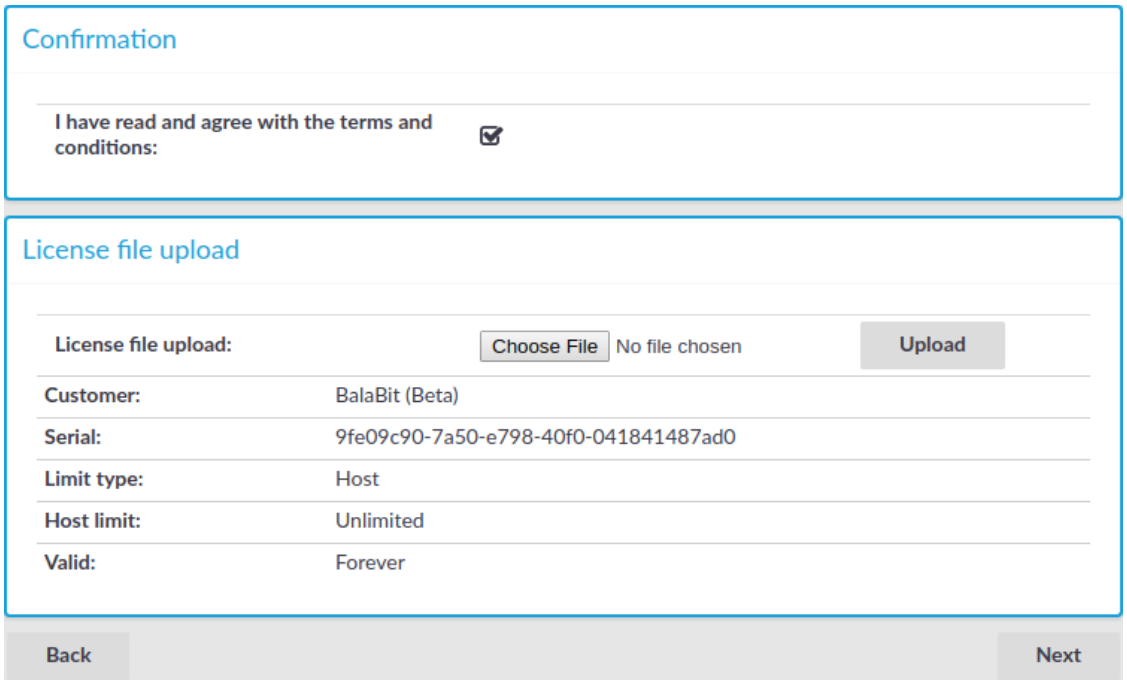
c. Click **Import**.

> **CAUTION:**
>
> **If you use the Import function to copy a configuration from one SPS to another, do not forget to configure the IP addresses of the second SPS. Having two devices with identical IP addresses on the same network leads to errors.**

3. Accept the End User License Agreement and install the SPS license.

Figure 19: The EULA and the license key





a. Read the End User License Agreement and select **I have read and agree with the terms and conditions**. The License Agreement covers both the traditional license, and subscription-based licensing as well. Clicking **I have read and agree with the terms and conditions** means that you accept the agreement that corresponds to the license you purchased. After the installation is complete, you can read the End User License Agreement at **Basic Settings > System > License**.

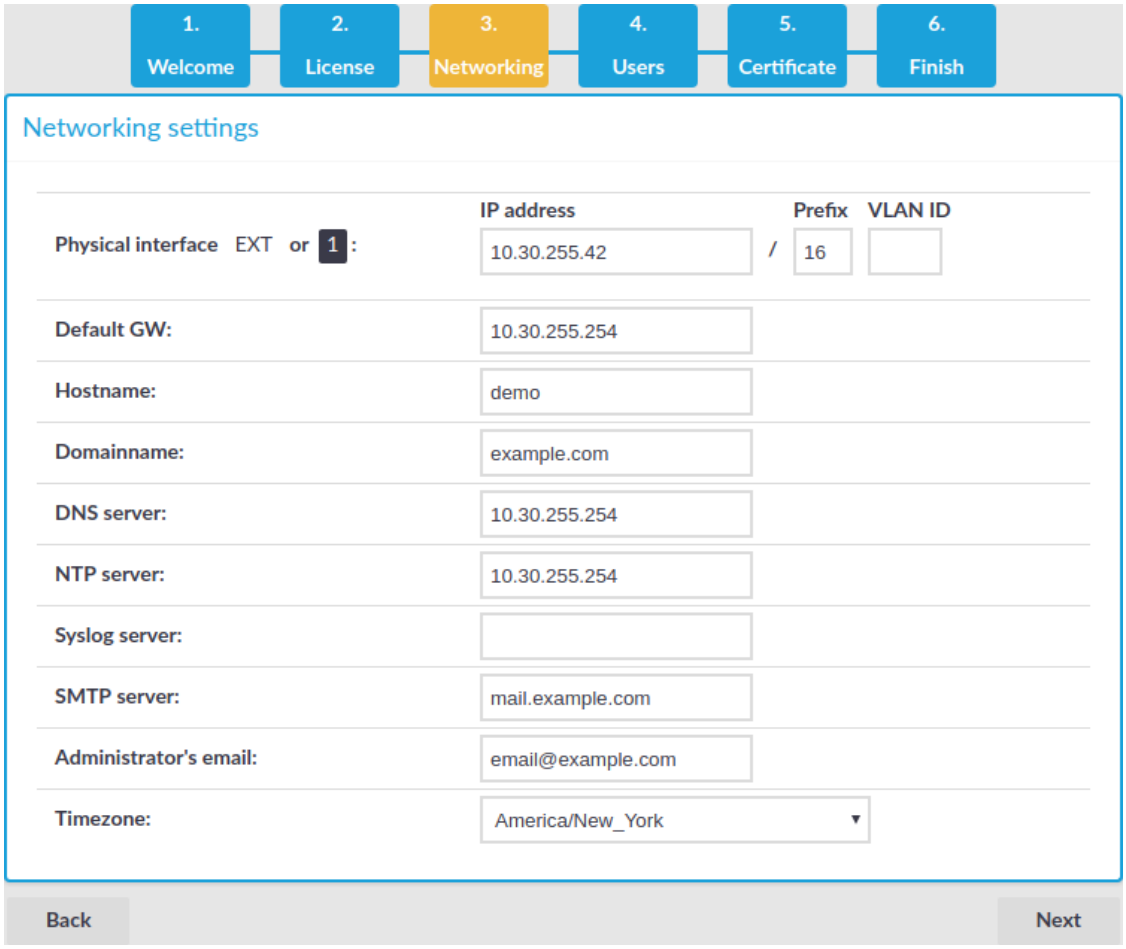
b. Click **Browse**, select the SPS license file received with SPS, then click **Upload**.

> NOTE:
>
> It is not required to manually decompress the license file. Compressed licenses (for example `.zip` archives) can also be uploaded.

c. Click **Next**.

4. Configure networking. All settings can be modified later using the web interface of SPS.

Figure 20: Initial networking configuration



a. **Physical interface EXT or 1 — IP address**: The IP address of interface 1 (or EXT, for older hardware) of SPS (for example, `192.168.1.1`). The IP address can be chosen from the range of the corresponding physical subnet. Clients will connect to this interface, therefore it must be accessible to them.

   Use an IPv4 address.

   > NOTE:
   >
   > Do not use IP addresses that fall into the following ranges:
   >
   > - `1.2.0.0/16` (reserved for communication between SPS cluster nodes)
   > - `127.0.0.0/8` (localhost IP addresses)

b. **Physical interface EXT or 1 — Prefix**: The IP prefix of the given range. For example, general class C networks have the /24 prefix.

c. **Physical interface EXT or 1 — VLAN ID**: The VLAN ID of interface 1 (or EXT). Optional.

> **CAUTION:**
>
> **Do not set the VLAN ID unless your network environment is already configured to use this VLAN. Otherwise, your SPS appliance will be unavailable using this interface.**

d. **Default GW**: IP address of the default gateway.

   Use an IPv4 address.

e. **Hostname**: Name of the machine running SPS (for example, SPS).

f. **Domainname**: Name of the domain used on the network.

g. **DNS server**: The IP address of the name server used for domain name resolution.

   Use an IPv4 address.

h. **NTP server**: The IP address or the hostname of the NTP server.

   Use an IPv4 address.

i. **Syslog server**: The IP address or the hostname of the syslog server.

   Use an IPv4 address.

j. **SMTP server**: The IP address or the hostname of the SMTP server used to deliver e-mails.

   Use an IPv4 address.

k. **Administrator's email**: E-mail address of the SPS administrator.

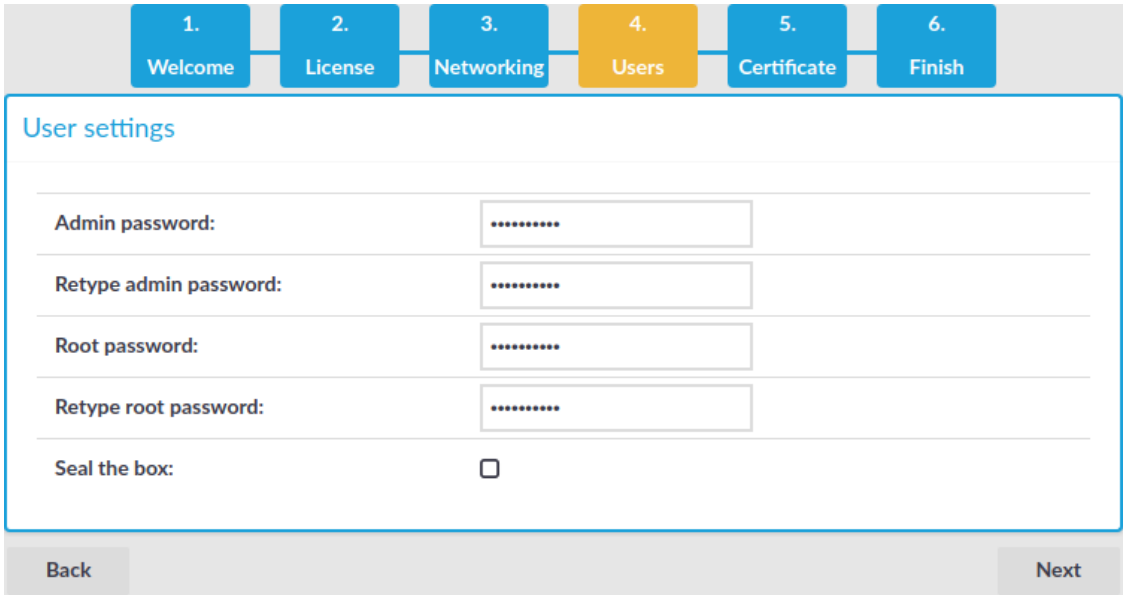
l. **Timezone**: The timezone where the SPS is located.

m. **HA address**: The IP address of the high availability (HA) interface. Leave this field on auto unless specifically requested by the support team.

n. Click **Next**.

5. Enter the passwords used to access SPS.

Figure 21: Passwords



NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

a. **Admin password**: The password of the `admin` user who can access the web interface of SPS.

b. **Root password**: The password of the `root` user, required to access SPS via SSH or from the local console.

   NOTE:

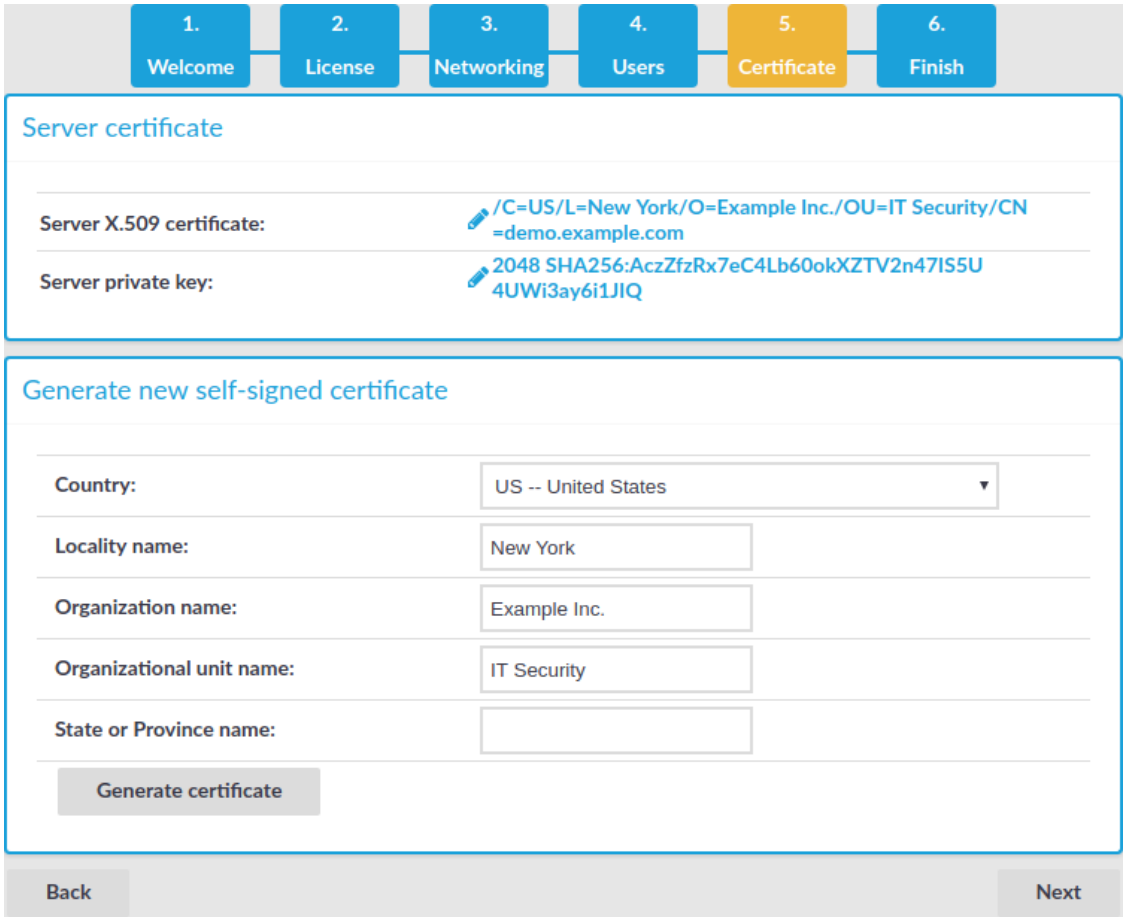
   Accessing SPS using SSH is rarely needed, and One Identity recommends it only for advanced users for troubleshooting situations.

c. If you want to prevent users from accessing SPS remotely via SSH or changing the root password of SPS, select the **Seal the box** checkbox. Sealed mode can be activated later from the web interface as well. For details, see "Sealed mode" in the Administration Guide.

d. Click **Next**.

6. Upload or create a certificate for the SPS web interface. This SSL certificate will be displayed by SPS to authenticate HTTPS connections to the web and the REST interface.

Figure 22: Creating a certificate for SPS



To create a self-signed certificate, fill the fields of the **Generate new self-signed certificate** section and click **Generate certificate**. The certificate will be self-signed by the SPS appliance. The hostname of SPS will be used as the issuer and common name.

a. **Country**: Select the country where SPS is located (for example, HU-Hungary).

b. **Locality name**: The city where SPS is located (for example, Budapest).

c. **Organization name**: The company who owns SPS (for example, Example Inc.).

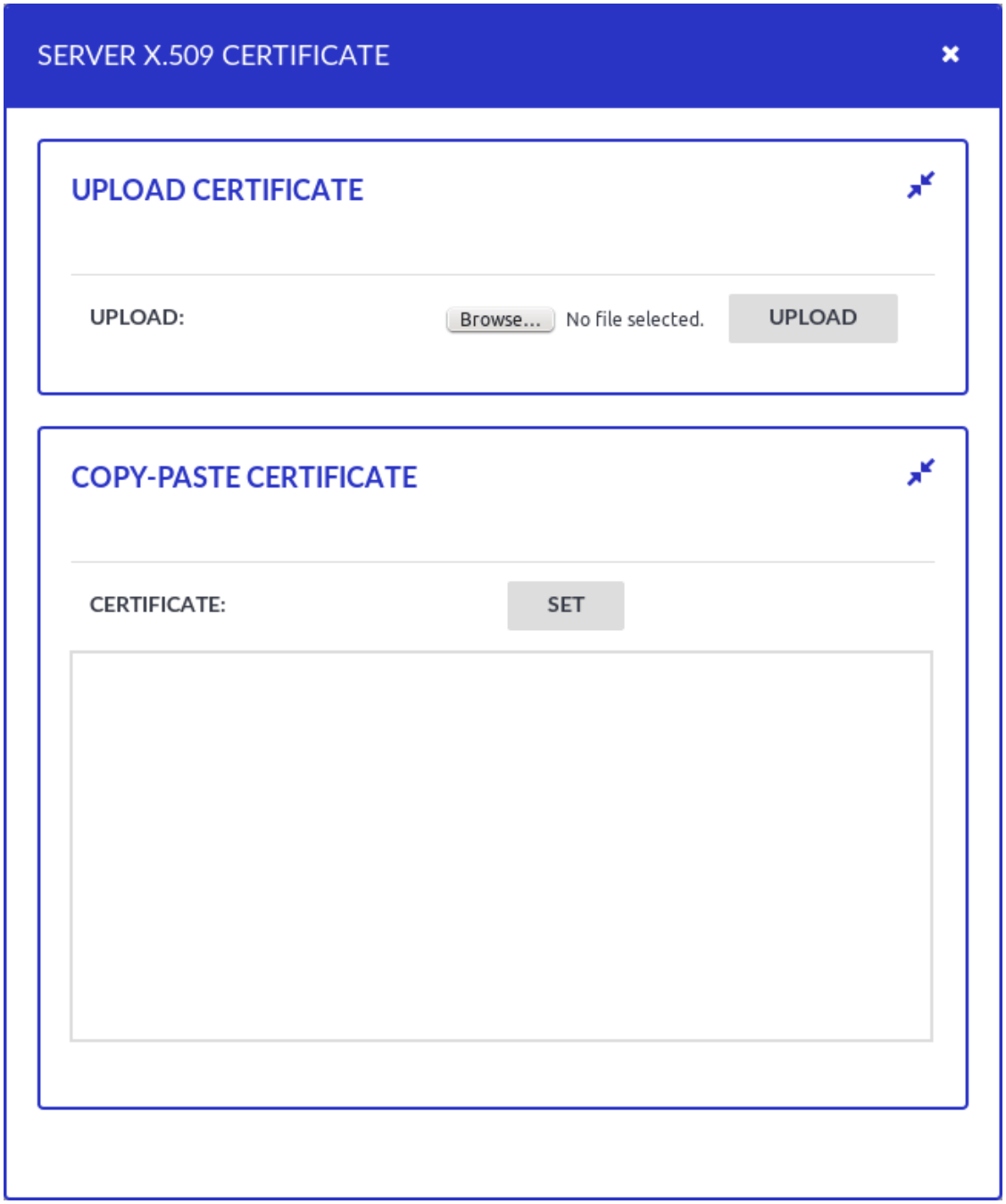
d. **Organizational unit name**: The division of the company who owns SPS (for example, IT Security Department).

e. **State or Province name**: The state or province where SPS is located.

f. Click **Generate certificate**.

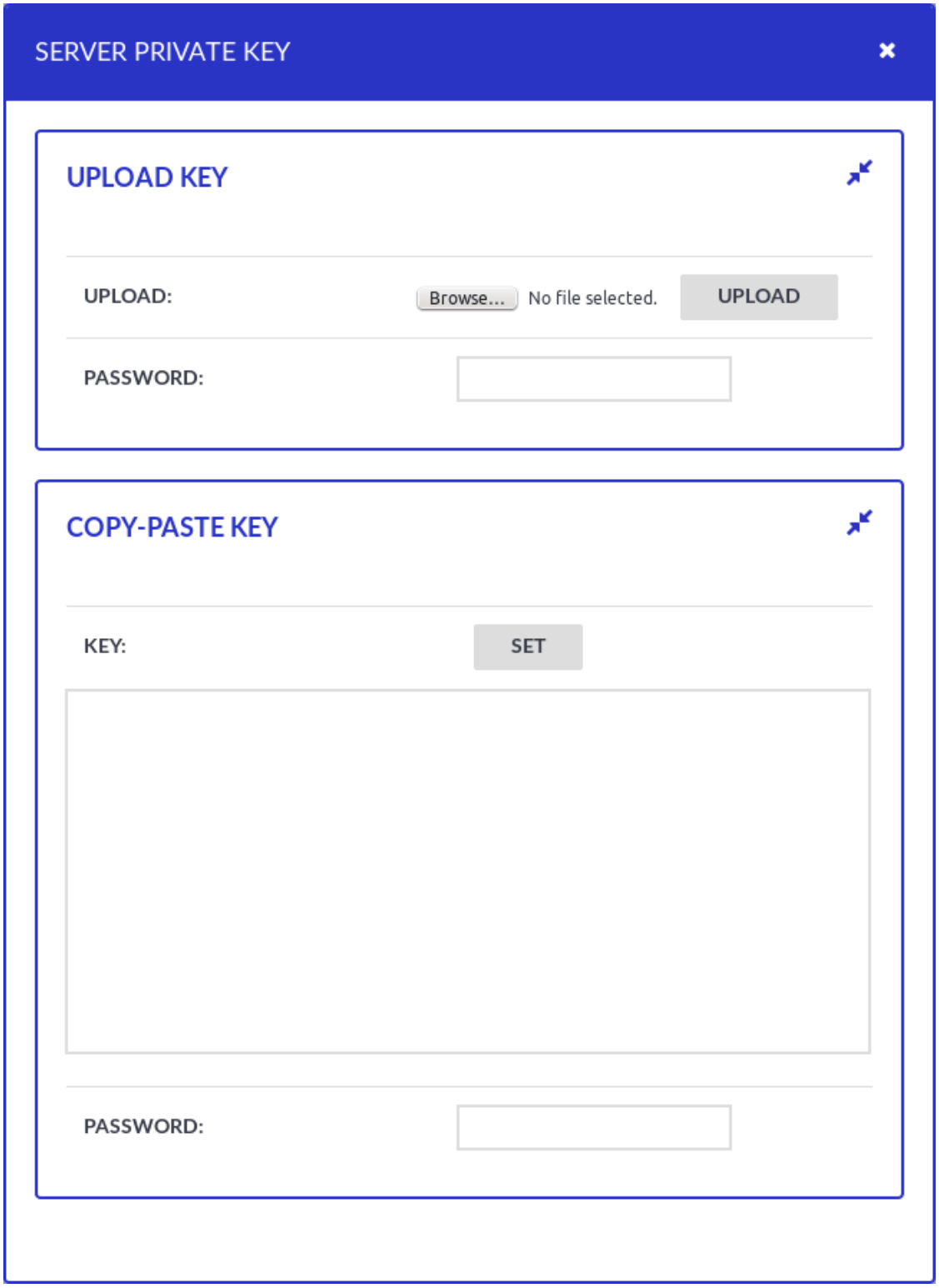
If you want to use a certificate that is signed by an external Certificate Authority, in the **Server X.509 certificate** field, click to upload the certificate.

Figure 23: Uploading a certificate for SPS



Then in the **Server private key** field click , upload the private key, and enter the password protecting the private key.

Figure 24: Uploading a private key



NOTE:

SPS accepts private keys in PEM (RSA), and PUTTY format. Password-protected private keys are also supported.

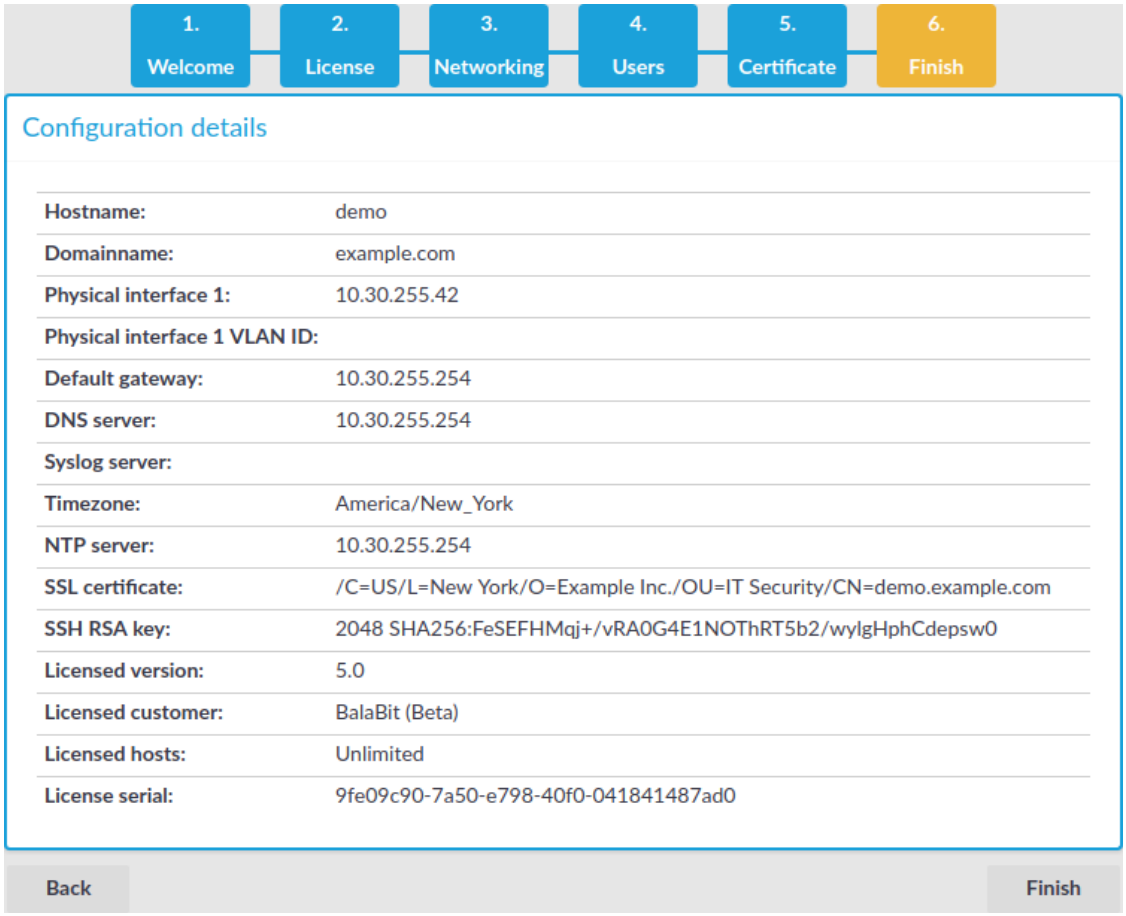
One Identity recommends using 2048-bit RSA keys (or stronger).

NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

7. Review the data entered in the previous steps. This page also displays the certificate generated in the last step, the SSH RSA key of SPS, and information about the license file.

Figure 25: Review configuration data



If all information is correct, click **Finish**.

CAUTION:

The configuration takes effect immediately after clicking Finish. Incorrect network configuration data can render SPS unaccessible.

SPS is now accessible from the regular web interface via the IP address of interface 1 (or EXT).

8. Your browser is automatically redirected to the IP address set for interface 1 (or EXT) of SPS, where you can login to the web interface of SPS using the `admin` username and the password you set for this user in the Welcome Wizard.

Logging in to One Identity Safeguard for Privileged Sessions (SPS) and configuring the first connection

Purpose

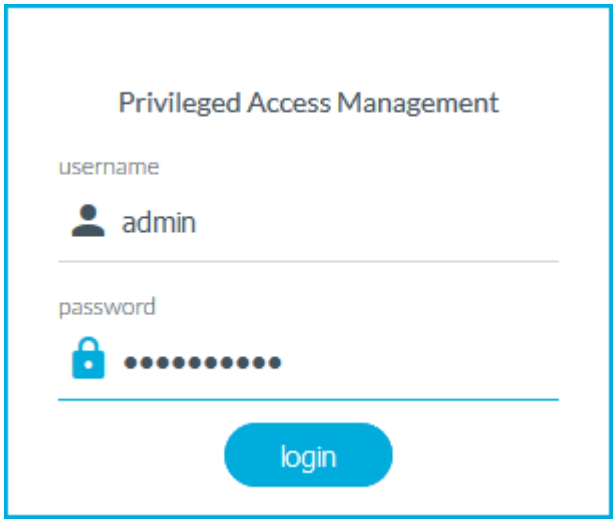
After finishing the initial configuration of One Identity Safeguard for Privileged Sessions (SPS) using the Welcome Wizard, connections must be configured between the clients and the servers. SPS inspects only the connections that are configured from the web interface, all other connections are forwarded without any inspection.

Steps

To enable a simple SSH terminal or a Remote Desktop session over a transparent and a non-transparent connection

1. Login to SPS's web interface.

   Figure 26: The first login

   a. Open the `https://IP-address-of-interface-1/` page from your browser to access the web interface of SPS. Replace the `IP-address-of-the-interface-1` string with the IP set for interface 1 in the **Networking settings** section of the Welcome Wizard (see Configuring interface 1) (for example, `192.168.1.1`).

   b. The certificate created in the **Certificate** section of the Welcome Wizard (see Creating the web interface certificate) is displayed. Accept it.

   c. Log in to the SPS web interface using the displayed login screen.

      - Enter `admin` into the **Login** field.
      - Enter the password set in the **Users** section of the Welcome Wizard (see Setting the administrator password) for the `admin` user into the **Password** field.
      - Click **Login**. The main page of the SPS administration interface is displayed.

2. Configure a new transparent connection.

   a.
      - To configure an SSH connection, select **SSH Control > Connections** from the Main Menu. Only terminal sessions will be permitted.
      - To configure an RDP connection, click on the **RDP Control > Connections** from the Main Menu. Only basic Remote Desktop sessions will be permitted (no file-sharing).

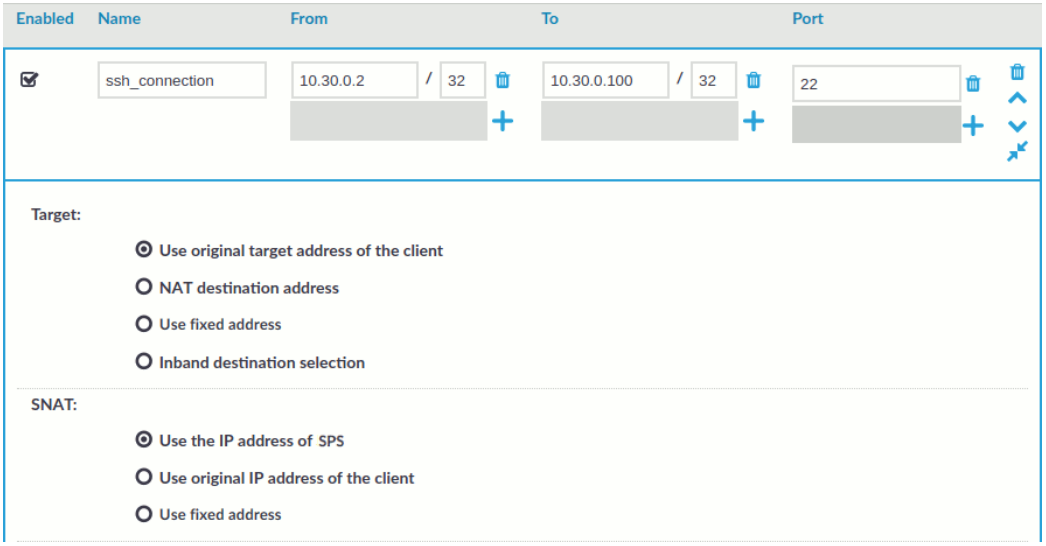
   b. Click the + icon on the right to create a new connection.

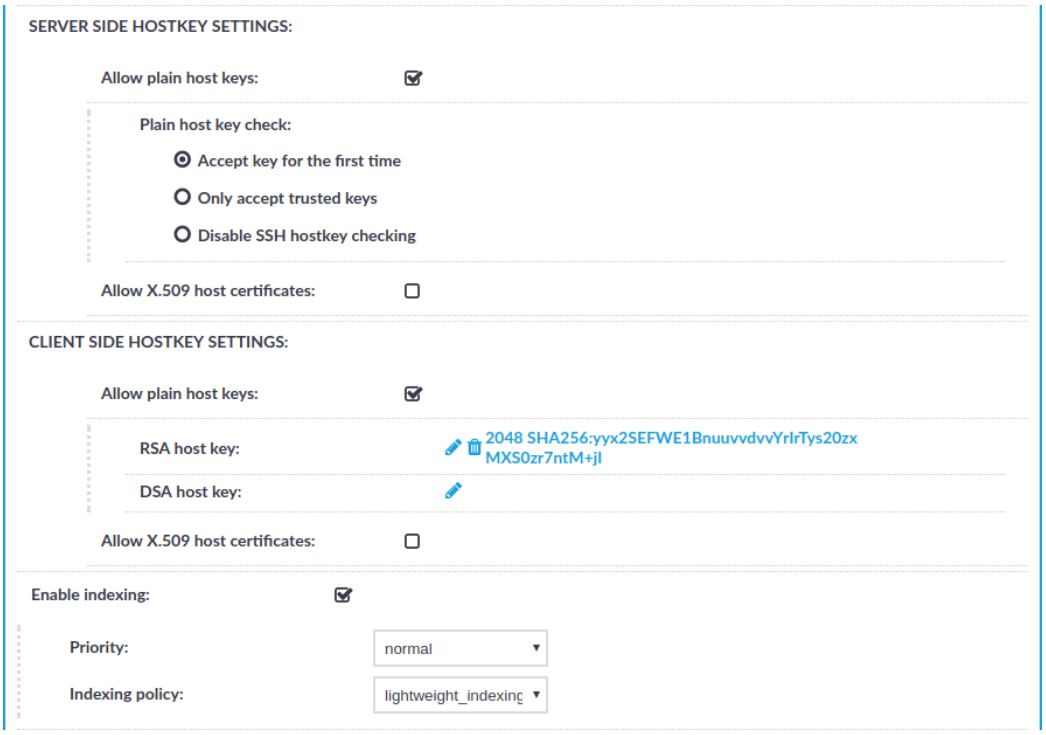
c. Enter a name into the **Name** field that will identify the connection (for example, `admin-server-transparent`).

> TIP:
>
> One Identity recommends that you use descriptive names that give information about the connection (that is, they refer to the name of the accessible server, the allowed users, and so on).

d. Enter the IP addresses defining the connection:

Figure 27: <Protocol name> Control > Connections — Configuring an SSH connection in transparent mode





- Enter the IP address of the client that will be permitted to access the server into the **From** field.

  You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).
- Enter the IP address of the server into the **To** field.

  You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).
- Enter the port number where the server is accepting connections into the **Port** field.

e. Select **Enable indexing**.

f. Click Commit.

This connection allows any user from the client machine to connect to the specified server, but permits only terminal sessions — other SSH channels like TCP forwarding are disabled.

3. Configure a new non-transparent connection.

   a.
   - To configure an SSH connection, select **SSH Control > Connections** from the Main Menu. Only terminal sessions will be permitted.
   - To configure an RDP connection, click on the **RDP Control > Connections** from the Main Menu. Only basic Remote Desktop sessions will be permitted (that is, no clipboard or file-sharing).

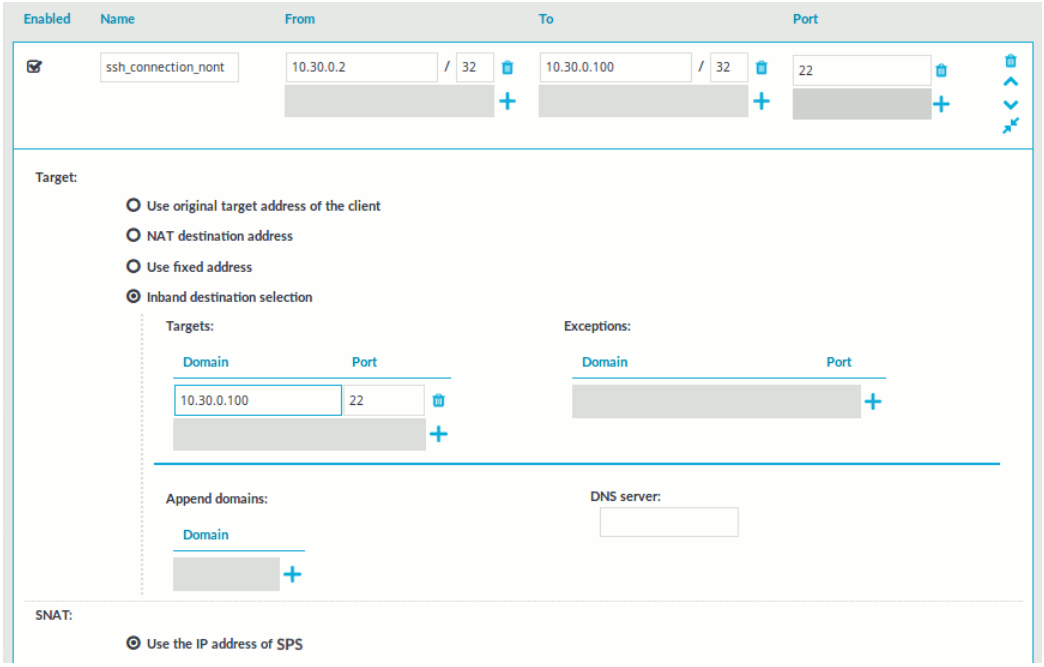
b. Click the + icon on the right to create a new connection.

c. Enter a name into the **Name** field that will identify the connection (for example, `admin-server-nontransparent`).

> TIP:
>
> One Identity recommends that you use descriptive names that give information about the connection (that is, they refer to the name of the accessible server, the allowed users, and so on).

d. Enter the IP addresses defining the connection:

Figure 28: <Protocol name> Control > Connections — Configuring an SSH connection in non-transparent mode



- Enter the IP address of the client that will be permitted to access the server into the **From** field.

  You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).

- Enter the IP address of SPS's physical interface 1 into the **To** field.

  You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).

- Enter a port number into the **Port** field.

- Enter the IP address of the server into the **Use fixed address** field of the **Target** section.

You can use an IPv4 or an IPv6 address.

- Enter the port number where the server is accepting connections into the **Port** field of the **Target** section.

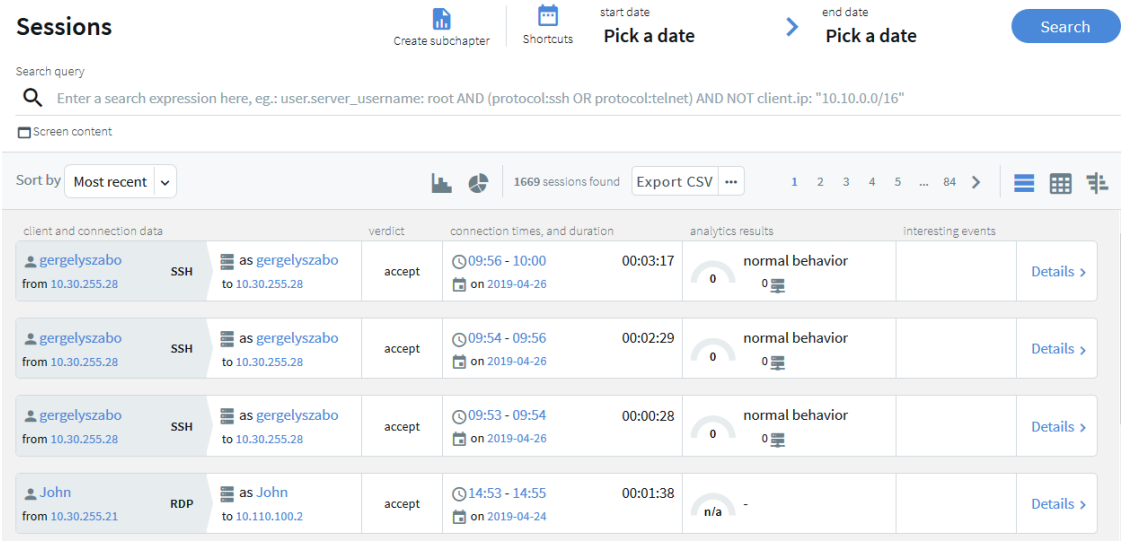
e. Select **Enable indexing**.

f. Click Commit.

This connection allows any user from the client machine to connect to the specified server, but permits only terminal sessions — other SSH channels like TCP forwarding are disabled.

4. Test the new configuration: try to initiate an SSH or and RDP connection from the client to the server.
   - For the transparent connection, use the IP address of the server (as configured in Configuring a connection in transparent mode).
   - For the non-transparent connection, use the IP address and port of SPS (as configured in Configuring a connection in non-transparent mode).
5. After successfully connecting to the server, do something in the connection, for example, execute a simple command in SSH (for example, `ls /tmp`), or launch an application in RDP (for example, the Windows Explorer), then disconnect from the server.
6. To access the Search interface, navigate to **Search**.

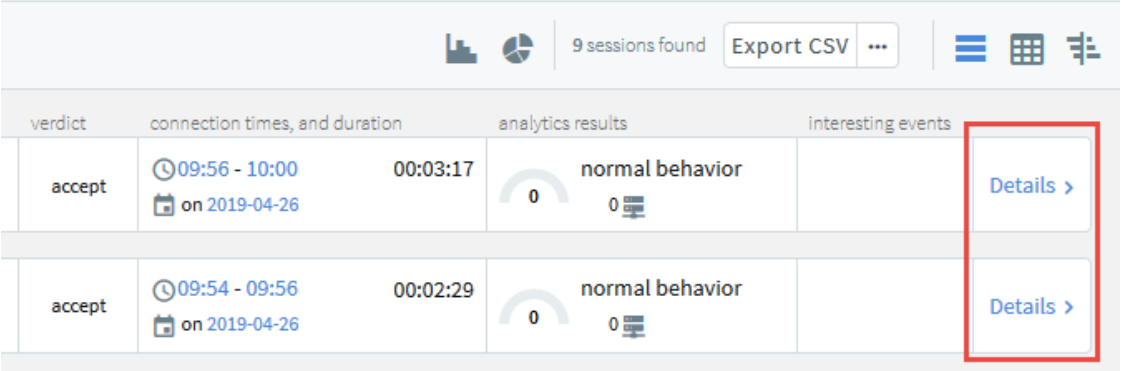
Figure 29: The Search interface



7. Find the session you want to replay on the **Search** page.

   For more information about search criteria and other search-related options, see Using the Search interface.

Figure 30: Search — Accessing session details



For more information about the **session info** window and its contents, see Viewing session details.

8. Click Details › to display the details of the connection.

   Alternatively, in the table view, click ••• .

9. Click Start rendering to generate a video file from the audit trail you want to replay. Depending on the load of the indexer and the length and type of the audit trail, this can take several minutes.

   When the video is available, Start rendering changes to Play video and Delete video . You can use the Delete video button if you want to remove the generated video. After you remove the video file, the Start rendering button is available and you can use it to recreate the video file.

10. (Optional) If you have encrypted audit trails but the necessary certificates and private keys are not uploaded into your private keystore, you have to upload the keys first. After uploading them, click Unlock events . The Unlock events feature decrypts the encrypted upstream traffic elements. As a result, they will be displayed distributed in the generated video.

11. To replay the video, click Play video .

    The Player window opens.

12. Play the audit trail, and review your actions.

For more information about audit trails, see sections Encrypting audit trails, Replaying audit trails in your browserand Replaying encrypted audit trails in your browser.

5

Basic settings

One Identity Safeguard for Privileged Sessions (SPS) is configured through the web interface. Configuration changes take effect automatically after clicking 

. Only the modifications of the current page or tab are activated — each page and tab must be committed separately.

- For details about the supported browsers, see Supported web browsers and operating systems on page 105.
- For details on how to use the web interface of SPS, see The structure of the web interface on page 106.
- For details on how to configure the network interfaces, name resolution, and other networking-related settings, see Network settings on page 114.
- For details on how to control (for example reboot) SPS, upload a new firmware or license, export the current configuration, or stop the incoming syslog traffic, see Network settings on page 114.
- For details on how to set the system time and automatic time synchronization to an NTP server, see Configuring date and time on page 124.
- For details on how to configure where SNMP and e-mail alerts are sent, see System logging, SNMP and e-mail alerts on page 126.
- For details on how to configure system monitoring and alerts, see Configuring system monitoring on SPS on page 137.
- For details on how to configure data and configuration backups, see Data and configuration backups on page 146.
- For details on how to configure archiving, see Archiving and cleanup on page 159.

Supported web browsers and operating systems

CAUTION:

Since the official support of Internet Explorer 9 and 10 ended in January, 2016, they are not supported in One Identity Safeguard for Privileged Sessions (SPS) version 4 F3 and later.

CAUTION:

Even though the One Identity Safeguard for Privileged Sessions (SPS) web interface supports Internet Explorer and Microsoft Edge in general, to replay audit trails you need to use Internet Explorer 11, and install the Google WebM Video for Microsoft Internet Explorer plugin. If you cannot install Internet Explorer 11 or another supported browser on your computer, use the the Safeguard Desktop Player application. For details, see "Replaying audit trails in your browser" in the Administration Guide and Safeguard Desktop Player User Guide.

NOTE: SPS displays a warning message if your browser is not supported or JavaScript is disabled.

NOTE: The minimum recommended screen resolution for viewing 's ('s) web interface is 1366 x 768 pixels on a 14-inch widescreen (standard 16:9 ratio) laptop screen. Screen sizes and screen resolutions that are equal to or are above these values will guarantee an optimal display of the web interface.

Supported browsers

The current version of Mozilla Firefox and Google Chrome, Microsoft Edge, and Microsoft Internet Explorer 11 or newer. The browser must support TLS-encrypted HTTPS connections, JavaScript, and cookies. Make sure that both JavaScript and cookies are enabled.

Supported operating systems

Windows 2008 Server, Windows 7, Windows 2012 Server, Windows 2012 R2 Server, Windows 8, Windows 8.1, Windows 10, Windows 2016, and Linux.

The SPS web interface can be accessed only using TLS-encryption and strong cipher algorithms.

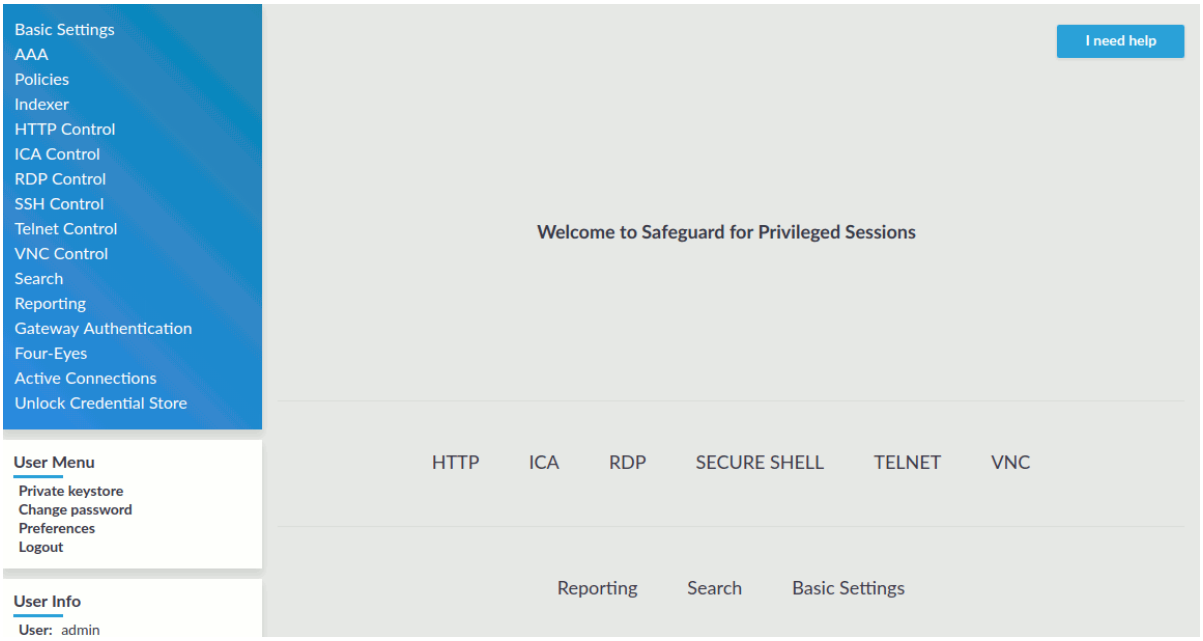
Opening the web interface in multiple browser windows or tabs is not supported.

The structure of the web interface

NOTE:

NOTE: The minimum recommended screen resolution for viewing 's ('s) web interface is 1366 x 768 pixels on a 14-inch widescreen (standard 16:9 ratio) laptop screen. Screen sizes and screen resolutions that are equal to or are above these values will guarantee an optimal display of the web interface.

Figure 31: Structure of the web interface



The web interface consists of the following main sections:

Main menu

Each menu item displays its options in the main workspace on one or more tabs. Click a **Main menu** item to display the list of tabs available under that particular menu item.

User Menu

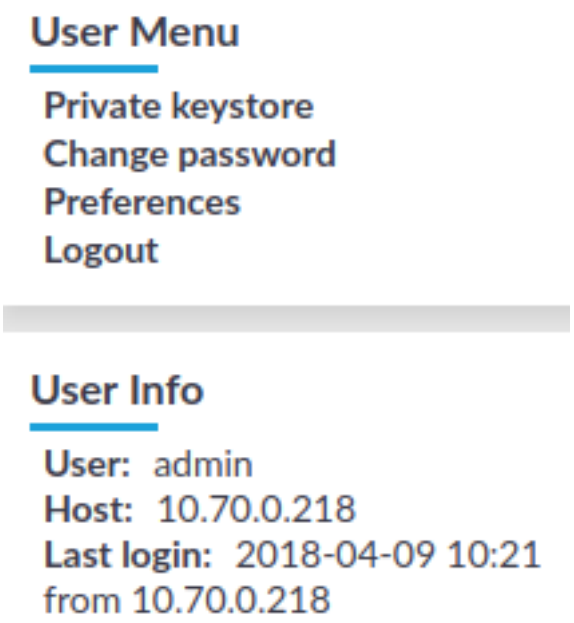
Provides possibilities to upload your security passphrase and permanent or temporary keys, to change your SPS password, to log out, and disable confirmation dialogs and tooltips using the **Preferences** option. For details, see Preferences on page 112.

User Info

Provides information about the user currently logged in:

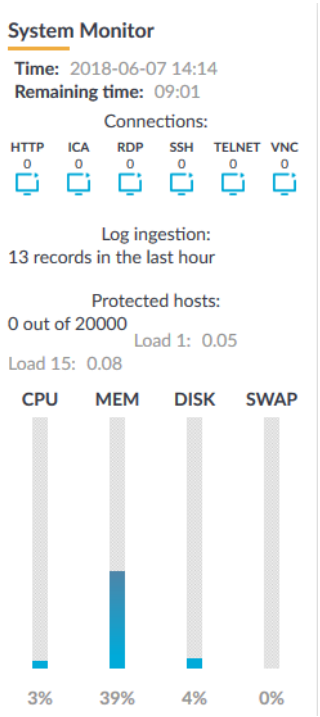
- **User**: username
- **Host**: IP address of the user's computer
- **Last login**: date and IP address of the user's last login

Figure 32: User Menu and User Info



System Monitor

Figure 33: System Monitor



Displays accessibility and system health information about SPS, including the following:

- **Time**: System date and time.
- **Remaining time**: The time remaining before the session to the web interface times out.

  NOTE:

  To change timeout settings, navigate to **Basic Settings > Management > Web interface timeout** and enter the timeout value in minutes.
- **Locked**: Indicates that the interface is locked by another administrator (for details, see Multiple users and locking on page 111).
- Indicators if **HTTP**, **ICA**, **RDP**, **SSH**, **Telnet**, and **VNC** traffic is permitted to the protected servers.
- **Connections**: The number of active **ICA**, **RDP**, **SSH**, **Telnet**, and **VNC** connections. For **HTTP**, the number of active HTTP sessions is displayed.
- **License**: License information if the license is not valid, or an evaluation version license has expired.
- The status of the RAID devices, if synchronization between the disks is in progress.
- **HA**: The HA status and the ID of the active node if two SPS units are running in a High Availability cluster. If there are redundant Heartbeat interfaces configured, their status is displayed as well. If the nodes of the cluster are synchronizing data between each other, the progress and the time remaining from the synchronization process is also displayed.
- **Protected hosts** or **Concurrent sessions**: Displays license usage, that is, the number of hosts that have been accessed through SPS in case of host-based licensing, or the number of active sessions in case of session-based licensing.
- Average system load during the
  - **Load 1**: last minute
  - **Load 15**: last fifteen minutes
- CPU, memory, hard disk, and swap use. Hover the mouse above the graphical bars to receive a more details in a tooltip, or navigate to **Basic Settings > Dashboard** for detailed reports.

The **System Monitor** displays current information about the state of SPS. To display a history of these parameters, go to **Basic Settings > Dashboard**. For details, see Status history and statistics on page 863.

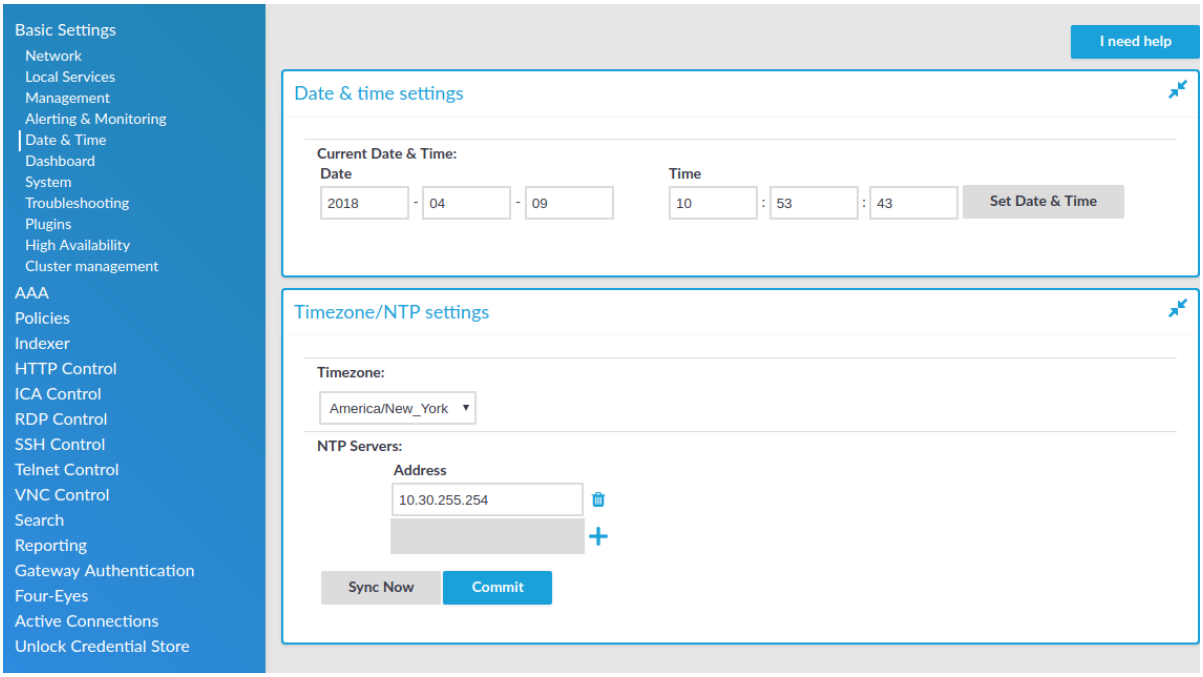
Elements of the main workspace

The main workspace displays the configuration settings related to the selected **Main menu** item grouped into one or more submenus. Related parameters of a submenu are organized into labeled groups or sections, marked with blue outline



.

Figure 34: Main workspace

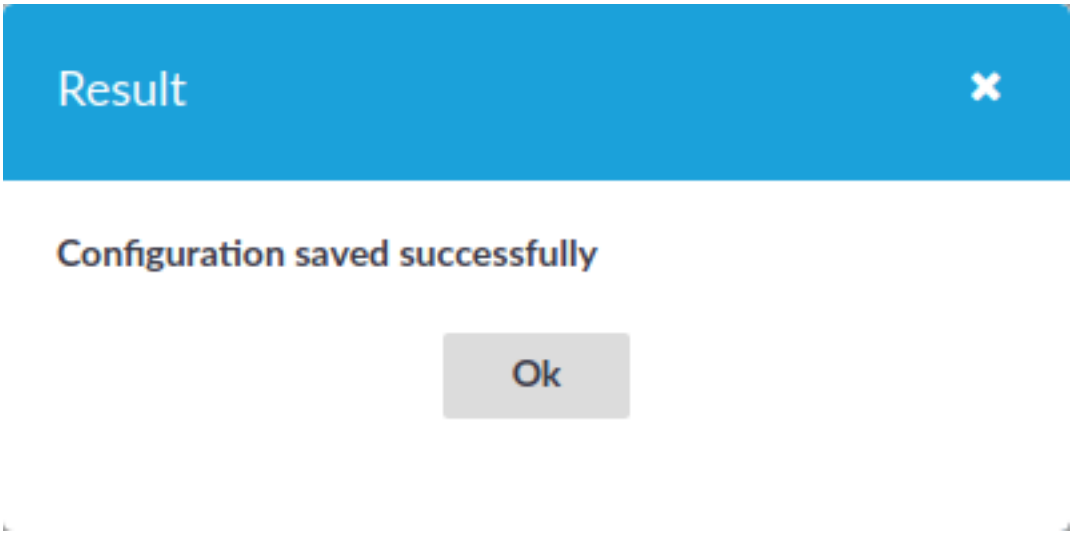


- Commit Each page includes one or more orange action buttons. The most common action button is Commit, which saves and activates the changes of the page.
- / *Show/Hide details*: Displays or hides additional configuration settings and options.
- *Create entry*: Create a new row or entry (for example an IP address or a policy).
- *Delete entry*: Delete a row or an entry (for example an IP address or a policy).
- *Modify entries or upload files*: Edit an entry (for example a host key, a list, and so on), or upload a file (for example a private key). These actions open a pop-up window where the actual modification can be performed.
- , *Position an item in a list*: Modify the order of items in a list. The order of items in a list (for example the order of connections, permitted channels in a channel policy, and so on) is important because when One Identity Safeguard for Privileged Sessions (SPS) is looking for a policy, it evaluates the list from top to down, and selects the first item completely matching the search criteria. For example, when a client initiates a connection to a protected server, SPS selects the first connection policy matching the client's IP address, the server's IP address, and the target port (the From, To, and Port fields of the connection).

Message window: This pop-up window displays the responses of SPS to the user's actions, for example **Configuration saved successfully**. Error messages are also displayed here. All messages are included in the system log. For detailed system logs (including message history), see the **Troubleshooting** tab of the **Basic Settings**. To make the window

appear only for failed actions, navigate to **User menu > Preferences** and enable the **Autoclose successful commit messages** option.

Figure 35: Message window

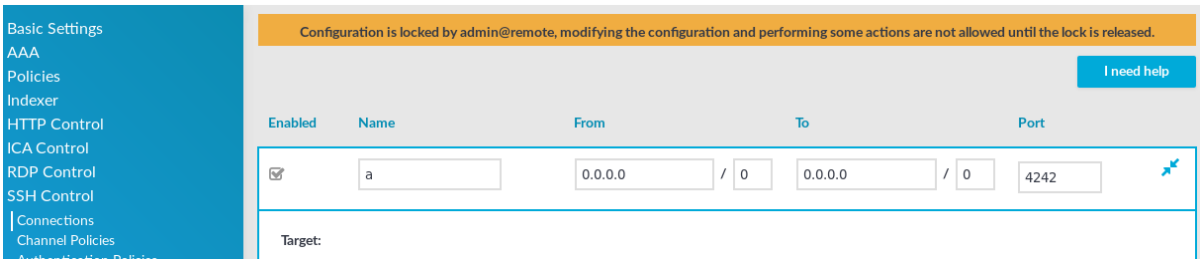


Multiple users and locking

Multiple administrators can access the One Identity Safeguard for Privileged Sessions (SPS) web interface simultaneously, but only one of them can modify the configuration. This means that the configuration of SPS is automatically locked when the first administrator who can modify the configuration opens a configuration page (for example the **Basic Settings** or the **AAA** menu).

The warning message displays the username of the administrator locking the configuration as shown in the image below:

Figure 36: Configuration lock by remote administrator



Other administrators can continue as read-only but must wait until the locking administrator navigates to an SPS page that does not require locking, the administrator logs out, or the session of the administrator times out. However, it is possible to access the **Search** and **Reporting** menus, and to perform gateway authentication and 4-eyes authorization or browse the configuration with only View rights (for details, see Managing user rights and usergroups on page 320).

Accessing SPS using the RPC API or starting a transaction in the REST API locks the configuration similarly to accessing SPS from the web interface.

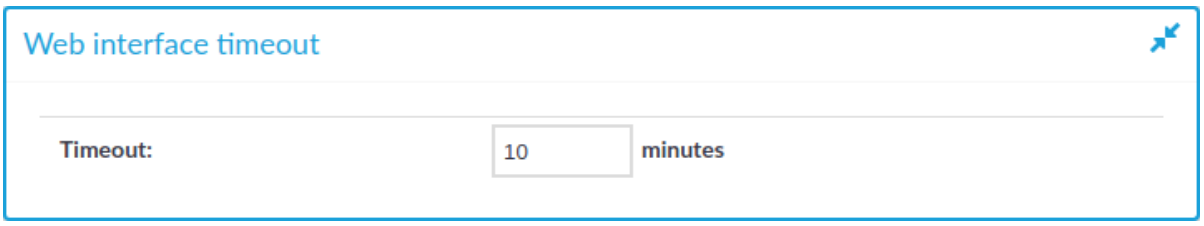
NOTE:

If an administrator logs in to SPS using the local console or a remote SSH connection, the configuration is also locked. Inactive local and SSH connections timeout just like web connections. For details, see Accessing the One Identity Safeguard for Privileged Sessions (SPS) console on page 389.

Web interface timeout

By default, One Identity Safeguard for Privileged Sessions (SPS) terminates the web session of a user after ten minutes of inactivity. To change value of this timeout, adjust the **Basic Settings > Management > Web interface timeout** option.

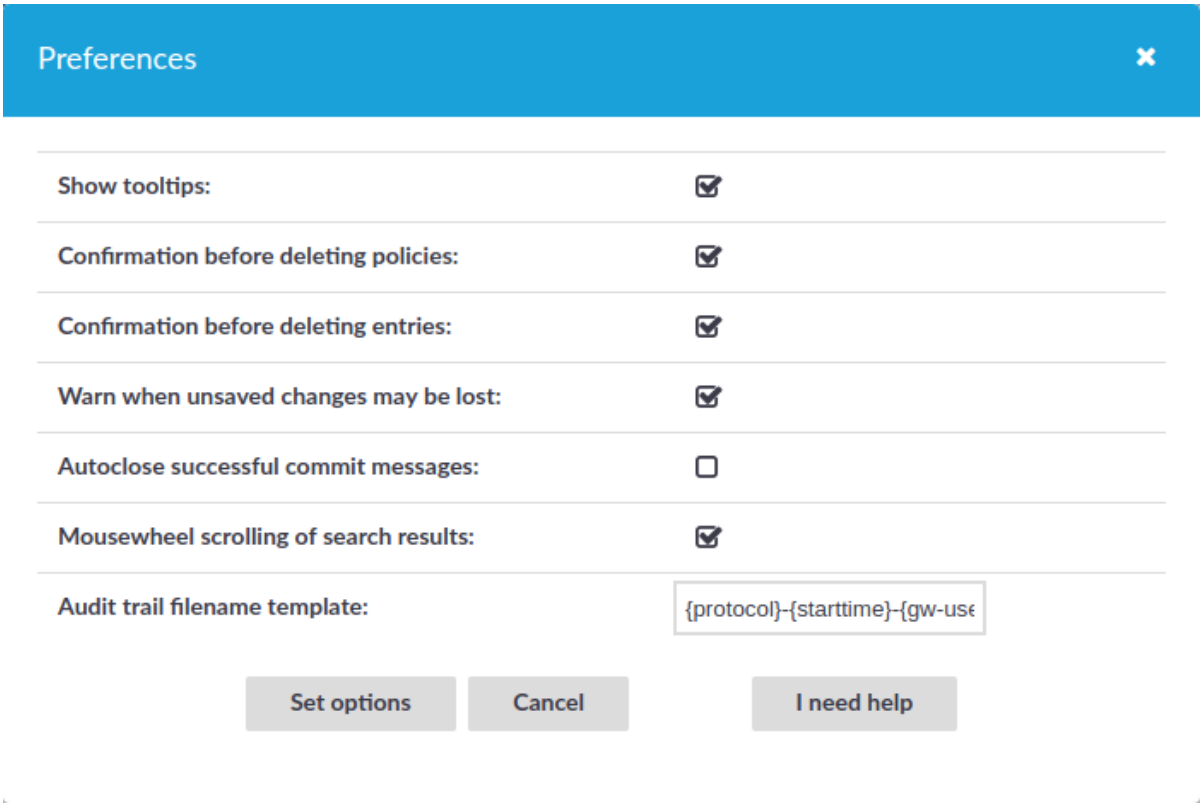
Figure 37: Basic Settings > Management > Web interface timeout — Web interface timeout



Preferences

To configure your preferences about the web interface, navigate to **User Menu > Preferences**.

Figure 38: User Menu > Preferences



- **Show tooltips**: Display tooltips for user interface elements to help using the product.
- **Confirmation before deleting policies**: Display a pop-up window when you attempt to delete policies to prevent deleting policies accidentally.
- **Confirmation before deleting entries**: Display a pop-up window when you attempt to delete entries to prevent deleting entries accidentally.
- **Warn when unsaved changes may be lost**: Display a pop-up window to warn when you navigate to another window without committing your changes to prevent losing unsaved changes.
- **Autoclose successful commit messages**: General confirmation windows will not appear. (For example, **Configuration saved successfully** that appears after successfully committing a change). As a result, pop-up windows appear only for failed actions or errors.
- **Mousewheel scrolling of search results**: When there are several pages of displayable search results on the **Search** page, navigate between search result pages with the mousewheel. When turned off, using the mousewheel on the **Search** page scrolls the whole page.
- **Audit trail filename template**:

  To change the filename of the audit trails, navigate to **User menu > Preferences** and change the **Audit trail filename template**. The default template is `{protocol}-{starttime}-{gw-username}-{remote-username}-{dst-ip}.zat`. The

template can include anything, the keys (inside {} brackets) are replaced with their actual values. These keys are the following:

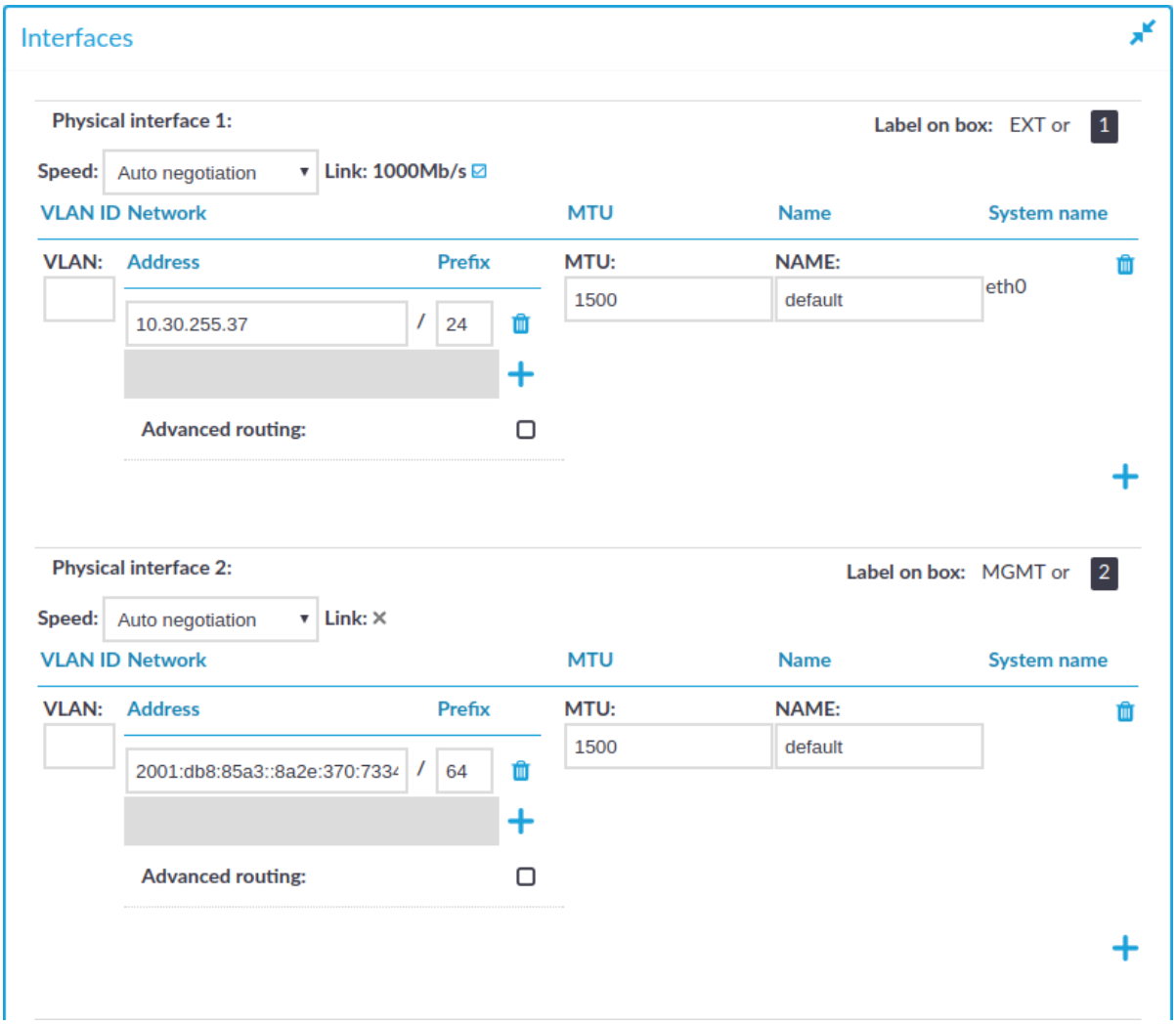
- `connection-policy`: The connection policy
- `dst-ip`: Destination IP address
- `dst-port`: Destination port
- `gw-username`: Gateway username
- `protocol`: Protocol
- `remote-username`: Remote username
- `session-id`: Session ID
- `src-ip`: Source IP address
- `starttime`: Start time of the session

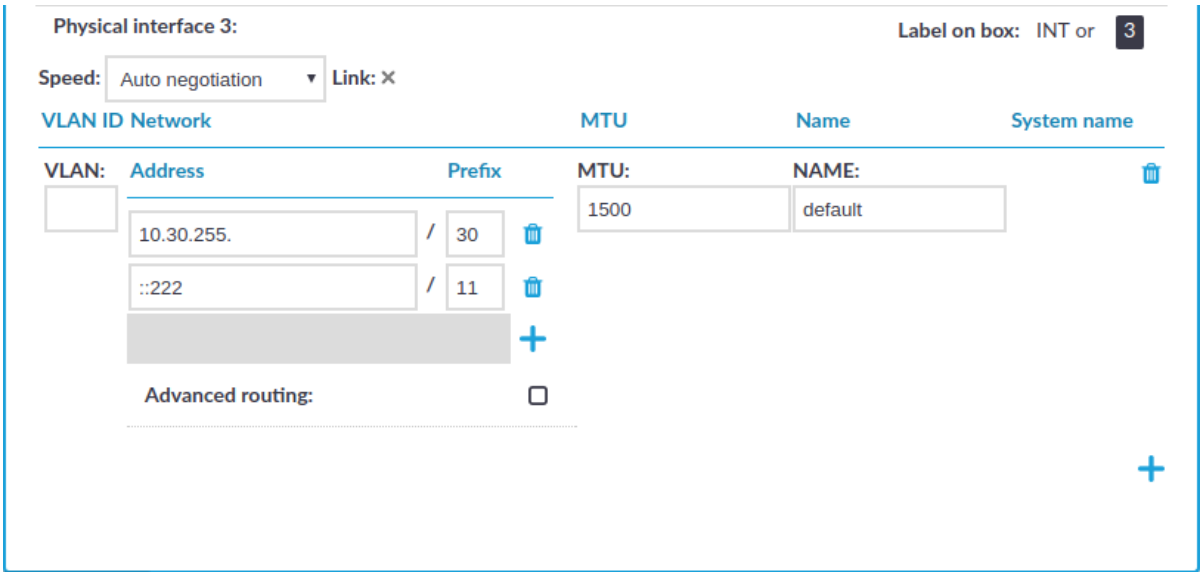
Network settings

The **Basic Settings > Network** tab contains the network interface and naming settings of One Identity Safeguard for Privileged Sessions (SPS).

Interfaces:

Figure 39: Basic Settings > Network > Interfaces





Lists all of the logical interfaces (VLAN IDs, IP addresses, netmasks, and names) assigned to the three physical interfaces of SPS. For more information on managing logical interfaces, see Managing logical interfaces on page 120.

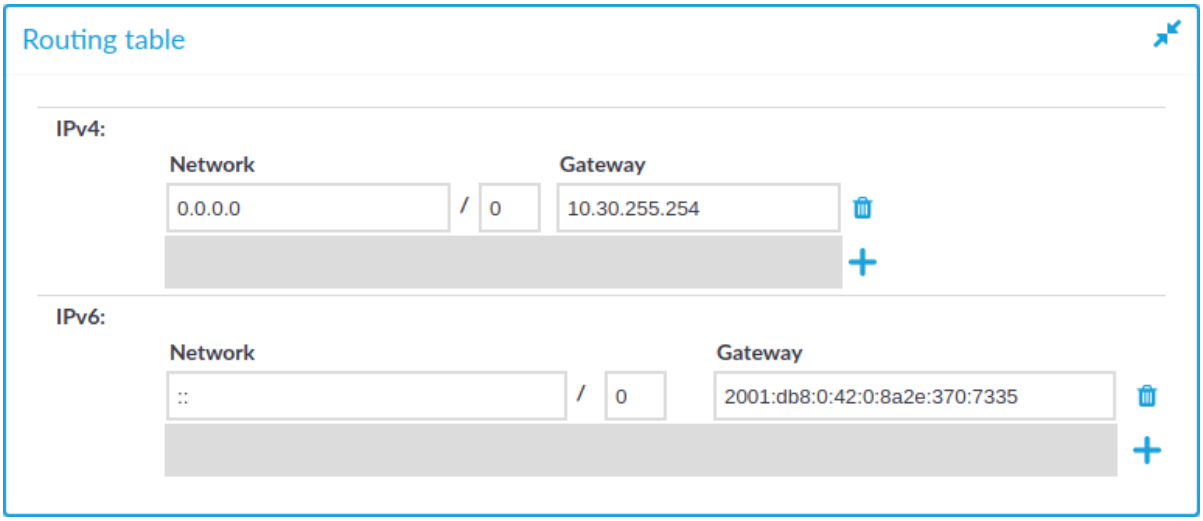
In addition, it is also possible to set the Maximum Transmission Unit (**MTU**) for each network interface (VLAN or network interface card) individually. The default value is `1500`.

Speed is displayed for every physical interface. To explicitly set the speed of the interface, select the new value from the **Speed** field. Modifying the speed of an interface is recommended only for advanced users.

You can add interface-specific network routes using the **Advanced routing** option of each interface. Otherwise, use the **Routing table** option to manage networking routes.

Routing table:

Figure 40: Basic Settings > Network > Routing table



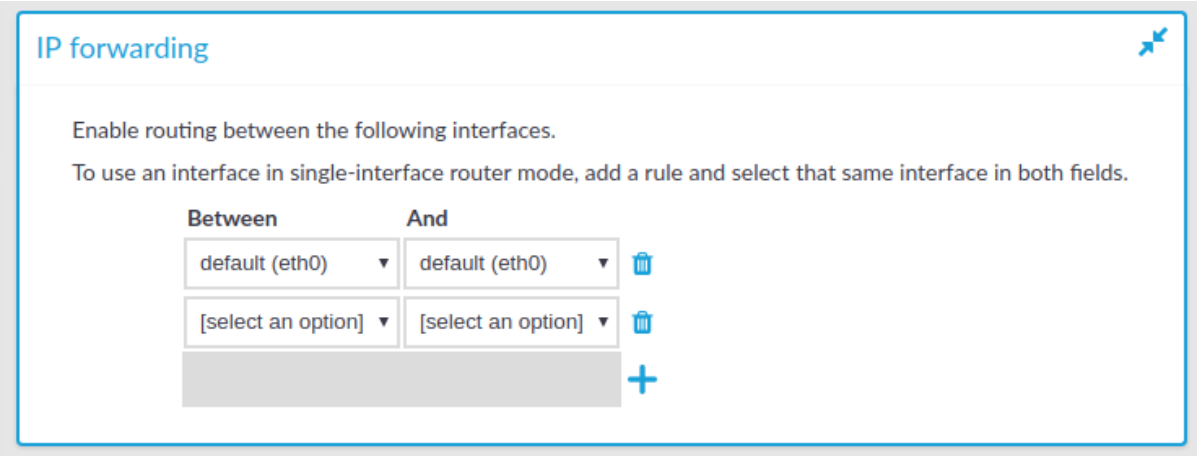
When sending a packet to a remote network, SPS consults the routing table to determine the path it should be sent. If there is no information in the routing table then the packet is sent to the default gateway. Use the routing table to define static routes to specific hosts or networks. You have to use the routing table if SPS interfaces are connected to multiple subnets.

Click the + and 🗑 icons to add new routes or delete existing ones. A route means that messages sent to the **Address**/**Netmask** network should be delivered to **Gateway**.

For detailed examples, see Configuring the routing table on page 123.

IP forwarding:

Figure 41: Basic Settings > Network > IP forwarding

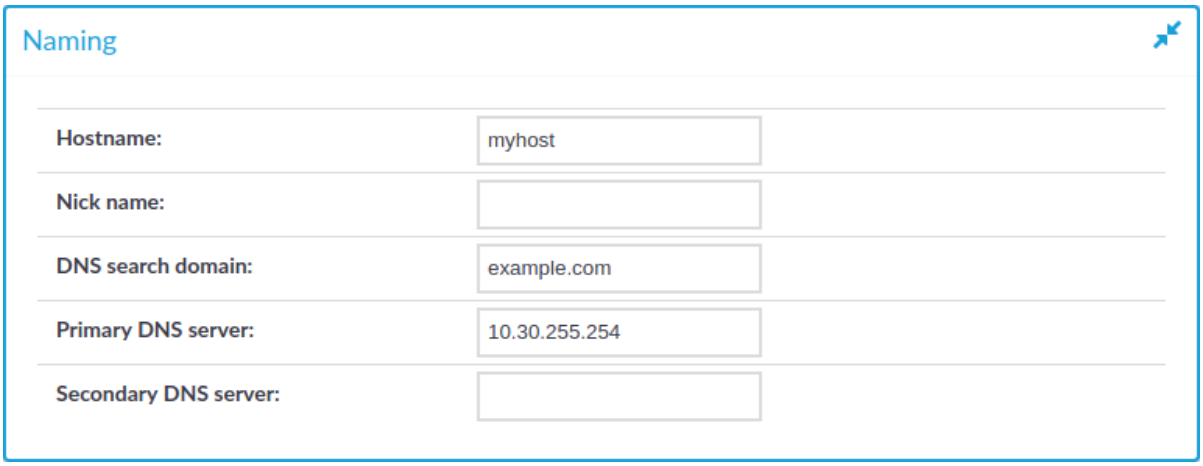


You can enable routing between logical interfaces, which allows you to direct uncontrolled traffic through SPS. For more information, see Routing uncontrolled traffic between logical interfaces on page 123.

To mimic the functionality of the deprecated Router mode, configure a logical interface for each physical interface you want to connect, and enable IP forwarding between them.

Naming:

Figure 42: Basic Settings > Network > Naming



- **Hostname**: Name of the machine running SPS.
- **Nick name**: The nickname of SPS. Use it to distinguish the devices. It is displayed in the core and boot login shells.
- **DNS search domain**: Name of the domain used on the network. When resolving the domain names of the audited connections, SPS will use this domain to resolve the target hostname if the appended domain entry of a target address is empty.
- **Primary DNS server**: IP address of the name server used for domain name resolution.
- **Secondary DNS server**: IP address of the name server used for domain name resolution if the primary server is unaccessible.

Configuring user and administrator login addresses

You can configure two separate login addresses for accessing the web interface of One Identity Safeguard for Privileged Sessions (SPS):

- Web login for administrators and users: On this address, users can, depending on their access privileges, modify the configuration of SPS, and perform authentication-related activities (gateway authentication, 4-eyes authorization).
- Web login for users only: The configuration of SPS cannot be viewed or altered from this address. Users (even ones with administrator privileges) can only perform gateway authentication and 4-eyes authorization.

NOTE:

You can find more information about gateway authentication and 4-eyes authorization in Advanced authentication and authorization techniques on page 740.

Both login addresses can be configured to restrict connections to a configured set of IP addresses only.

NOTE:

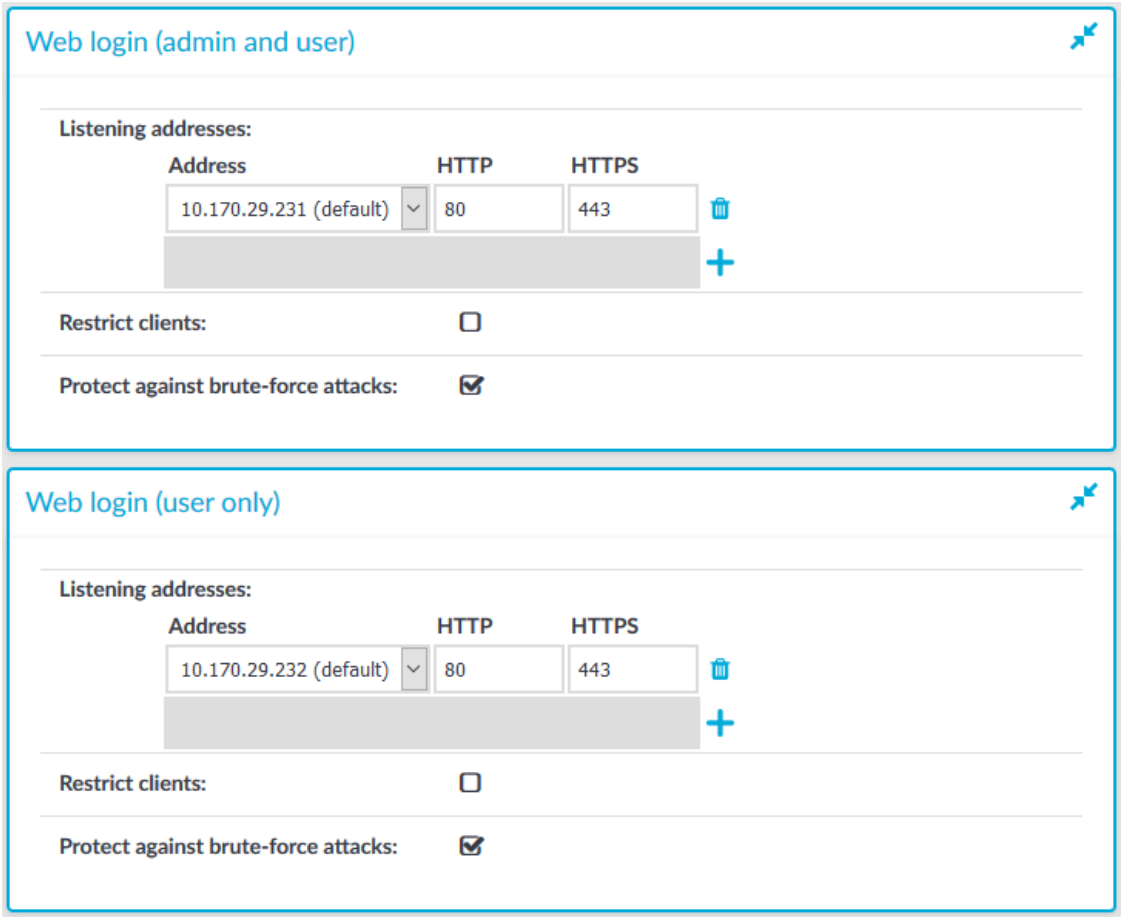
Avoid using the IP address configured for administrator or user login on One Identity Safeguard for Privileged Sessions (SPS) when configuring HTTP or SSH connections.

The login addresses are, by default, protected against brute-force attacks: after five unsuccessful login attempts, all following attempts are denied for increasing periods of time. You can turn this off by unselecting the **Protect against brute-force attacks** option for the web login addresses.

To configure two separate login addresses for accessing the web interface of SPS

1. Navigate to **Basic Settings > Local Services > Web login**.

Figure 43: Basic Settings > Local Services > Web login — Configuring web login address



2. Choose + in the **Listening addresses** field.

3. Enter the IP address to use for connecting to SPS's user interface into the **Address** field.

   The available addresses correspond to the interface addresses configured in **Basic Settings > Network > Interfaces**. Only IPv4 addresses can be selected.
4. Enter the port number for HTTP connections into the **HTTP** field.
5. Enter the port number for HTTPS connections into the **HTTPS** field.
6. (Optional) To permit access to the SPS web interface only from selected subnets or IP addresses, select **Restrict clients**, click + and enter the IP address and netmask of the allowed clients. Note that these settings do not affect the SSH access to SPS.

   CAUTION:

   Permit administrative access to SPS only from trusted networks. If possible, monitored connections and administrative access to the SPS web interface should originate from separate networks.

   After comitting the changes, the web interface will be available only from the configured subnets or IP addresses.

   Use an IPv4 address.
7. Recommended: configure a separate login address for user connections in **Web login (user only)**. The configuration settings of SPS cannot be viewed or modified from this address.
8. Click Commit.

Managing logical interfaces

You can assign logical interfaces to a physical interface. Each logical interface must have its own VLAN ID, and can have its own set of (alias) IP addresses and prefixes. The configured name for each logical interface is visible on One Identity Safeguard for Privileged Sessions (SPS)'s user interface only.

You can configure IPv4 and IPv6 addresses as well. IPv6 is intended for configuring monitored connections. Local services (including the web login) require IPv4 addresses. An interface can have multiple IP addresses, including a mix of IPv4 and IPv6 addresses.

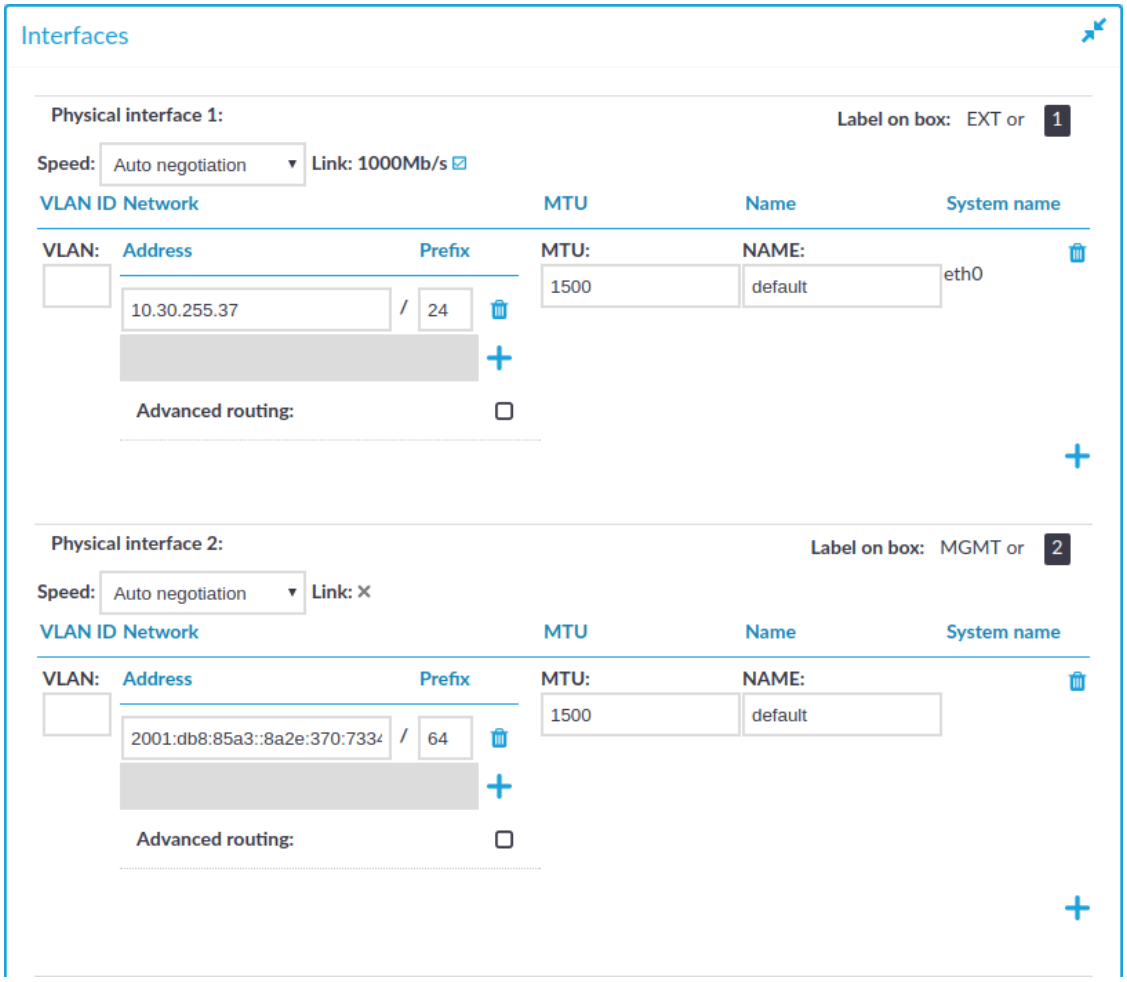
NOTE:

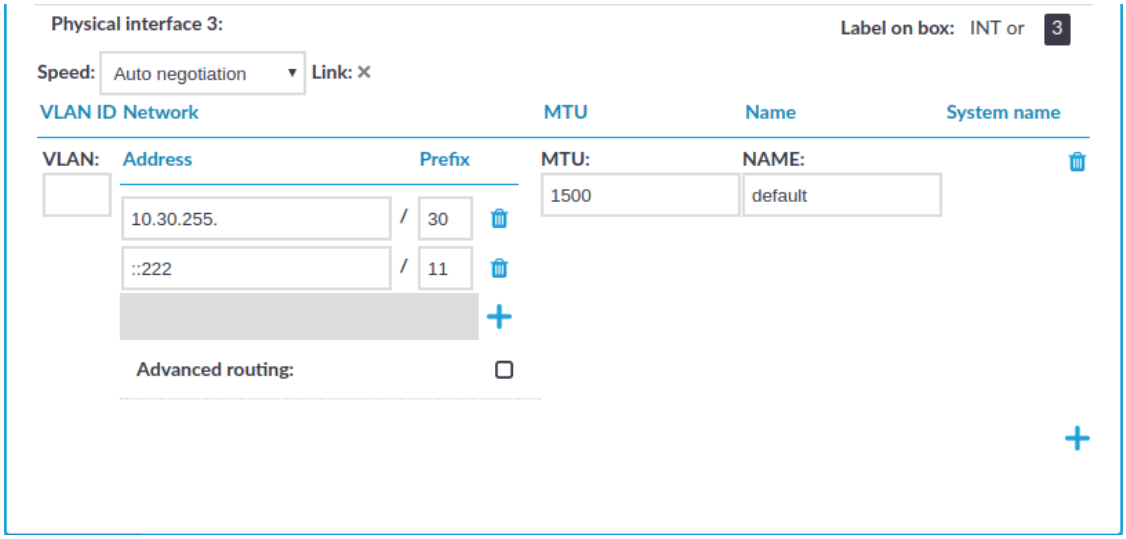
SPS does not support scenarios with two hosts using the same IP address on different VLAN groups.

To manage logical interfaces

1. Navigate to **Basic Settings > Network > Interfaces**.

Figure 44: Basic Settings > Network > Interfaces — Managing the logical interfaces





2. If necessary, use the label on the SPS hardware to identify the physical interface to which you want to assign a logical interface.
3. Choose + to add a new logical interface. Provide the following:
   - **VLAN**: The VLAN ID of the logical interface. Optional.

     CAUTION:

     Do not set the VLAN ID unless your network environment is already configured to use this VLAN. Otherwise, your SPS appliance will be unavailable using this interface.
   - **Address**: The IP address of the logical interface.

     You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:
     - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.
     - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.

     NOTE:

     Do not use IP addresses that fall into the following ranges:
     - `1.2.0.0/16` (reserved for communication between SPS cluster nodes)
     - `127.0.0.0/8` (localhost IP addresses)
   - **Prefix**: The IP range of the logical interface.

- Optional: To add additional (alias) IP addresses and prefixes to a logical interface, click +. To remove an alias IP address, click the corresponding 🗑.
- **MTU**: Maximum Transmission Unit (MTU) to set per network interface (VLAN or network interface card). The default value is `1500`.
- **Name**: The name of the logical interface. This name is visible on SPS's user interface only.

To remove a logical interface, choose the 🗑 on the right side.

4. Click Commit.

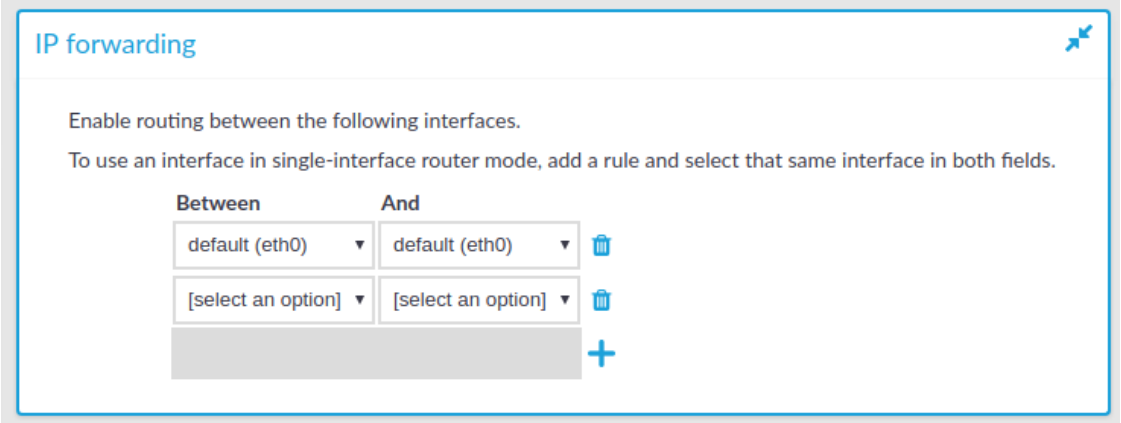
Routing uncontrolled traffic between logical interfaces

You can enable routing between logical interfaces, which allows you to direct uncontrolled traffic through SPS.

To enable routing between logical interfaces

1. Navigate to **Basic Settings > Network > IP forwarding**.

Figure 45: Basic Settings > Network > IP forwarding — IP forwarding between interfaces



2. To add a new forwarding rule, choose + and select the two logical interfaces to connect. You can select the same interface in both fields to use that logical interface in single-interface router mode.

   To delete an existing rule, choose 🗑.

3. Click Commit.

Configuring the routing table

The routing table contains the network destinations SPS can reach. You have to make sure that both the monitored connections, and the local services of SPS (including connections made to the backup and archive servers, the syslog server, and the SMTP server) are routed properly.

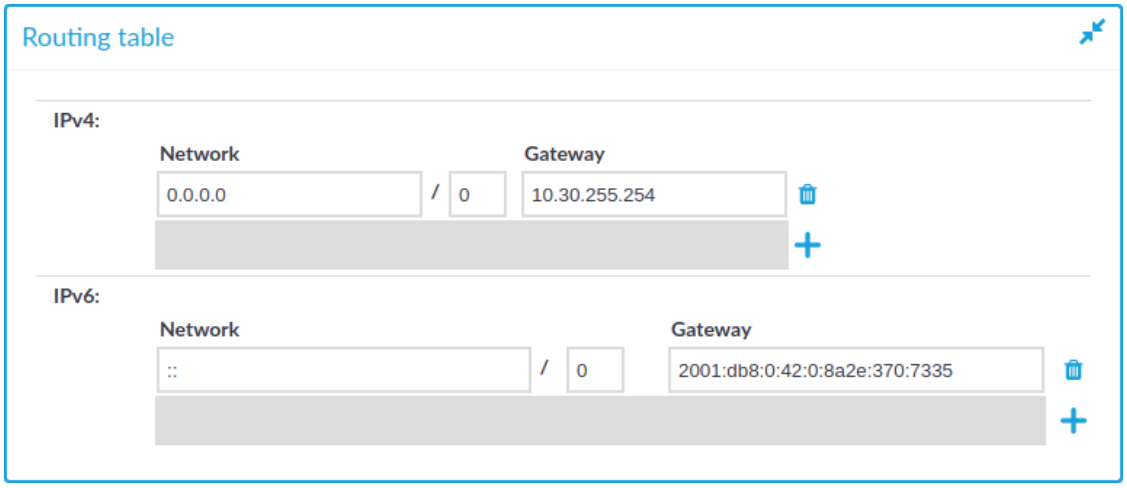
You can add multiple IPv4 and IPv6 addresses and address ranges along with their respective gateways.

To configure the routing table

1. To add a new routing entry, navigate to **Basic Settings > Network**.

   You can add interface-specific network routes using the **Advanced routing** option of each interface. Otherwise, use the **Routing table** option to manage networking routes.

   Figure 46: Basic Settings > Network > Routing table — Routing

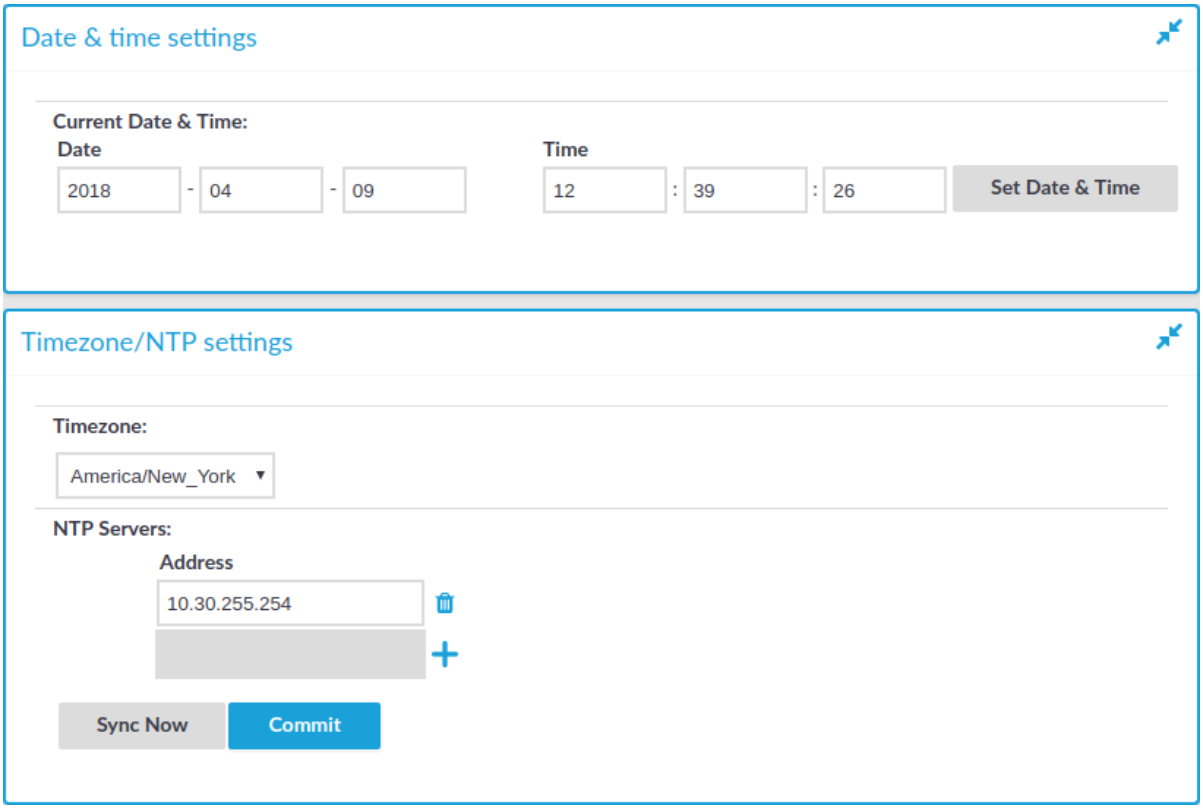
   

2. Click +, then enter the IP address and the network prefix into the **Network** field.
3. Enter the IP address of the gateway used on that subnetwork into the **Gateway** field.
4. Click Commit.

Configuring date and time

To configure the date and time-related settings of SPS, navigate to **Basic Settings > Date & Time**.

Figure 47: Basic Settings > Date & Time — Date and time management



CAUTION:

It is essential to set the date and time correctly on SPS, otherwise the date information of the logs and audit trails will be inaccurate.

SPS displays a warning on this page and sends an alert if the time becomes out of sync.

To explicitly set the date and time on SPS, enter the current date into respective fields of the **Date & Time settings** group and click **Set Date & Time**.

When two SPS units are operating in High Availability mode, the secondary node automatically synchronizes its time and date to the primary node. To manually synchronize the time between the nodes, click **Sync Master** (available only in High Availability mode).

To retrieve the date automatically from a time server, complete the following steps:

1. Select your timezone in the **Timezone** field.
2. Enter the IP address of an NTP time server into the **Address** field.

   Use an IPv4 address.
3. Click Commit.
4. Click the + and 🗑 icons to add new servers or delete existing ones.

5. Optional: If the time setting of SPS is very inaccurate (that is, the difference between the system time and the actual time is great), it might take a long time to retrieve the date from the NTP server. In this case, click **Sync Now** or **Sync Master** to sync the time immediately using SNTP.

System logging, SNMP and e-mail alerts

E-mail alerts and system logging can be configured on the **Basic Settings > Management** page.

Configuring system logging

One Identity Safeguard for Privileged Sessions (SPS) can send its system log messages to remote syslog servers (for example, syslog-ng Premium Edition, syslog-ng Store Box, Splunk, or HPE ArcSight Data Platform).

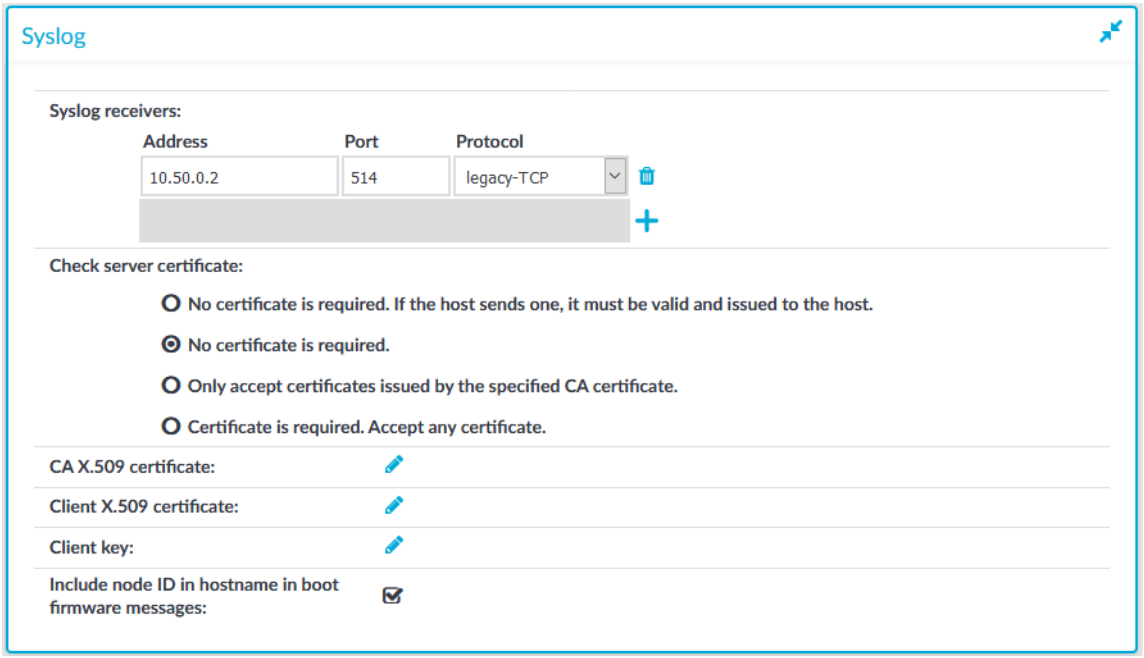
NOTE:

To send log messages in any custom format, contact our Support Team.

CAUTION:

The retention time for local logs of SPS is seven days. To retain them longer, forward them to a remote logserver.

Figure 48: Basic Settings > Management > Syslog — Configuring system logging



To configure logging to a remote server

1. Navigate to **Basic Settings > Management**.
2. Click + in the **Syslog > Syslog receivers** field to add a new syslog server.
3. Enter the IP address and port of the syslog server into the respective fields.

   Use an IPv4 address.
4. Select the network protocol used to transfer the messages in the **Protocol** field. The `legacy-` prefix corresponds to the legacy BSD-syslog protocol described in RFC3164, while the `syslog-` prefix corresponds to the new IETF-syslog protocol described in RFC5424. Note that not every syslog server supports the IETF protocol yet.

   Select `TCP+TLS` to send the log messages using a TLS-encrypted connection.

   > TIP:
   >
   > Transferring the syslog messages using TCP ensures that the server receives them.
   >
   > Transferring the syslog messages using TLS encryption ensures that third parties cannot read the messages. However, not every syslog server accepts encrypted connections. The syslog-ng Premium Edition and Open Source Edition applications, and the syslog-ng Store Box (which is a log-collector appliance similar to SPS) support both encrypted connections and the new IETF-syslog protocol as well. For details on these products, see syslog-ng Premium Edition and syslog-ng Store Box.
5. To display separate hostnames for syslog messages sent by the nodes of a SPS HA cluster, select the **Include node ID in hostname in boot firmware messages** option. The node ID included in the hostname file of the syslog message is the MAC address of the node's HA interface. (Messages of the core firmware are always sent by the primary node.)

   The boot firmware boots up SPS, provides high availability support, and starts the core firmware. The core firmware, in turn, handles everything else: provides the web interface, manages the connections, and so on.
6. If you have selected the `TCP+TLS` protocol, complete the following steps.

   Otherwise, click Commit.

   a. If you want SPS to verify the certificate of the syslog server, select **Only accept certificates issued by the specified CA certificate** in the **Check server certificate** field and proceed to the next step.

      If you want SPS to simply accept any certificate shown by the server, select **No certificate is required** in the **Check server certificate** field.

NOTE:

Alternatively, you can use the following, less strict options to check the certificate of the server:

- **No certificate is required. If the host sends one, it must be valid and issued to the host**: If the remote host sends a certificate, SPS checks if it is valid (not expired) and that the Common Name of the certificate contains the domain name or the IP address of the host. If these checks fail, SPS rejects the connection. However, SPS accepts the connection if the host does not send a certificate.
- **Certificate is required. Accept any certificate**: SPS requests a certificate from the server, and rejects the connection if no certificate is received, if the certificate is not valid (expired), or if the Common Name of the certificate does not contain the domain name or the IP address of the server.

b. Click the icon in the **CA X.509 certificate** field. A pop-up window is displayed.

   You can choose to upload a single certificate or a certificate chain (where one member of the chain is the CA that will sign the certificates). To upload a certificate chain, copy the certificates one after the other in a single file, and upload the file using **Browse**, and then **Upload**. Alternatively, you can copy and paste the certificates one after the other into the **Certificate Chain** field, and click **Set**. The certificates do not have to be in order, SPS will order them and validate the chain: if a member of the chain is missing, an error message is displayed.

   SPS will use this CA certificate to verify the certificate of the server, and reject the connections if the verification fails.

   To download previously uploaded certificates, click on the certificate and either download the certificate (or certificate chain) in one single PEM or DER file, or you can download single certificate files separately (if it is a certificate chain).

c. If the syslog server requires mutual authentication, that is, it expects a certificate from SPS, generate and sign a certificate for SPS, then click the icon in the **Client X.509 certificate** field to upload the certificate. After that, click the icon in the **Client key** field and upload the private key corresponding to the certificate.

d. Click Commit.

7. Click the + and icons to add new servers or delete existing ones.

   NOTE:

   To reduce the risk of the syslog server not receiving log messages from SPS because of a network outage or other problem with the syslog server, SPS buffers up to 10 Megabytes of log messages to its hard disk in case the syslog server becomes unaccessible.

Configuring e-mail alerts

The following describes how to configure e-mail alerts.

To configure e-mail alerts

1. Navigate to **Basic Settings > Management > Mail settings**.
2. If you want to encrypt the communication between SPS and the SMTP server, in **Encryption**, select the **STARTTLS** option and complete the following steps:
   - If you want SPS to verify the certificate of the server, select **Only accept certificates issued by the specified CA certificate** and click the icon in the **CA X.509 certificate** field. A pop-up window is displayed.

     Click **Browse**, select the certificate of the Certificate Authority (CA) that issued the certificate of the SMTP server, then click **Upload**. Alternatively, you can paste the certificate into the **Copy-paste** field and click **Set**.

     SPS will use this CA certificate to verify the certificate of the server, and reject the connections if the verification fails.
   - If the SMTP server requires mutual authentication, that is, it expects a certificate from SPS, enable **Authenticate as client**. Generate and sign a certificate for SPS, then click in the **Client X.509 certificate** field to upload the certificate. After that, click in the **Client key** field and upload the private key corresponding to the certificate.

   One Identity recommends using 2048-bit RSA keys (or stronger).
3. If you want SPS to authenticate to the SMTP server, in **Authentication**, select the **Enabled** option. Enter the **Username** to authenticate with.

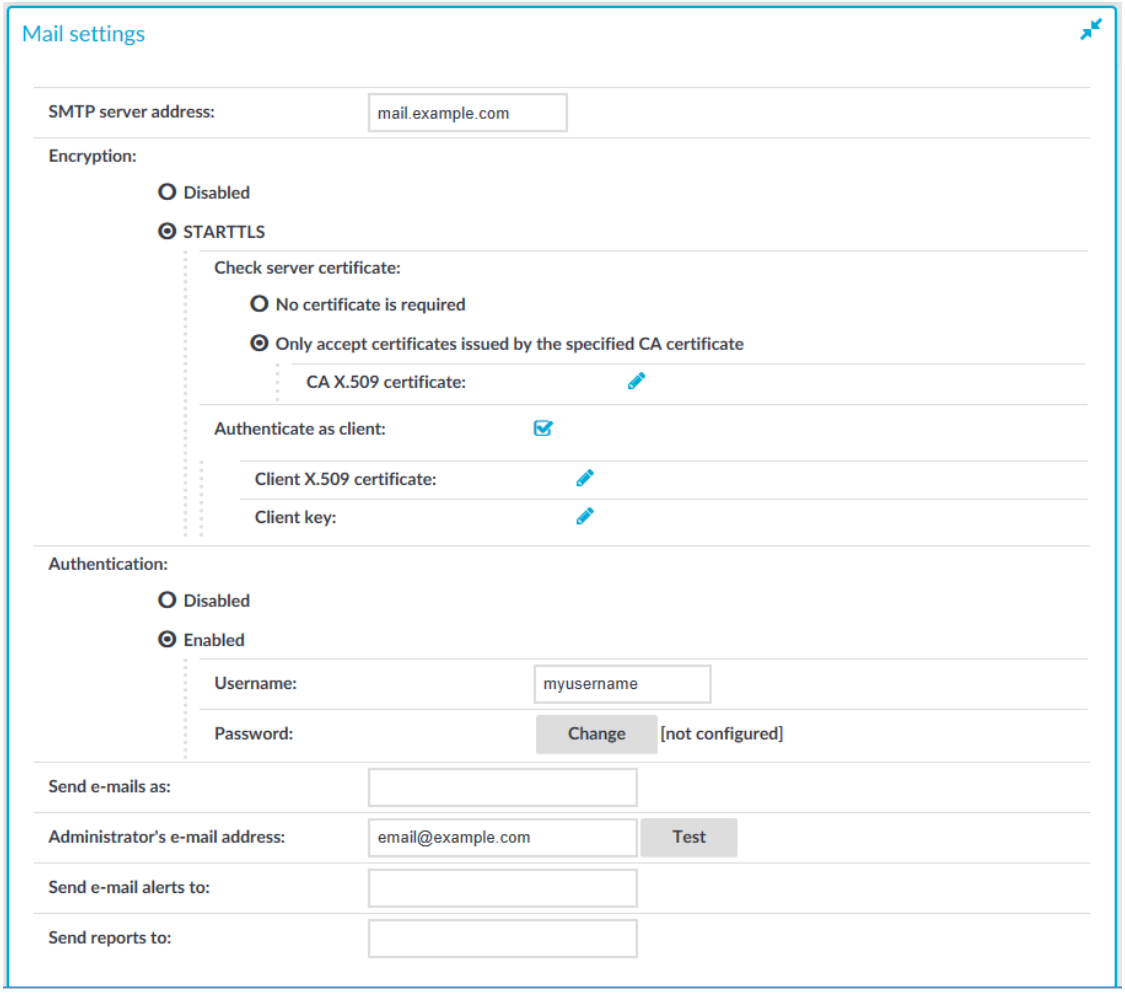
   To configure or change the password to use to authenticate to the SMTP server, click **Change** and enter the password. Click **Update**. Click Commit.

   NOTE:

   One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`
4. Enter the IP address or the hostname of the mail server into the **SMTP server address** field.

   Use an IPv4 address.

Figure 49: Basic Settings > Management > Mail settings — Configuring e-mail sending



5. Enter the e-mail address where you want to receive e-mails from into the **Send e-mails as** field. This can be useful for e-mail filtering purposes. SPS sends e-mails from the address provided here. If no e-mail address is entered, e-mails will be sent from the default e-mail address.
6. Enter the e-mail address of the administrator into the **Administrator's e-mail address** field. SPS sends notifications related to system-events (but not alerts and reports) to this address.
7. Enter the e-mail address of the administrator into the **Send e-mail alerts to** field. SPS sends monitoring alerts to this address.
8. Enter the e-mail address the person who should receive traffic reports from SPS into the **Send reports to** field. For details on reports, see Reports on page 777.
9. Click Commit.

10. Click **Test** to send a test message.

    If the test message does not arrive to the server, check if SPS can access the server. For details, see Troubleshooting One Identity Safeguard for Privileged Sessions (SPS) on page 854.

11. Navigate to **Basic Settings > Alerting & Monitoring** and select in which situations should SPS send an e-mail alert. For details, see Configuring system monitoring on SPS on page 137.

12. Click Commit.

Configuring SNMP alerts

SPS can send alerts to a central monitoring server through SNMP (Simple Network Management Protocol).

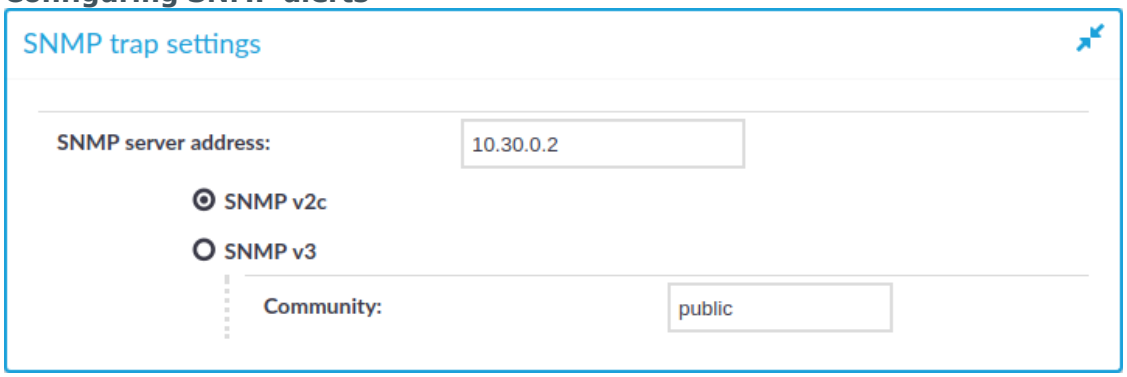
To configure SNMP alerts

1. Navigate to **Basic Settings > Management > SNMP trap settings**.

   Enter the IP address or the hostname of the SNMP server into the **SNMP server address** field.

   Use an IPv4 address.

   Figure 50: Basic Settings > Management > SNMP trap settings — Configuring SNMP alerts

2.

3. Select the SNMP protocol to use.

   (Optional) To use the SNMP v2c protocol for SNMP queries, select **SNMP v2c**, and enter the community to use into the **Community** field. Otherwise, skip these steps.

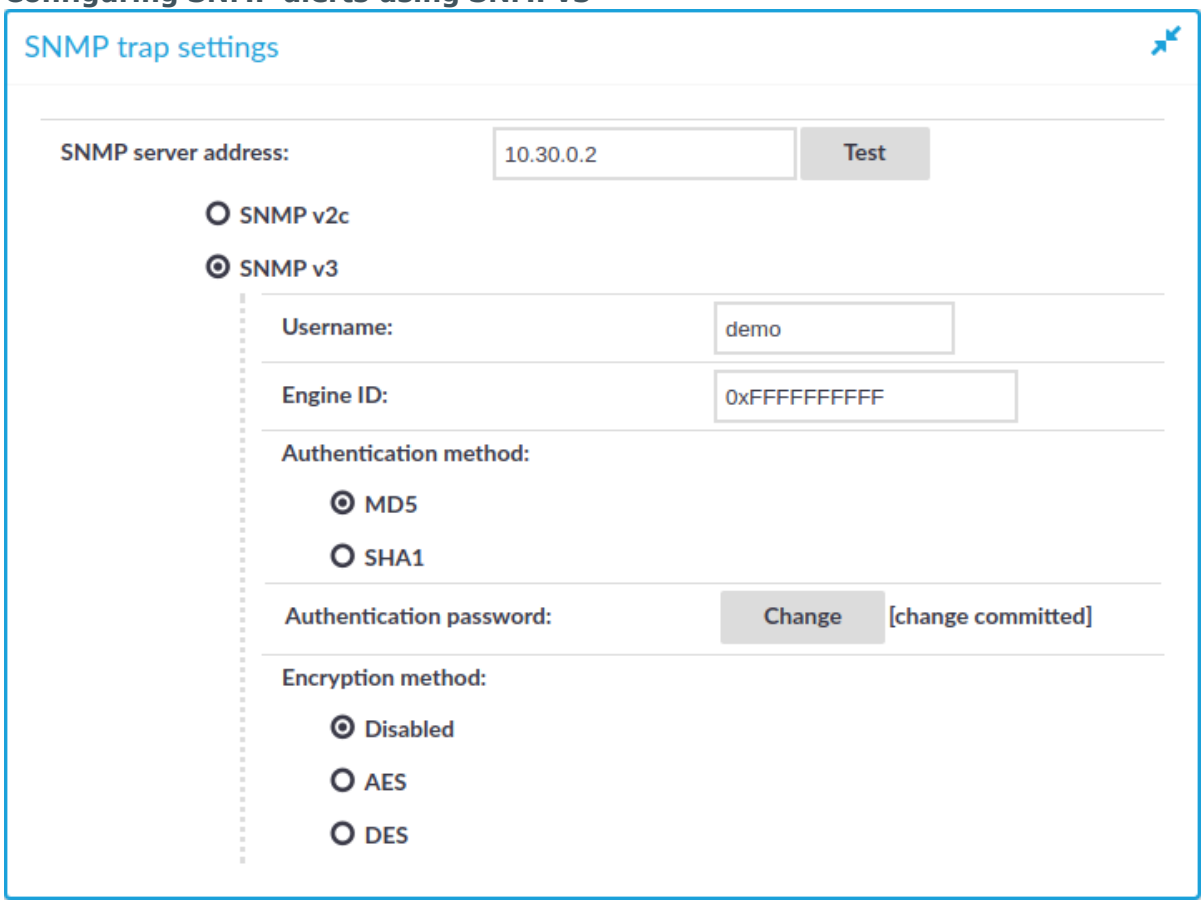
   (Optional) To use the SNMP v3 protocol, select **SNMP v3** and complete the following steps. Otherwise, skip these steps.

   1. Enter the username to use into the **Username** field.
   2. Enter the engine ID to use into the **Engine ID** field. The engine ID is a hexadecimal number at least 10 digits long, starting with `0x`. For example,

0xABABABABAB.

3. Select the authentication method (**MD5 or SHA1**) to use from the options under the **Authentication method:** field.
4. Enter the password to use into the **Authentication password** field.
5. Select the encryption method (**Disabled, DES or AES**) to use from the options under the **Encryption method:** field .
6. Enter the encryption password to use into the **Encryption password:** field.

Figure 51: Basic Settings > Management > SNMP trap settings — Configuring SNMP alerts using SNMPv3



NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

4. Click Commit.

5. Navigate to **Basic Settings > Alerting & Monitoring** and select in which situations SPS should send an SNMP alert. For details, see Configuring system monitoring on SPS on page 137.
6. Click Commit.

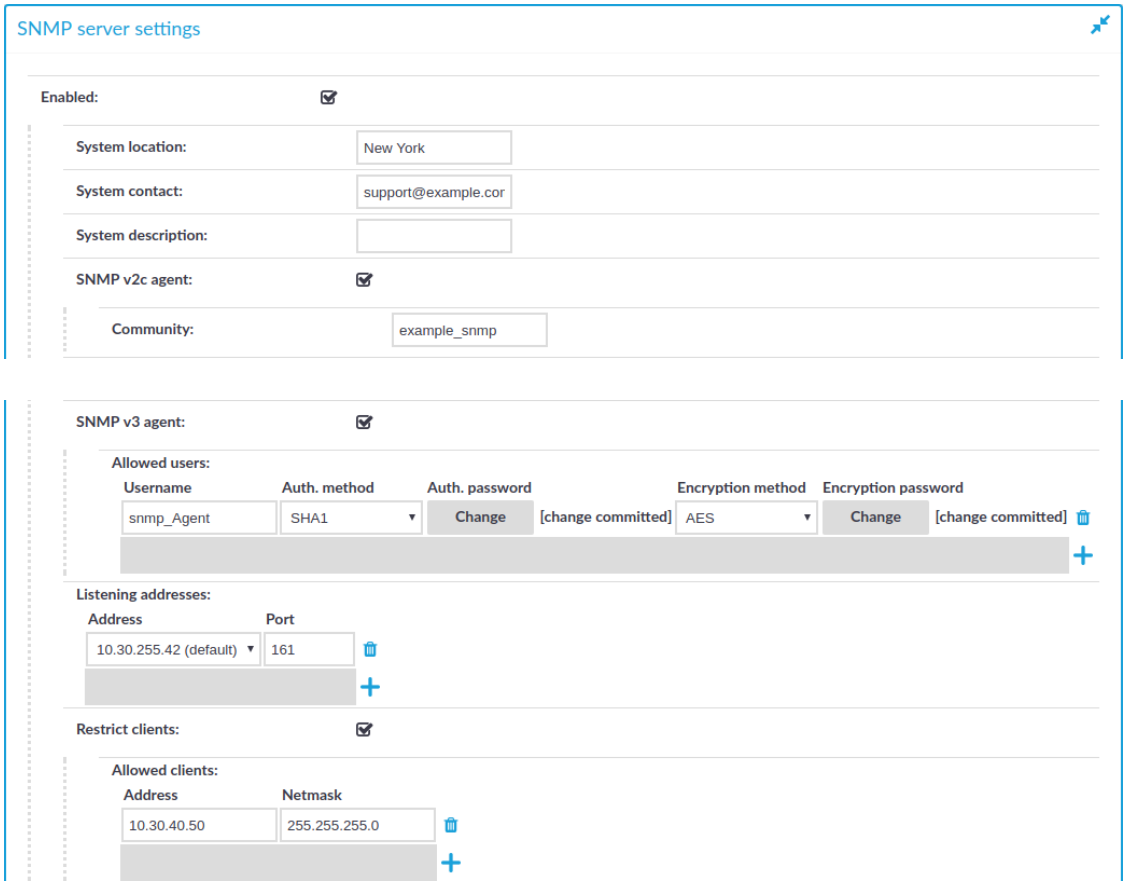
Querying SPS status information using agents

External SNMP agents can query the basic status information of SPS.

To configure which clients can query status information

1. Navigate to **Basic Settings > Local Services > SNMP server settings**.

Figure 52: Basic Settings > Local Services > SNMP server settings — Configuring SNMP agent access



2. Enable the SNMP server.

3. Optionally, you can enter the details of the SNMP server into the **System location**, **System contact**, and **System description** fields.
4. To use the SNMP v2c protocol for SNMP queries, enable **SNMP v2c agent**, and enter the community to use into the **Community** field.
5. To use the SNMP v3 protocol, select **SNMP v3 agent** and complete the following steps:
    a. Click +
    b. Enter the username used by the SNMP agent into the **Username** field.
    c. Select the authentication method (**MD5 or SHA1**) to use from the options under the **Auth. method** field.
    d. Enter the password used by the SNMP agent into the **Auth. password** field.
    e. Select the encryption method (**Disabled, DES or AES**) to use from the options under the **Encryption method** field.
    f. Enter the encryption password to use into the **Encryption password** field.
    g. To add other agents, click +.

    NOTE:

    One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`
6. In the **Listening addresses** field, choose + and select the IP address and port for the SNMP server.

    The available addresses correspond to the interface addresses configured in **Basic Settings > Network > Interfaces**. Only IPv4 addresses can be selected.

    Repeat this step to add multiple addresses.
7. (Optional) To permit access to the SNMP server only from selected subnets or IP addresses, select **Restrict clients**, click + and enter the IP address and netmask of the allowed clients.

    Use an IPv4 address.

    Repeat this step to add multiple addresses.
8. Click Commit.

Customize system logging in One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) uses the syslog-ng Open Source Edition application (version 3.16) for system logging. Starting with SPS 5 LTS, you can customize its configuration to better integrate SPS into your logging infrastructure. If you are not familiar with syslog-ng Open Source Edition, read how syslog-ng OSE works.

Customizing the configuration of syslog-ng Open Source Edition allows you to better integrate the log messages of SPS into your environment, for example, to:

- change the message format or rename message fields,
- send the messages to multiple logservers or SIEMs, or to
- select (filter) which messages to send to your logserver.

Limitations

Note that not every feature described in the syslog-ng Open Source Edition documentation is available on SPS. Typically, features that are only rarely used on logging clients are not available (for example, Java-based destinations like HDFS and Elasticsearch). For a detailed list of available modules, execute the **syslog-ng --module-registry** command.

Customize the syslog-ng configuration

Do not change the syslog configuration of SPS unless you know exactly what you are doing. Incorrect changes can decrease the performance of SPS, deactivate system logging, or cause message loss.

While customizing the syslog-ng configuration, note the following points in particular:

1. Create a SPS configuration snippet in a file. Make sure that the filename ends with `.conf`. Note that syslog-ng OSE uses the configuration objects defined in these files only if they are used in a log path as well, so make sure to include a log path.

   Do not loop messages. That is, make sure that the destination does not send a message back to the original source of the message (doing so would cause an infinite loop).
2. Copy it to the `/etc/syslog-ng/conf.d/` directory of the core firmware. (If you are using a high-availability SPS cluster, SPS automatically copies the file to the secondary node as well.)

   Files located in this directory do not taint the SPS configuration and SPS automatically includes them in the configuration of syslog-ng Open Source Edition.

   Do not modify the original configuration files (for example, `/etc/syslog-ng/syslog-ng.conf` or `/etc/syslog-ng/conf.d/message-queue-client.conf`).
3. Verify that the resulting syslog-ng OSE configuration file is syntactically valid. The configuration is valid if executing the following command does not show any syntax errors: **syslog-ng --syntax-only --no-caps**
4. Your changes will take effect only after you reload the configuration of syslog-ng Open Source Edition using the following command: **syslog-ng-ctl reload**

   If there are any errors in the configuration, SPS keeps on using the earlier configuration. In this case, correct the configuration, because if SPS reboots while the syslog-ng OSE configuration is invalid, SPS will not be able to log messages.

Available sources

You can use the following sources in your custom configuration. These sources are defined in the stock configuration file of syslog-ng OSE, and are in regular syslog message format (except for `s_message_queue_client`).

- `s_core_journal`: Logs of the SPS host, including log messages about the audited sessions.
- `s_message_queue_client`: Logs about the audited sessions in JSON format.
- `s_slave_boot`: Logs from the boot firmware of the secondary node in a high-availability SPS cluster.
- `src`: Logs messages of local SPS services.
- `src-internal`: The internal logs of syslog-ng OSE running on SPS.

Certificates and encrypted connections

If you are using a custom destination that requires a certificate (either to authenticate SPS, or to verify the identity of the logserver). In this case, copy the certificates to SPS into the `/etc/syslog-ng/conf.d/` directory. In the custom syslog configuration you cannot use the certificates uploaded to SPS using the web interface.

SIEM integration

Customizing the syslog configuration of SPS allows you to send log messages directly to your SIEM (for example, Splunk) in a format that your SIEM can understand.

One Identity can provide you the configuration files needed to send the log messages of SPS to Splunk in the Splunk Common Information Model (CIM) format. If you are interested, contact our Support Team.

If you need assistance to use another SIEM format, contact professionalservices@balabit.com.

Examples

The following configuration snippet reads the messages from the built-in `s_message_queue_client` source, parses the JSON message, and sends the messages to a remote destination using the RFC5424 message format (the body of the message remains in JSON).

```
parser json {
    json-parser(
        prefix(".scb.")
        template("$MSG")
    );
};

destination d_custom_remote {
    syslog(
        "192.168.1.1"
```

```
        transport(tcp)
        port(6514)
        template("$(format-json --key .scb.*)\n")
    );
};

log {
    source(s_message_queue_client);
    parser(json);
    destination(d_custom_remote);
};
```

A sample log message using the above configuration is the following (line-breaks added for clarity):

```
{"_scb":
    {
       "username":"frobert",
             "timestamp":"1499673286.380022",
             "session_id":"svc/hm4soFrNYeAmkyzLVkzi1o/ssh_demo:4",
             "server_port":"22",
             "server_ip":"10.30.255.70",
             "protocol":"ssh",
             "event_type":"server_authentication_success",
             "connection_name":"ssh_demo",
             "client_port":"50830",
             "client_ip":"10.30.255.70"
       }
 }
```

To use this configuration snippet on your SPS, copy it to a file (make sure that the filename ends with `.conf`), change the IP address and port number to match your environment, copy it to the core firmware of your SPS into the `/etc/syslog-ng/conf.d` directory, then reload the syslog-ng configuration using **syslog-ng-ctl reload**.

Configuring system monitoring on SPS

SPS supports the SNMPv2c and SNMPv3 protocols. The SNMP server set on the **Management** tab can query status information from SPS.

TIP:

In order to have your central monitoring system to recognize the SNMP alerts sent by SPS, import the SPS-specific Management Information Base (MIB) into your monitoring system. Download all MIBs by navigating to **Basic Settings > Alerting & Monitoring** and clicking **Download MIBs** and import them into your monitoring system. For details, see the documentation of your monitoring system.

Configuring monitoring

The following describes how to configure monitoring.

To configure monitoring

1. Navigate to **Basic Settings > Alerting & Monitoring**.
2. The default threshold values of the parameters are suitable for most situations. Adjust the threshold values only if needed.
3. Click Commit.
4. Navigate to **Basic Settings > Management** and verify that the **SNMP settings** and **Mail settings** of SPS are correct. SPS sends alerts only to the alert e-mail address and to the SNMP server.

   CAUTION:

   Sending alerts fails if these settings are incorrect.

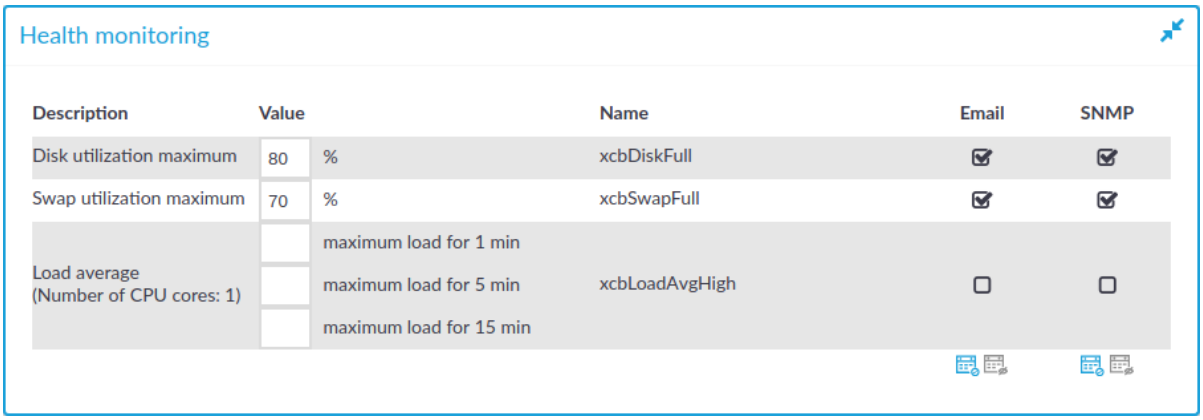
The following sections describe the parameters you can receive alerts on.

- For details on health-monitoring alerts, see Health monitoring on page 138.
- For details on system-monitoring alerts, see System related traps on page 140.
- For details on traffic-monitoring alerts, see Traffic related traps on page 143.

Health monitoring

SPS continuously monitors a number of parameters of the SPS hardware and its environment. If a parameter reaches a critical level (set in its respective **Maximum** field), SPS sends e-mail or SNMP messages to alert the administrator.

Figure 53: Basic Settings > Alerting & Monitoring — Health monitoring



- **Disk utilization maximum**: Ratio of free space available on the hard disk. SPS sends an alert if the audit trails use more space than the set value. Archive the audit trails to a backup server to free disk space. For details, see Archiving and cleanup on page 159.

  NOTE:

  The alert message includes the actual disk usage, not the limit set on the web interface. For example, you set SPS to alert if the disk usage increases above `10` percent. If the disk usage of SPS increases above this limit (for example to `17` percent), you receive the following alert message: `less than 90% free (= 17%)`. This means that the amount of used disk space increased above 10% (what you set as a limit, so it is less than 90%), namely to 17%.

- **Load average**: The average load of SPS during the last one, five, or 15 minutes.
- **Swap utilization maximum**: Ratio of the swap space used by SPS. SPS sends an alert if it uses more swap space than the set value.

Preventing disk space fill-up

The following describes how to prevent disk space from filling up.

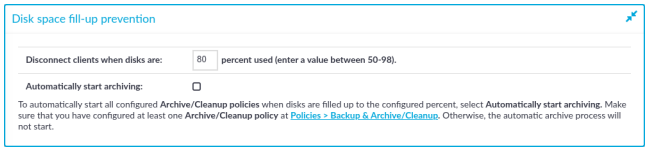
NOTE:

One Identity highly recommends this if One Identity Safeguard for Privileged Sessions (SPS) is hosted in a virtual environment.

To prevent disk space from filling up

1. Navigate to **Basic Settings > Management > Disk space fill-up prevention**.

   Figure 54: Basic Settings > Management > Disk space fill-up prevention — Preventing disk space fill-up

2. Enter the limit of maximum disk utilization in percents in the **Disconnect clients when disks are: x percent used** field. Make sure to enter a value between `50-98` percent. When disk space is used above the configured limit, SPS disconnects all clients. The default value is `80`.
3. (Optional) To automatically start all configured archiving/cleanup jobs when disk usage goes over the limit, select the **Automatically start archiving** option.

   For more information on configuring an archiving policy, see Archiving and cleanup on page 159.

NOTE:

If there is no archiving policy configured, selecting this option will not trigger automatic archiving.

4. Click Commit.
5. Navigate to **Basic Settings > Alerting & Monitoring > Health monitoring** and enable alert **Disk utilization maximum**.
6. Click Commit.

System related traps

SPS can send the following system related alerts in e-mail or as SNMP trap. To configure these alerts, see Configuring e-mail alerts on page 129 and Configuring SNMP alerts on page 131.

NOTE:

Configure **Disk space fill-up prevention**, and configure SPS to send an alert if the free space on the disks of SPS is low. For details, see "Preventing disk space fill-up" in the Administration Guide.

Configure SPS to send an alert if a user fails to login to SPS. For details, see the **Login failed** alert in "System related traps" in the Administration Guide.

Figure 55: Basic Settings > Alerting & Monitoring — health monitoring

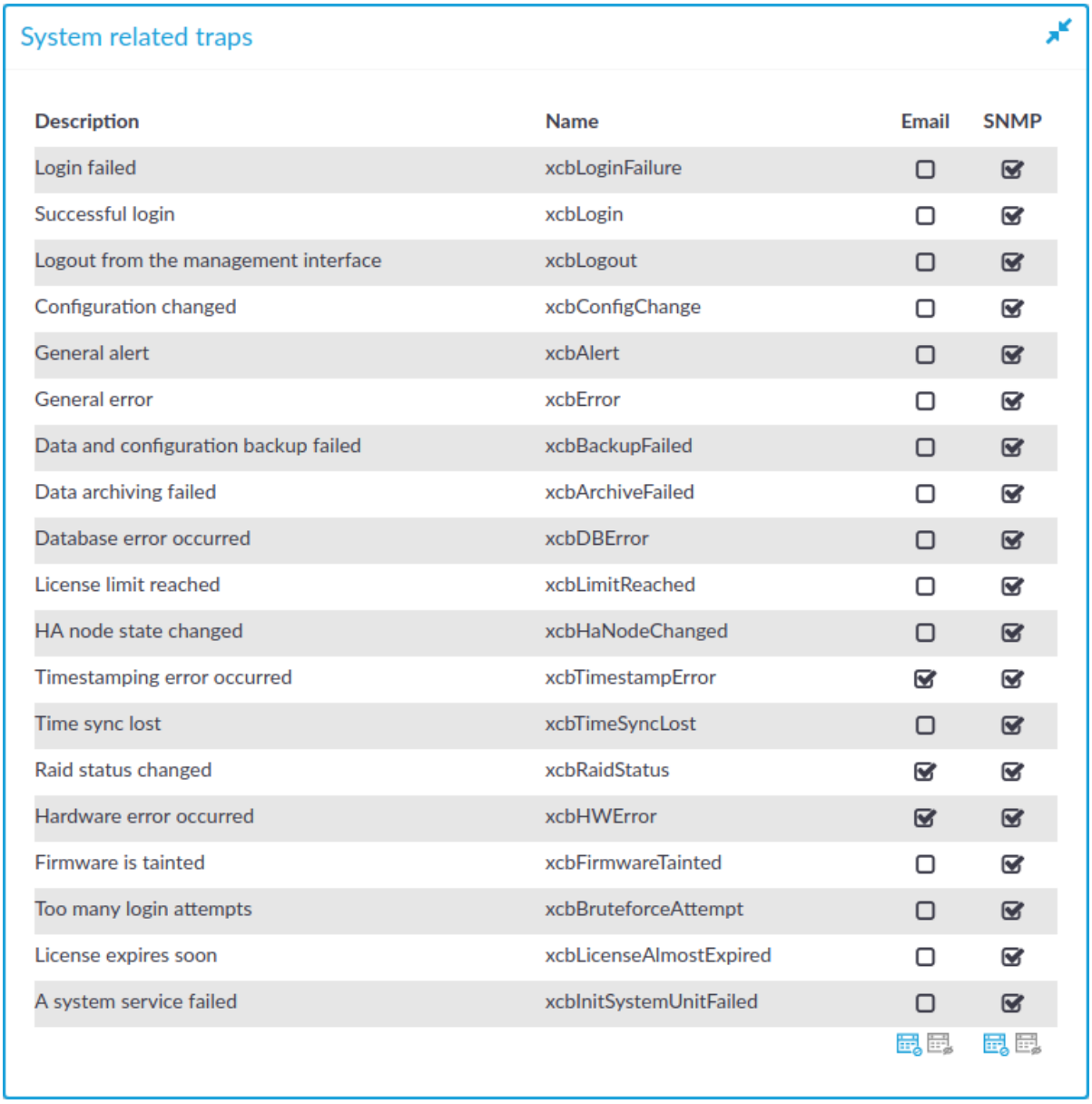


Table 1: System related traps

Name	SNMP alert ID	Description
Login failed	`xcbLoginFailure`	Failed login attempts from SPS web interface.
Successful login	`xcbLogin`	Successful login attempts into SPS web interface.
Logout from the manage-	`xcbLogout`	Logouts from SPS web interface.

Name	SNMP alert ID	Description
ment interface		
Configuration changed	`xcbConfigChange`	Any modification of SPS's configuration.
General alert	`xcbAlert`	General alerts and error messages occurring on SPS. Note that alerts on general alerts and errors are sent whenever there is an alert or error level message in the SPS system log. These messages are very verbose and mainly useful only for debugging purposes. Enabling these alerts may result in multiple e-mails or SNMP traps sent about the same event.
General error	`xcbError`	
Data and configuration backup failed	`xcbBackupFailed`	Alerts if the backup procedure is unsuccessful.
Data archiving failed	`xcbArchiveFailed`	Alerts if the archiving procedure is unsuccessful.
Database error occurred	`xcbDBError`	An error occurred in the database where SPS stores the connection metadata. For assistance, contact our Support Team.
License limit reached	`xcbLimitReached`	The number of protected servers (or concurrent sessions) reached the limit set in the SPS license. Clients cannot connect to new servers using SPS.
HA node state changed	`xcbHaNodeChanged`	A node of the SPS cluster changed its state (for example, a takeover occurred).
Timestamping error occurred	`xcbTimestampError`	An error occurred during the timestamping process (for example, the timestamping server did not respond).
Time sync lost	`xcbTimeSyncLost`	The system time became out of sync.
Raid status changed	`xcbRaidStatus`	The status of the node's RAID device changed its state.
Hardware error occurred	`xcbHWError`	SPS detected a hardware error.

Name	SNMP alert ID	Description
Firmware is tainted	`xcbFirmwareTainted`	A user has locally modified a file from the console.
Too many login attempts	`xcbBruteforceAttempt`	SPS has detected a possible brute-force attack.
License expires soon	`xcbLicenseAlmostExpired`	Your SPS license will expire within 60 days.
A system service failed	`xcbInitSystemUnitFailed`	A system service has failed. Note that one alert is sent for each failed service.

Traffic related traps

SPS can send the following traffic related alerts in e-mail or as SNMP trap. To configure these alerts, see Configuring e-mail alerts on page 129 and Configuring SNMP alerts on page 131.

Figure 56: Basic Settings > Alerting & Monitoring — health monitoring

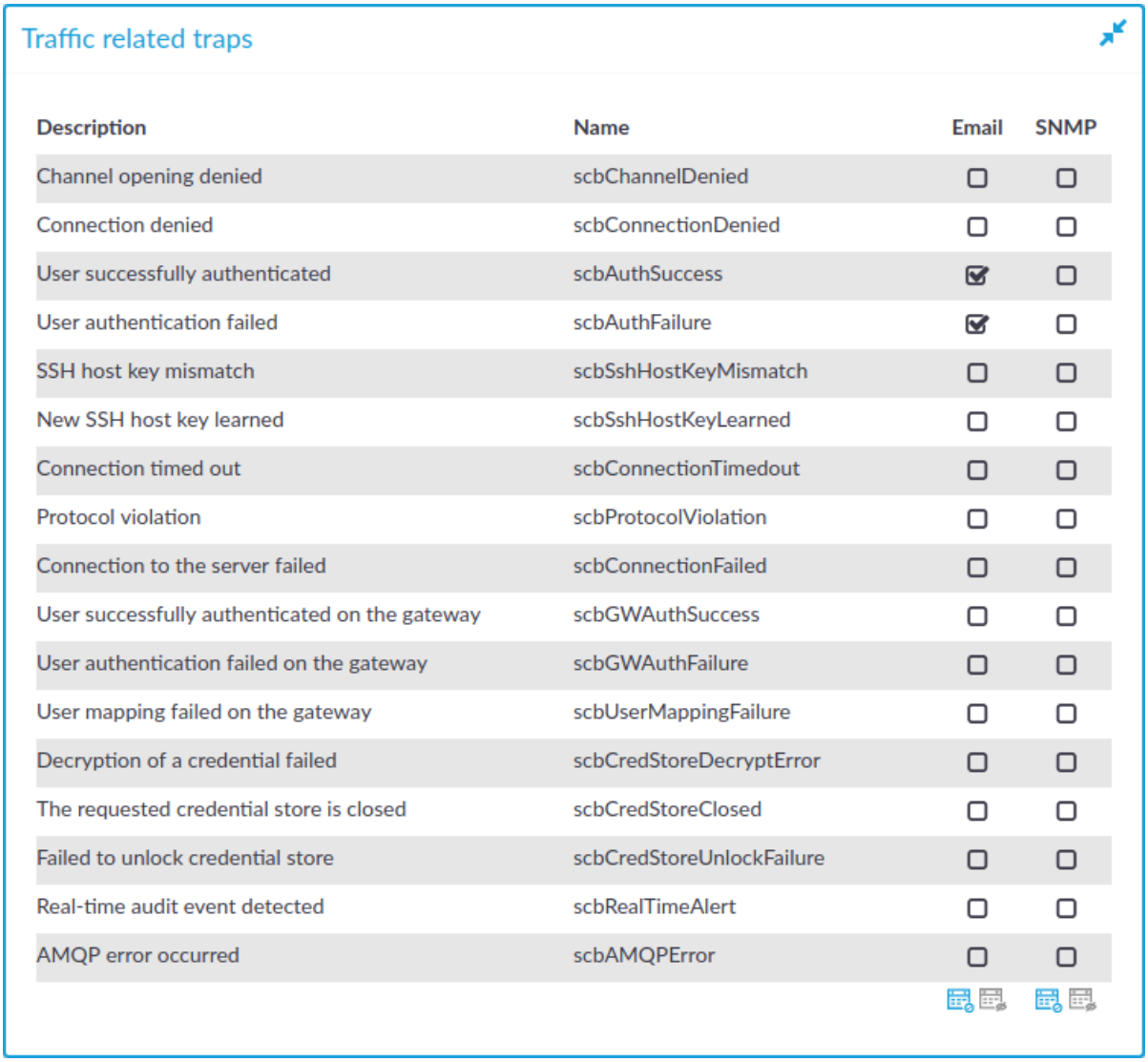


Table 2: Traffic related traps

Name	SNMP alert ID	Description
Channel opening denied	`scbChannelDenied`	A user attempted to open a channel not permitted by the channel policy.
Connection denied	`scbConnectionDenied`	A user attempted to connect a server not permitted in the connection policies.
User successfully authenticated	`scbAuthSuccess`	A user successfully authenticated on a protected server.

Name	SNMP alert ID	Description
User authentication failed	`scbAuthFailure`	A user failed to complete the authentication on a protected server.
SSH host key mismatch	`scbSshHostKeyMismatch`	The SSH host key of a server did not match the key stored on SPS.
New SSH host key learned	`scbHostKeyLearned`	SPS learned a new SSH host key.
Connection timed out	`scbConnectionTimedout`	A connection to a protected server timed out.
Protocol violation	`scbProtocolViolation`	A connection violated the protocol as specified in the RFC or protocol documentation. This may have been caused by an incompatible application or a deliberate attack.
Connection to the server failed	`scbConnectionFailed`	A connection to a protected server failed.
User successfully authenticated on the gateway	`scbGWAuthSuccess`	A user has successfully authenticated a connection on SPS as part of a gateway-authentication process.
User authentication failed on the gateway	`scbGWAuthFailure`	The gateway-authentication of a connection has failed.
User mapping failed on the gateway	`scbUserMappingFailure`	A usermapping policy did not find a suitable mapping for the connection.
Decryption of a credential store failed	`scbCredStoreDecrpytError`	SPS could not unlock a password-protected Credential Store. Navigate to **Unlock Credential Store** and enter the password(s) to open the Credential Store.
The requested credential store is closed	`scbCredStoreClosed`	A user attempted to access a connection policy that uses a password-protected Credential Store, and the Credential Store has not been unlocked. Navigate to **Unlock Credential Store** and enter the password(s) to open the Credential Store.

Name	SNMP alert ID	Description
Failed to unlock credential store	`scbCredStoreUnlockFailure`	A user attempted to unlock a password-protected Credential Store with an incorrect password. Navigate to **Unlock Credential Store** and enter the correct password(s) to open the Credential Store.
Real time audit event detected	`scbRealTimeAlert`	A real-time audit event has occurred.
AMQP error occurred	`scbAMQPError`	An error occurred in the event queue where SPS forwards session data. contact our Support Team.

Data and configuration backups

Backups create a snapshot of the configuration of One Identity Safeguard for Privileged Sessions (SPS) or the data which can be used for recovery in case of errors. SPS can create automatic backups of its configuration and the stored audit-trails to a remote server.

Configuring backups is a two-step process:

1. Create a backup policy.
2. Assign that policy to the system or a connection — depending on what it is that you wish to back up, SPS's configuration or a connection.

Creating a backup policy:

Backup policies define the address of the backup server, which protocol to use to access it, and other parameters. SPS can be configured to use the Rsync, SMB/CIFS, and NFS protocols to access the backup server:

- To configure backups using Rsync over SSH, see Creating a backup policy using Rsync over SSH on page 147.
- To configure backups using SMB/CIFS, see Creating a backup policy using SMB/CIFS on page 150.
- To configure backups using NFS, see Creating a backup policy using NFS on page 154.

The different backup protocols assign different file ownerships to the files saved on the backup server. The owners of the backup files created using the different protocols are the following:

- *Rsync*: The user provided on the web interface.
- *SMB/CIFS*: The user provided on the web interface.
- *NFS*: `root` with `no-root-squash`, `nobody` otherwise.

CAUTION:

SPS cannot modify the ownership of a file that already exists on the remote server. If you change the backup protocol but you use the same directory of the remote server to store the backups, make sure to adjust the ownership of the existing files according to the new protocol. Otherwise SPS cannot overwrite the files and the backup procedure fails.

Assigning a backup policy:

Once you have configured a backup policy, set it as a system backup policy (for configuration backups) or data backup policy (for connections backups):

- To configure a system backup policy, see Creating configuration backups on page 157.
- To configure a data backup policy, see Creating data backups on page 158.

NOTE:

Backup deletes all other data from the target directory. Restoring a backup deletes all other data from SPS. For details on restoring configuration and data from backup, see Restoring One Identity Safeguard for Privileged Sessions (SPS) configuration and data on page 883.

Creating a backup policy using Rsync over SSH

The **Rsync over SSH** backup method connects the target server with SSH and executes the **rsync** UNIX command to copy the data to the remote server. SPS authenticates itself with a public key — password-based authentication is not supported.

CAUTION:

The backup server must run rsync version 3.0 or newer.

To create a backup policy using Rsync over SSH

1. Navigate to **Policies > Backup & Archive/Cleanup** and click + in the **Backup policies** section to create a new backup policy.
2. Enter a name for the backup policy (for example, `config-backup`).
3. Enter the time when the backup process should start into the **Start time** field in HH:MM format (for example, `23:00`).

   You can add the start time for additional backup processes.

   CAUTION:

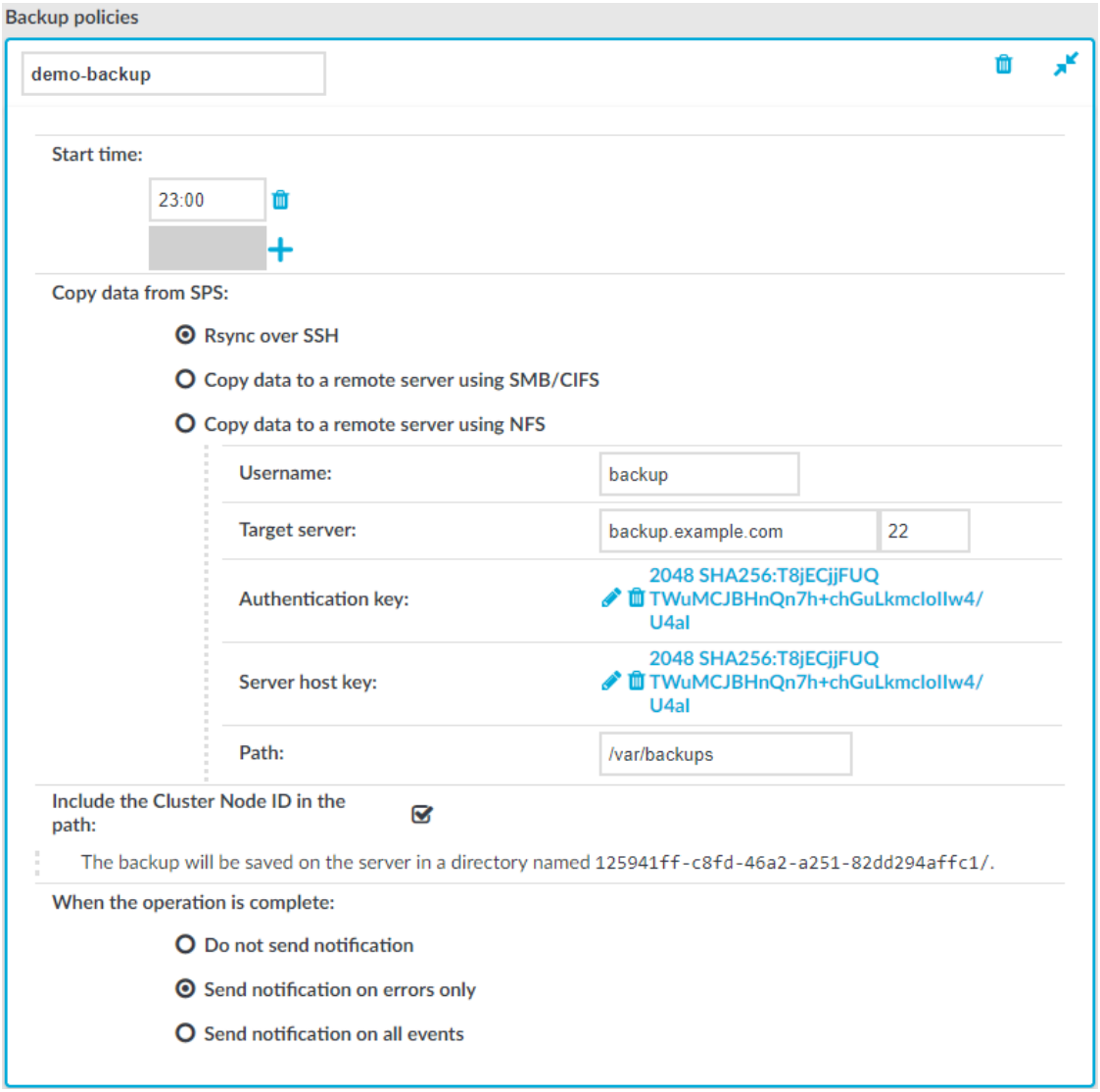
   When specifying an additional start time, ensure that the previous backup process finishes before the new backup process starts.

4. Enter the IP address or the hostname of the remote server into the **Target server** field (for example, `backup.example.com`).

   Use an IPv4 address.

5. Select **Rsync over SSH** from the **Copy data from PSM** radio buttons.

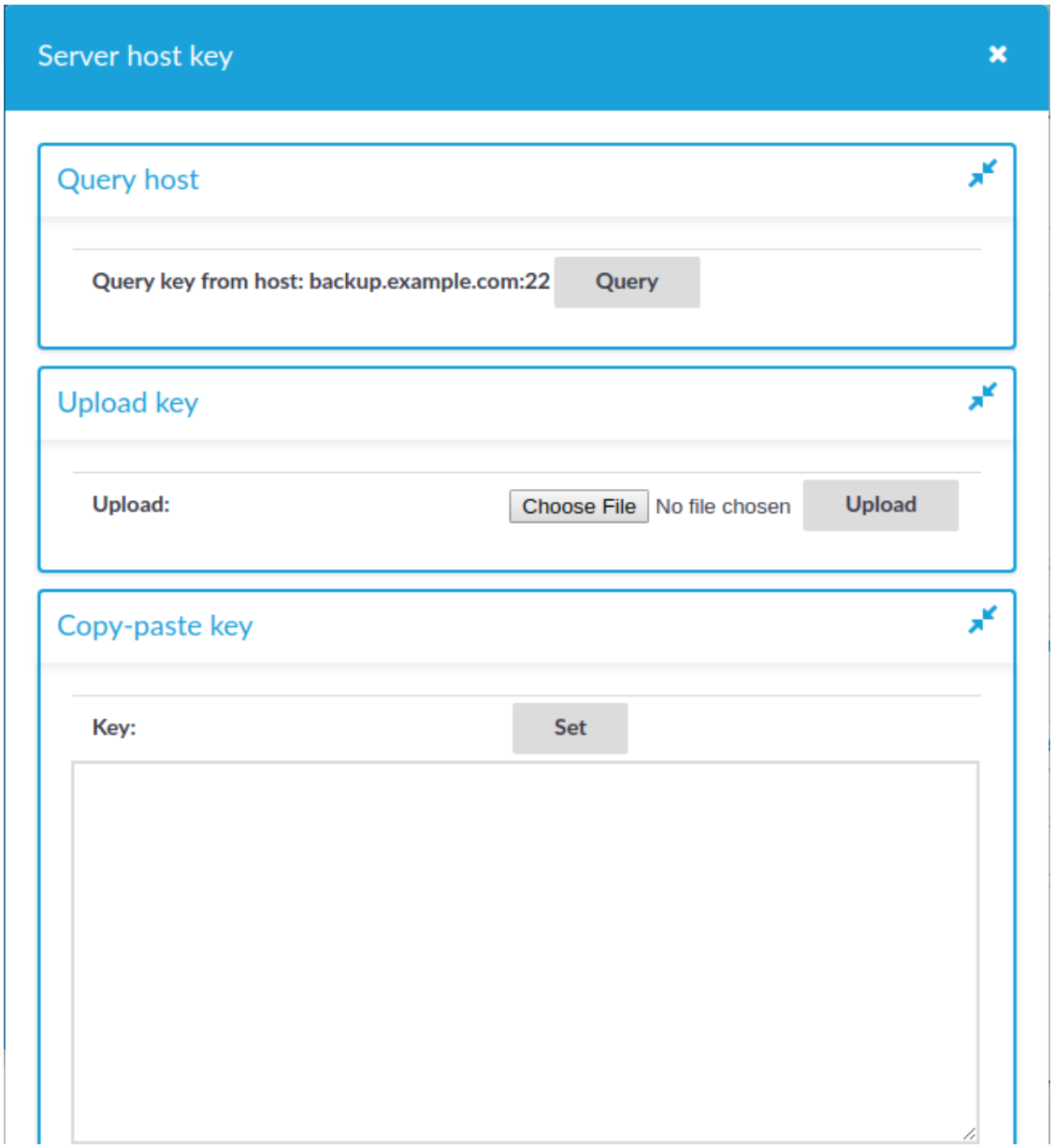
Figure 57: Policies > Backup & Archive/Cleanup > Backup policies — Configuring backups using rsync



6. Enter the username used to log on to the remote server into the **Username** field.
7. Click in the **Authentication key** field. A popup window is displayed.
8. Generate a new keypair by clicking **Generate** or upload or paste an existing one. This key will be used to authenticate SPS on the remote server. The public key of this keypair must be imported to the remote server.

9. Click in the **Server host key** field. A popup window is displayed.
10. Click **Query** to download the host key of the server, or upload or paste the host key manually. SPS will compare the host key shown by the server to this key, and connect only if the two keys are identical.

Figure 58: Policies > Backup & Archive/Cleanup > Backup policies — Configuring SSH keys



11. Enter the port number of the SSH server running on the remote machine into the **Port** field.

12. Enter the path to the backup directory on the target server into the **Path** field (for example `/backups`).

    SPS saves all data into this directory, automatically creating the subdirectories. Backups of audit-trails are stored in the `data`, configuration backups in the `config` subdirectory.

13. When your SPS instance is a node in a cluster, select **Include the Cluster Node ID in the path**. This ensures that the ID of the node is included in the path of the relevant directory, which is required to prevent cluster nodes from backing up data to the same location, and so overwriting each other's data and resulting in data loss. In addition, having the node's ID in the directory name also enables easy identification.

    CAUTION:

    Hazard of data loss

    Unchecking Include the Cluster Node ID in the path when your SPS is a node in a cluster can result in data loss if you have configured configuration synchronization across your cluster nodes.

14. To receive e-mail notification of the backup, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab.

    To include the list of files in the e-mail, select **Send notification on all events** and enable the **Include file list** option. However, note that if the list is very long (for example, SPS stores over 20000 audit trails), the SPS web interface might become unaccessible. In this case, set the **Maximum number of files in notification** lower. After this number has been reached, file names will be omitted from the notification.

    NOTE:

    This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see Configuring system monitoring on SPS on page 137).

15. Click Commit.
16. To assign the backup policy to a connection, see Creating data backups on page 158.

Creating a backup policy using SMB/CIFS

The **Copy data to a remote server using SMB/CIFS** backup method connects to a share on the target server with Server Message Block protocol. SMB/CIFS is mainly used on Microsoft Windows Networks.

When deployed from the Azure Marketplace, you can use Azure File storage shares in your for Backup and Archive Policies. This is very useful as the quota for the files storage can be changed dynamically, so the cumulative size of the audit trails is not limited to the OS disk

size. You can set up this share as a normal SMB shares in your Backup and Archive policies. The parameters for the policy can be obtained from the Azure portal.

NOTE:

Backup and archive policies only work with existing shares and subdirectories.

If a server has a share at, for example, `archive` and that directory is empty, when the user configures `archive/scb1` (or similar) as a backup/archive share, it will fail.

CAUTION:

When you try to create backups and archives from SPS to NetApp devices using the CIFS protocol, the operation may fail with a similar error message: `/opt/scb/mnt/14719217504d41370514043/reports/2010": Permission denied (13) '2010/day/' rsync: failed to set times on.`

To overcome this problem, grant the SPS user "Full Control" access rights to the CIFS share on the NetApp device.

CAUTION:

When using the CIFS protocol to backup or archive files to a target server running Windows 2008 R2 that uses NTLMv2 authentication, the operation may fail with a similar error message:

```
CIFS VFS: Unexpected SMB signature
Status code returned 0xc000000d NT_STATUS_INVALID_PARAMETER
CIFS VFS: Send error in SessSetup = -22
CIFS VFS: cifs_mount failed w/return code = -22
CIFS VFS: Server requires packet signing to be enabled in
/proc/fs/cifs/SecurityFlags.
CIFS VFS: cifs_mount failed w/return code = -95
CIFS VFS: Server requires packet signing to be enabled in
/proc/fs/cifs/SecurityFlags.
CIFS VFS: cifs_mount failed w/return code = -95
```

To overcome this problem, either:

- **use the NFS protocol to access your Windows 2008 R2 servers, or**
- **edit the registry of the Windows 2008 R2 server or apply a hotfix. For details, see Article 957441 in the Microsoft Support site.**

1. Navigate to **Policies > Backup & Archive/Cleanup** and click + in the **Backup policies** section to create a new backup policy.
2. Enter a name for the backup policy (for example, `config-backup`).
3. Enter the time when the backup process should start into the **Start time** field in HH:MM format (for example, `23:00`).

   You can add the start time for additional backup processes.

CAUTION:

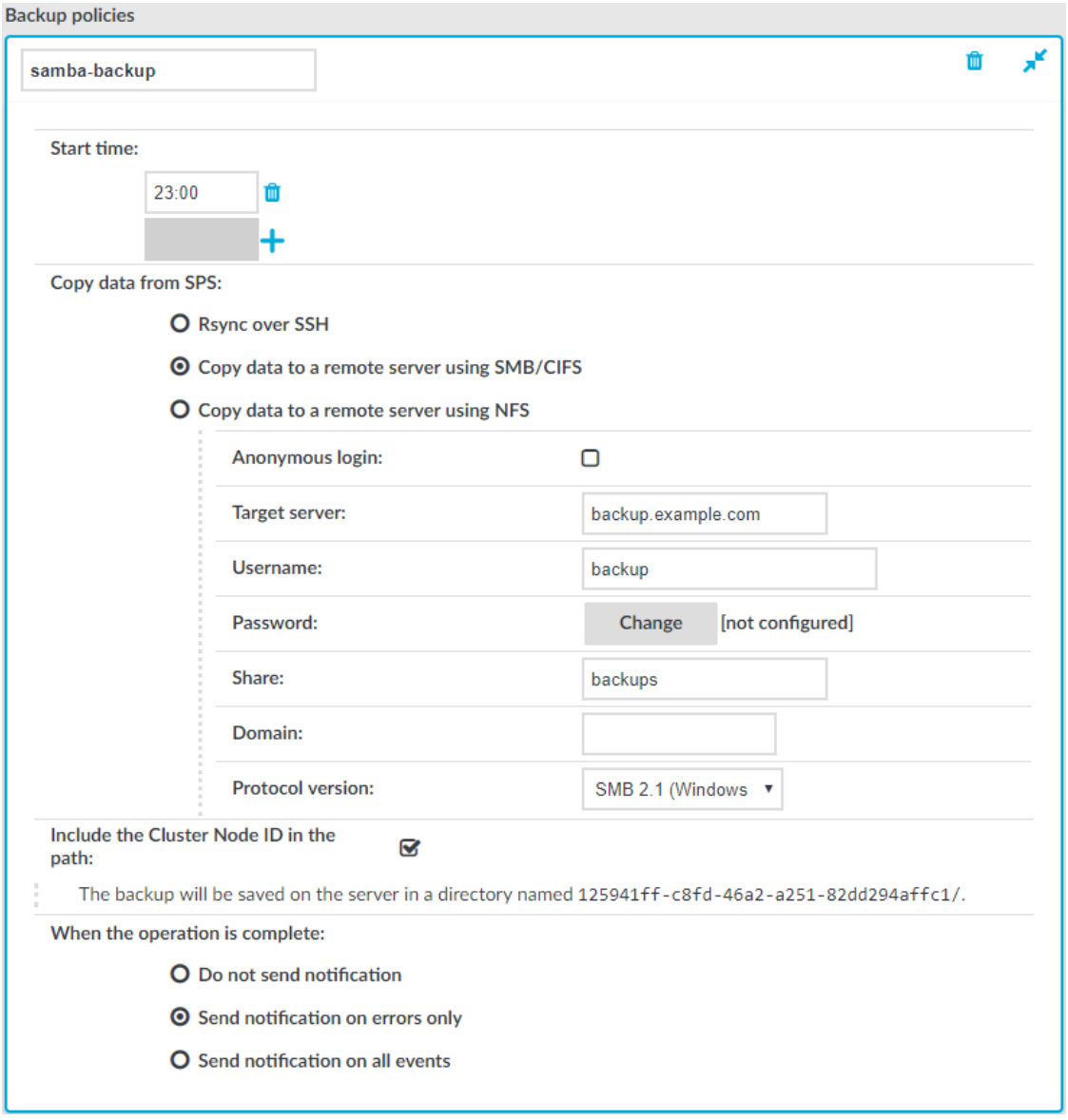
When specifying an additional start time, ensure that the previous backup process finishes before the new backup process starts.

4. Enter the IP address or the hostname of the remote server into the **Target server** field (for example, `backup.example.com`).

   Use an IPv4 address.

5. Select **Copy data to a remote server using SMB/CIFS** from the **Copy data from PSM** radio buttons.

Figure 59: Policies > Backup & Archive/Cleanup > Backup policies — Configuring backups through SMB/CIFS



6. Enter the username used to log on to the remote server into the **Username** field, or select the **Anonymous login** option.

   Usernames can contain space.

7. Enter the password corresponding to the username into the **Password** field.

   > NOTE:
   >
   > One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

8. Enter the name and directory path of the share into the **Share** field. Use the following format:

   ```
   share_name/path/to/directory
   ```

   You can use backslashes and forward slashes as well.

   SPS saves all data into this directory, automatically creating the subdirectories. Backups of audit-trails are stored in the `data`, configuration backups in the `config` subdirectory.

9. Enter the domain name of the target server into the **Domain** field.

10. Select which SMB protocol to use when SPS connects to the server in the **Protocol version** field. Servers are usually backwards compatible with earlier protocol versions (for example, a server that supports version 2.1 supports versions 2.0 and 1.0 as well).

11. When your SPS instance is a node in a cluster, select **Include the Cluster Node ID in the path**. This ensures that the ID of the node is included in the path of the relevant directory, which is required to prevent cluster nodes from backing up data to the same location, and so overwriting each other's data and resulting in data loss. In addition, having the node's ID in the directory name also enables easy identification.

    > **CAUTION:**
    >
    > **Hazard of data loss**
    >
    > **Unchecking Include the Cluster Node ID in the path when your SPS is a node in a cluster can result in data loss if you have configured configuration synchronization across your cluster nodes.**

12. To receive e-mail notification of the backup, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab.

    To include the list of files in the e-mail, select **Send notification on all events** and enable the **Include file list** option. However, note that if the list is very long (for example, SPS stores over 20000 audit trails), the SPS web interface might become unaccessible. In this case, set the **Maximum number of files in notification** lower. After this number has been reached, file names will be omitted from the notification.

NOTE:

This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see Configuring system monitoring on SPS on page 137).

13. Click 

.
14. To assign the backup policy to a connection, see Creating data backups on page 158.

Creating a backup policy using NFS

The **Copy data to a remote server using NFS** backup method connects to a shared directory of the target server with the Network File Share protocol.

NOTE:

Backup and archive policies only work with existing shares and subdirectories.

If a server has a share at, for example, `archive` and that directory is empty, when the user configures `archive/scb1` (or similar) as a backup/archive share, it will fail.

The version of NFS used is automatically detected. All versions of NFS, up to and including NFS version 4 protocol, are supported.

1. Navigate to **Policies > Backup & Archive/Cleanup** and click + in the **Backup policies** section to create a new backup policy.
2. Enter a name for the backup policy (for example, `config-backup`).
3. Enter the time when the backup process should start into the **Start time** field in HH:MM format (for example, `23:00`).

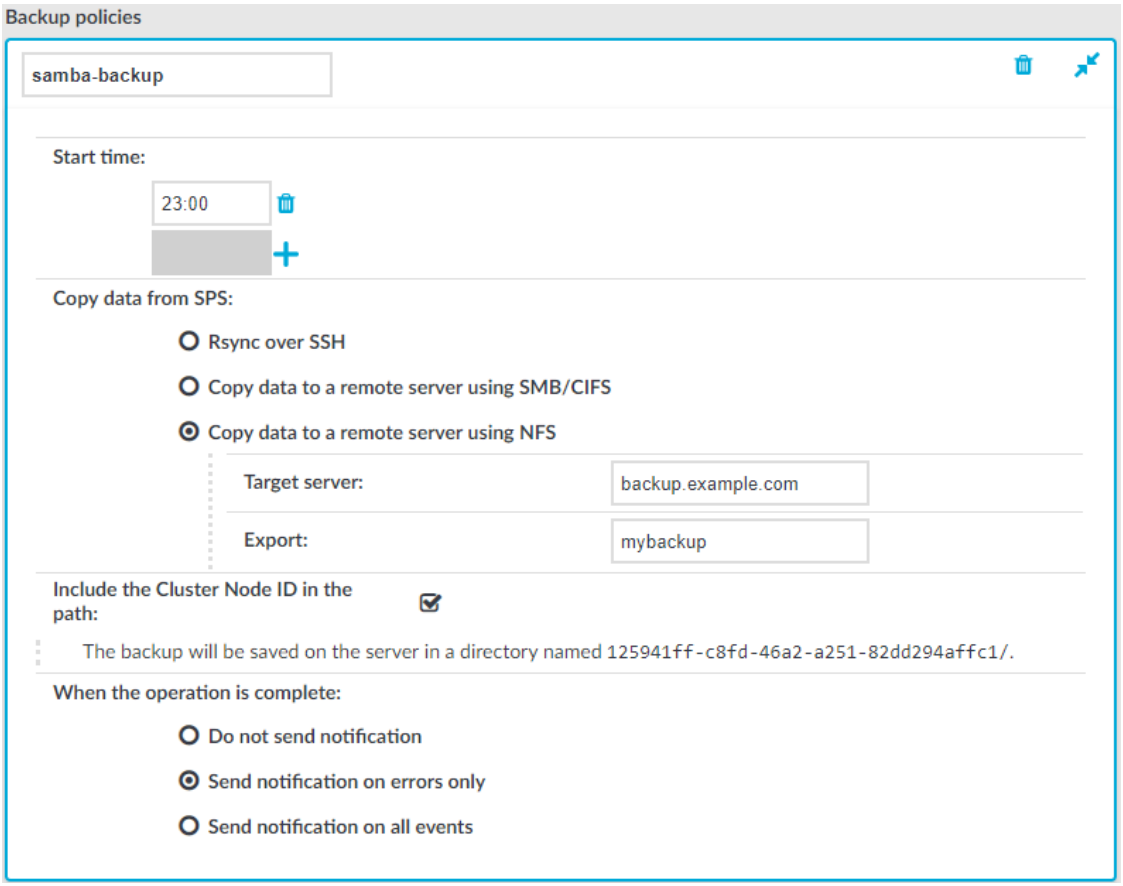
   You can add the start time for additional backup processes.

   CAUTION:

   When specifying an additional start time, ensure that the previous backup process finishes before the new backup process starts.
4. Enter the IP address or the hostname of the remote server into the **Target server** field (for example, `backup.example.com`).

   Use an IPv4 address.
5. Select **Copy data to a remote server using NFS** from the **Copy data from PSM** radio buttons.

Figure 60: Policies > Backup & Archive/Cleanup > Backup policies — Configuring NFS backups



6. Enter the domain name of the remote server into the **Target server** field.
7. Enter the name of the NFS export into the **Export** field.

   SPS saves all data into this directory, automatically creating the subdirectories. Audit-trail backups are stored in the `data`, configuration backups in the `config` subdirectory.
8. The remote server must also be configured to accept backups from SPS.

   Add a line that corresponds to the settings of SPS to the `/etc/exports` file of the backup server. This line should contain the following parameters:

   - The path to the backup directory as set in the **Export** field of the SPS backup policy.
   - The IP address of the SPS interface that is used to access the remote server. For more information on the network interfaces of SPS, see Network settings on page 114.

Use an IPv4 address.

- The following parameters: `(rw,no_root_squash,sync)`.

Example: Configuring NFS on the remote server

For example, if SPS connects the remote server from the `192.168.1.15` IP address and the data is saved into the `/var/backups/SPS` directory, add the following line to the `/etc/exports` file:

```
/var/backups/SPS 192.168.1.15(rw,no_root_squash,sync)
```

9. On the remote server, execute the following command:

   ```
   exportfs -a
   ```

   Verify that the `rpc portmapper` and `rpc.statd` applications are running.

10. When your SPS instance is a node in a cluster, select **Include the Cluster Node ID in the path**. This ensures that the ID of the node is included in the path of the relevant directory, which is required to prevent cluster nodes from backing up data to the same location, and so overwriting each other's data and resulting in data loss. In addition, having the node's ID in the directory name also enables easy identification.

    CAUTION:

    Hazard of data loss

    Unchecking Include the Cluster Node ID in the path when your SPS is a node in a cluster can result in data loss if you have configured configuration synchronization across your cluster nodes.

11. To receive e-mail notification of the backup, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab.

    To include the list of files in the e-mail, select **Send notification on all events** and enable the **Include file list** option. However, note that if the list is very long (for example, SPS stores over 20000 audit trails), the SPS web interface might become unaccessible. In this case, set the **Maximum number of files in notification** lower. After this number has been reached, file names will be omitted from the notification.

    NOTE:

    This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see Configuring system monitoring on SPS on page 137).

12. Click Commit.
13. To assign the backup policy to a connection, see Creating data backups on page 158.

Creating configuration backups

To create a configuration backup, assign a backup policy as the **System backup policy** of SPS.

TIP:

To create an immediate backup of SPS's configuration to your machine (not to the backup server), select **Basic Settings > System > Export configuration**. Note that the configuration export contains only the system settings and configuration files (including changelogs). System backups includes additional information like reports and alerts, and also the connection database.

When exporting the configuration of SPS, or creating configuration backups, always use encryption. Handle the exported data with care, as it contains sensitive information, including credentials. For details on encrypting the configuration, see "Encrypting configuration backups with GPG" in the Administration Guide.

To encrypt your configuration backups, see Encrypting configuration backups with GPG on page 158.

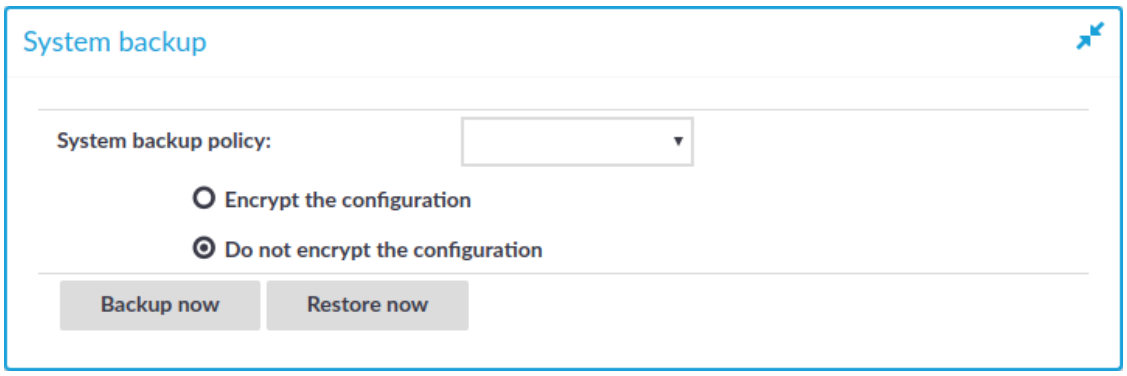
Prerequisites:

You have to configure a backup policy before starting this procedure. For details, see Data and configuration backups.

To create a configuration backup

1. Navigate to **Basic Settings > Management > System backup**.

Figure 61: Basic Settings > Management > System backup — Configuring system backups



2. Select the backup policy you want to use for backing up the configuration of SPS in the **System backup policy** field.
3. Click 

.
4. *Optional:* To start the backup process immediately, click **Backup now**. The **Backup now** functionality works only after a backup policy has been selected and committed.

Creating data backups

To configure data backups, assign a backup policy to the connection.

NOTE:

When exporting the configuration of SPS, or creating configuration backups, always use encryption. Handle the exported data with care, as it contains sensitive information, including credentials. For details on encrypting the configuration, see "Encrypting configuration backups with GPG" in the Administration Guide.

Prerequisites:

- Configure the system backup. Restoring a data backup works only if a matching system configuration and metadata is available, that is, if a system backup is restored first. For details, see Creating configuration backups on page 157.
- Configure a backup policy before starting this procedure. For details, see Data and configuration backups on page 146.

To configure data backups

1. Navigate to **[Your chosen protocol] Control > Connections**.
2. Select the connection you want to back up.
3. Select a backup policy in the **Backup policy** field.
4. Click 

.
5. *Optional:* To start the backup process immediately, click **Backup** or **Backup ALL**. The **Backup** and **Backup ALL** functionalities work only after a backup policy has been selected and committed.

Encrypting configuration backups with GPG

You can encrypt the configuration file of SPS during system backups using the public-part of a GPG key. The system backups of SPS contain other information as well (for example, databases), but only the configuration file is encrypted. Note that system backups do not contain audit-trail data.

When exporting the configuration of SPS, or creating configuration backups, always use encryption. Handle the exported data with care, as it contains sensitive information,

including credentials. For details on encrypting the configuration, see "Encrypting configuration backups with GPG" in the Administration Guide.

For details on restoring configuration from a configuration backup, see Restoring One Identity Safeguard for Privileged Sessions (SPS) configuration and data on page 883.

NOTE:

It is not possible to directly import a GPG-encrypted configuration into SPS, it has to be decrypted locally first.

Prerequisites:

You have to configure a backup policy before starting this procedure. For details, see Data and configuration backups on page 146.

You need a GPG key which must be permitted to encrypt data. Keys that can be used only for signing cannot be used to encrypt the configuration file.

To encrypt the configuration file of SPS during system backup

1. Navigate to **Basic Settings > Management > System backup**.
2. Select **Encrypt configuration**.
3. Click .
   - To upload a key file, click **Browse**, select the file containing the public GPG key, and click **Upload**. SPS accepts both binary and ASCII-armored GPG keys.
   - To copy-paste the key from the clipboard, copy it, paste it into the **Key** field, then click **Set**.
4. Click Commit.

Archiving and cleanup

Archiving transfers data from SPS to an external storage solution, cleanup removes (deletes) old files. Archived data can be accessed and searched, but cannot be restored (moved back) to the SPS appliance. Only those closed audit-trail files are archived where the retention time has already elapsed.

To configure archiving and cleanup, you first have to create an archive/cleanup policy. Archive/cleanup policies define the retention time, the address of the remote backup server, which protocol to use to access it, and other parameters. SPS can be configured to use the SMB/CIFS and NFS protocols to access the backup server:

- To configure a cleanup policy that does not archive data to a remote server, see Creating a cleanup policy on page 160.
- To configure archiving using SMB/CIFS, see Creating an archive policy using SMB/CIFS on page 161.

- To configure archiving using NFS, see Creating an archive policy using NFS on page 166.

CAUTION:

Hazard of data loss Never delete an Archive Policy if data has been archived with it. This will make the already archived data inaccessible.

Do not "remake" an Archive Policy (that is, deleting an Archive Policy and then creating another one with the same name but different parameters). This will make data inaccessible, and identifying the root cause of the issue complicated.

If you want to change the connection parameters (that is when you perform a storage server migration), you must make sure that the share contents and file permissions are kept unmodified and there are no archiving or backup tasks running.

On the other hand, if you want to add a new network share to your archives, proceed with the following steps:

1. **Create a new empty SMB/NFS network share.**
2. **Create a new Archive Policy that points to this network share.**
3. **Modify your Connection Policy(es) to archive using the newly defined Archive Policy.**
4. **Make sure to leave the existing Archive Policy unmodified.**

It is also safe to extend the size of the network share on the server side.

The different protocols assign different file ownerships to the files saved on the remote server. The owners of the archives created using the different protocols are the following:

- *SMB/CIFS*: The user provided on the web interface.
- *NFS*: `root` with `no-root-squash`, `nobody` otherwise.

CAUTION:

SPS cannot modify the ownership of a file that already exists on the remote server.

Once you have configured an archive/cleanup policy, assign it to the connection you want to archive. For details, see Archiving or cleaning up the collected data on page 169.

Data about archived connections can be automatically deleted from the connection database. For details, see Configuring cleanup for the One Identity Safeguard for Privileged Sessions (SPS) connection database on page 488.

Creating a cleanup policy

Cleanup permanently deletes all audit trails and data that is older than **Delete data from SPS after** without creating a backup copy or an archive. Such data is irrecoverably lost. Use this option with care.

NOTE:

This policy does not delete existing archives from an external CIFS or NFS server.

1. Navigate to **Policies > Backup & Archive/Cleanup** and click + in the **Archive/Cleanup policies** section to create a new cleanup policy.
2. Enter a name for the cleanup policy.
3. Enter the time when the cleanup process should start into the **Start time** field in HH:MM format (for example `23:00`).

   You can add the start time for additional cleanup processes.

   CAUTION:

   When specifying an additional start time, ensure that the previous cleanup process finishes before the new cleanup process starts.
4. To cleanup the data collected on SPS more than once a day, click +. You can schedule multiple cleanup times.

   NOTE:

   In case a cleanup process is not finished before the next one would start, the next cleanup process waits for the previous process to be completed.
5. Fill the **Delete data from SPS after** field. Data older than this value is deleted from SPS.
6. To receive e-mail notifications, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab, and include the list of the files that were backed up.

   NOTE:

   This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see Configuring system monitoring on SPS on page 137).
7. Click Commit.
8. To assign the cleanup policy to the connection you want to clean up, see Archiving or cleaning up the collected data on page 169.

Creating an archive policy using SMB/CIFS

The **Move data to a remote server using SMB/CIFS** archive method connects to a share on the target server with Server Message Block protocol. SMB/CIFS is mainly used on Microsoft Windows Networks.

NOTE:

Backup and archive policies only work with existing shares and subdirectories.

If a server has a share at, for example, `archive` and that directory is empty, when the user configures `archive/scb1` (or similar) as a backup/archive share, it will fail.

When deployed from the Azure Marketplace, you can use Azure File storage shares in your for Backup and Archive Policies. This is very useful as the quota for the files storage can be changed dynamically, so the cumulative size of the audit trails is not limited to the OS disk size. You can set up this share as a normal SMB shares in your Backup and Archive policies. The parameters for the policy can be obtained from the Azure portal.

CAUTION:

When you try to create backups and archives from SPS to NetApp devices using the CIFS protocol, the operation may fail with a similar error message: `/opt/scb/mnt/14719217504d41370514043/reports/2010": Permission denied (13) '2010/day/' rsync: failed to set times on.`

To overcome this problem, grant the SPS user "Full Control" access rights to the CIFS share on the NetApp device.

CAUTION:

When using the CIFS protocol to backup or archive files to a target server running Windows 2008 R2 that uses NTLMv2 authentication, the operation may fail with a similar error message:

```
CIFS VFS: Unexpected SMB signature
Status code returned 0xc000000d NT_STATUS_INVALID_PARAMETER
CIFS VFS: Send error in SessSetup = -22
CIFS VFS: cifs_mount failed w/return code = -22
CIFS VFS: Server requires packet signing to be enabled in
/proc/fs/cifs/SecurityFlags.
CIFS VFS: cifs_mount failed w/return code = -95
CIFS VFS: Server requires packet signing to be enabled in
/proc/fs/cifs/SecurityFlags.
CIFS VFS: cifs_mount failed w/return code = -95
```

To overcome this problem, either:

- **use the NFS protocol to access your Windows 2008 R2 servers, or**
- **edit the registry of the Windows 2008 R2 server or apply a hotfix. For details, see Article 957441 in the Microsoft Support site.**

1. Navigate to **Policies > Backup & Archive/Cleanup** and click + in the **Archive/Cleanup policies** section to create a new archive policy.
2. Enter a name for the archive policy.
3. Enter the time when the archive process should start into the **Start time** field in HH:MM format (for example `23:00`).

You can add the start time for additional archive processes.

> **CAUTION:**
>
> **When specifying an additional start time, ensure that the previous archive process finishes before the new archive process starts.**

4. To archive the data collected on SPS more than once a day, click +. You can schedule multiple archive times.

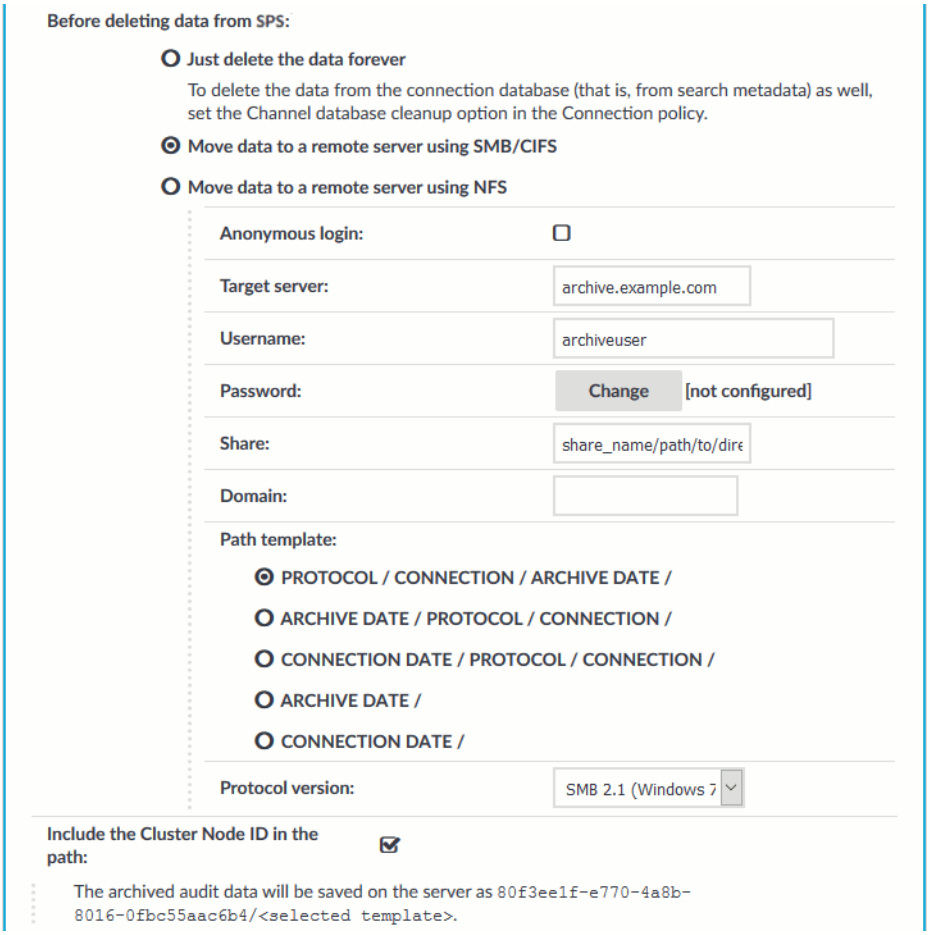
   > NOTE:
   >
   > In case an archive process is not finished before the next one would start, the next archive process waits for the previous process to be completed.

5. Fill the **Delete data from SPS after** field. Data older than this value is archived to the external server.

   > NOTE:
   >
   > The archived data is deleted from SPS.

6. Select **Move data to a remote server using SMB/CIFS** from the **Before deleting data from PSM** radio buttons.

Figure 62: Policies > Backup & Archive/Cleanup — Configuring cleanup and archiving



7. Enter the username used to log on to the remote server into the **Username** field, or select the **Anonymous login** option.

   Usernames can contain space.

8. Enter the password corresponding to the username into the **Password** field.

   NOTE:

   One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

9. Enter the name and directory path of the share into the **Share** field. Use the following format:

   ```
   share_name/path/to/directory
   ```

   You can use backslashes and forward slashes as well.

SPS saves all data into this directory, automatically creating the subdirectories. Archives of audit-trails are stored in the `data`, configuration backups in the `config` subdirectory.

10. Enter the domain name of the target server into the **Domain** field.
11. Select which SMB protocol to use when SPS connects to the server in the **Protocol version** field. Servers are usually backwards compatible with earlier protocol versions (for example, a server that supports version 2.1 supports versions 2.0 and 1.0 as well).
12. SPS organizes the audit trails into directories based on the date or the protocol. The subdirectories are created directly into the archive directory. Select one of the following directory structures:
    - **Protocol/Connection/Archive Date/**
    - **Archive Date/Connection/Protocol/**
    - **Connection Date/Protocol/Connection/**
    - **Archive Date/**
    - **Connection Date/**

    For example, the **Protocol/Connection/Archive Date** template will have create subdirectories for the audited protocols (that is, `ssh, rdp, telnet, vnc`), for the name of the connection policy, and finally, for the date (YEAR-MONTH-DAY in `YYYY-MM-DD` format).

    NOTE:

    Connection Date refers to the time the connection started, while **Archive Date** to the time it was archived. The difference between the two dates depends on the retention time set for the archiving policy.
13. When your SPS instance is a node in a cluster, select **Include the Cluster Node ID in the path**. This ensures that the ID of the node is included in the path of the relevant directory, which is required to prevent cluster nodes from archiving data to the same location, and so overwriting each other's data and resulting in data loss. In addition, having the node's ID in the directory name also enables easy identification.

    CAUTION:

    Hazard of data loss

    Unchecking Include the Cluster Node ID in the path when your SPS is a node in a cluster can result in data loss if you have configured configuration synchronization across your cluster nodes.
14. To receive e-mail notifications, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab, and include the list of the files that were backed up.

NOTE:

This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see Configuring system monitoring on SPS on page 137).

15. Click Commit.
16. To assign the archive policy to the connection you want to archive, see Archiving or cleaning up the collected data on page 169.

Creating an archive policy using NFS

The **Move data to a remote server using NFS** archive method connects to a shared directory of the target server with the Network File Share protocol.

NOTE:

Backup and archive policies only work with existing shares and subdirectories.

If a server has a share at, for example, `archive` and that directory is empty, when the user configures `archive/scb1` (or similar) as a backup/archive share, it will fail.

The version of NFS used is automatically detected. All versions of NFS, up to and including NFS version 4 protocol, are supported.

1. Navigate to **Policies > Backup & Archive/Cleanup** and click + in the **Archive/Cleanup policies** section to create a new archive policy.
2. Enter a name for the archive policy.
3. Enter the time when the archive process should start into the **Start time** field in HH:MM format (for example `23:00`).

   You can add the start time for additional archive processes.

   CAUTION:

   When specifying an additional start time, ensure that the previous archive process finishes before the new archive process starts.

4. To archive the data collected on SPS more than once a day, click +. You can schedule multiple archive times.

   NOTE:

   In case an archive process is not finished before the next one would start, the next archive process waits for the previous process to be completed.

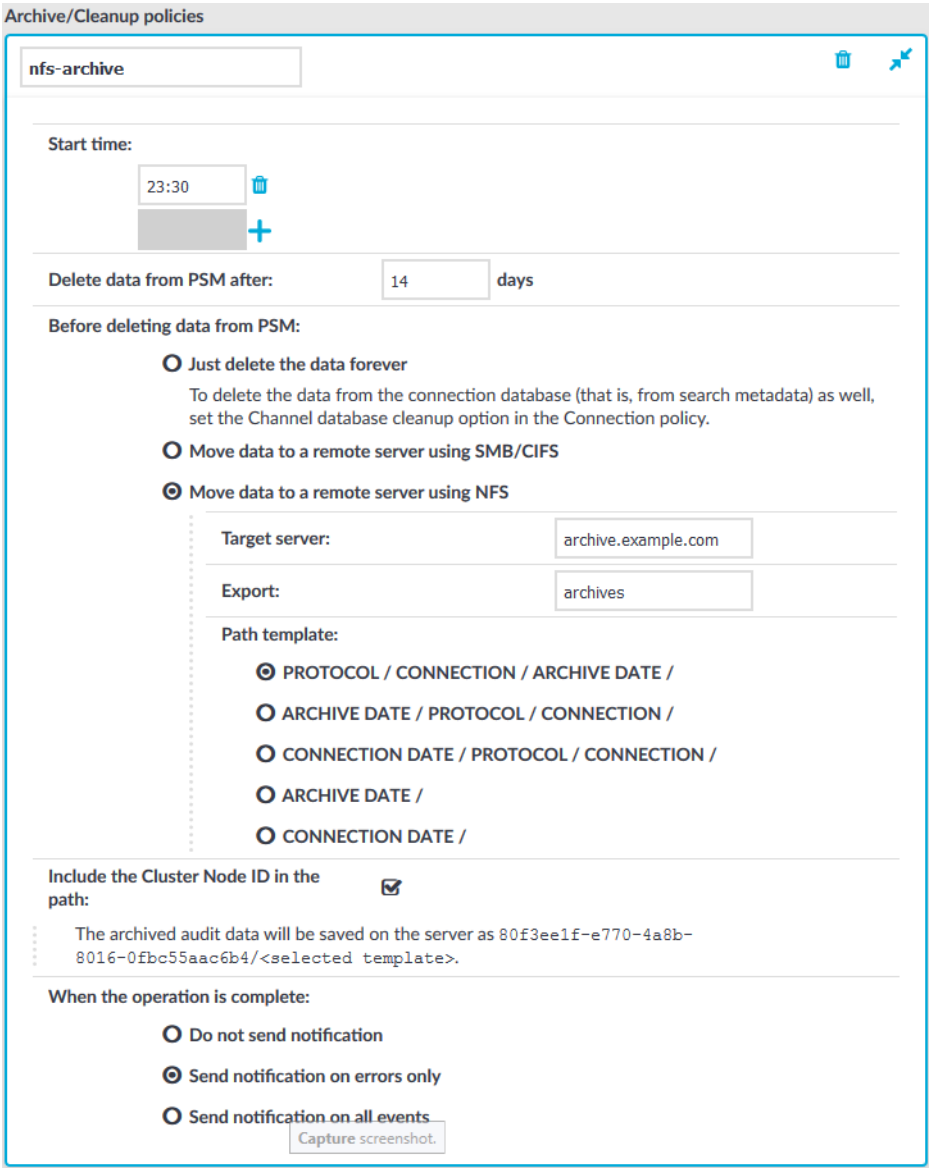
5. Fill the **Delete data from SPS after** field. Data older than this value is archived to the external server.

NOTE:

The archived data is deleted from SPS.

6. Select **Move data to a remote server using NFS** from the **Before deleting data from PSM** radio buttons.

Figure 63: Policies > Backup & Archive/Cleanup —Configuring cleanup and archiving



7. Enter the domain name of the remote server into the **Target server** field.
8. Enter the name of the NFS export into the **Export** field.

   SPS saves all data into this directory, automatically creating the subdirectories.

9. The remote server must also be configured to accept connections from SPS.

   Add a line that corresponds to the settings of SPS to the `/etc/exports` file of the remote server. This line should contain the following parameters:

   - The path to the archive directory as set in the **Export** field of the SPS archive policy.
   - The IP address of the SPS interface that is used to access the remote server. For more information on the network interfaces of SPS, see Network settings on page 114.

     Use an IPv4 address.
   - The following parameters: `(rw,no_root_squash,sync)`.

   Example: Configuring NFS on the remote server

   For example, if SPS connects the remote server from the `192.168.1.15` IP address and the data is saved into the `/var/backups/SPS` directory, add the following line to the `/etc/exports` file:

   ```
   /var/backups/SPS 192.168.1.15(rw,no_root_squash,sync)
   ```

10. On the remote server, execute the following command:

    ```
    exportfs -a
    ```

    Verify that the `rpc portmapper` and `rpc.statd` applications are running.

11. SPS organizes the audit trails into directories based on the date or the protocol. The subdirectories are created directly into the archive directory. Select one of the following directory structures:

    - **Protocol/Connection/Archive Date/**
    - **Archive Date/Connection/Protocol/**
    - **Connection Date/Protocol/Connection/**
    - **Archive Date/**
    - **Connection Date/**

    For example, the **Protocol/Connection/Archive Date** template will have create subdirectories for the audited protocols (that is, `ssh, rdp, telnet, vnc`), for the name of the connection policy, and finally, for the date (YEAR-MONTH-DAY in `YYYY-MM-DD` format).

NOTE:

Connection Date refers to the time the connection started, while **Archive Date** to the time it was archived. The difference between the two dates depends on the retention time set for the archiving policy.

12. When your SPS instance is a node in a cluster, select **Include the Cluster Node ID in the path**. This ensures that the ID of the node is included in the path of the relevant directory, which is required to prevent cluster nodes from archiving data to the same location, and so overwriting each other's data and resulting in data loss. In addition, having the node's ID in the directory name also enables easy identification.

    CAUTION:

    Hazard of data loss

    Unchecking Include the Cluster Node ID in the path when your SPS is a node in a cluster can result in data loss if you have configured configuration synchronization across your cluster nodes.

13. To receive e-mail notifications, select the **Send notification on errors only** or the **Send notification on all events** option. Notifications are sent to the administrator e-mail address set on the **Management** tab, and include the list of the files that were backed up.

    NOTE:

    This e-mail notification is different from the one set on the **Alerting & Monitoring** tab. This notification is sent to the administrator's e-mail address, while the alerts are sent to the alert e-mail address (see Configuring system monitoring on SPS on page 137).

14. Click 

.
15. To assign the archive policy to the connection you want to archive, see Archiving or cleaning up the collected data on page 169.

Archiving or cleaning up the collected data

To configure data archiving/cleanup, assign an archive/cleanup policy to the connection.

Prerequisites:

You have to configure an archive/cleanup policy before starting this procedure. For details, see Archiving and cleanup on page 159.

To assign an archive/cleanup policy to the connection

1. Navigate to the connection (for example to **SSH Control > Connections**).
2. Select the connection.

3. Select the archive/cleanup policy you want to use in the **Archive/Cleanup policy** field.
4. Click 

.
5. *Optional:* To start the archiving or clean up process immediately, click **Archive now**. This functionality works only after a corresponding policy has been configured.

Forwarding data to third-party systems

SPS can forward session data to Splunk, ArcSight, or other third-party systems that enable you to search, analyze, and visualize the forwarded data.

Using the Splunk forwarder

The Splunk forwarder can automatically send file-based data to Splunk. Using the Balabit Privileged Account Analytics, you can integrate this data with your other sources, and access all your data related to privileged user activities from a single interface.

Unlike the universal SIEM forwarder, the Splunk forwarder can forward data based on various criteria such as source or type of event, and, as a result, it is more resource-heavy.

Use the Splunk forwarder if you need to analyze or make changes to the data before you forward it, or you need to control where the data goes based on its contents. For more information, see Using the Splunk forwarder.

NOTE:

Since SPS version 5.11, the universal SIEM forwarder supports Splunk easier than in previous versions. If you want to integrate your SPS with Splunk, One Identity recommends using the universal SIEM forwarder instead of the Splunk forwarder (which will be deprecated as of SPS version 6.4).

Using the universal SIEM forwarder

The universal SIEM forwarder can automatically send data about the audited sessions to Splunk, ArcSight, or other third-party systems. The messages are standard syslog messages in RFC3164 format (also called legacy-syslog or BSD-syslog format). The body of the syslog message (the MESSAGE part) can be formatted as JavaScript Object Notation (JSON), Common Event Format (CEF), or JSON-CIM format. For information about the details of the messages that the universal SIEM forwarder sends to the external SIEM network elements, see Message format forwarded to SIEMs.

One of the main advantages of the universal SIEM forwarder is that it has a lower impact on network and performance.

Each message contains the minimal information relevant to the event. Use the built-in correlation feature of the SIEM to combine events by session ID and view all information in one place.

Use the universal SIEM forwarder if you need a less resource-heavy solution. For more information, see Using the universal SIEM forwarder.

Using the Splunk forwarder

SPS can forward session data to Splunk near real-time. Using the One Identity Safeguard for Privileged Sessions App for Splunk you can integrate this data with your other sources, and access all your data related to privileged user activities from a single interface. To configure SPS to forward session data to Splunk, complete the following steps.

Prerequisites and restrictions:

- SPS version 5 F5 or later
- Splunk version 6.5 or later
- SPS does not send historical data to Splunk, only data from the sessions started after you complete this procedure.

NOTE:

The Splunk forwarder will be deprecated as of version 6.4 of SPS and will be removed in that feature release. One Identity recommends using the Universal SIEM forwarder instead.

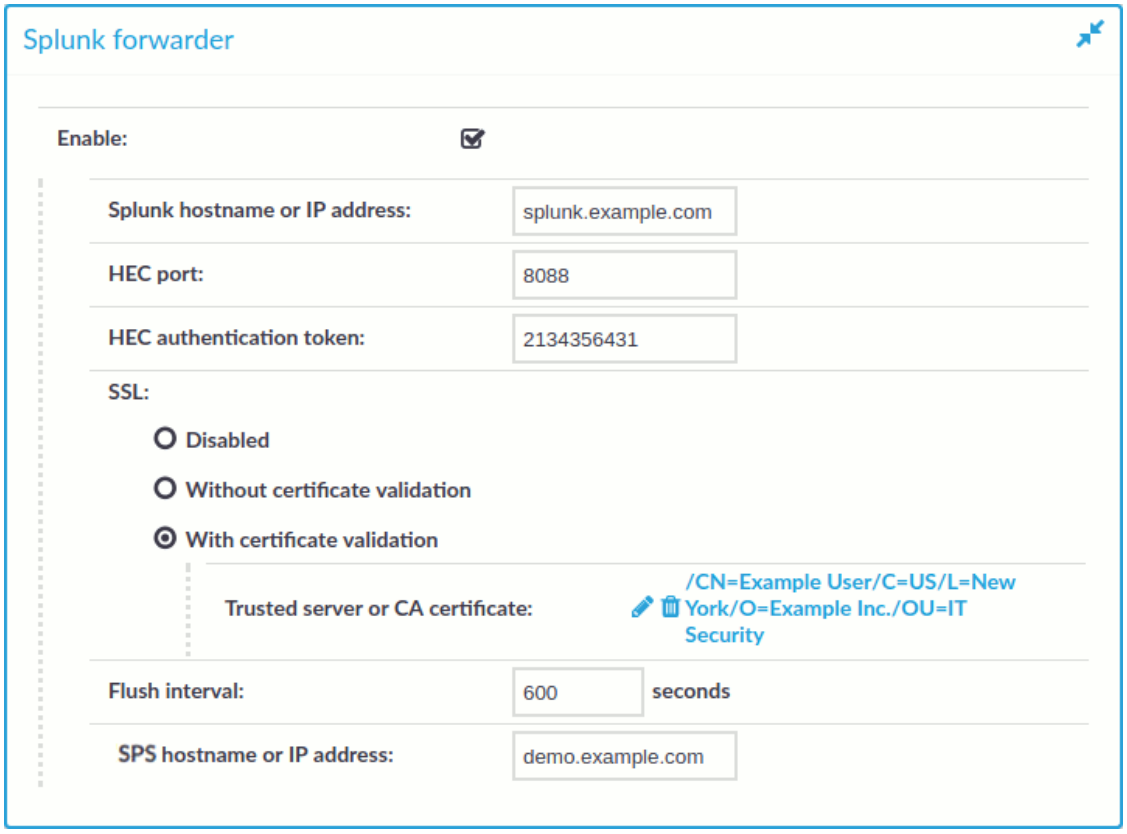
To configure SPS to forward session data to Splunk

1. Install the One Identity Safeguard for Privileged Sessions App for Splunk to your Splunk installation. This will automatically enable and configure the HTTP Event Collector (HEC) in your Splunk installation, and create an HTTP Event Collector authentication token ("HEC token") that SPS will use.

   To help identify the source of the received data, the following settings are configured automatically in the One Identity Safeguard for Privileged Sessions App for Splunk:

   - **index**: The One Identity Safeguard for Privileged Sessions App for Splunk creates the index automatically, with the name `balabit_events`.
   - **sourcetype**: The source type of the events the SPS fowards is `balabit:event`.
2. On your Splunk interface, navigate to **Settings > Data inputs > HTTP Event Collector**. Copy the **Token Value** from the Balabit_HEC field. This is the HTTP Event Collector authentication token and you will need it when configuring SPS.
3. Log in to SPS and navigate to **Basic Settings > Management > Splunk forwarder**.

Figure 64: Basic Settings > Management > Splunk forwarder — Sending session data to Splunk

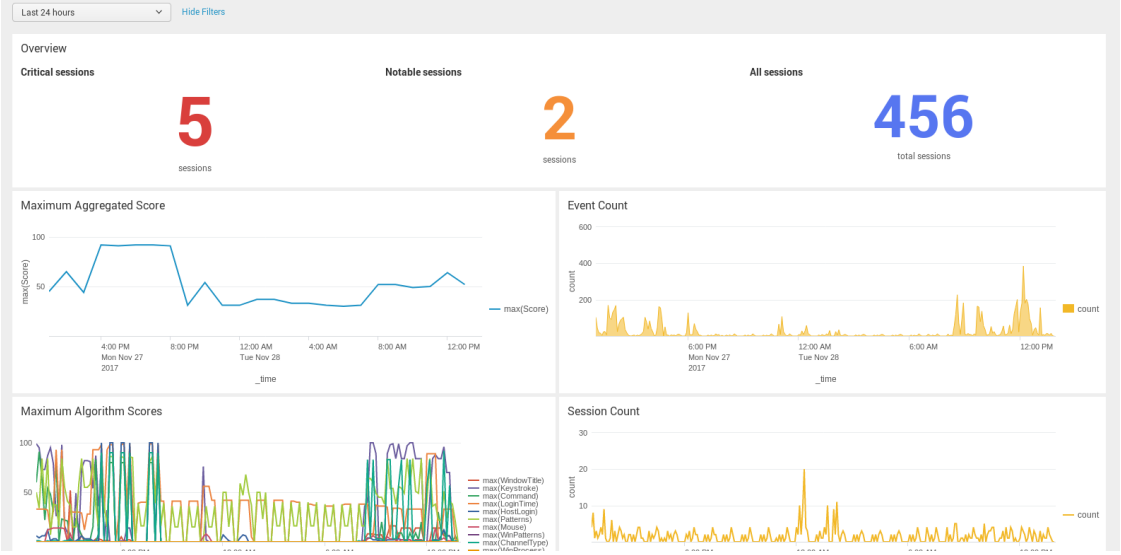


4. Enter the IPv4 address or hostname of your Splunk installation into the **Splunk hostname or IP address** field.
5. Enter the port number where your Splunk HTTP Event Collector is accepting connections into the **HEC port** field. By default, Splunk uses port 8088.
6. Copy the HTTP Event Collector authentication token you have generated for SPS into the **HEC authentication token** field.
7.
   - If your Splunk HTTP Event Collector accepts unencrypted HTTP connections, select **SSL > Disabled**.

     Since the data forwarded to Splunk contains sensitive information, One Identity recommends to use HTTPS encryption between SPS and Splunk.
   - To use HTTPS encryption between SPS and Splunk, select **SSL > Without certificate validation**.
   - To use HTTPS encryption between SPS and Splunk and also verify the identity of the Splunk server, select **SSL > With certificate validation**, then click and upload the certificate of the Splunk server, or the certificate of the CA that issued the certificate of the Splunk server.

8. Splunk will display the data received from SPS as it was received from the host set in the **PAM hostname or IP address** field. By default, this is the hostname and domain name of the SPS appliance as set on the **Basic Settings > Network > Naming** page. Adjust this field as needed for your environment.
9. Click Commit. From now on, SPS will forward session data to Splunk. If the Splunk server becomes unaccessible, SPS will try to resend the data when the period set in **Flush interval** expires.
10. Start a session that SPS will audit to test your configuration, and verify that the data of the session appears in Splunk.

Figure 65: Balabit Privileged Account Analytics



Using the universal SIEM forwarder

The universal SIEM forwarder can automatically send data about the audited sessions to Splunk, ArcSight, or other third-party systems. The messages are standard syslog messages in RFC3164 format (also called legacy-syslog or BSD-syslog format). The body of the syslog message (the MESSAGE part) can be formatted as JavaScript Object Notation (JSON), Common Event Format (CEF), or JSON-CIM format. For information about the details of the messages that the universal SIEM forwarder sends to the external SIEM network elements, see Message format forwarded to SIEMs.

One of the main advantages of the universal SIEM forwarder is that it has a lower impact on network and performance.

Each message contains the minimal information relevant to the event. Use the built-in correlation feature of the SIEM to combine events by session ID and view all information in one place.

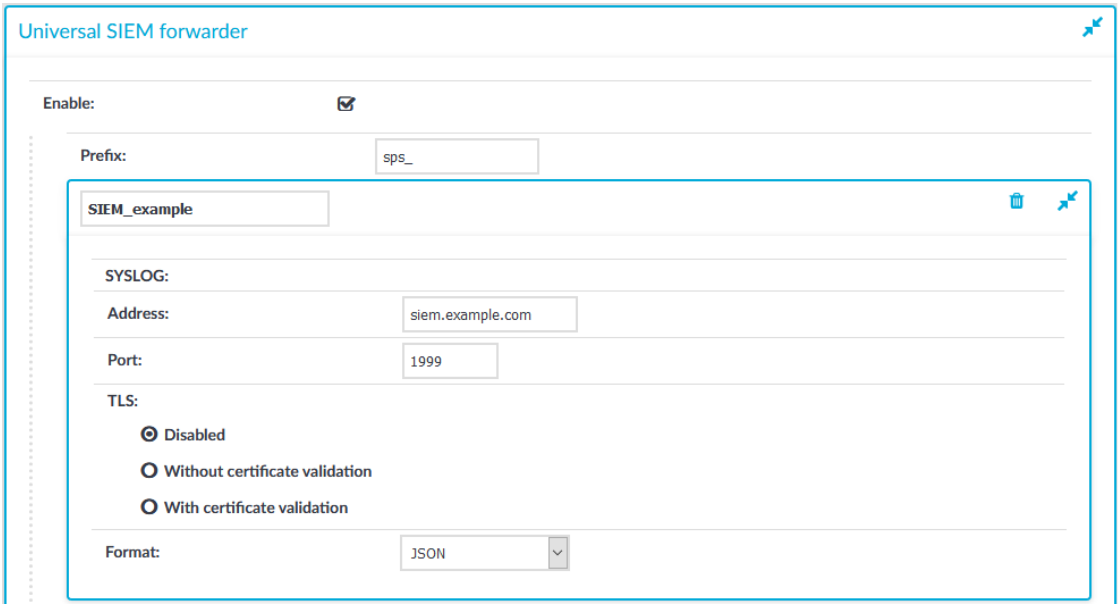
Prerequisites and restrictions

- SPS version 5 F9 or later
- Splunk version 6.5 or later
- The CEF format is supported on all currently supported versions of ArcSight ESM, IBM QRadar and Microsoft Azure Sentinel.
- SPS does not send historical data, only data from the sessions started after you complete this procedure.

To use the universal SIEM forwarder

1. Log in to SPS and navigate to **Basic Settings > Management > Universal SIEM forwarder**.

   Figure 66: Basic Settings > Management > Universal SIEM forwarder — Sending session data to SIEM

2. Enter the IPv4 address or hostname of your third-party system, into the **Address** field.
3. Enter the port number where your third-party system is accepting connections into the **Port** field. For example, if you use Splunk, use port **1999**.
4. - If your third-party system accepts unencrypted connections, select **TLS > Disabled**.

     Since the data forwarded contains sensitive information, One Identity recommends to use TLS encryption between SPS and your SIEM.
   - To use TLS encryption between SPS and your third-party system, select **TLS > Without certificate validation**.

- To use TLS encryption between SPS and your third-party system and also verify the identity of your third-party system server, select **TLS > With certificate validation**, then select the CA list you want to use to validate the certificate of the third-party system in the **Trusted CAs** field. For details on creating trusted CA lists, see Verifying certificates with Certificate Authorities.

5. Select the format of the message as follows:
   - **JSON-CIM**: if using Splunk.
   - **CEF**: if using CEF-compatible SIEMs, for example, Microsoft Sentinel.
   - **JSON**: for general use.
6. (Optional) You can specify a prefix to make the data more readable. Enter the prefix you want to use into the **Prefix** field.

   The prefix is added to each JSON key. For example, if you use **sps_** as a prefix, in the forwarded JSON message the {"protocol": "ssh"} key changes to {"**sps_**protocol": "ssh"}, which allows you to identify the forwarded data more easily.

   Other formats ignore the Prefix option.
7. Click 

. From now on, SPS forwards session data to your third-party system.

Message types forwarded to SIEMs

There are three major categories of messages that One Identity Safeguard for Privileged Sessions (SPS): forwards to the SIEM: content, meta, and score.

- Content messages represents events when SPS detects interesting textual content in the session, such as a command execution or new window title.
- Meta messages represent events that change the session state and/or carry new information about a session.
- Score messages represent scoring events when SPS has calculated an initial score for the session, or updated the score for the session.

The following tables provide a summary of events for the different message types.

Content messages

Table 3: Summary of events for content messages

Event Id	Event Name	Description
127084214	CommandChannelEvent	Emitted when a command is detected in the session text.
911383355	WindowTitleChannelEvent	Emitted when a window title is detected in a graphical session.

Event Id	Event Name	Description
1127618380	FileTransfer	Emitted when SCP file transfer is detected in the SSH protocol.

Meta messages

Table 4: Summary of events for meta messages

Event Id	Event Name	Description
1843867026	GatewayAuthenticationFailure	Emitted if gateway authentication is configured and the user failed to authenticate through the gateway.
1865245228	ServerAuthenticationSuccess	Comes after the server authentication successfully happened.
1262825953	ServerAuthenticationFailure	Emitted if the server authentication failed.
107115592	ServerConnect	Comes after the server authentication successfully happened.
998298775	RdpEmbeddedInTsg	Emitted when the gateway user is acquired in a Terminal Service Gateway authentication scenario. This message will only contain the gateway_username optional field.
1639978560	ServerNameResolved	Emitted when the server_name field was successfully resolved to an ip address. This message will only contain the server_address optional field.
449510124	SessionClosed	Emitted when the session ends.

Score messages

Table 5: Summary of events for score messages

Event Id	Event Name	Description
1991765353	SessionScored	The message contains the aggregate score and one scoring algorithm name and score.

Message format forwarded to SIEMs

The messages are standard syslog messages in RFC3164 format (also called legacy-syslog or BSD-syslog format). The body of the syslog message (the MESSAGE part) can be formatted as one of:

- Common Event Format (CEF), based on the ArcSight CEF specification rev. 16, 22 July 2010
- JavaScript Object Notation (JSON)
- JSON-CIM format (available in SPS version 5.11 and later).

CEF

CEF (Common Event Format): the mapping to CEF will be described in terms of mapping from the JSON format to CEF. In CEF all relevant keys are present, but the value may be empty if it is not known.

Header

Here <...> is substituted with the actual values.

CEF:0|OneIdentity|SPS|<SPS_version>|<event_type_id>|<event_name>|<severity>|

Extensions

CEF extensions that are always present:

cs1: string, equal to session_id

cs1Label: string, equal to literal "Session ID"

start: equal to timestamp

For details on the exact messages and the fields they contain, see CEF messages on page 178.

JSON

JSON (JavaScript Object Notation): the generated JSON structure is flat and the keys in the JSON depend on what kind of event is described. There are some keys that are always present in all messages. There are also keys that are message type specific, but may be missing if the related information is not available.

Keys that are always present and filled:

base_type_name: string, specifies the main category of the message, one of "meta", "content" or "score".

event_type_id: integer, a unique number specifying the message type (primarily for CEF).

event_name: string, the name of the event type.

session_id: string, the unique identifier of the session.

severity: integer, 0-10, the score of the session divided by 10 at the time of the message was created. The value is 0 if the score is not available.

timestamp: string, milliseconds since Unix epoch.

For details on the exact messages and the fields they contain, see JSON messages on page 225.

JSON-CIM

In One Identity Safeguard for Privileged Sessions (SPS) version 5.11 and later versions of SPS, the JSON-CIM external message format is also supported. The JSON-CIM format is a JSON format following Splunk's CIM field names. As a result, Splunk applications can interpret the JSON-CIM format.

For details on the exact messages and the fields they contain, see JSON_CIM messages on page 266.

CEF messages

ServerConnect on initial contact

Description of the message: Emitted when SPS connects to the serverfor the first time in the session

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|107115592|ServerConnect|0|app=SSH cs1=svc-
hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com
dpt=22 dst=10.170.255.206 duser= dvc=10.30.24.20 shost=client.acme.com spt=38014
src=10.30.0.24 start=1554470650290 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
`index 0`	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `107115592`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `ServerConnect`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`cs1`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`cs1Label`	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
`dst`	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
dhost	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
duser	Destination username	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dpt	Destination port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
shost	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
spt	Source port	session	always

Description: the port number on the client

Example: 38014

ServerConnect for secondary channels

Description of the message: Emitted when SPS connects to the serverfor opening further channels. The difference from initial connection is that the server user name is known and authenticated this time.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|107115592|ServerConnect|0|app=SSH cs1=svc-
hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com
dpt=22 dst=10.170.255.206 duser=root dvc=10.30.24.20 shost=client.acme.com
spt=38014 src=10.30.0.24 start=1554470650290 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
index 0	CEF version	product	always

Description:

Example: CEF:0

Field	Name	Scope	Present
index 1	Device vendor	product	always

Description: fixed to OneIdentity

Example: OneIdentity

Field	Name	Scope	Present
index 2	Device product	product	always

Description: fixed to SPS

Example: SPS

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `107115592`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `ServerConnect`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
dvc	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
cs1	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
cs1Label	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
dst	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
dhost	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
duser	Destination username	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`dpt`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`shost`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`suser`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`spt`	Source port	session	always

Description: the port number on the client

Example: `38014`

ServerAuthenticationSuccess

Description of the message: Emitted after the server authentication successfully happened

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|1865245228|ServerAuthenticationSuccess|0|app=SSH
cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID
dhost=server.acme.com dpt=22 dst=10.170.255.206 duser=root dvc=10.30.24.20
shost=client.acme.com spt=38014 src=10.30.0.24 start=1554470652340
suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
`index 0`	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `1865245228`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `ServerAuthenticationSuccess`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
cs1	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
cs1Label	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
dst	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
dhost	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
duser	Destination username	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
dpt	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`shost`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`suser`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`spt`	Source port	session	always

Description: the port number on the client

Example: `38014`

ServerAuthenticationFailure

Description of the message: Emitted after the server authentication failed

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|1262825953|ServerAuthenticationFailure|0|app=SSH
cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID
dhost=server.acme.com dpt=22 dst=10.170.255.206 duser=root dvc=10.30.24.20
shost=client.acme.com spt=38014 src=10.30.0.24 start=1554470652340
suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
`index 0`	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `1262825953`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `ServerAuthenticationFailure`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
index 7	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
start	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
dvc	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
cs1	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
cs1Label	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
dst	Destination address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
dhost	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
duser	Destination username	session	always

Description: contains the non authenticated server username

Example: root

Field	Name	Scope	Present
dpt	Destination port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
shost	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`spt`	Source port	session	always

Description: the port number on the client

Example: `38014`

GatewayAuthenticationFailure

Description of the message: Emitted after a failed gateway authentication. Note that the gateway username here is not authenticated and will not be retained in further messages to avoid confusion with an authenticated gateway user.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|1843867026|GatewayAuthenticationFailure|0|app=SSH
cs1=svc-mBbMWzauBWHQN9TpoZz8mD-my_connection-3 cs1Label=Session ID dhost= dpt=
dst= duser= dvc=10.30.24.20 shost=client.acme.com spt=46296 src=10.30.0.24
start=1557912667169 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
`index 0`	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `1843867026`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `GatewayAuthenticationFailure`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
dvc	Device address	device	always

Description: IP address of SPS

Example: 10.30.24.20

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
cs1	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
cs1Label	Session ID label	product	always

Description: fixed to Session ID

Example: Session ID

Field	Name	Scope	Present
dst	Destination address	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dhost	Destination host name	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
duser	Destination username	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dpt	Destination port	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
shost	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
suser	Source username	message	always

Description: the non authenticated gateway username

Example: `gwtestauto`

Field	Name	Scope	Present
spt	Source port	session	always

Description: the port number on the client

Example: `38014`

SessionClosed of successfully authenticated session

Description of the message: Emitted when the session ends and server authentication and any gateway authentication was successful. There may be further messages related to the session after this message due to post processing of session data!

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|449510124|SessionClosed|0|app=SSH cs1=svc-
hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com
dpt=22 dst=10.170.255.206 duser=root dvc=10.30.24.20 shost=client.acme.com
spt=38014 src=10.30.0.24 start=1554470652340 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
`index 0`	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `449510124`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `SessionClosed`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`cs1`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`cs1Label`	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
`dst`	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`dhost`	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`duser`	Destination username	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`dpt`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
shost	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
spt	Source port	session	always

Description: the port number on the client

Example: 38014

SessionClosed after a failed gateway authentication

Description of the message: Emitted when the session ends because gateway authentication failed.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|449510124|SessionClosed|0|app=SSH cs1=svc-
iiCfsG48oJG5smpuocBLAN-my_connection-25 cs1Label=Session ID dhost= dpt= dst=
duser= dvc=10.30.24.20 shost=client.acme.com spt=54632 src=10.30.0.24
start=1557913042048 suser=
```

The message contains the following fields.

Field	Name	Scope	Present
index 0	CEF version	product	always

Description:

Example: CEF:0

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `449510124`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `SessionClosed`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`cs1`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`cs1Label`	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
dst	Destination address	session	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
dhost	Destination host name	session	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
duser	Destination username	message	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
dpt	Destination port	session	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client
Example: `10.30.0.24`

Field	Name	Scope	Present
shost	Source host name	session	always

Description: the client hostname or IP address if hostname is not known
Example: `client.acme.com`

Field	Name	Scope	Present
suser	Source username	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
spt	Source port	session	always

Description: the port number on the client

Example: 38014

SessionClosed after a failed server authentication

Description of the message: Emitted when the session ends because server authentication failed.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|449510124|SessionClosed|0|app=SSH cs1=svc-
iiCfsG48oJG5smpuocBLAN-my_connection-27 cs1Label=Session ID
dhost=server.acme.com dpt=22 dst=10.170.255.206 duser= dvc=10.30.24.20
shost=client.acme.com spt=55084 src=10.30.0.24 start=1557913066163
suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
index 0	CEF version	product	always

Description:

Example: CEF:0

Field	Name	Scope	Present
index 1	Device vendor	product	always

Description: fixed to OneIdentity

Example: OneIdentity

Field	Name	Scope	Present
index 2	Device product	product	always

Description: fixed to SPS

Example: SPS

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `449510124`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `SessionClosed`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
dvc	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
cs1	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
cs1Label	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
dst	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
dhost	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
duser	Destination username	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dpt	Destination port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
shost	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
spt	Source port	session	always

Description: the port number on the client

Example: 38014

RdpEmbeddedInTsg

Description of the message: Emitted when the gateway user is acquired in a Terminal Service Gateway authentication scenario.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|998298775|RdpEmbeddedInTsg|0|app=RDP cs1=svc-
oUDm7arcL8zNb3t2CVwSQr-my_connection-44-1 cs1Label=Session ID dhost= dpt= dst=
duser= dvc=10.30.24.20 shost=client.acme.com spt=51083 src=10.30.0.24
start=1558006199668 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
`index 0`	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `998298775`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `RdpEmbeddedInTsg`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`cs1`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`cs1Label`	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
`dst`	Destination address	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`dhost`	Destination host name	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`duser`	Destination username	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`dpt`	Destination port	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`src`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
shost	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
suser	Source username	session	always

Description: the authenticated gateway username

Example: gwtestauto

Field	Name	Scope	Present
spt	Source port	session	always

Description: the port number on the client

Example: 38014

SessionScored

Description of the message: Score messages represent scoring events when SPS has calculated an initial or changed score for the session.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|1991765353|SessionScored|7|app=SSH cs1=svc-
822TNSfws1M6qixvRjQX8b-my_connection-4 cs1Label=Session ID cs2=70
cs2Label=Aggregated session score cs3=keystroke cs3Label=Scorer algorithm name
cs4=18 cs4Label=Score given by algorithm dst=10.170.255.206 duser=root
dvc=10.30.24.20 src=10.30.0.24 start=1558008998716 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
index 0	CEF version	product	always

Description:

Example: CEF:0

Field	Name	Scope	Present
index 1	Device vendor	product	always

Description: fixed to OneIdentity

Example: OneIdentity

Field	Name	Scope	Present
index 2	Device product	product	always

Description: fixed to SPS

Example: SPS

Field	Name	Scope	Present
index 3	Device version	product version	always

Description: version of SPS

Example: 5.11.0

Field	Name	Scope	Present
index 4	Signature ID	message	always

Description: numeric identifier of message type

Example: 1991765353

Field	Name	Scope	Present
index 5	Name	message	always

Description: the type of the message

Example: SessionScored

Field	Name	Scope	Present
index 6	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
index 7	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
start	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
dvc	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
cs1	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
cs1Label	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
dst	Destination address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
duser	Destination username	session	always

Description: the server username

Example: root

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
cs2	Aggregated score	message	always

Description: the average score from all enabled analytics algorithms

Example: 50

Field	Name	Scope	Present
cs2Label	Aggregated score label	message	always

Description: fixed to Aggregated session score

Example: Aggregated session score

Field	Name	Scope	Present
cs3	Algorithm name	message	always

Description: the name of the algorithm that changed value

Example: keystroke

Field	Name	Scope	Present
cs3Label	Algorithm name label	message	always

Description: fixed to Scorer algorithm name

Example: Scorer algorithm name

Field	Name	Scope	Present
cs4	Algorithm score	message	always

Description: the new score value of the algorithm that changed value

Example: 60

Field	Name	Scope	Present
cs4Label	Algorithm score label	message	always

Description: fixed to Score given by algorithm

Example: Score given by algorithm

CommandChannelEvent

Description of the message: Emitted when a command is detected in the session channel text.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|127084214|CommandChannelEvent|0|app=SSH cs1=svc-
sZZoAcZZz9CbtCzTKWXgao-my_connection-0 cs1Label=Session ID cs2=exit
cs2Label=Command dst=10.170.255.206 duser=root dvc=10.30.24.20 src=10.30.0.24
start=1556287687858 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
index 0	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `127084214`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `CommandChannelEvent`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`cs1`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`cs1Label`	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
dst	Destination address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
duser	Destination username	session	always

Description: the server username

Example: root

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
cs2	Command	message	always

Description: the full command detected

Example: exit

Field	Name	Scope	Present
cs2Label	Command label	message	always

Description: fixed to Command

Example: Command

WindowTitleChannelEvent

Description of the message: Emitted when a command is detected in the session channel text.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|911383355|WindowTitleChannelEvent|0|app=RDP
cs1=svc-oUDm7arcL8zNb3t2CVwSQr-my_connection-44-4 cs1Label=Session ID
cs2=Shortcut Tools Application Tools Administrative Tools cs2Label=Window title
dst=10.170.255.206 duser=Administrator dvc=10.30.24.20 src=10.30.0.24
start=1558006237095 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
`index 0`	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
`index 1`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`index 2`	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `911383355`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `WindowTitleChannelEvent`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
cs1	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
cs1Label	Session ID label	product	always

Description: fixed to Session ID

Example: Session ID

Field	Name	Scope	Present
dst	Destination address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
duser	Destination username	session	always

Description: the server username

Example: root

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
cs2	Window title	message	always

Description: the window title detected in graphical protocol

Example: `firefox`

Field	Name	Scope	Present
cs2Label	Window title label	message	always

Description: fixed to Window title

Example: `Window title`

FileTransfer

Description of the message: Emitted when a command is detected in the session channel text.

Example message:

```
CEF:0|OneIdentity|SPS|5.11.0|1127618380|FileTransfer|0|act=UPLOAD app=SSH
cs1=svc-2L83Phh9J6GKLWTc881awk-my_connection-308 cs1Label=Session ID
dst=10.170.255.206 duser=root dvc=10.30.24.20 filePath=/cpuinfo fname=cpuinfo
src=10.30.0.24 start=1558023621127 suser=gwtestauto
```

The message contains the following fields.

Field	Name	Scope	Present
index 0	CEF version	product	always

Description:

Example: `CEF:0`

Field	Name	Scope	Present
index 1	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
index 2	Device product	product	always

Description: fixed to SPS

Example: `SPS`

Field	Name	Scope	Present
`index 3`	Device version	product version	always

Description: version of SPS

Example: `5.11.0`

Field	Name	Scope	Present
`index 4`	Signature ID	message	always

Description: numeric identifier of message type

Example: `1127618380`

Field	Name	Scope	Present
`index 5`	Name	message	always

Description: the type of the message

Example: `FileTransfer`

Field	Name	Scope	Present
`index 6`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`index 7`	CEF extensions	product	always

Description: contains the payload in key-value form

Example: `app=SSH cs1=svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0 cs1Label=Session ID dhost=server.acme.com dpt=22 ...`

Field	Name	Scope	Present
`start`	Start time	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`dvc`	Device address	device	always

Description: IP address of SPS

Example: `10.30.24.20`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`cs1`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`cs1Label`	Session ID label	product	always

Description: fixed to Session ID

Example: `Session ID`

Field	Name	Scope	Present
`dst`	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`duser`	Destination username	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
src	Source address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
suser	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
act	Operation	message	always

Description: the operation on the file such as UPLOAD/DOWNLOAD. It may contain the suffix 'WARNING', if the operation failed

Example: UPLOAD

Field	Name	Scope	Present
fname	Filename	message	always

Description: the file name

Example: foobar.txt

Field	Name	Scope	Present
filePath	Full file path	message	always

Description: the name of the file including its path on the server

Example: /tmp/foobar.txt

JSON messages

ServerConnect on initial contact

Description of the message: Emitted when SPS connects to the serverfor the first time in the session

Example message:


```
{"timestamp":"1557913242888","severity":"0","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-43","server_port":"22","server_
name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"107115592","event_
name":"ServerConnect","connection_policy":"my_connection","client_
port":"59190","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: meta

Example: meta

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: 107115592

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: ServerConnect

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`timestamp`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`gateway_username`	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`gateway_domain`	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: `acme.com`

Field	Name	Scope	Present
`server_name`	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`server_address`	Server address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`server_port`	Server port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`client_name`	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`client_address`	Client address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`client_port`	Client port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`protocol`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`connection_policy`	Connection policy name	session	always

Description: SPS connection policy name

Example: `my_connection`

ServerConnect for secondary channels

Description of the message: Emitted when SPS connects to the serverfor opening further channels. The difference from initial connection is that the server user name is known and authenticated this time.

Example message:

```
{"timestamp":"1557913242888","severity":"0","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-43","server_port":"22","server_
name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"107115592","event_
name":"ServerConnect","connection_policy":"my_connection","server_
username":"root","client_port":"59190","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: meta

Example: meta

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: 107115592

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: ServerConnect

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`timestamp`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`server_username`	Server user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`server_domain`	Server user domain if known	session	sometimes

Description: the server domain, if known

Example: `acme.com`

Field	Name	Scope	Present
`gateway_username`	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`gateway_domain`	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: `acme.com`

Field	Name	Scope	Present
`server_name`	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
server_address	Server address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
server_port	Server port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: `my_connection`

ServerAuthenticationSuccess

Description of the message: Emitted after the server authentication successfully happened

Example message:


```
{"timestamp":"1557913243423","severity":"0","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-43","server_username":"root","server_
port":"22","server_name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"1865245228","event_
name":"ServerAuthenticationSuccess","connection_policy":"my_connection","client_
port":"59190","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
`base_type_name`	Basic type	message	always

Description: basic message type: meta

Example: `meta`

Field	Name	Scope	Present
`event_type_id`	Signature ID	message	always

Description: numeric identifier of message type

Example: `1865245228`

Field	Name	Scope	Present
`event_name`	Event name	message	always

Description: the type of the message

Example: `ServerAuthenticationSuccess`

Field	Name	Scope	Present
`session_id`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`severity`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`timestamp`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`server_username`	Server user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`server_domain`	Server user domain if known	session	sometimes

Description: the server domain, if known

Example: `acme.com`

Field	Name	Scope	Present
`gateway_username`	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`gateway_domain`	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: `acme.com`

Field	Name	Scope	Present
`server_name`	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`server_address`	Server address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`server_port`	Server port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`client_name`	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`client_address`	Client address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`client_port`	Client port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`protocol`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`connection_policy`	Connection policy name	session	always

Description: SPS connection policy name

Example: `my_connection`

ServerAuthenticationFailure

Description of the message: Emitted after the server authentication failed

Example message:


```
{"timestamp":"1557913134598","severity":"0","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-33","server_username":"root","server_
port":"22","server_name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"1262825953","event_
name":"ServerAuthenticationFailure","connection_policy":"my_connection","client_
port":"56692","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
`base_type_name`	Basic type	message	always

Description: basic message type: meta

Example: `meta`

Field	Name	Scope	Present
`event_type_id`	Signature ID	message	always

Description: numeric identifier of message type

Example: `1262825953`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: ServerAuthenticationFailure

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
timestamp	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
server_username	Server user	session	always

Description: contains the non authenticated server username

Example: root

Field	Name	Scope	Present
server_domain	Server user domain if known	session	sometimes

Description: the non authenticated server domain, if known

Example: acme.com

Field	Name	Scope	Present
gateway_username	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
gateway_domain	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: acme.com

Field	Name	Scope	Present
server_name	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
server_address	Server address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
server_port	Server port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: my_connection

GatewayAuthenticationFailure

Description of the message: Emitted after a failed gateway authentication. Note that the gateway username here is not authenticated and will not be retained in further messages to avoid confusion with an authenticated gateway user.

Example message:


```
{"timestamp":"1557913110027","severity":"0","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-31","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"1843867026","event_
name":"GatewayAuthenticationFailure","connection_policy":"my_
connection","client_port":"56020","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: meta

Example: meta

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: 1843867026

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: GatewayAuthenticationFailure

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
timestamp	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
gateway_username	Gateway username	message	always

Description: the non authenticated gateway username

Example: gwtestauto

Field	Name	Scope	Present
gateway_username	Gateway user domain	session	sometimes

Description: the non authenticated gateway user domain if known

Example: acme.com

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: `my_connection`

SessionClosed of successfully authenticated session

Description of the message: Emitted when the session ends and server authentication and any gateway authentication was successful. There may be further messages related to the session after this message due to post processing of session data!

Example message:


```
{"timestamp":"1557912701233","severity":"0","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-6","server_username":"root","server_
port":"22","server_name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"449510124","event_
name":"SessionClosed","connection_policy":"my_connection","client_
port":"46958","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta","auth_method":"password"}
```


The message contains the following fields.

Field	Name	Scope	Present
`base_type_name`	Basic type	message	always

Description: basic message type: meta

Example: `meta`

Field	Name	Scope	Present
`event_type_id`	Signature ID	message	always

Description: numeric identifier of message type

Example: `449510124`

Field	Name	Scope	Present
`event_name`	Event name	message	always

Description: the type of the message

Example: `SessionClosed`

Field	Name	Scope	Present
`session_id`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`severity`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`timestamp`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`server_username`	Server user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`server_domain`	Server user domain if known	session	sometimes

Description: the server domain, if known

Example: `acme.com`

Field	Name	Scope	Present
`gateway_username`	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`gateway_domain`	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: `acme.com`

Field	Name	Scope	Present
`server_name`	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`server_address`	Server address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`server_port`	Server port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`client_name`	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`client_address`	Client address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`client_port`	Client port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`protocol`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`connection_policy`	Connection policy name	session	always

Description: SPS connection policy name

Example: `my_connection`

Field	Name	Scope	Present
`auth_method`	Authentication method	session	always

Description: the type of authentication used in gateway authentication

Example: `password`

SessionClosed after a failed gateway authentication

Description of the message: Emitted when the session ends because gateway authentication failed.

Example message:


```
{"timestamp":"1557912725391","severity":"0","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-9","protocol":"SSH","event_type_
id":"449510124","event_name":"SessionClosed","connection_policy":"my_
connection","client_port":"47444","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
`base_type_name`	Basic type	message	always

Description: basic message type: meta

Example: `meta`

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: 449510124

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: SessionClosed

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
timestamp	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: my_connection

SessionClosed after a failed server authentication

Description of the message: Emitted when the session ends because server authentication failed.

Example message:


```
{"timestamp":"1557912748990","severity":"0","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-11","server_port":"22","server_
name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"449510124","event_
name":"SessionClosed","connection_policy":"my_connection","client_
port":"47840","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: meta

Example: meta

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: 449510124

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: SessionClosed

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
timestamp	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
gateway_username	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
gateway_domain	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: acme.com

Field	Name	Scope	Present
server_name	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
server_address	Server address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
server_port	Server port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: my_connection

RdpEmbeddedInTsg

Description of the message: Emitted when the gateway user is acquired in a Terminal Service Gateway authentication scenario.

Example message:


```
{"timestamp":"1558007294417","severity":"0","session_id":"svc-
oUDm7arcL8zNb3t2CVwSQr-my_connection-50-4","protocol":"RDP","gateway_
username":"gwtestauto","event_type_id":"998298775","event_
name":"RdpEmbeddedInTsg","connection_policy":"my_connection","client_
port":"51270","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"meta"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: meta

Example: meta

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: 998298775

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: RdpEmbeddedInTsg

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
timestamp	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
gateway_username	Gateway username	session	always

Description: the authenticated gateway username

Example: gwtestauto

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: my_connection

SessionScored

Description of the message: Score messages represent scoring events when SPS has calculated an initial or changed score for the session.

Example message:


```
{"timestamp":"1558009822701","severity":"7","session_id":"svc-
62a6XGcPzaFvLYDhVYDYXj-my_connection-0","server_username":"root","server_
port":"22","server_name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"1991765353","event_
name":"SessionScored","connection_policy":"my_connection","client_
port":"35620","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"score","auth_
method":"password","algorithm_score":"18","algorithm_
name":"keystroke","aggregated_score":"70"}
```


The message contains the following fields.

Field	Name	Scope	Present
`base_type_name`	Basic type	message	always

Description: basic message type: score

Example: `score`

Field	Name	Scope	Present
`event_type_id`	Signature ID	message	always

Description: numeric identifier of message type

Example: `1991765353`

Field	Name	Scope	Present
`event_name`	Event name	message	always

Description: the type of the message

Example: `SessionScored`

Field	Name	Scope	Present
`session_id`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
timestamp	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
server_username	Server user	session	always

Description: the server username

Example: root

Field	Name	Scope	Present
server_domain	Server user domain if known	session	sometimes

Description: the server domain, if known

Example: acme.com

Field	Name	Scope	Present
gateway_username	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
gateway_domain	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: acme.com

Field	Name	Scope	Present
server_name	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
server_address	Server address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
server_port	Server port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`connection_policy`	Connection policy name	session	always

Description: SPS connection policy name

Example: `my_connection`

Field	Name	Scope	Present
`auth_method`	Authentication method	session	always

Description: the type of authentication used in gateway authentication

Example: `password`

Field	Name	Scope	Present
`aggregated_score`	Aggregated score	message	always

Description: the average score from all enabled analytics algorithms

Example: `50`

Field	Name	Scope	Present
`algorithm_name`	Algorithm name	message	always

Description: the name of the algorithm that changed value

Example: `keystroke`

Field	Name	Scope	Present
`algorithm_score`	Algorithm score	message	always

Description: the new score value of the algorithm that changed value

Example: `60`

CommandChannelEvent

Description of the message: Emitted when a command is detected in the session channel text.

Example message:

```
{"timestamp":"1557912701166","severity":"0","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-6","server_username":"root","server_
port":"22","server_name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","event_type_id":"127084214","event_
name":"CommandChannelEvent","connection_policy":"my_
connection","command":"exit","client_port":"46958","client_
name":"client.acme.com","client_address":"10.30.0.24","base_type_
name":"content","auth_method":"password"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: content

Example: content

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: 127084214

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: CommandChannelEvent

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`timestamp`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`server_username`	Server user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`server_domain`	Server user domain if known	session	sometimes

Description: the server domain, if known

Example: `acme.com`

Field	Name	Scope	Present
`gateway_username`	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`gateway_domain`	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: `acme.com`

Field	Name	Scope	Present
`server_name`	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`server_address`	Server address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`server_port`	Server port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`client_name`	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`client_address`	Client address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`client_port`	Client port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`protocol`	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: my_connection

Field	Name	Scope	Present
auth_method	Authentication method	session	always

Description: the type of authentication used in gateway authentication

Example: password

Field	Name	Scope	Present
command	Command	message	always

Description: the full command detected

Example: exit

WindowTitleChannelEvent

Description of the message: Emitted when a command is detected in the session channel text.

Example message:


```
{"window_title":"Shortcut Tools Application Tools Administrative
Tools","timestamp":"1558007305516","severity":"0","session_id":"svc-
oUDm7arcL8zNb3t2CVwSQr-my_connection-50-4","server_
username":"Administrator","server_port":"3389","server_
name":"server.acme.com","server_
address":"10.170.255.206","protocol":"RDP","gateway_
username":"gwtestauto","event_type_id":"911383355","event_
name":"WindowTitleChannelEvent","connection_policy":"my_connection","client_
port":"51270","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"content"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: content

Example: `content`

Field	Name	Scope	Present
`event_type_id`	Signature ID	message	always

Description: numeric identifier of message type

Example: `911383355`

Field	Name	Scope	Present
`event_name`	Event name	message	always

Description: the type of the message

Example: `WindowTitleChannelEvent`

Field	Name	Scope	Present
`session_id`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`severity`	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: `0`

Field	Name	Scope	Present
`timestamp`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`server_username`	Server user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
server_domain	Server user domain if known	session	sometimes

Description: the server domain, if known

Example: acme.com

Field	Name	Scope	Present
gateway_username	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
gateway_domain	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: acme.com

Field	Name	Scope	Present
server_name	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
server_address	Server address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
server_port	Server port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: SSH

Field	Name	Scope	Present
connection_policy	Connection policy name	session	always

Description: SPS connection policy name

Example: my_connection

Field	Name	Scope	Present
window_title	Window title	message	always

Description: the window title detected in graphical protocol

Example: firefox

FileTransfer

Description of the message: Emitted when a command is detected in the session channel text.

Example message:


```
{"timestamp":"1558023671115","severity":"0","session_id":"svc-
2L83Phh9J6GKLWTc881awk-my_connection-316","server_username":"root","server_
port":"22","server_name":"server.acme.com","server_
address":"10.170.255.206","protocol":"SSH","gateway_
username":"gwtestauto","filepath":"","filename":"cpuinfo","file_
operation":"UPLOAD","event_type_id":"1127618380","event_
name":"FileTransfer","connection_policy":"my_connection","client_
port":"44292","client_name":"client.acme.com","client_
address":"10.30.0.24","base_type_name":"content","auth_method":"password"}
```


The message contains the following fields.

Field	Name	Scope	Present
base_type_name	Basic type	message	always

Description: basic message type: content

Example: `content`

Field	Name	Scope	Present
event_type_id	Signature ID	message	always

Description: numeric identifier of message type

Example: `1127618380`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: `FileTransfer`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
severity	Severity	message	always

Description: number between 0-10 inclusive, equal to aggregated analytics score divided by 10 or 0 if analytics is disabled

Example: 0

Field	Name	Scope	Present
timestamp	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
server_username	Server user	session	always

Description: the server username

Example: root

Field	Name	Scope	Present
server_domain	Server user domain if known	session	sometimes

Description: the server domain, if known

Example: acme.com

Field	Name	Scope	Present
gateway_username	Gateway username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: gwtestauto

Field	Name	Scope	Present
gateway_domain	Gateway user domain	session	sometimes

Description: the authenticated gateway user domain if there was a successful gateway authentication and known

Example: acme.com

Field	Name	Scope	Present
server_name	Server name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
server_address	Server address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
server_port	Server port	session	always

Description: the port number on the server

Example: 22

Field	Name	Scope	Present
client_name	Client name	session	always

Description: the client hostname or IP address if hostname is not known

Example: client.acme.com

Field	Name	Scope	Present
client_address	Client address	session	always

Description: the IP address of the client

Example: 10.30.0.24

Field	Name	Scope	Present
client_port	Client port	session	always

Description: the port number on the client

Example: 38014

Field	Name	Scope	Present
protocol	Application protocol	session	always

Description: SPS supported protocol

Example: `SSH`

Field	Name	Scope	Present
`connection_policy`	Connection policy name	session	always

Description: SPS connection policy name

Example: `my_connection`

Field	Name	Scope	Present
`auth_method`	Authentication method	session	always

Description: the type of authentication used in gateway authentication

Example: `password`

Field	Name	Scope	Present
`file_operation`	Operation	message	always

Description: the operation on the file such as UPLOAD/DOWNLOAD. It may contain the suffix 'WARNING', if the operation failed

Example: `UPLOAD`

Field	Name	Scope	Present
`filename`	Filename	message	always

Description: the file name

Example: `foobar.txt`

Field	Name	Scope	Present
`filepath`	File path	message	always

Description: the path to the file on the server

Example: `/tmp`

JSON_CIM messages

ServerConnect on initial contact

Description of the message: Emitted when SPS connects to the serverfor the first time in the session

Example message:

```
{"vendor":"OneIdentity","user":"","transport":"tcp","src_
user":"gwtestauto","src_port":"58140","src_
ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-39","product":"SPS-5.11.0","event_
name":"ServerConnect","dvc":"sps1.acme.com","dest_port":"22","dest_
ip":"10.170.255.206","dest":"server.acme.com","app":"ssh","action":"added","_
time":"1557913195000"}
```

The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: `ServerConnect`

Field	Name	Scope	Present
`action`	Action	message	always

Description: the taken by the device according to CIM model

Example: `added`

Field	Name	Scope	Present
`_time`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `ssh`

Field	Name	Scope	Present
`dest_ip`	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`dest`	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`user`	Name of the user	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`dest_port`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`transport`	Transport	session	always

Description: the layer 3 protocol

Example: `tcp`

ServerConnect for secondary channels

Description of the message: Emitted when SPS connects to the serverfor opening further channels. The difference from initial connection is that the server user name is known and authenticated this time.

Example message:


```
{"vendor":"OneIdentity","user":"","transport":"tcp","src_
user":"gwtestauto","src_port":"58140","src_
ip":"10.30.0.24","src":"client.acme.com","user":"root","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-39","product":"SPS-5.11.0","event_
name":"ServerConnect","dvc":"sps1.acme.com","dest_port":"22","dest_
ip":"10.170.255.206","dest":"server.acme.com","app":"ssh","action":"added","_
time":"1557913195000"}
```


The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: ServerConnect

Field	Name	Scope	Present
action	Action	message	always

Description: the taken by the device according to CIM model

Example: added

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: ssh

Field	Name	Scope	Present
dest_ip	Destination address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
dest	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
user	Name of the user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`dest_port`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`transport`	Transport	session	always

Description: the layer 3 protocol

Example: `tcp`

ServerAuthenticationSuccess

Description of the message: Emitted after the server authentication successfully happened

Example message:

```
{"vendor":"OneIdentity","user":"root","transport":"tcp","src_
user":"gwtestauto","src_port":"57982","src_
ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-38","product":"SPS-5.11.0","event_
name":"ServerAuthenticationSuccess","dvc":"sps1.acme.com","dest_
port":"22","dest_
ip":"10.170.255.206","dest":"server.acme.com","app":"ssh","action":"success","_
time":"1557913189329"}
```

The message contains the following fields.

Field	Name	Scope	Present
`vendor`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`product`	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
`dvc`	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
`session_id`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
`event_name`	Event name	message	always

Description: the type of the message

Example: `ServerAuthenticationSuccess`

Field	Name	Scope	Present
`action`	Action	message	always

Description: marks a successful authentication

Example: `success`

Field	Name	Scope	Present
`_time`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `ssh`

Field	Name	Scope	Present
`dest_ip`	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`dest`	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`user`	Name of the user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`dest_port`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
transport	Transport	session	always

Description: the layer 3 protocol

Example: tcp

ServerAuthenticationFailure

Description of the message: Emitted after the server authentication failed

Example message:


```
{"vendor":"OneIdentity","user":"root","transport":"tcp","src_
user":"gwtestauto","src_port":"58140","src_
ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
iiCfsG48oJG5smpuocBLAN-my_connection-39","product":"SPS-5.11.0","event_
name":"ServerAuthenticationFailure","dvc":"sps1.acme.com","dest_
port":"22","dest_
ip":"10.170.255.206","dest":"server.acme.com","app":"ssh","action":"failure","_
time":"1557913197211"}
```


The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: OneIdentity

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: SPS-5.11.0

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: sps1.acme.com

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: ServerAuthenticationFailure

Field	Name	Scope	Present
action	Action	message	always

Description: marks a failed authentication

Example: failure

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: ssh

Field	Name	Scope	Present
dest_ip	Destination address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
dest	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`user`	Name of the user	session	always

Description: contains the non authenticated server username

Example: `root`

Field	Name	Scope	Present
`dest_port`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`transport`	Transport	session	always

Description: the layer 3 protocol

Example: `tcp`

GatewayAuthenticationFailure

Description of the message: Emitted after a failed gateway authentication. Note that the gateway username here is not authenticated and will not be retained in further messages to avoid confusion with an authenticated gateway user.

Example message:


```
{"vendor":"OneIdentity","user":"","transport":"tcp","src_
user":"gwtestauto","src_port":"49070","src_
ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-15","product":"SPS-5.11.0","event_
name":"GatewayAuthenticationFailure","dvc":"sps1.acme.com","dest_port":"","dest_
ip":"","dest":"","app":"ssh","action":"failure","_time":"1557912792360"}
```


The message contains the following fields.

Field	Name	Scope	Present
`vendor`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`product`	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
`dvc`	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: GatewayAuthenticationFailure

Field	Name	Scope	Present
action	Action	message	always

Description: marks a failed authentication

Example: failure

Field	Name	Scope	Present
dest	Destination host name	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dest_ip	Destination address	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
user	Name of the user	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dest_port	Destination port	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`_time`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `ssh`

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	message	always

Description: the non authenticated gateway username

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
transport	Transport	session	always

Description: the layer 3 protocol

Example: tcp

SessionClosed of successfully authenticated session

Description of the message: Emitted when the session ends and server authentication and any gateway authentication was successful. There may be further messages related to the session after this message due to post processing of session data!

Example message:


```
{"vendor":"OneIdentity","user":"root","transport":"tcp","src_
user":"gwtestauto","src_port":"48302","src_
ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-12","product":"SPS-5.11.0","event_
name":"SessionClosed","dvc":"sps1.acme.com","dest_port":"22","dest_
ip":"10.170.255.206","dest":"server.acme.com","app":"ssh","_
time":"1557912765545"}
```


The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: OneIdentity

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: SPS-5.11.0

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: sps1.acme.com

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: SessionClosed

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: ssh

Field	Name	Scope	Present
dest_ip	Destination address	session	always

Description: the IP address of the server

Example: 10.170.255.206

Field	Name	Scope	Present
dest	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: server.acme.com

Field	Name	Scope	Present
user	Name of the user	session	always

Description: the server username

Example: `root`

Field	Name	Scope	Present
`dest_port`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`transport`	Transport	session	always

Description: the layer 3 protocol

Example: `tcp`

SessionClosed after a failed gateway authentication

Description of the message: Emitted when the session ends because gateway authentication failed.

Example message:


```
{"vendor":"OneIdentity","user":"","transport":"tcp","src_user":"","src_
port":"49070","src_ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-15","product":"SPS-5.11.0","event_
name":"SessionClosed","dvc":"sps1.acme.com","dest_port":"","dest_
ip":"","dest":"","app":"ssh","_time":"1557912792398"}
```


The message contains the following fields.

Field	Name	Scope	Present
`vendor`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`product`	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
`dvc`	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
`session_id`	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: SessionClosed

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
app	Application protocol	session	always

Description: SPS supported protocol

Example: ssh

Field	Name	Scope	Present
dest_ip	Destination address	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dest	Destination host name	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
user	Name of the user	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
dest_port	Destination port	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`transport`	Transport	session	always

Description: the layer 3 protocol

Example: `tcp`

SessionClosed after a failed server authentication

Description of the message: Emitted when the session ends because server authentication failed.

Example message:

```
{"vendor":"OneIdentity","user":"","transport":"tcp","src_
user":"gwtestauto","src_port":"49426","src_
ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
mBbMWzauBWHQN9TpoZz8mD-my_connection-17","product":"SPS-5.11.0","event_
name":"SessionClosed","dvc":"sps1.acme.com","dest_port":"22","dest_
ip":"10.170.255.206","dest":"server.acme.com","app":"ssh","_
time":"1557912813792"}
```

The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: `SessionClosed`

Field	Name	Scope	Present
`_time`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol

Example: `ssh`

Field	Name	Scope	Present
`dest_ip`	Destination address	session	always

Description: the IP address of the server

Example: `10.170.255.206`

Field	Name	Scope	Present
`dest`	Destination host name	session	always

Description: the server hostname or IP address if hostname is not known

Example: `server.acme.com`

Field	Name	Scope	Present
`user`	Name of the user	message	always

Description: empty, not known in this message type

Example:

Field	Name	Scope	Present
`dest_port`	Destination port	session	always

Description: the port number on the server

Example: `22`

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	sometimes

Description: the authenticated gateway username if there was a successful gateway authentication

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`transport`	Transport	session	always

Description: the layer 3 protocol

Example: `tcp`

RdpEmbeddedInTsg

Description of the message: Emitted when the gateway user is acquired in a Terminal Service Gateway authentication scenario.

Example message:


```
{"vendor":"OneIdentity","user":"","transport":"tcp","src_
user":"gwtestauto","src_port":"51204","src_
ip":"10.30.0.24","src":"client.acme.com","session_id":"svc-
oUDm7arcL8zNb3t2CVwSQr-my_connection-47-4","product":"SPS-5.11.0","event_
name":"RdpEmbeddedInTsg","dvc":"sps1.acme.com","dest_port":"","dest_
ip":"","dest":"","app":"rdp","action":"allowed","_time":"1558006936608"}
```

The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: `RdpEmbeddedInTsg`

Field	Name	Scope	Present
action	Action	message	always

Description: the action taken by the device according to CIM model

Example: `allowed`

Field	Name	Scope	Present
`_time`	Timestamp	message	always

Description: the UNIX time stamp when the event occurred
Example: `1554470652340`

Field	Name	Scope	Present
`app`	Application protocol	session	always

Description: SPS supported protocol
Example: `ssh`

Field	Name	Scope	Present
`dest_ip`	Destination address	session	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
`dest`	Destination host name	session	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
`user`	Name of the user	message	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
`dest_port`	Destination port	session	always

Description: empty, not known in this message type
Example:

Field	Name	Scope	Present
`src_ip`	Source address	session	always

Description: the IP address of the client

Example: `10.30.0.24`

Field	Name	Scope	Present
`src`	Source host name	session	always

Description: the client hostname or IP address if hostname is not known

Example: `client.acme.com`

Field	Name	Scope	Present
`src_user`	Source username	session	always

Description: the authenticated gateway username

Example: `gwtestauto`

Field	Name	Scope	Present
`src_port`	Source port	session	always

Description: the port number on the client

Example: `38014`

Field	Name	Scope	Present
`transport`	Transport	session	always

Description: the layer 3 protocol

Example: `tcp`

SessionScored

Description of the message: Score messages represent scoring events when SPS has calculated an initial or changed score for the session.

Example message:

```
{"vendor":"OneIdentity","signature":"keystroke","session_id":"svc-
416YVFZMy7rT8RA7T7yeAs-my_connection-0","product":"SPS-5.11.0","event_
name":"SessionScored","dvc":"sps1.acme.com","algorithm_score":"18","algorithm_
name":"keystroke","aggregated_score":"70","action":"allowed","_
time":"1558010880806"}
```

The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: OneIdentity

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: SPS-5.11.0

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: sps1.acme.com

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: SessionScored

Field	Name	Scope	Present
action	Action	message	always

Description: the action taken by the device according to CIM model

Example: allowed

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`aggregated_score`	Aggregated score	message	always

Description: the average score from all enabled analytics algorithms

Example: `50`

Field	Name	Scope	Present
`algorithm_name`	Algorithm name	message	always

Description: the name of the algorithm that changed value

Example: `keystroke`

Field	Name	Scope	Present
`signature`	Signature	message	always

Description: the algorithm name as CIM intrusion detection signature

Example: `hostlogin`

Field	Name	Scope	Present
`algorithm_score`	Algorithm score	message	always

Description: the new score value of the algorithm that changed value

Example: `60`

CommandChannelEvent

Description of the message: Emitted when a command is detected in the session channel text.

Example message:

```
{"vendor":"OneIdentity","session_id":"svc-mBbMWzauBWHQN9TpoZz8mD-my_connection-12","product":"SPS-5.11.0","event_name":"CommandChannelEvent","dvc":"sps1.acme.com","command":"exit","action":"allowed","_time":"1557912765461"}
```

The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: `CommandChannelEvent`

Field	Name	Scope	Present
action	Action	message	always

Description: the action taken by the device according to CIM model

Example: `allowed`

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: `1554470652340`

Field	Name	Scope	Present
`command`	Command	message	always

Description: the full command detected

Example: `exit`

WindowTitleChannelEvent

Description of the message: Emitted when a command is detected in the session channel text.

Example message:

```
{"window_title":"Shortcut Tools Application Tools Administrative
Tools","vendor":"OneIdentity","session_id":"svc-oUDm7arcL8zNb3t2CVwSQr-my_
connection-47-4","product":"SPS-5.11.0","event_
name":"WindowTitleChannelEvent","dvc":"sps1.acme.com","action":"allowed","_
time":"1558007001482"}
```

The message contains the following fields.

Field	Name	Scope	Present
`vendor`	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
`product`	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
`dvc`	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: WindowTitleChannelEvent

Field	Name	Scope	Present
action	Action	message	always

Description: the action taken by the device according to CIM model

Example: allowed

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
window_title	Window title	message	always

Description: the window title detected in graphical protocol

Example: firefox

FileTransfer

Description of the message: Emitted when a command is detected in the session channel text.

Example message:

```
{"vendor":"OneIdentity","session_id":"svc-2L83Phh9J6GKLWTc881awk-my_connection-
324","product":"SPS-5.11.0","file_path":"/cpuinfo","file_
operation":"UPLOAD","file_name":"cpuinfo","event_
name":"FileTransfer","dvc":"sps1.acme.com","action":"allowed","_
time":"1558023721326"}
```

The message contains the following fields.

Field	Name	Scope	Present
vendor	Device vendor	product	always

Description: fixed to OneIdentity

Example: `OneIdentity`

Field	Name	Scope	Present
product	Product version	product	always

Description: short product name with version

Example: `SPS-5.11.0`

Field	Name	Scope	Present
dvc	Device fqdn	device	always

Description: the hostname of SPS

Example: `sps1.acme.com`

Field	Name	Scope	Present
session_id	Session ID	session	always

Description: the unique identifier of the session

Example: `svc-hjdBxA2UWkTadH3juDVwrT-my_connection-0`

Field	Name	Scope	Present
event_name	Event name	message	always

Description: the type of the message

Example: `FileTransfer`

Field	Name	Scope	Present
action	Action	message	always

Description: the action taken by the device according to CIM model

Example: allowed

Field	Name	Scope	Present
_time	Timestamp	message	always

Description: the UNIX time stamp when the event occurred

Example: 1554470652340

Field	Name	Scope	Present
file_operation	Operation	message	always

Description: the operation on the file such as UPLOAD/DOWNLOAD. It may contain the suffix 'WARNING', if the operation failed

Example: UPLOAD

Field	Name	Scope	Present
file_name	Filename	message	always

Description: the file name

Example: foobar.txt

Field	Name	Scope	Present
file_path	Full file path	message	always

Description: the name of the file including its path on the server

Example: /tmp/foobar.txt

Joining to One Identity Starling

One Identity Starling helps to combine products from the One Identity line to create a secure and customizable cloud service. For details on One Identity Starling, see Starling - Technical Documentation.

If you are using a Starling 2FA plugin, (that is, you have uploaded it to **Basic Settings > Plugins** and then configured it at **Policies > AA Plugin Configurations**) and the SPS node is joined to One Identity Starling, you do not have to specify `api_key` and `api_url` in the Starling 2FA plugin configuration. This configuration method is more secure.

NOTE:

Currently the solution does not support using an HTTP proxy.

Joining SPS to One Identity Starling with Credential String

The following describes how to join SPS to One Identity Starling and take advantage of companion features from Starling products such as 2FA and Identity Analytics.

Prerequisites

- An existing Starling organization (tenant)
- A One Identity Hybrid Subscription

To join SPS to One Identity Starling

1. Navigate to **Basic Settings > Management > Join to Starling**.
2. *Optional:* If you have received your TIMS License from the Licensing Department (TIMS.License@quest.com), enter your TIMS License into **Product TIMS License**.
3. Click **Start join process**. The One Identity Starling site will open in a new tab.
4. Enter your One Identity Starling credentials.
5. Click **Next**.

   NOTE:

   By clicking **Next**, you have joined your SPS machine to Starling. Now you have to store this information in SPS to finish the join process.

6. Copy your **Credential String** from the page. For example,

   ```
   8dc0d6d4-b062-4357-abe2-8634523a91d9:4c0321bf-a099-4f95-86a6-20c6a4eb9298
   ```

7. Navigate back to the SPS tab.
8. Paste your Credential String into the **Credential String** field.

   NOTE:

   If for some reason you cannot paste the Credential String, you can re-retrieve it by refreshing this page and repeating the join process. You will receive the same Credential String if you did not change your host name.

9. Click **Finalize join process**.
10. The following will be displayed automatically:
    - **Product Name**
    - **Product Instance**
    - **Product TIMS License** if you have entered it before starting the join process

Unjoining SPS from One Identity Starling

If you intend to decommission an SPS machine, or replace it with another one, you have to unjoin that machine and join the new machine. The following describes how to unjoin SPS from One Identity Starling.

Prerequisites

- An existing Starling organization (tenant)
- A One Identity Hybrid Subscription
- A SPS that is already joined to One Identity Starling.

To unjoin SPS from One Identity Starling

1. Navigate to **Basic Settings > Management > Join to Starling**.
2. Click **Unjoin from Starling**.
3. To join the new machine, see Joining SPS to One Identity Starling with Credential String.

6

User management and access control

The **AAA** menu (Authentication, Authorization, and Accounting) allows you to control the authentication, authorization, and accounting settings of the users accessing One Identity Safeguard for Privileged Sessions (SPS). The following will be discussed in the next sections:

- For details on how to authenticate locally on SPS — see Managing One Identity Safeguard for Privileged Sessions (SPS) users locally on page 303.
- For details on how to authenticate users using an external LDAP (for example Microsoft Active Directory) database, see Managing One Identity Safeguard for Privileged Sessions (SPS) users from an LDAP database on page 310.
- For details on how to authenticate users using an external RADIUS server, see Authenticating users to a RADIUS server on page 316.
- For details on how to control the privileges of users and usergroups, see Managing user rights and usergroups on page 320.
- For details on how to display the history of changes of SPS configuration, see Listing and searching configuration changes on page 332.

Managing One Identity Safeguard for Privileged Sessions (SPS) users locally

By default, One Identity Safeguard for Privileged Sessions (SPS) users are managed locally on SPS. In order to add local users in SPS, all steps of the following procedure need to be completed:

1. Create users.

   For detailed instructions on how to create local users, see Creating local users in One Identity Safeguard for Privileged Sessions (SPS) on page 304.

2. Assign users to groups.

   For details about how to add a usergroup, see Managing local usergroups on page 308.

3. Assign privileges to groups.

For information on how to control the privileges of usergroups, see Managing user rights and usergroups on page 320.

Creating local users in One Identity Safeguard for Privileged Sessions (SPS)

The following describes how to create a local user.

NOTE:

The `admin` user is available by default and has all possible privileges. It is not possible to delete this user.

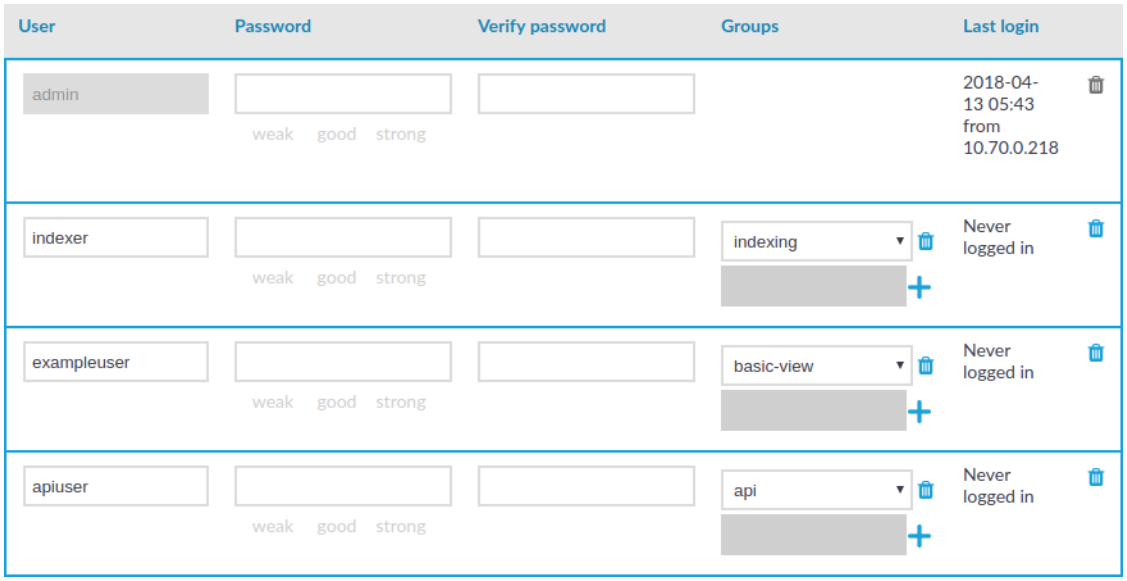
Local users cannot be managed when LDAP authentication is used (see Managing One Identity Safeguard for Privileged Sessions (SPS) users from an LDAP database on page 310). When LDAP authentication is enabled, the accounts of local users is disabled, they are not displayed on the **AAA > Local Users** page, but they are not deleted, either.

When using RADIUS authentication together with local users, the users are authenticated to the RADIUS server, only their group memberships must be managed locally on One Identity Safeguard for Privileged Sessions (SPS). For details, see Authenticating users to a RADIUS server on page 316.

To create a local user

1. Navigate to **AAA > Local Users** and click +.

Figure 67: AAA > Local Users — Creating local users



2. Enter the username into the **User** field.

NOTE:

For the username of SSH users, only valid UTF-8 strings are allowed.

The following characters cannot be used in usernames: `<>\/[]:;|=,+*?`

3. Enter a password for the user into the **Password** and **Verify password** fields.

   The strength of the password is indicated below the **Password** field as you type. To set a policy for password strength, see Setting password policies for local users on page 306. The user can change the password later from the SPS web interface, and you can modify the password of the user here.

   Use strong passwords: at least 8 characters that include numbers, letters, special characters, and capital letters. For local One Identity Safeguard for Privileged Sessions (SPS) users, require the use of strong passwords (set **AAA > Settings > Minimal password strength** to `strong`). For details, see "Setting password policies for local users" in the Administration Guide.

   NOTE:

   One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

4. Click + in the **Groups** section and select a group that the user will be member of. Repeat this step to add the user to multiple groups.

   If you wish to modify the group membership of a local user later on, you can do that here.

   To remove a user from a group, click 🗑 next to the group.

5. Click Commit.

Deleting a local user from One Identity Safeguard for Privileged Sessions (SPS)

The following describes how to delete a local user from One Identity Safeguard for Privileged Sessions (SPS).

To delete a local user from SPS

1. Navigate to **AAA > Local Users**.
2. Find the user you wish to delete.
3. Click 🗑 next to the user, at the right edge of the screen.
4. Click Commit.

Setting password policies for local users

One Identity Safeguard for Privileged Sessions (SPS) can use password policies to enforce minimal password strength and password expiry.

Limitations

Note the following important points about password policies.

- Password policies do not apply to the built-in `admin` user.
- Password policies apply only for locally managed users, and have no effect if you manage your users from an LDAP database, or if you authenticate your users to a RADIUS server.

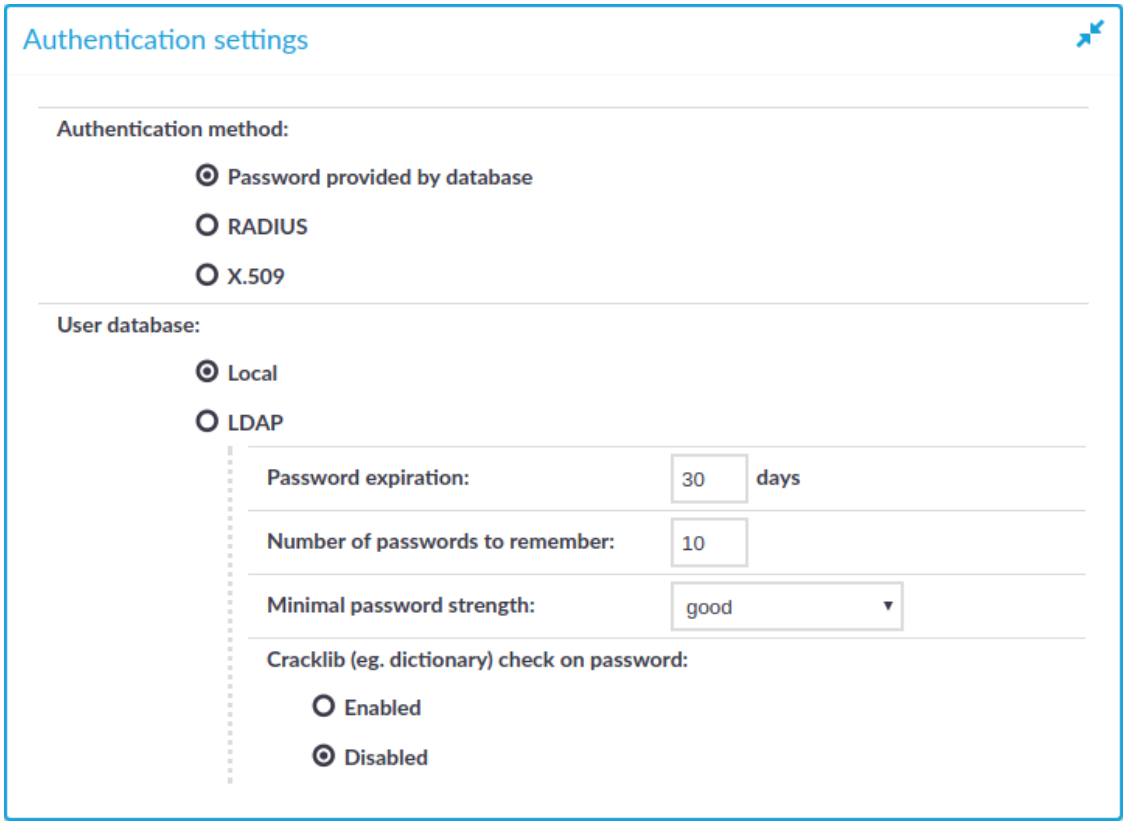
  NOTE:

  One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

To create a password policy

1. Navigate to **AAA > Settings**.

Figure 68: AAA > Settings — Configuring password policies



2. Verify that the **Authentication method** is set to **Password provided by database** and that the **User database** is set to **Local**.

   NOTE:

   If the setting of these fields is different (for example LDAP or RADIUS), then SPS is not configured to manage passwords locally.

3. Set how long the passwords are valid in the **Password expiration** field. After this period, SPS users will have to change their password. To disable password expiry, enter `0`.
4. To prevent password-reuse (for example when a user has two password and instead of changing to a new password only switches between the two), set how many different passwords must the user use before reusing an old password.
5. To enforce the use of strong password, select the level of password-complexity from

the **Minimal password strength** field.

NOTE:

The strength of the password is determined by its entropy: the variety of numbers, letters, capital letters, and special characters used, not only by its length.

To execute some simple dictionary-based attacks to find weak passwords, set **Cracklib (eg. dictionary) check on password** to **Enabled**.

6. Click 

.

NOTE:

Changes to the password policy do not affect existing passwords. However, setting password expiry will require every user to change their passwords after the expiry date, and the new passwords must comply with the strength requirements set in the password policy.

Managing local usergroups

You can use local groups to control the privileges of One Identity Safeguard for Privileged Sessions (SPS) local users — who can view and configure what.

For the description of built-in groups, see Built-in usergroups of One Identity Safeguard for Privileged Sessions (SPS) on page 328.

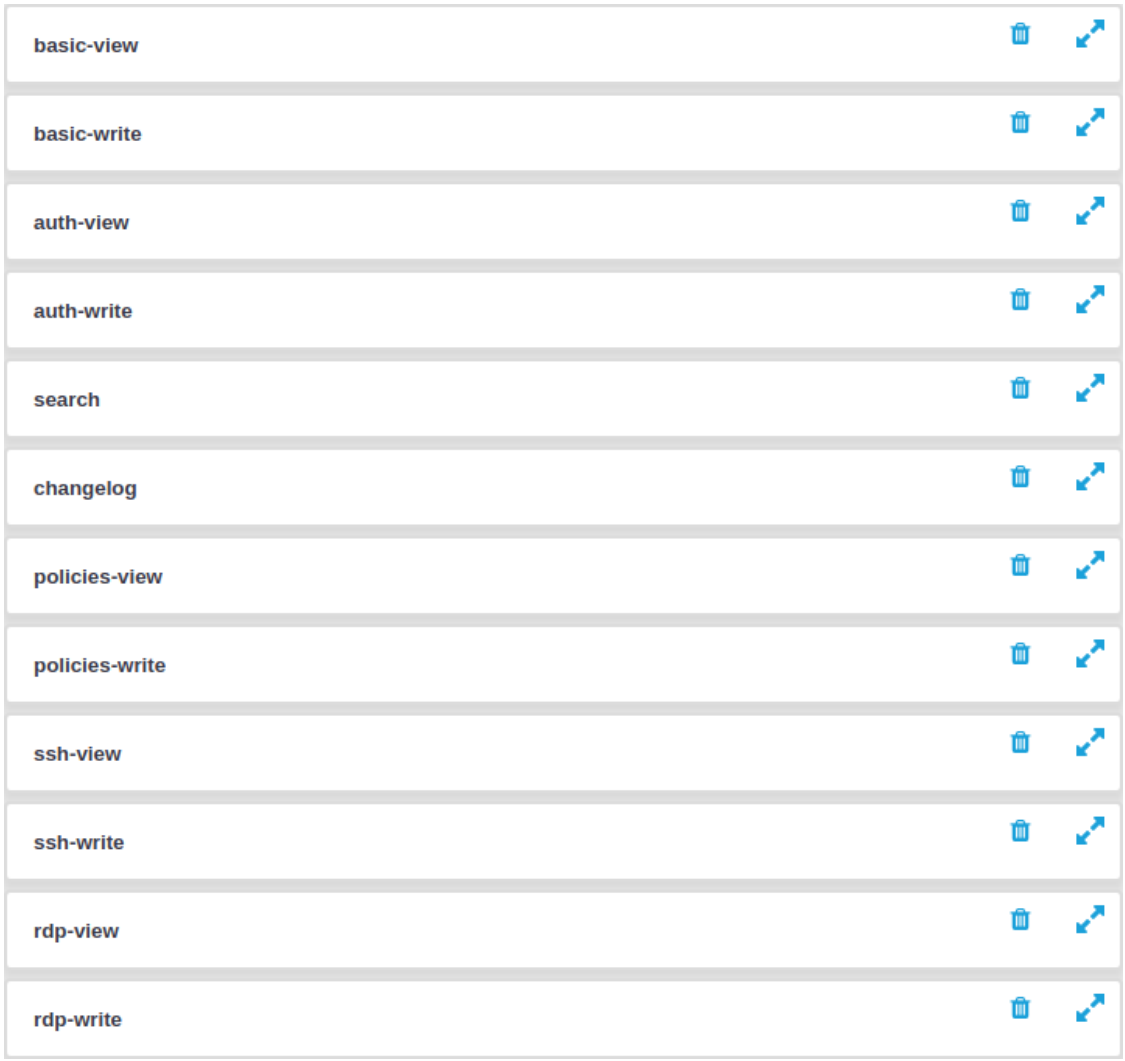
Use the **AAA > Group Management** page to:

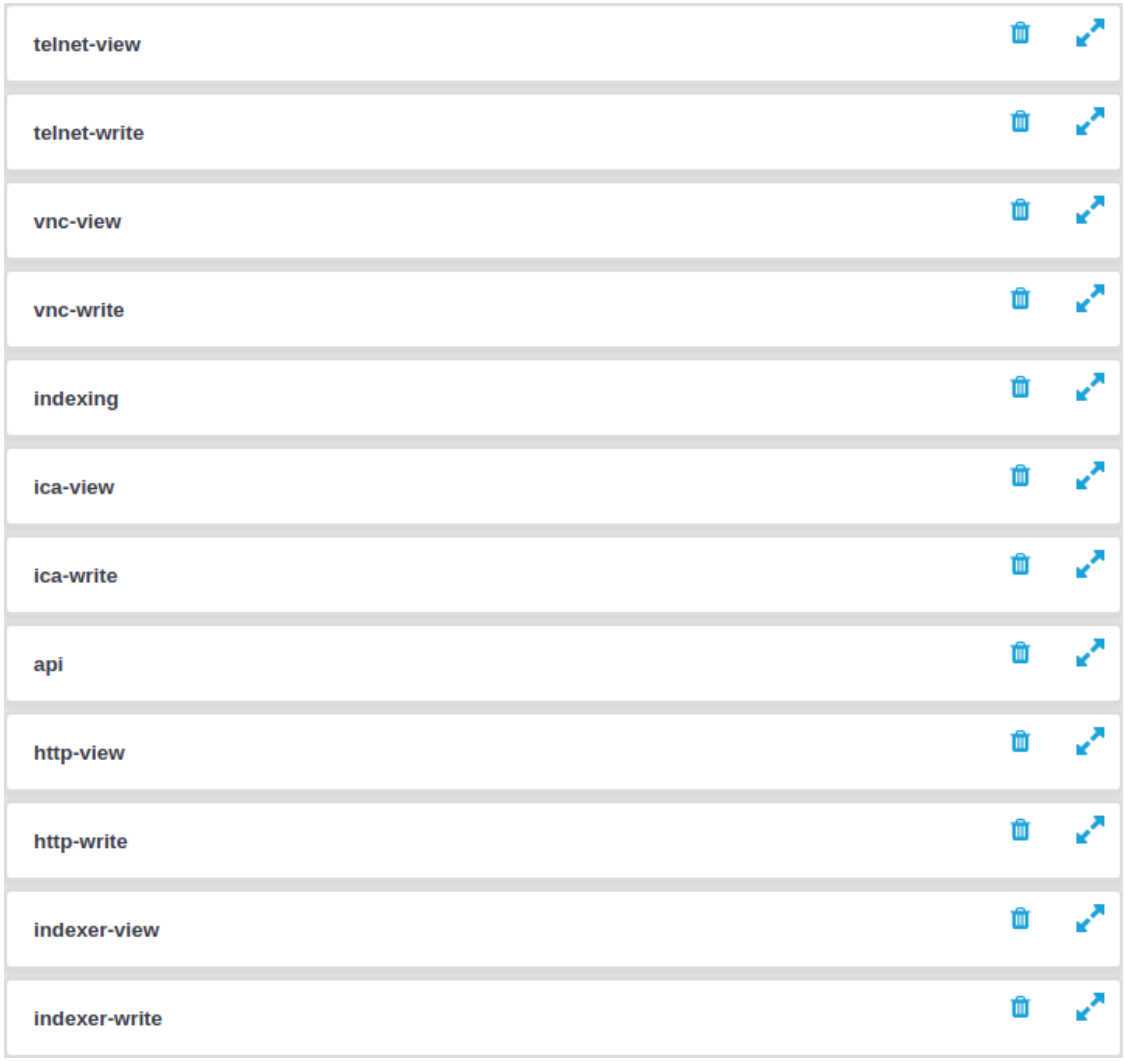
- Create a new usergroup.
- Display which users belong to a particular local usergroup.
- Edit group memberships.

To create a new group,

1. Navigate to **AAA > Group Management** and click **+**.

Figure 69: AAA > Group Management — Group management





2. Enter a name for the group.
3. Enter the names of the users belonging to the group. Click + to add more users.
4. Click Commit.

   Once you have added your usergroups, the next step is to start assigning privileges to them. For details on how to do that, see Assigning privileges to usergroups for the One Identity Safeguard for Privileged Sessions (SPS) web interface on page 323.

Managing One Identity Safeguard for Privileged Sessions (SPS) users from an LDAP database

The One Identity Safeguard for Privileged Sessions (SPS) web interface can authenticate users to an external LDAP database to simplify the integration of SPS to your existing infrastructure. You can also specify multiple LDAP servers: if the first server is unavailable, SPS will try to connect to the second server.

NOTE:

- The `admin` user is available by default and has all privileges. It is not possible to delete this user.
- Enabling LDAP authentication automatically disables the access of every local user except for `admin`. The `admin` user can login to SPS even if LDAP authentication is used.
- SPS accepts both pre-win2000-style and Win2003-style account names (User Principal Names). User Principal Names (UPNs) consist of a username, the at (@) character, and a domain name, for example `administrator@example.com`.
- For the username of SSH users, only valid UTF-8 strings are allowed.
- The following characters cannot be used in:
  - usernames: `/\[]:;|=+*?<>"`
  - group names: `/\[]:;|=+*?<>"@,`
- When using RADIUS authentication together with LDAP users, the users are authenticated to the RADIUS server, only their group memberships must be managed in LDAP. For details, see "Authenticating users to a RADIUS server" in the Administration Guide.
- SPS treats user and group names in a case insensitive manner if the matching rule for the attribute in question is case insensitive in the LDAP database.

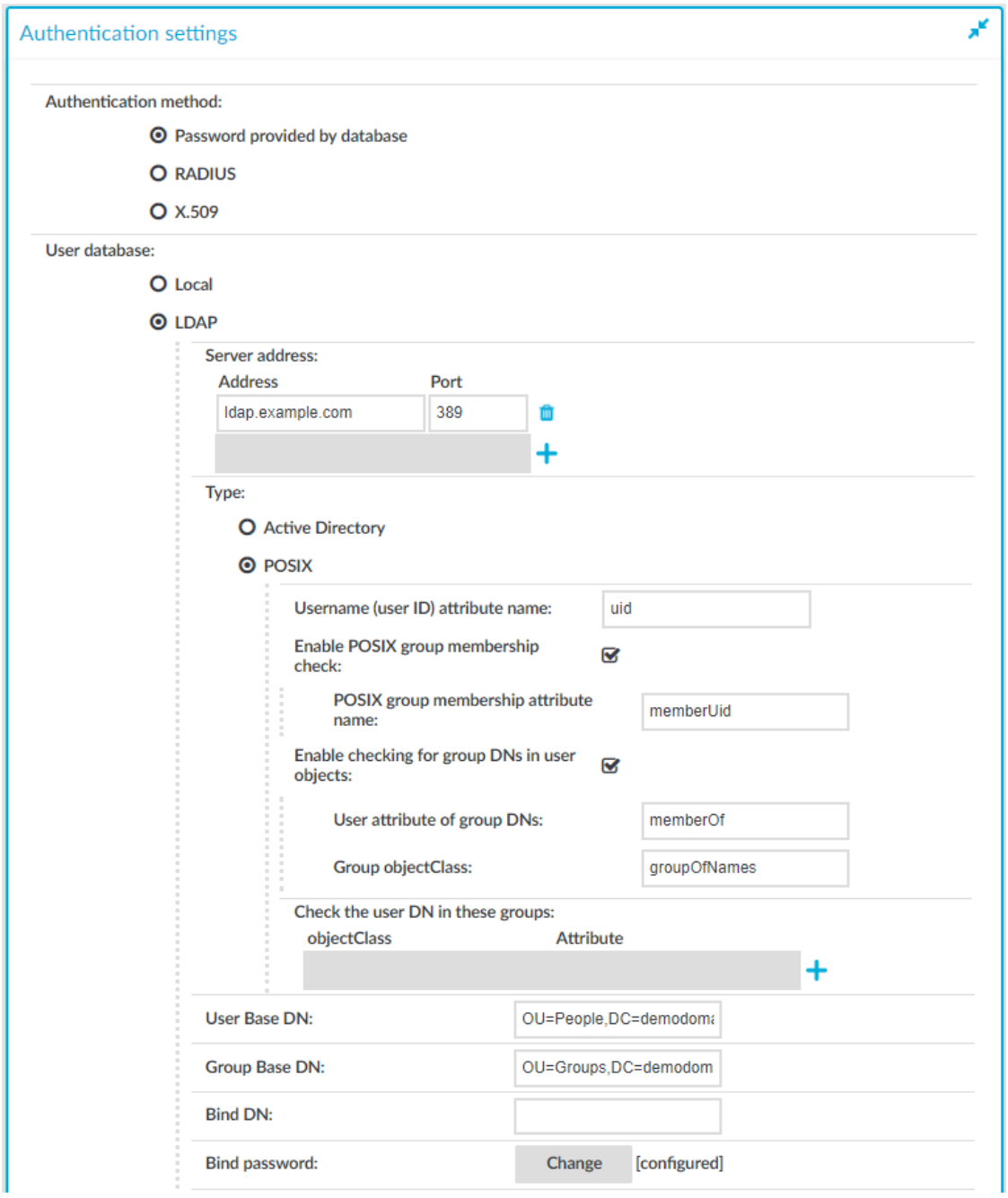
Prerequisites

Make sure that the response timeout of the LDAP/Active Directory server is at least 120 seconds.

To enable LDAP authentication

1. Navigate to **AAA > Settings > Authentication settings**.
2. Select the **LDAP** option and enter the parameters of your LDAP server.

Figure 70: AAA > Settings > Authentication settings — Configuring LDAP authentication



3. Enter the IP address or hostname and port of the LDAP server into the **Server Address** field. If you want to encrypt the communication between SPS and the LDAP server, in case of TLS, enter 636 as the port number, or in case of STARTTLS, enter 389 as the port number.

   Use an IPv4 address.

To add multiple servers, click + and enter the address of the next server. If a server is unreachable, SPS will try to connect to the next server in the list in failover fashion.

> **CAUTION:**
>
> **If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example `ldap.example.com`) in the Server Address field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.**

4. Select the type of your LDAP server in the **Type** field. Select:
   - **Active Directory** to connect to Microsoft Active Directory servers.

     You can enable nested groups. Select **Enable AD group membership check**, then **Enable nested groups**.

     > **CAUTION:**
     >
     > **Nested groups can slow down the query and cause the connection to timeout if the LDAP tree is very large. In this case, disable the Enable nested groups option.**

     To also check group membership based on group DNs in a user attribute, select **Enable checking for group DNs in user objects** and enter the name of the user attribute, for example, `memberOf` in the **User attribute of group DNs** field.

     > **CAUTION:**
     >
     > **Using this option significantly slows down log on to the SPS web interface if you have too many groups.**
     >
     > **Only use this option if you have an LDAP schema where the user groups can only be determined from a user attribute that contains the group DNs.**

     To check for group membership based on user DNs in group attributes, use the **Check the user DN in these groups** options.

     For more information, see Active Directory LDAP backend¶.
   - **POSIX** to connect to servers that use the POSIX LDAP scheme.

     If your LDAP server uses a custom POSIX LDAP scheme, you might need to set which LDAP attributes store the username, or the attributes that set group memberships. For example, if your LDAP scheme does not use the `uid` attribute to store the usernames, set the **Username (user ID) attribute name** option.

     In addition to the primary group membership checking, you can allow checking for supplementary group memberships by selecting the **Enable POSIX group membership check** and specifying the **POSIX group membership attribute name** field.

To also check group membership based on group DNs in a user attribute, select **Enable checking for group DNs in user objects** and enter the name of the user attribute, for example, `memberOf` in the **User attribute of group DNs** field and objectClass, for example, `groupOfNames` in the **Group objectClass** field.

> **CAUTION:**
>
> **Using this option significantly slows down log on to the SPS web interface if you have too many groups.**
>
> **Only use this option if you have an LDAP schema where the user groups can only be determined from a user attribute that contains the group DNs.**

To check for group membership based on user DNs in group attributes, use the **Check the user DN in these groups** options.

For more information, see Posix LDAP backend¶.

For an overview about LDAP user and group resolution in SPS, see Overview.

5. In the **User Base DN** field, enter the name of the DN to be used as the base of queries regarding users (for example: `OU=People,DC=demodomain,DC=exampleinc`).

   > NOTE:
   >
   > You must fill in this field. It is OK to use the same value for **User Base DN** and **Group Base DN**.
   >
   > However, note that specifying a sufficiently narrow base for the LDAP subtrees where users and groups are stored can speed up LDAP operations.

6. In the **Group Base DN** field, enter the name of the DN to be used as the base of queries regarding groups (for example: `OU=Groups,DC=demodomain,DC=exampleinc`).

   > NOTE:
   >
   > You must fill in this field. It is OK to use the same value for **User Base DN** and **Group Base DN**.
   >
   > However, note that specifying a sufficiently narrow base for the LDAP subtrees where users and groups are stored can speed up LDAP operations.

7. In the **Bind DN** field, enter the Distinguished Name that SPS should use to bind to the LDAP directory (for example: `CN=Administrator,DC=demodomain,DC=exampleinc`).

   > NOTE:
   >
   > SPS accepts both pre-win2000-style and Win2003-style account names (User Principal Names), for example `administrator@example.com` is also accepted.

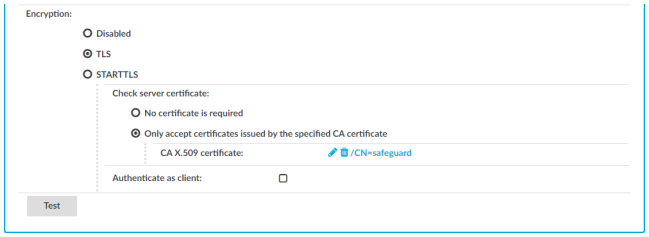
8. To configure or change the password to use when binding to the LDAP server, click **Change** and enter the password. Click **Update**. Click Commit.

NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: !"#$%&'()*+,-./:;<=>?@[\]^-`{|}

9. If you want to encrypt the communication between SPS and the LDAP server, in **Encryption**, select the **TLS** or the **STARTTLS** option and complete the following steps:

Figure 71: Policies > LDAP Servers — Configuring encryption



NOTE:

TLS-encrypted connection to Microsoft Active Directory is supported only on Windows 2003 Server and newer platforms. Windows 2000 Server is not supported.

- If you want SPS to verify the certificate of the server, select **Only accept certificates issued by the specified CA certificate** and click the icon in the **CA X.509 certificate** field. A pop-up window is displayed.

  Click **Browse**, select the certificate of the Certificate Authority (CA) that issued the certificate of the LDAP server, then click **Upload**. Alternatively, you can paste the certificate into the **Copy-paste** field and click **Set**.

  SPS will use this CA certificate to verify the certificate of the server, and reject the connections if the verification fails.

CAUTION:

According to recent cryptographic research, SHA-1 algorithm cannot be trusted as secure anymore, because signatures can be forged with reasonable costs. As a result, SHA-1 algorithm is not supported in SPS for X.509 certificate chains. Starting from SPS versions 6.0.4 and 6.5.0, certificates with SHA1-based signatures are no longer trusted for Active Directory or LDAP authentication, and future versions might refuse to validate SHA-1 signatures altogether.

Note that Root CA certificates may still contain SHA-1 signatures, because the signature is not validated for self-signed certificates. It is expected that other software such as clients and servers connected to SPS might reject SHA-1 signatures in a similar fashion.

Signing CAs in SPS generate certificates with SHA-256 since versions 4.3.4 and 5.0.0.

CAUTION:

If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example `ldap.example.com`) in the Server Address field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.

- If the LDAP server requires mutual authentication, that is, it expects a certificate from SPS, enable **Authenticate as client**. Generate and sign a certificate for SPS, then click in the **Client X.509 certificate** field to upload the certificate. After that, click in the **Client key** field and upload the private key corresponding to the certificate.

One Identity recommends using 2048-bit RSA keys (or stronger).

10. Click Commit.

NOTE:

You also have to configure the usergroups in SPS and possibly in your LDAP database. For details on using usergroups, see Using usergroups on page 327.

11. Click **Test** to test the connection.

Authenticating users to a RADIUS server

One Identity Safeguard for Privileged Sessions (SPS) can authenticate its users to an external RADIUS server. Group memberships of the users must be managed either locally on SPS or in an LDAP database.

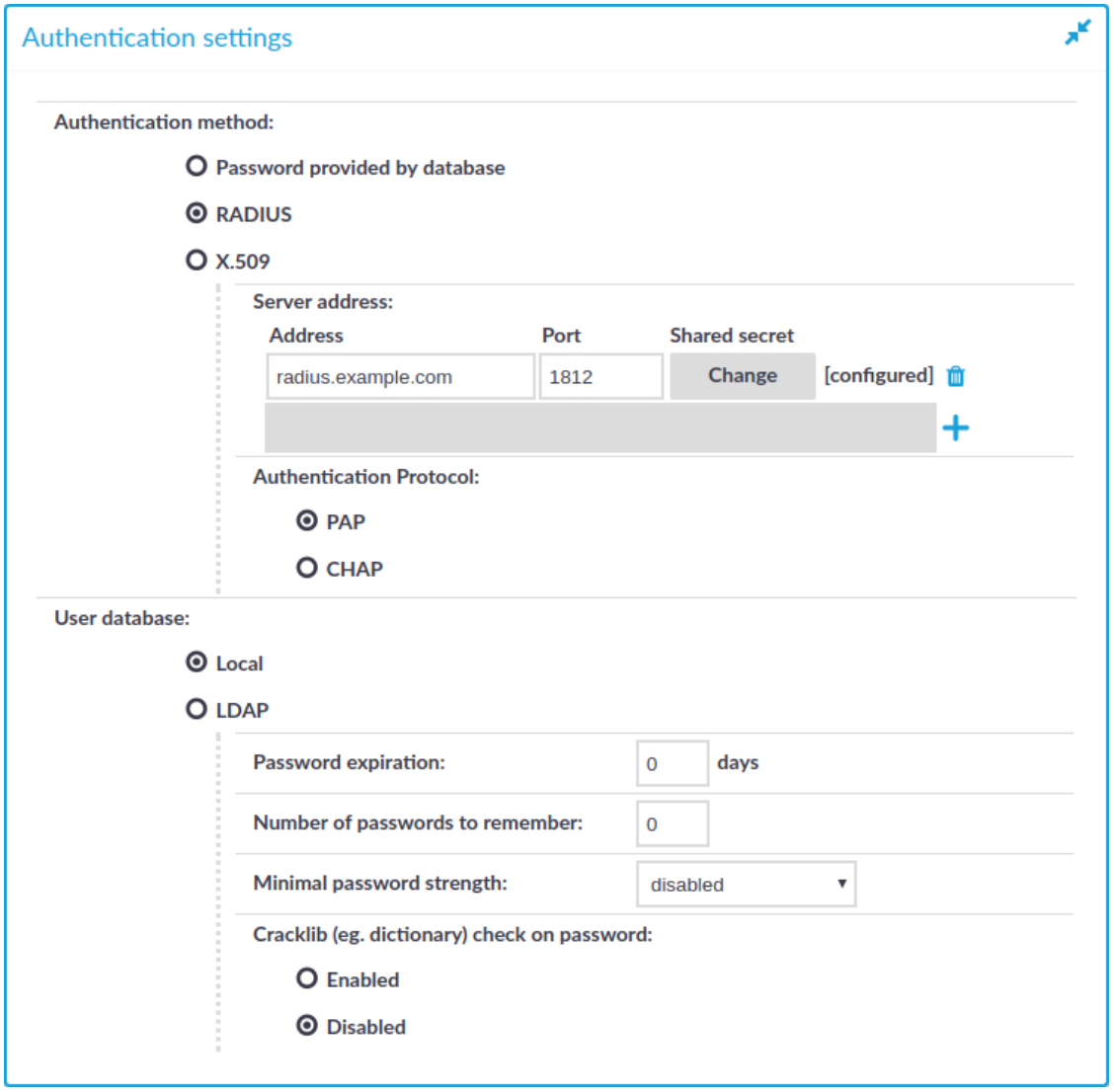
CAUTION:

The challenge/response authentication method is currently not supported. Other authentication methods (for example password, SecureID) should work.

To authenticate SPS users to a RADIUS server

1. Navigate to **AAA > Settings**.

Figure 72: Configuring RADIUS authentication



2. Set the **Authentication method** field to **RADIUS**.

   The status information displayed (**[NOT CONFIGURED]** and **[CONFIGURED]**) indicates whether or not you have provided the shared secret required to access the RADIUS server.

3. Enter the IP address or domain name of the RADIUS server into the **Address** field.

   Use an IPv4 address.

4. Click **Change**, and enter the password that SPS can use to access the server into the **Shared secret** field.

   > NOTE:
   >
   > One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

   Click **Update**.

5. To use the Password Authentication Protocol, select **PAP**. To use the Challenge-Handshake Authentication Protocol, select **CHAP**.

6. To add more RADIUS servers, click + and repeat Steps 2-4.

   Repeat this step to add multiple servers. If a server is unreachable, SPS will try to connect to the next server in the list in failover fashion.

7. When configuring RADIUS authentication with locally managed user accounts, complete the following steps.

   a. Set **Password expiration** to `0`.

   b. Set **Number of passwords to remember** to `0`.

   c. Set **Minimal password strength** to `disabled`.

   d. Set **Cracklib check on password** to `disabled`.

8. Click Commit.

   > **CAUTION:**
   >
   > **After you commit this configuration, the SPS web interface will be available only after successfully authenticating to the RADIUS server. Note that the default `admin` account of SPS will be able to login normally, even if the RADIUS server is unaccessible.**

Authenticating users with X.509 certificates

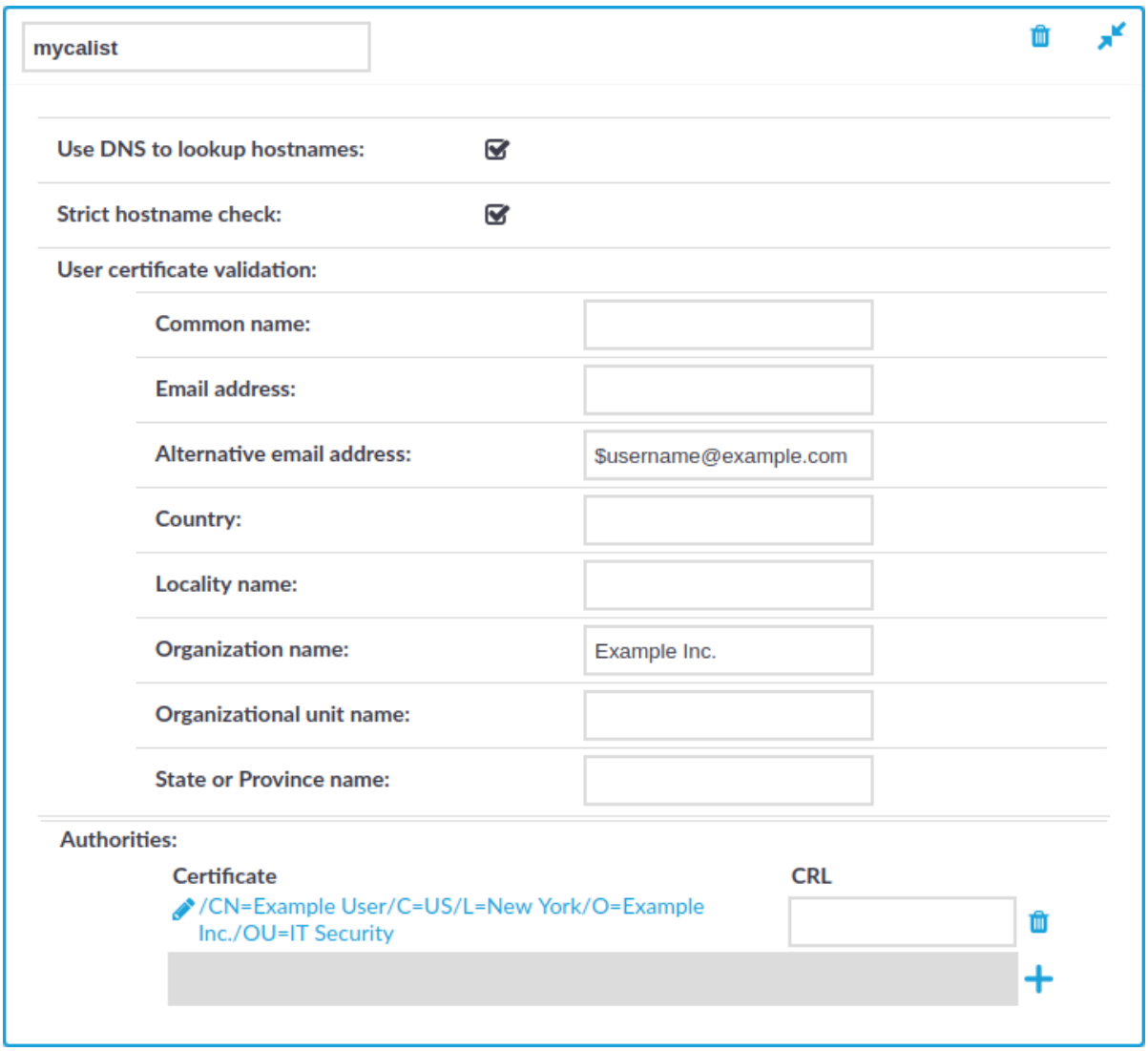
One Identity Safeguard for Privileged Sessions (SPS) provides a method to authenticate the users of the web interface with X.509 client certificates. The client certificate is validated against a CA list, and the username is exported from the client certificate for identification. One Identity recommends using 2048-bit RSA keys (or stronger).

To authenticate SPS users on the SPS web interface with X.509 client certificates, complete the following steps.

Prerequisites

- You will have to upload the CA certificates that issued the certificates of the users, so this CA certificate must be available on your computer in PEM format.
- The certificates of the users must contain the username used to authenticate on SPS. You must know which certificate field will contain the usernames (for example, `CN` or `UID`).
- The certificates must be imported into the browsers of the users. SPS offers the possibility to authenticate with a certificate only if a personal certificate is available in the browser.

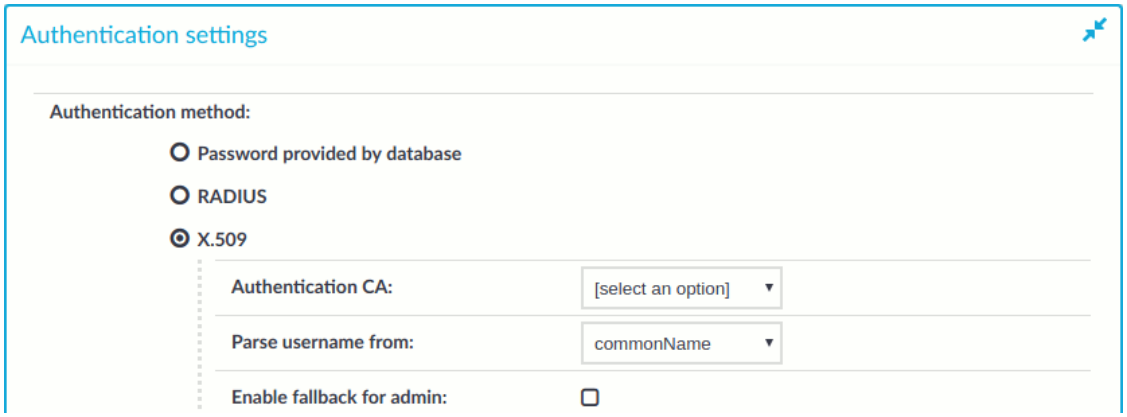
Figure 73: Policies > Trusted CA Lists — Creating Trusted CA lists



To authenticate users with X.509 certificates

1. Navigate to **Policies > Trusted CA Lists** and create a **Trusted CA List**.
2. If the user certificates contain the username in the Common Name field, make sure that the **Strict Hostname Check** is disabled.
3. Upload the CA certificate.
4. Adjust other settings as needed. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities.
5. Click Commit.
6. Navigate to **AAA > Settings > Authentication settings**.

   Figure 74: AAA > Settings > Authentication settings — Configuring X.509 authentication

7. Select **X.509**.
8. Select the trusted CA list created in the first step in **Authentication CA**.
9. Select which field of the user certificate contains the username in the **Parse username from** field. In most cases, it is the `commonName` or `userid` field, but SPS supports the `emailAddress` and `userPrincipalName` fields as well.
10. To allow the `admin` user to be able to log in without using X.509 authorization, select **Enable fallback for admin**. This will fallback to password authentication.
11. Click Commit.

Managing user rights and usergroups

In One Identity Safeguard for Privileged Sessions (SPS), user rights can be assigned to usergroups. SPS has numerous usergroups defined by default, but custom user groups can be defined as well. Every group has a set of privileges: which pages of the SPS web

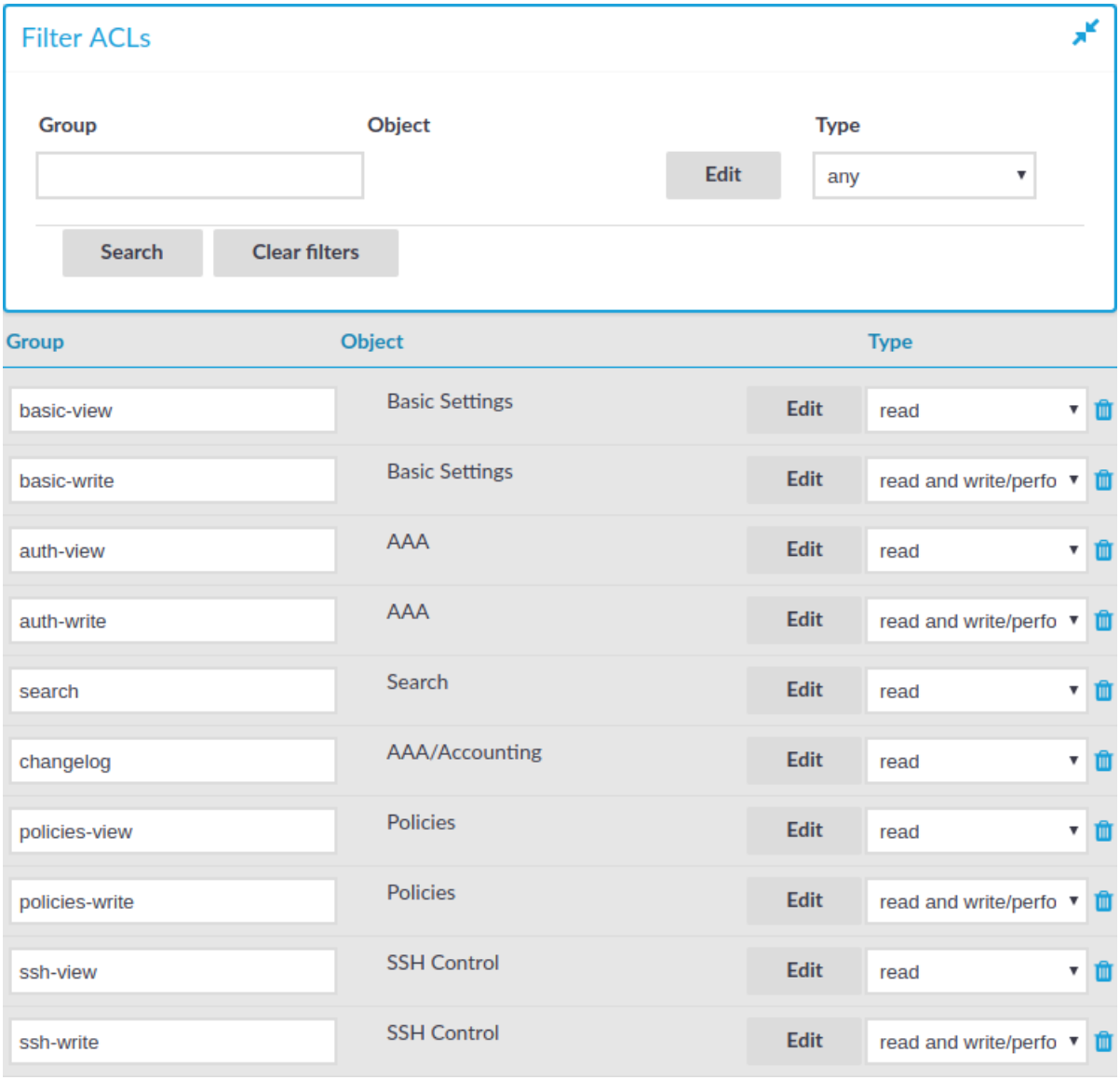
interface it can access, and whether it can only view (read) or also modify (read & write/perform) those pages or perform certain actions.

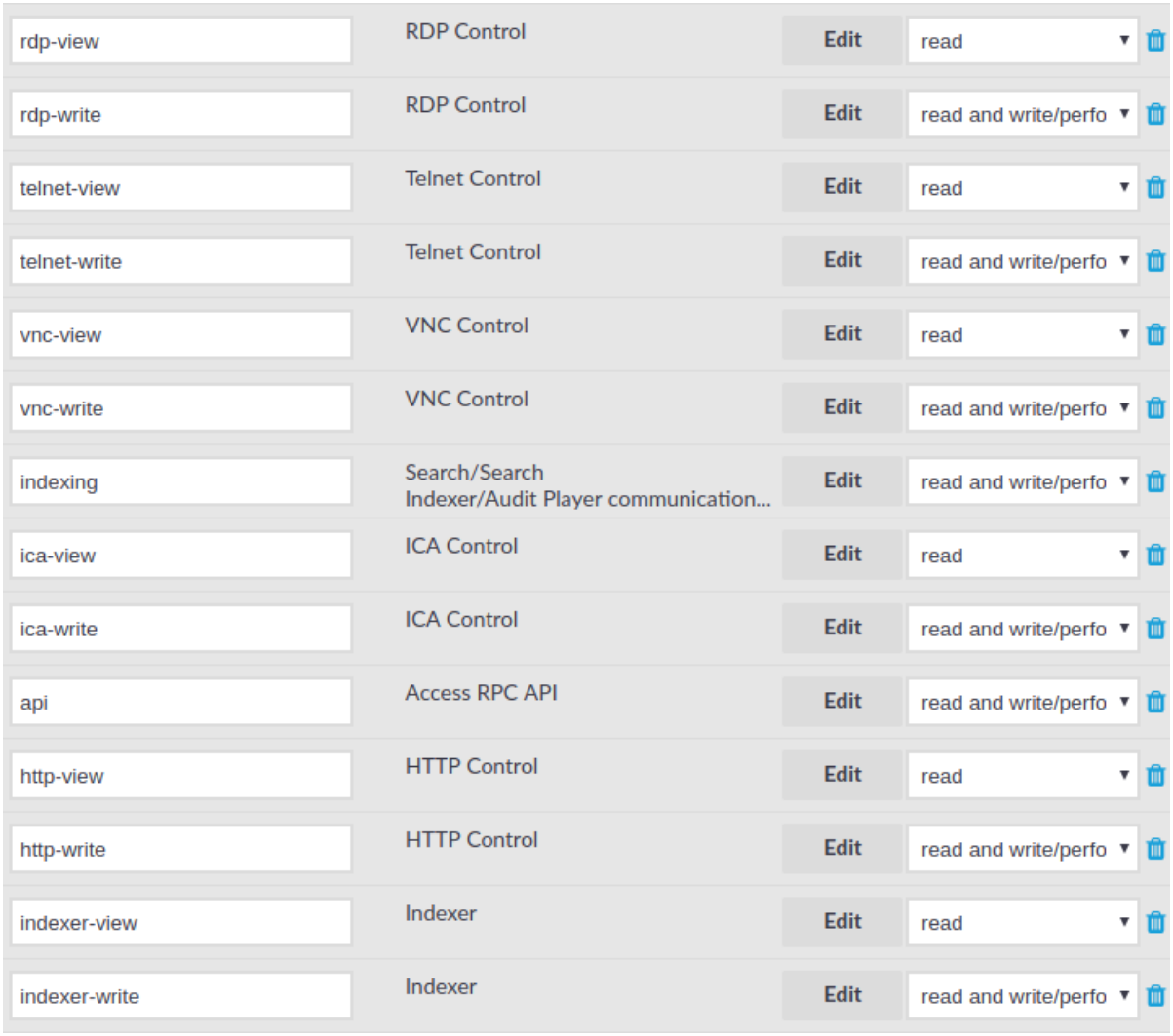
NOTE:

Every group has either read or read & write/perform privileges to a set of pages.

- For details on assigning privileges to a new usergroup, see Assigning privileges to usergroups for the One Identity Safeguard for Privileged Sessions (SPS) web interface on page 323.
- For details on modifying existing groups, see Modifying group privileges on page 324.
- For details on finding usergroups that have a specific privilege, see Finding specific usergroups on page 325.
- For tips on using usergroups, see Using usergroups on page 327.
- For a detailed description about the privileges of the built-in usergroups, see Built-in usergroups of One Identity Safeguard for Privileged Sessions (SPS) on page 328.

Figure 75: AAA > Access Control — Managing SPS users





Assigning privileges to usergroups for the One Identity Safeguard for Privileged Sessions (SPS) web interface

The following describes how to assign privileges to a new group.

To assign privileges to a new group

1. Navigate to **AAA > Access Control** and click +.
2. Find your usergroup. If you start typing the name of the group you are looking for, the autocomplete function will make finding your group easier for you.
3. Click **Edit** located next to the name of the group. The list of available privileges is displayed.

4. Select the privileges (that is, the pages of the One Identity Safeguard for Privileged Sessions (SPS) interface) to which the group will have access and click **Save**.

   NOTE:

   To export the configuration of SPS, the **Export configuration** privilege is required.

   To import a configuration to SPS, the **Import configuration** privilege is required.

   To update the firmware and set the active firmware, the **Firmware** privilege is required.

5. Select the type of access (**read** or **read & write**) from the **Type** field.
6. Click 

.

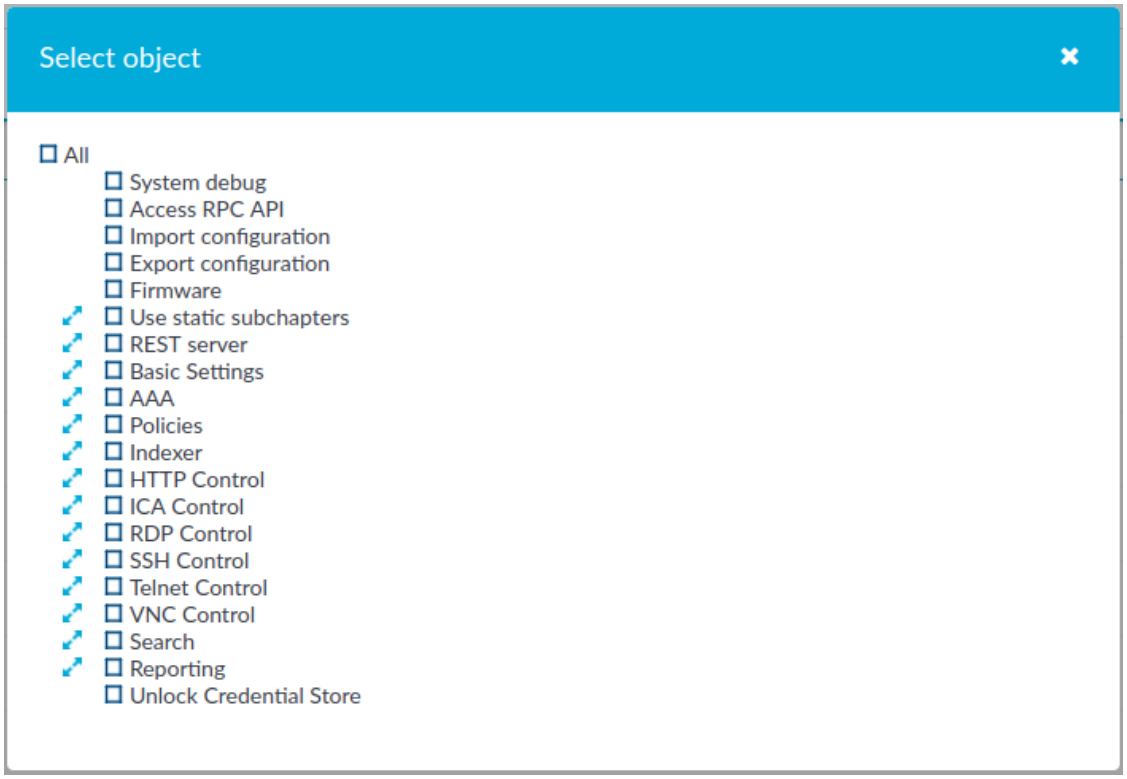
Modifying group privileges

The following describes how to modify the privileges of an existing group.

To modify the privileges of an existing group

1. Navigate to **AAA > Access Control**.
2. Find the group you want to modify and click **Edit**. The list of available privileges is displayed.
3. Select the privileges (pages of the One Identity Safeguard for Privileged Sessions (SPS) interface) to which the group will have access and click **Save**.

Figure 76: AAA > Access Control > Edit — Modifying group privileges



CAUTION:

Assigning the Search privilege to a user on the AAA page automatically enables the Search in all connections privilege, and grants the user access to every audit trail, even if the user is not a member of the groups listed in the Access Control option of the particular connection policy.

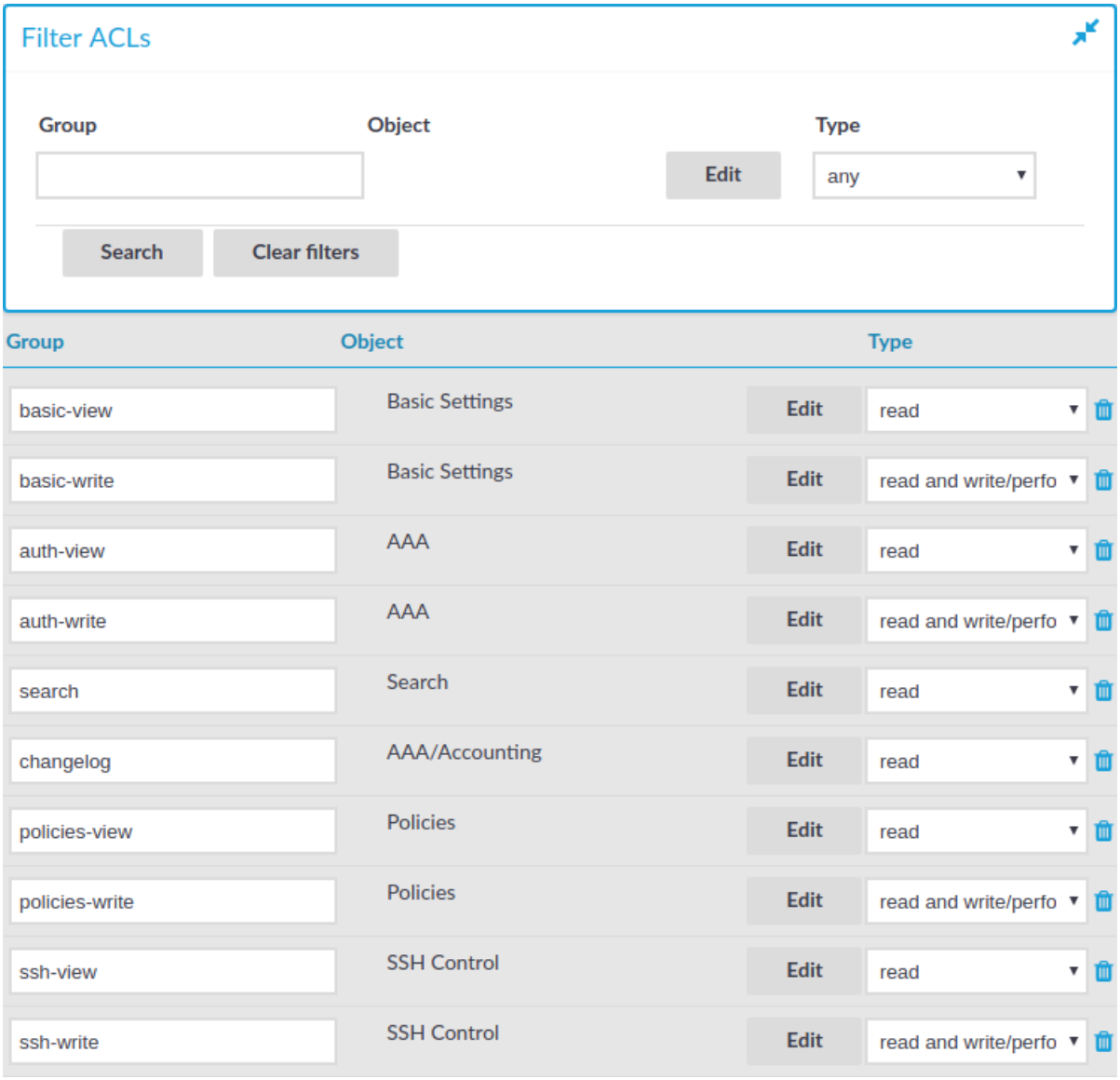
4. Select the type of access (**read** or **read & write**) from the **Type** field.
5. Click Commit.

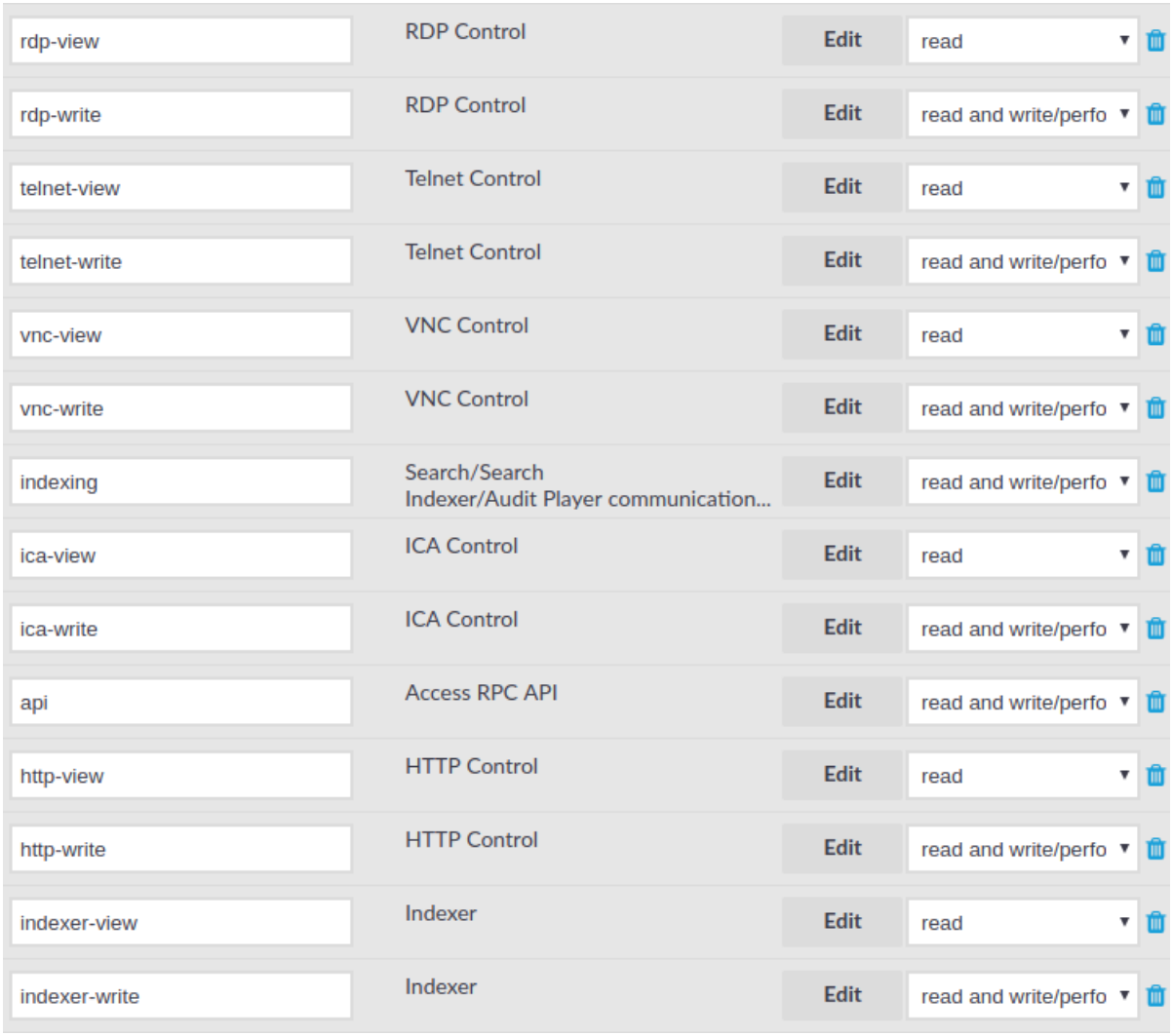
The `admin` user is available by default and has all privileges. It is not possible to delete this user.

Finding specific usergroups

The **Filter ACLs** section of the **AAA > Access Control** page provides you with a simple searching and filtering interface to search the names and privileges of usergroups.

Figure 77: AAA > Access Control — Finding specific usergroups





- To filter on a specific usergroup, enter the name of the group into the **Group** field and select **Search**.
- To select usergroups who have a specific privilege, click **Edit**, select the privilege or privileges you are looking for, and click **Search**.
- To filter for read or write access, use the **Type** option.

Using usergroups

How you should name usergroups depends on the way you manage your One Identity Safeguard for Privileged Sessions (SPS) users.

- *Local users*: If you use only local users, create or modify usergroups on the **AAA > Group Management** page, assign or modify privileges on the **AAA > Access Control** page, and add users to the groups on the **AAA > Local Users** or the **AAA >**

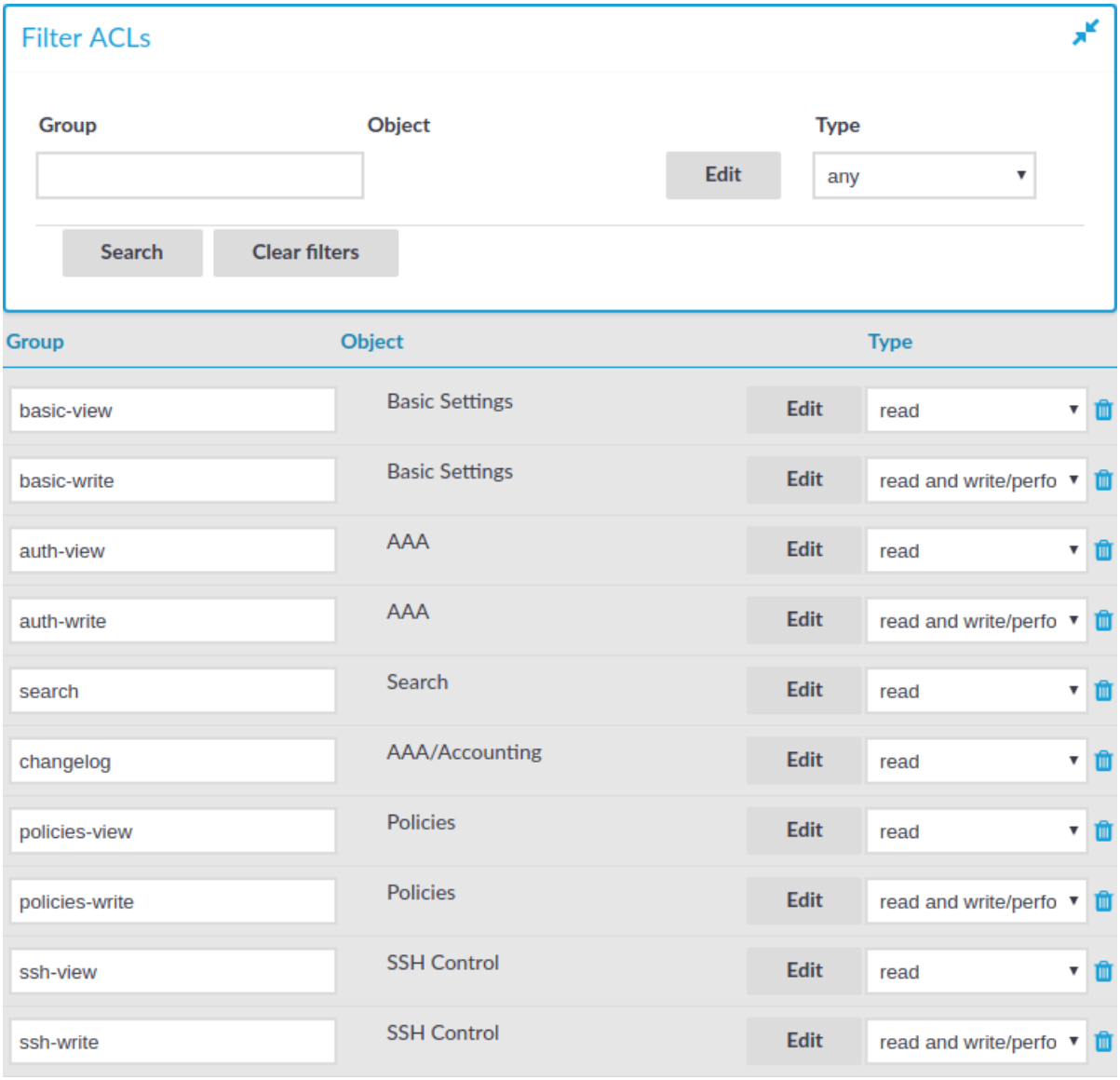
Group Management page.

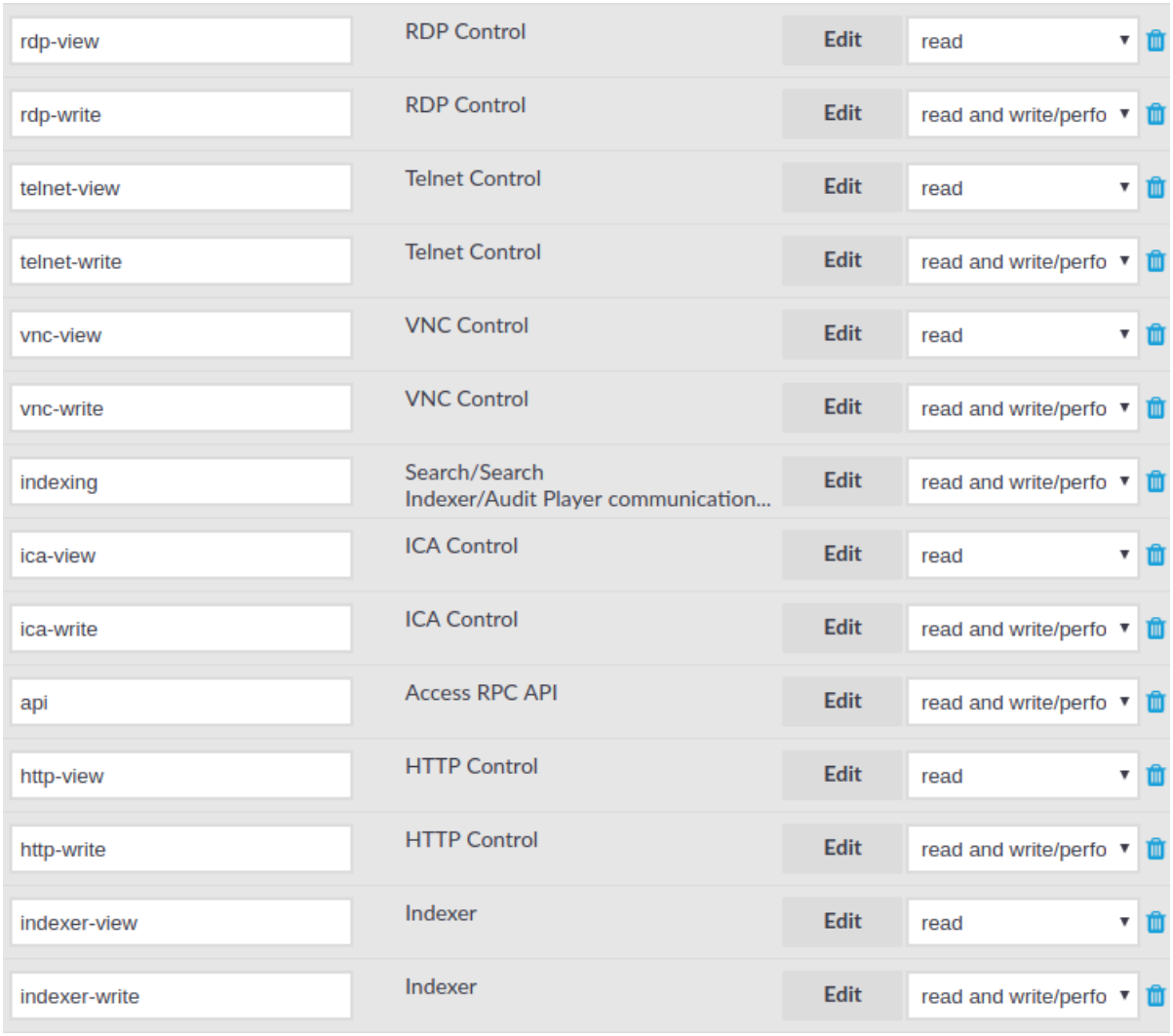
- *LDAP users and LDAP groups*: If you manage your users from LDAP, and also have LDAP groups that match the way you want to group your SPS users, create or modify your usergroups on the **AAA > Access Control** page and ensure that the name of your LDAP group and the SPS usergroup is the same. For example, to make members of the `admins` LDAP group be able to use SPS, create a usergroup called `admins` on the **AAA > Access Control** page and edit the privileges of the group as needed.
- *RADIUS users and local groups*: This is the case when you manage users from RADIUS, but you cannot or do not want to create groups in LDAP. Create your local groups on the **AAA > Access Control** page, and add your RADIUS users to these groups on the **AAA > Group Management** page.

Built-in usergroups of One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) has the following usergroups by default. Note that you can modify and delete these usergroups as you see fit.

Figure 78: AAA > Access Control — Built-in usergroups of SPS





CAUTION:

If you use LDAP authentication on the SPS web interface and want to use the default usergroups, you have to create these groups in your LDAP database and assign users to them. For details on using usergroups, see Using usergroups on page 327.

- **basic-view**: View the settings in the **Basic Settings** menu, including the system logs of SPS. Members of this group can also execute commands on the **Troubleshooting** tab.
- **basic-write**: Edit the settings in the **Basic Settings** menu. Members of this group can manage SPS as a host.
- **auth-view**: View the names and privileges of the SPS administrators, the configured usergroups, and the authentication settings in the **AAA** menu. Members of this group can also view the history of configuration changes.
- **auth-write**: Edit authentication settings and manage users and usergroups.

CAUTION:

Members of the `auth-write` group, or any other group with write privileges to the AAA menu are essentially equivalent to system administrators of SPS, because they can give themselves any privilege. Users with limited rights should never have such privileges.

If a user with write privileges to the AAA menu gives himself new privileges (for example gives himself group membership to a new group), then he has to relogin to the SPS web interface to activate the new privilege.

- **search**: Browse and download various logs and alerts in the **Search** menu. The members of this group have access to the audit trail files as well. Note that to open encrypted audit trail files, the proper encryption keys are required.
- **changelog**: View the history of SPS configuration changes in the **AAA > Accounting** menu.
- **report**: Browse, create and manage reports, and add statistics-based chapters to the reports in the **Reports** menu. Users with this privilege can access every report. To grant access to users only to specific reports, use the **Reports are accessible by the following groups** option of the report. For details, see Configuring custom reports on page 778.

  NOTE:

  To control exactly which statistics-based chapters and reports can the user include in a report, use the `Use static subchapters` privileges.
- **policies-view**: View the policies and settings in the **Policies** menu.
- **policies-write**: Edit the policies and settings in the **Policies** menu.
- **ssh-view**: View all connection and policy settings in the **SSH Control** menu.
- **ssh-write**: Edit all connection and policy settings in the **SSH Control** menu.
- **rdp-view**: View all connection and policy settings in the **RDP Control** menu.
- **rdp-write**: Edit all connection and policy settings in the **RDP Control** menu.
- **telnet-view**: View all connection and policy settings in the **Telnet Control** menu.
- **telnet-write**: Edit all connection and policy settings in the **Telnet Control** menu.
- **vnc-view**: View all connection and policy settings in the **VNC Control** menu.
- **vnc-write**: Edit all connection and policy settings in the **VNC Control** menu.
- **indexing**: Allows hosts running external indexers to access and download audit trails for automatic indexing. Note that the members of this group can access the SPS web interface as well, and download any audit trail directly.
- **ica-view**: View all connection and policy settings in the **ICA Control** menu.
- **ica-write**: Edit all connection and policy settings in the **ICA Control** menu.

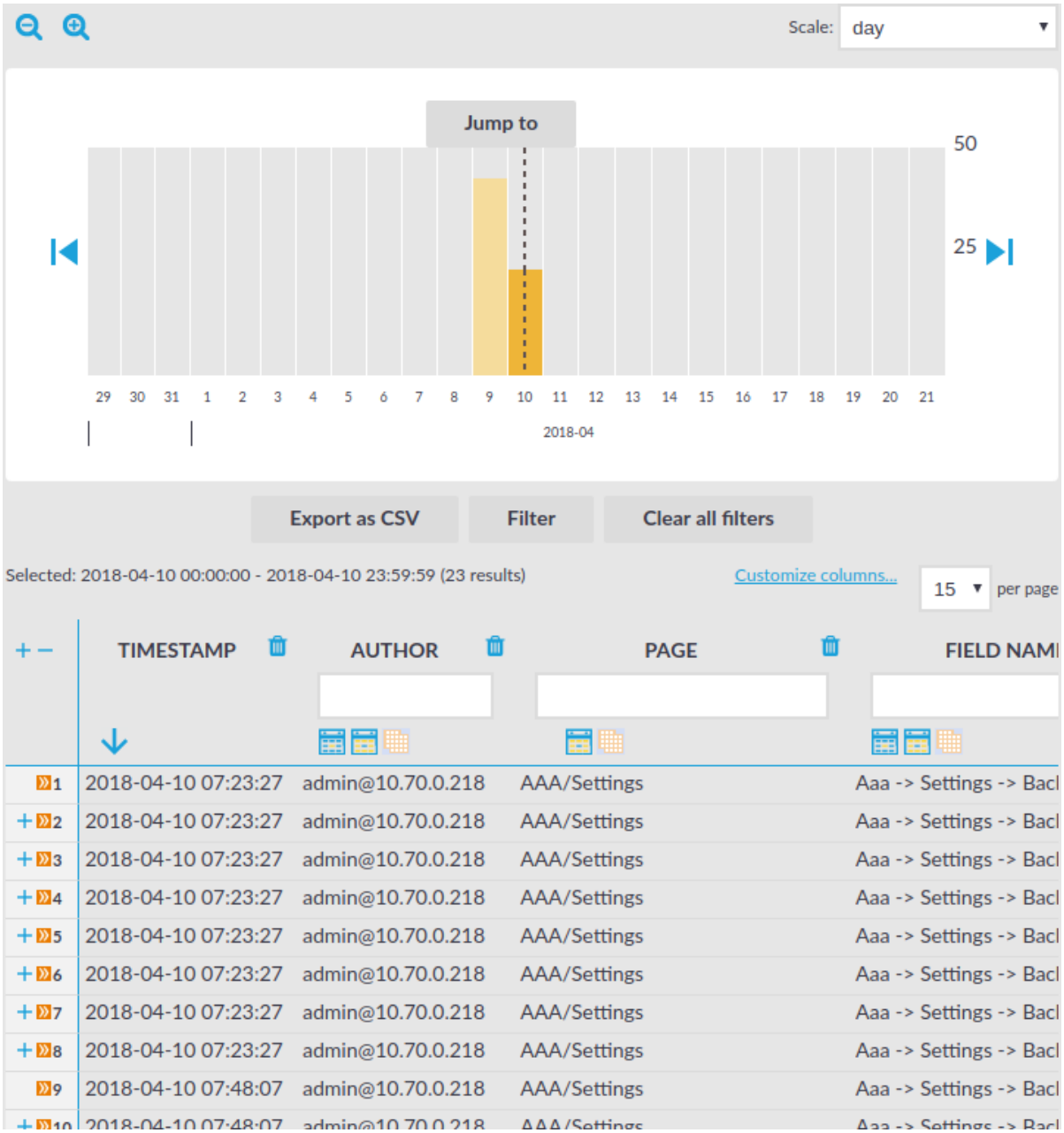
- **api**: View and edit rights for the **Access RPC API** privilege, to access SPS through RPC.
- **http-view**: View all connection and policy settings in the **HTTP Control** menu.
- **http-write**: Edit all connection and policy settings in the **HTTP Control** menu.
- **indexer-view**: View all connection and policy settings in the **Indexer** menu.
- **indexer-write**: Edit all connection and policy settings in the **Indexer** menu.

Listing and searching configuration changes

One Identity Safeguard for Privileged Sessions (SPS) automatically tracks every change of its configuration. To display the history of changes, select **AAA > Accounting**. The changes are displayed on a search interface. For more information on using and customizing this interface, see Using the internal search interface on page 334.

The following information is displayed about each modification:

Figure 79: AAA > Accounting — Browsing configuration changes



- **Timestamp**: The date of the modification.
- **Author**: Username of the administrator who modified the configuration of SPS.
- **Page**: The menu item that was modified.
- **Field name**: The name of the field or option that was modified.
- **New value**: The new value of the configuration parameter.
- **Message**: The changelog or commit log that the administrator submitted. This field is available only if the **Require commit log** option is enabled (see below).

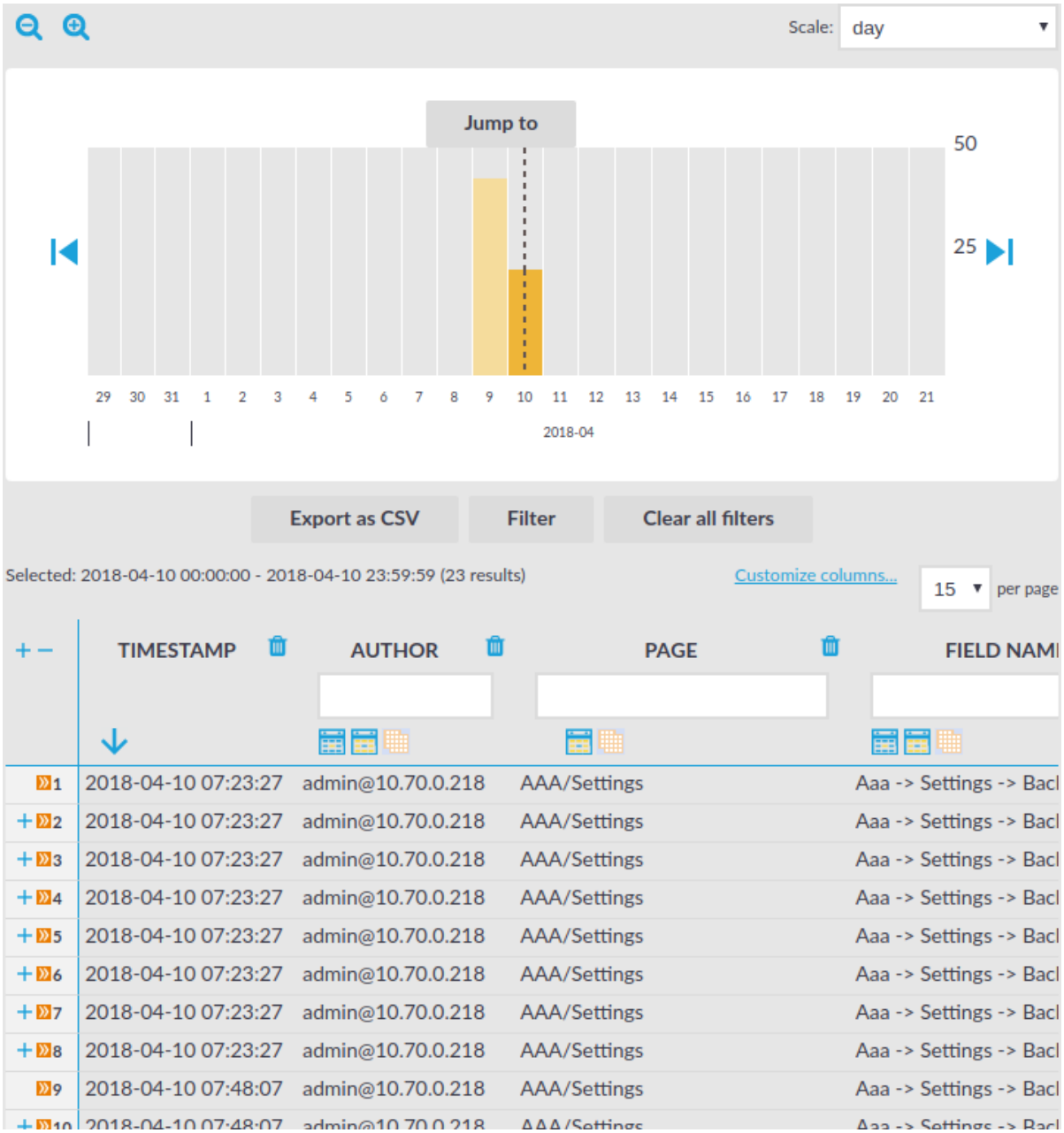
- **Old value**: The old value of the configuration parameter.
- **Swap**: Signs if the order of objects was modified on the page (for example the order of two policies in the list).

To request the administrators to write an explanation to every configuration change, navigate to **AAA > Settings > Accounting settings** and select the **Require commit log** option.

Using the internal search interface

The internal search interface is for browsing and filtering the configuration changes and reports of One Identity Safeguard for Privileged Sessions (SPS).

Figure 80: AAA > Accounting — The internal search interface



The bars display the number of results in the selected interval. Use the and icons to zoom, and the arrows to display the previous or the next intervals. To explicitly select a date, select **Jump to** and set the date in the calendar. You can change the length of the displayed interval with the **Scale** option.

Hovering the mouse above a bar displays the number of entries and the start and end date of the period that the bar represents. Click a bar to display the entries of that period in the table. Use Shift+Click to select multiple bars.

If data is too long to fit on one line, it is automatically wrapped and only the first line is displayed. To expand a row, click +. To shrink the row back to its original size, click −. To

expand/shrink all rows, click the respective button on the header of the table. The rows can also be expanded/shrunk by double clicking on the respective row.

Filtering

The tables can be filtered for any parameter, or a combination of parameters. To filter the list, enter the filter expression in the input field of the appropriate column, and press `Enter`, or click on an entry in the table.

> NOTE:
>
> When you use filters, the bars display the statistics of the filtered results.

Filtering displays also partial matches. For example, filtering the **Author** column on the **AAA > Accounting** screen for `adm` displays all changes performed by users whose username contains the `adm` string.

You can use the  icon to perform an exact search, and the  icon for inverse filtering ("does not include"). To clear filters from a column, click .

To restore the original table, click **Clear all filters**.

Exporting the results

To save the table of search results as a file, click **Export as CSV**. This saves the table as a text file containing comma-separated values. Note that if an error occurs when exporting the data, the exported CSV file will include a line (usually as the last line of the file) starting with a zero and the details of the problem, for example `0;description_of_the_error`.

> **CAUTION:**
>
> **Do not use Export as CSV to export large amounts of data, as exporting data can be very slow, especially if the system is under heavy load.**

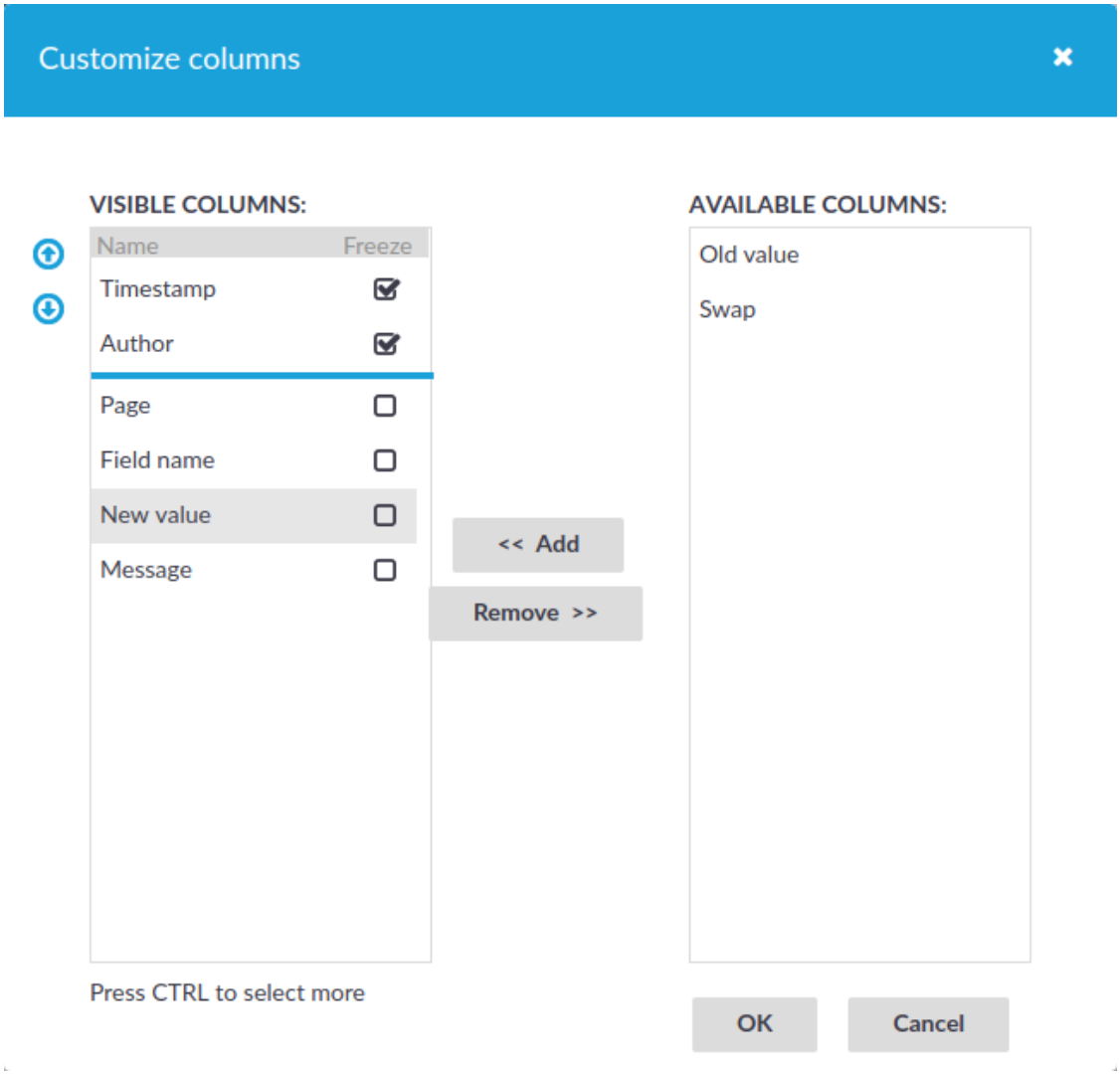
Customizing columns of the internal search interface

The following describes how to customize the data displayed on the interface.

To customize the data displayed on the interface

1. Navigate to the database you want to browse, for example **AAA > Accounting**.
2. Click **Customize Columns**. A pop-up window containing the list of visible and available columns is displayed.

Figure 81: AAA > Accounting — Customizing columns of the general search interfaces



3. The displayed parameters are enlisted in the **Visible columns** field. All other available parameters are enlisted in the **Available columns** field.
    - To add parameters to the **Visible columns** field, select the desired parameter (s) and click **Add**.
    - To remove parameters from the **Visible columns** field, select the desired parameter(s) and click **Remove**.
    - To freeze columns (to make them permanently visible, even when scrolling horizontally), enable the **Freeze** option next to the desired parameter.

    NOTE:

    To select multiple parameters, press`Ctrl`while clicking the items.

4. Click **OK**. The selected information is displayed.

Displaying the privileges of users and user groups

One Identity Safeguard for Privileged Sessions (SPS) version 3.2 and later provides an interface to query the user-rights and privileges of individual users and user groups. To display the privileges of a user or usergroup, navigate to **AAA > Permission Query**, enter the name of the user or group into the respective field, then click **Filter**. Note that:

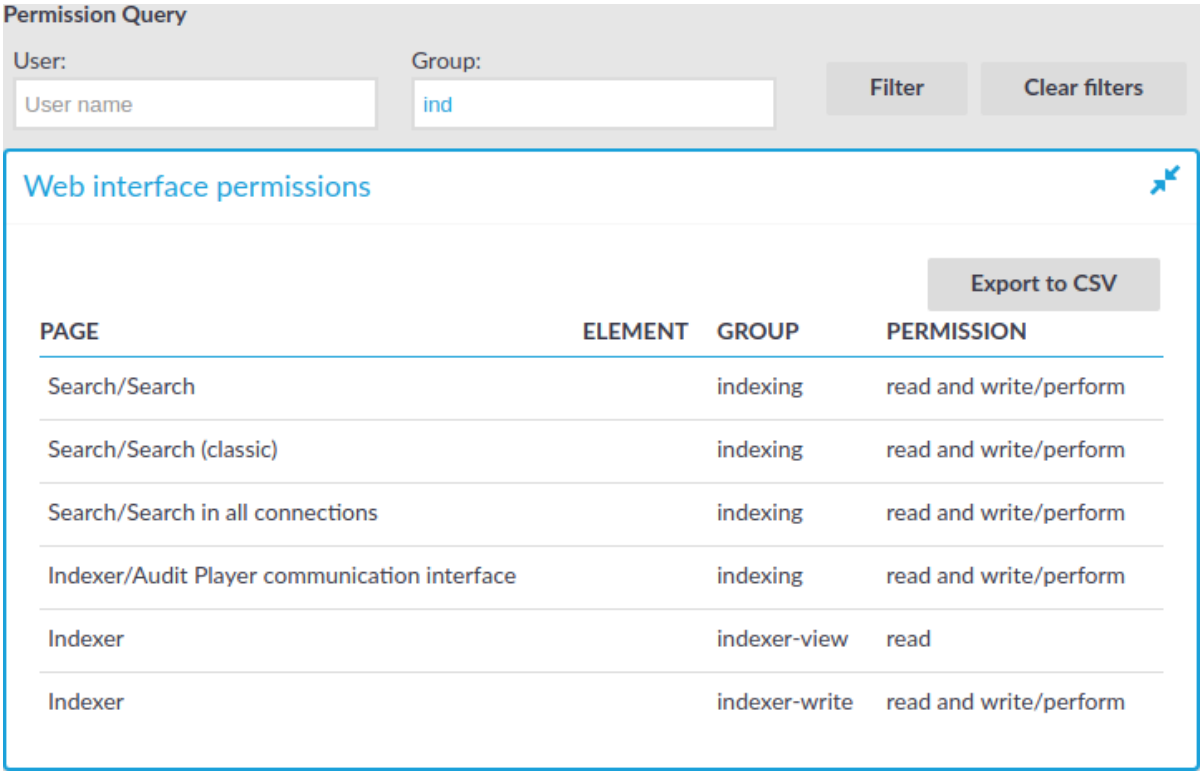
- It is not possible to filter on both the username and the group at the same time.
- Partial matches are also displayed.
- Usergroups can be local usergroups, userlists, or LDAP usergroups.

Web interface permissions

For usergroups accessing the SPS web interface, a table is displayed that lists the pages of the SPS web interface that the user or usergroup can access. The following information is displayed:

- **Page**: The name of the page or group of pages, for example, **Basic Settings**.
- **Element**: If a group has access only to a section of a page, the name of the element is listed here. For example, a particular Channel Policy.
- **Group**: The name of the usergroup.
- **Permission**: The type of access that the user or usergroup has to the page: `read` or `read and write/perform`.

Figure 82: AAA > Permission Query — Displaying web interface permissions



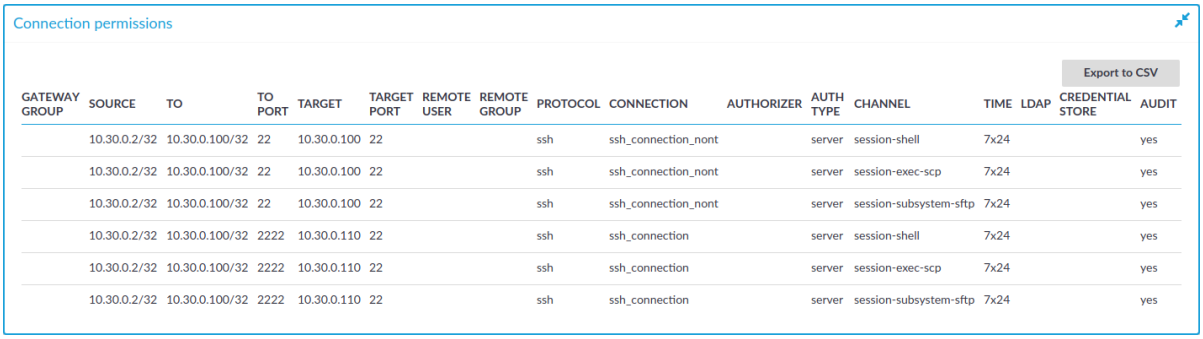
Connection permissions

To review which servers a user or usergroup can access, SPS collects the main information about the connections the user or group is permitted to use. The following information is displayed.

NOTE:

To display the usergroups that can access a specific Connection Policy, open the Connection Policy, then select **Show connection permissions > Show** on the Connections page.

Figure 83: AAA > Connection permissions — Displaying connection permissions



- **Gateway group**: Lists the group memberships required to access the connection. Group memberships can be restricted at the following places:
  - **Connection > Gateway authentication > Groups**
  - **Channel Policies > Gateway group**
  - **Policies > Usermapping Policies > Groups**
- **Source**: Refers to the following field from the session database:

  Source IP: The IP address of the client.
- **To**: Refers to the following field from the session database:

  Destination IP: The IP address of the server as requested by the client.
- **To port**: Refers to the following field from the session database:

  Destination port: The port number of the server as requested by the client.
- **Target**: Refers to the following field from the session database:

  Server IP: The IP address of the server connected by SPS.
- **Target port**: Refers to the following field from the session database:

  Server port: The port number of the server connected by SPS.
- **Remote user**: Refers to the following field from the session database:

  Username on server: The username used to log in to the remote server. This username can differ from the client-side username if usermapping is used in the connection. For details on usermapping, see Configuring usermapping policies on page 740.
- **Remote group**: The group that can access the destination server, as set in the Usermapping Policy (if any).
- **Protocol**: The protocol used in the connection (Citrix ICA, HTTP, RDP, SSH, Telnet, or VNC).
- **Connection**: Refers to the following field from the session database:

  Connection policy ID: The identifier of the connection policy.
- **Authorizer**: Refers to the following field from the session database:

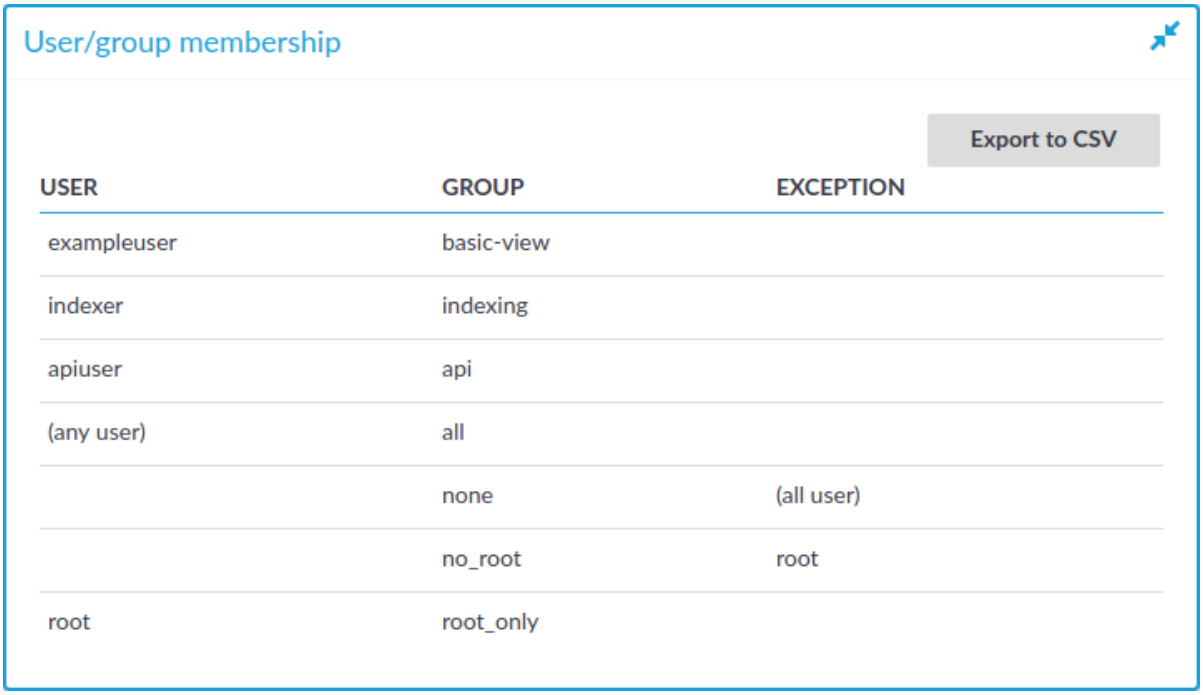
  Four-eyes authorizer: The username of the user who authorized the session. Available only if 4-eyes authorization is required for the channel. For details on 4-eyes authorization, see Configuring four-eyes authorization on page 752.
- **Auth type**: The authentication method used in the client-side connection during gateway authentication.
- **Channel**: The type of the channel, for example, `session-shell`.
- **Time**: The name of the Time Policy used in the connection.
- **LDAP**: The name of the LDAP Server used in the connection (if any).
- **Credential store**: The name of the Credential Store used in the connection (if any).
- **Audit**: Indicates if the connection is recorded into audit trails.

Usergroup memberships

When searching for users, the table displays the group memberships of the matching users. When searching for usergroups, the table displays the members of the matching groups. The following information is displayed:

- **User**: The username of the user.
- **Group**: The name of the usergroup or userlist.
- **Exception**: Usernames that are denied in case of default-deny userlists managed locally on SPS.

Figure 84: AAA > Connection permissions — Displaying usergroup and userlist memberships



Managing One Identity Safeguard for Privileged Sessions (SPS)

The following sections explain the basic management tasks of .

One Identity Safeguard for Privileged Sessions (SPS).

- For basic management tasks (reboot and shutdown, disabling traffic), see Controlling One Identity Safeguard for Privileged Sessions (SPS): reboot, shutdown on page 343.
- For information on managing a cluster of two or more SPS instances, see Managing Safeguard for Privileged Sessions (SPS) clusters on page 346.
- For managing a high availability cluster, see Managing a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster on page 368.
- For instructions on upgrading SPS, see Upgrading One Identity Safeguard for Privileged Sessions (SPS) on page 377.
- For instructions on accessing SPS through console and SSH, see Accessing the One Identity Safeguard for Privileged Sessions (SPS) console on page 389.
- For enabling sealed mode (which disables basic configuration changes from a remote host), see Sealed mode on page 397.
- For information on configuring the out-of-band (IPMI) interface, see Out-of-band management of One Identity Safeguard for Privileged Sessions (SPS) on page 398.
- For managing certificates used on SPS, see Managing the certificates used on One Identity Safeguard for Privileged Sessions (SPS) on page 407.

You can configure your SPS cluster in the following ways:

- Configuration synchronization without a central search.

  It allows you to perform your configuration settings on your Central Management node. Managed Host nodes periodically fetch and merge the settings into their own (configuration synchronization). Central search is not configured and you can search for sessions on each node, including the Central Management node.

  For more information, see Configuration synchronization without a central search.
- Central search with configuration synchronization.

IMPORTANT: One Identity does not recommend having a central search configuration without configuration synchronization.

It allows you to use a Central Management node with a Search Master role to view session data recorded by the minion nodes of your cluster, as well as manage all the nodes in the cluster from one central location.

For more information, see Central search with configuration synchronization.

Controlling One Identity Safeguard for Privileged Sessions (SPS): reboot, shutdown

To reboot or shut down One Identity Safeguard for Privileged Sessions (SPS)

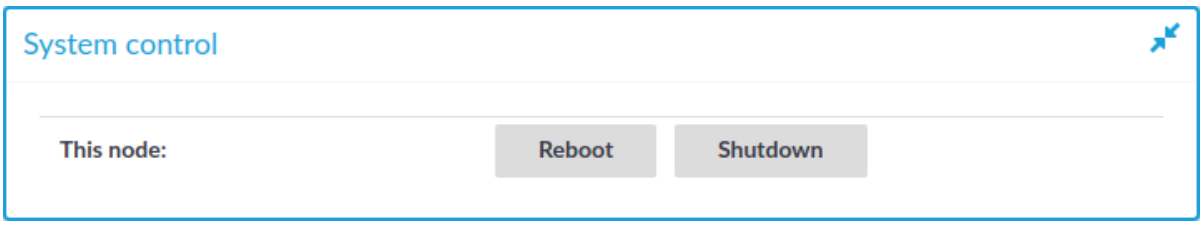
1. Navigate to **Basic Settings > System > System control > This node**.
2. Click the respective action button.

The **Other node** refers to the secondary node of a high availability SPS cluster. For details on high availability clusters, see Managing a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster on page 368.

CAUTION:

- **When rebooting the nodes of a cluster, reboot the other (secondary) node first to avoid unnecessary takeovers.**
- **When shutting down the nodes of a cluster, shut down the other (secondary) node first. When powering on the nodes, start the primary node first to avoid unnecessary takeovers.**
- **When both nodes are running, avoid interrupting the connection between the nodes: do not unplug the Ethernet cables, reboot the switch or router between the nodes (if any), or disable the HA interface of SPS.**

Figure 85: Basic Settings > System > System Control — Performing basic management



NOTE:

Web sessions to the SPS interface are persistent and remain open after rebooting SPS, so you do not have to relogin after a reboot.

During the reboot process, SPS displays information about the progress of the reboot and any possible problems in the following places:

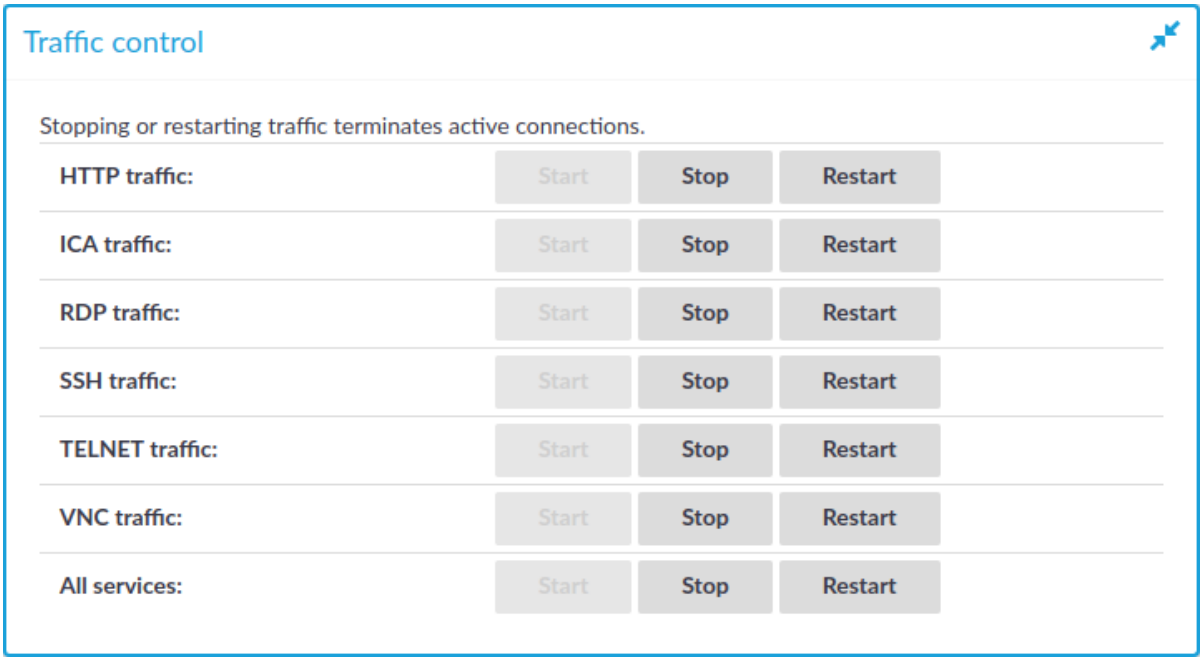
- On the web interface of SPS, at any of the **Listening addresses** configured at **Basic settings > Local Services > Web login (admin and user)**. (After booting, you are directed to the login screen of SPS.)
- On the console, which you can monitor with IPMI (ILOM) or console access.

The information displayed in the browser and on the console is the same.

Disabling controlled traffic

The following describes how to temporarily disable some or all of the controlled traffic to the protected servers.

Figure 86: Basic Settings > System > Traffic control — Disabling the controlled traffic



CAUTION:

Disabling traffic that way is only temporary. Connections will be enabled again after committing any other change from the SPS web interface. For details on how to permanently disable a type of traffic, see Disabling controlled traffic permanently on page 345.

NOTE:

Disabling the traffic affects only the traffic configured in the Connection policies, other traffic can pass SPS even if the all traffic is disabled. For details on configuring Connection policies, see General connection settings on page 434.

To temporarily disable some or all of the controlled traffic to the protected servers

1. Navigate to the **Basic Settings > System > Traffic control**.
2. - To disable SSH traffic, click **Stop** in the **SSH traffic** field. Note that this also stops all other traffic forwarded in SSH, for example X11.
   - To disable RDP traffic, click **Stop** in the **RDP traffic** field.
   - To disable Telnet and TN3270 traffic, click **Stop** in the **Telnet traffic** field.
   - To disable VNC traffic, click **Stop** in the **VNC traffic** field.
   - To disable all types of traffic, click **Stop** in the **All services** field.

   The **System monitor** displays the status of all types of traffic.

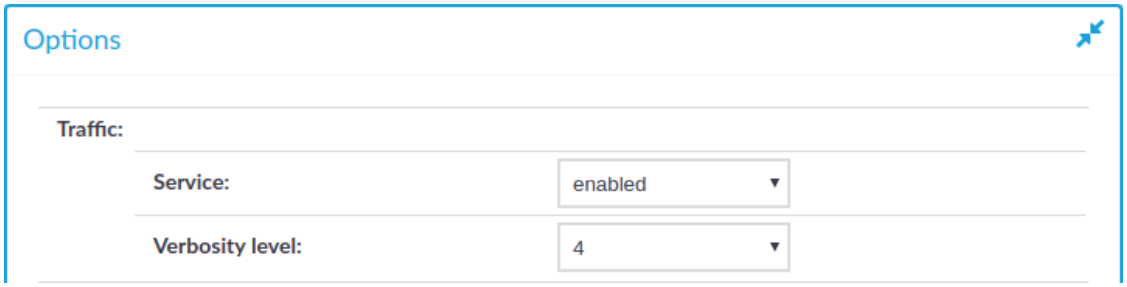
Disabling controlled traffic permanently

NOTE:

Disabling the traffic affects only the traffic configured in the Connection policies, other traffic can pass SPS even if the all traffic is disabled. For details on configuring Connection policies, see General connection settings on page 434.

To disable controlled traffic permanently

1. **Figure 87: <Protocol name> Control > Global Options — Disabling the controlled traffic persistently**

   Navigate to the **Global Options** page of the traffic type you want to disable, for example to **SSH Control > Global Options** to disable SSH traffic.
2. Set the **Traffic > Service** field to `disabled`.
3. Click Commit.

Managing Safeguard for Privileged Sessions (SPS) clusters

When you have a set of two or more Safeguard for Privileged Sessions (SPS instances in your deployment, you can join them into a cluster. This has several advantages. You can:

- Manage the nodes from one central location.
- Monitor their status and update their configuration centrally.
- Search all session data recorded by all nodes in the cluster on a single node.
- Scale the performance of the cluster by adding new nodes and joining them to the cluster easily.
- Extend auditing to other networks by adding new nodes to the cluster and joining them to the cluster.

This is achieved by assigning roles to the individual nodes in your cluster: you can set one of your Safeguard for Privileged Sessions nodes to be the Central Management node and the rest of the nodes are managed from this central node.

NOTE:

All nodes in a cluster must run the same version of SPS.

NOTE:

One Identity recommends managing not more than a few tens of instances from the Central Management node.

Nodes in the cluster connect to each other using IPsec.

Cluster roles

You can assign any of the following roles to your nodes:

- *Central Management*: There can be only one node with this role in the cluster.

  The purpose of having a Central Management node is to have a node with a central configuration, which can be synchronized to the other nodes of the cluster. Any changes that you make in the cluster's configuration on this node (for example, role changes, host address changes, and so on) are fetched by all the other nodes and merged into their configuration.

  The Central Management node also has status information about all the other nodes in the cluster, so you can check the health of the cluster on this node. Status information contains:

  - the result and time of the last attempt at configuration fetch and configuration update

    - errors and warnings that may have occurred during configuration fetch and configuration update
- *Managed Host*: There can be several nodes with this role.

  Nodes with the Managed Host role synchronize their entire configuration from the Central Management node, not only those elements of the configuration that are related to the cluster.

  Managed Host nodes send their status information to the Central Management node every 10 seconds.
- *Search Master*: There can be only one node with this role.

  The Search Master node is the one node in the cluster on which you can search all the session data recorded by other nodes in the cluster, provided that the other nodes have been assigned the Search Minion role.

  > NOTE: A One Identity Safeguard for Privileged Sessions (SPS) node with the Search Master role cannot be used for monitoring network traffic, or for session recording and auditing purposes.
  >
  > Before assigning this role to a node, read the limitations that apply to Search Master nodes carefully: Searching session data on a central node in a cluster

  This role can only be assigned to nodes that either have the Managed Host or Central Management role. This is required so that the configuration of Search Minion nodes and the Search Master node are always in sync.

  If there is no configuration synchronization between the node acting as the Search Master and the Search Minion nodes, then session data may show up on the Search interface of the Search Master that come from connections that do not match the connection policies set up on the Search Master (because they come from session data recorded by the Search Minions).
- *Search Minion*: There can be several nodes with this role.

  Nodes with the Search Minion role send session data that they recorded to the Search Master for central search purposes. The session data recorded by a Search Minion node is not searchable on the node itself, only on the Search Master.

  This role can only be assigned to nodes that either have the Managed Host or Central Management role. This is required so that the configuration of Search Minion nodes and the Search Master node are always in sync.

  If there is no configuration synchronization between the node acting as the Search Master and the Search Minion nodes, then session data may show up on the Search interface of the Search Master that come from connections that do not match the connection policies set up on the Search Master (because they come from session data recorded by the Search Minions).
- *Search Local*: There can be several nodes with this role.

  Nodes with the Search Local role keep the session data that they recorded for local searching. The session data recorded by a Search Local node is searchable on the node itself, but not on the Search Master (if there is one).

- *No role*: Nodes without any role fetch only the cluster-related elements of the configuration from the Central Management node.

  Nodes with no roles send their status information to the Central Management node every 10 seconds.

Nodes keep their role in the cluster after a system restore.

For more information about configuration synchronization, see Configuration synchronization across nodes in a cluster on page 355.

For more information about central search, see Searching session data on a central node in a cluster on page 738.

Enabling cluster management

To enable cluster management, enable the cluster interface on all nodes that you want to be part of your Safeguard for Privileged Sessions (SPS) cluster. Complete the following steps on each and every node of the cluster.

NOTE:

All nodes in a cluster must run the same version of SPS.

Prerequisites

Nodes in the cluster connect to each other using IPsec, which requires UDP ports 500 and 4500 to be open in the firewalls between the nodes.

Prerequisite

The nodes in the cluster must connect to each other using IPsec.

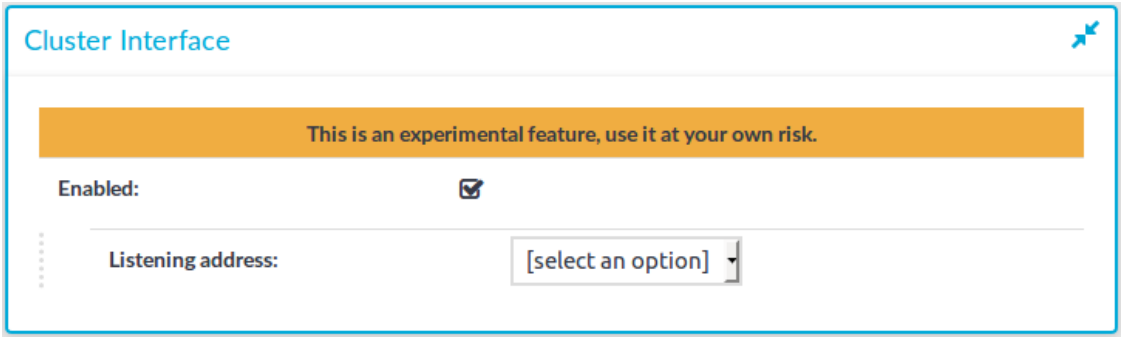
NOTE: IPsec requires UDP ports `500` and `4500` to be open bidirectionally in the firewalls between the nodes.

To enable cluster management

1. Navigate to **Basic Settings > Local Services > Cluster Interface**.
2. Select **Enabled**.

   The **Listening address** field is displayed.
3. Select a cluster interface for the node to listen on.

Figure 88: Basic Settings > Local Services > Cluster Interface — Enabling cluster management



4. Click Commit.

Building a cluster

Build a cluster by promoting a Safeguard for Privileged Sessions (SPS) node to the role of the Central Management node, and then join other nodes to your cluster.

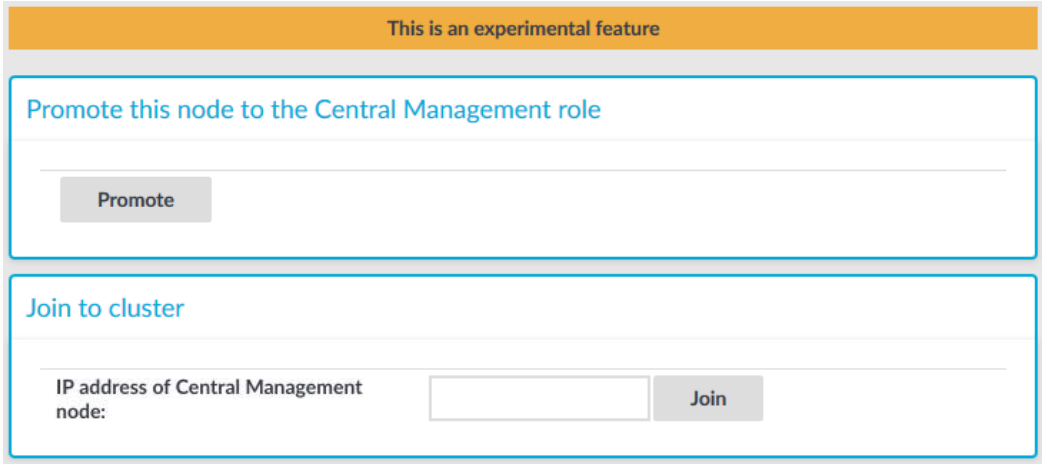
Prerequisites

Enable the cluster interface on all nodes that you want to be part of your cluster. For details on how to do that, see Enabling cluster management on page 348.

To build a cluster

1. Promote a node to be the Central Management node:
    a. Navigate to **Basic Settings > Cluster management > Promote this node to the Central Management role**.

Figure 89: Basic Settings > Cluster management — Promote node to become Central Management node in cluster

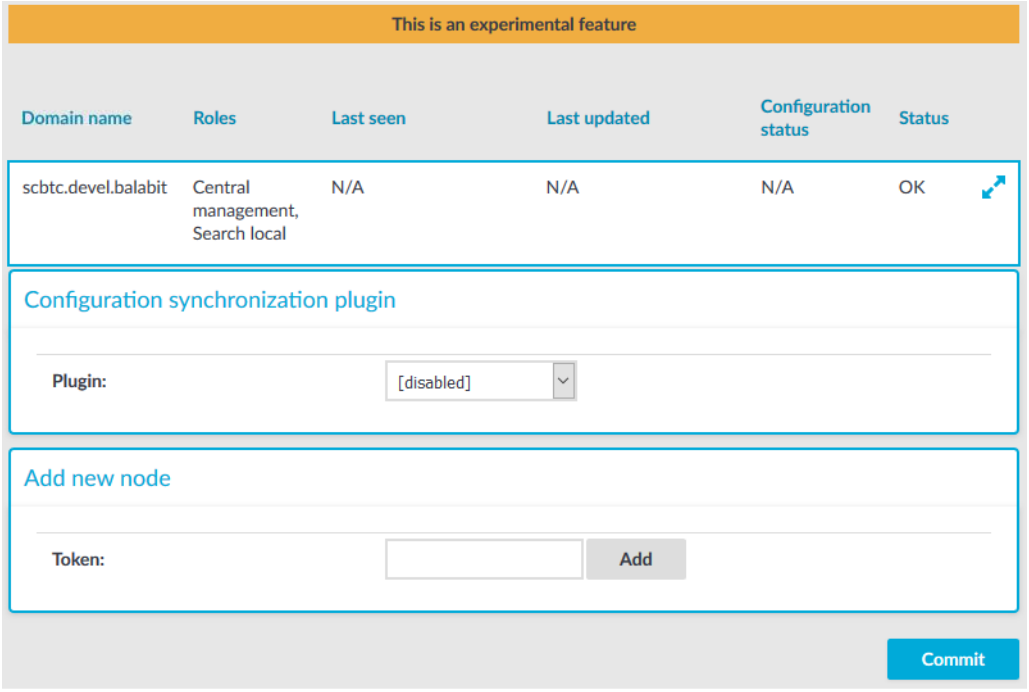


b. Click **Promote**.

> NOTE:
>
> This is an action that you cannot undo or modify.

The Central Management node you have just added is displayed.

Figure 90: Basic Settings > Cluster management — View Central Management node in cluster



You can also promote a node through the REST API. For details, see "Promote a Safeguard for Privileged Sessions node to be the Central Management node in a new cluster" in the REST API Reference Guide.

2. Join additional nodes to your cluster:

   CAUTION:

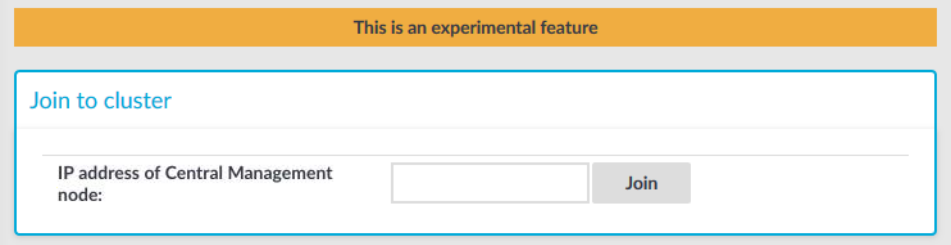
   Configuration options that you set on a node before joining it to the cluster will be overwritten by the configuration of the Central Management node. For example, policies and protocol-specific settings will be overwritten once you assign the Managed Host role to the node. Managed Host roles periodically fetch the configuration of the Central Management node and merge it into their own. This is called configuration synchronization.

   To avoid the loss of policies and settings that are specific to your Managed Host node, use a configuration synchronization plugin. Such plugins enable you to limit the scope of configuration synchronization.

   For more information, see Configuration synchronization across nodes in a cluster on page 355.

   a. On the node that you want to join to the cluster, navigate to **Basic Settings > Cluster management**.

      Figure 91: Basic Settings > Cluster management — Join node to cluster

   b. In the **IP address of Central Management node** field, enter the IP address of the Central Management node.

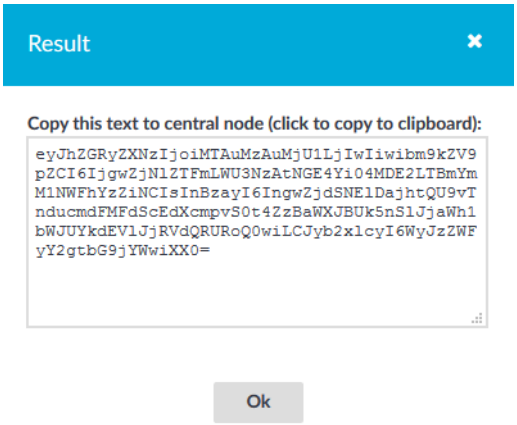
   c. Click **Join**.

      NOTE:

      This is an action that you cannot undo. After clicking **Join**, you will still be able to change the IP address. However, you will not be able to promote this node to be the Central Management node.

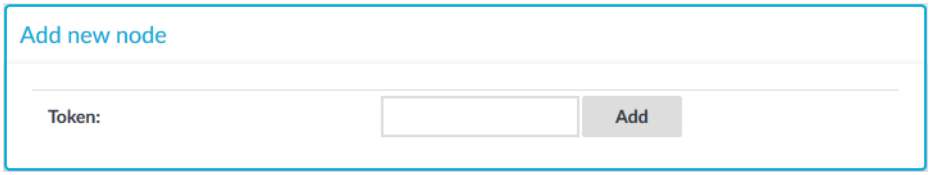
      A dialog box with the token of the node pops up.

Figure 92: Token of node to join to cluster



d. Copy the token, and click **Ok**.

e. On the Central Management node, navigate to **Basic Settings > Cluster management > Add new node**.

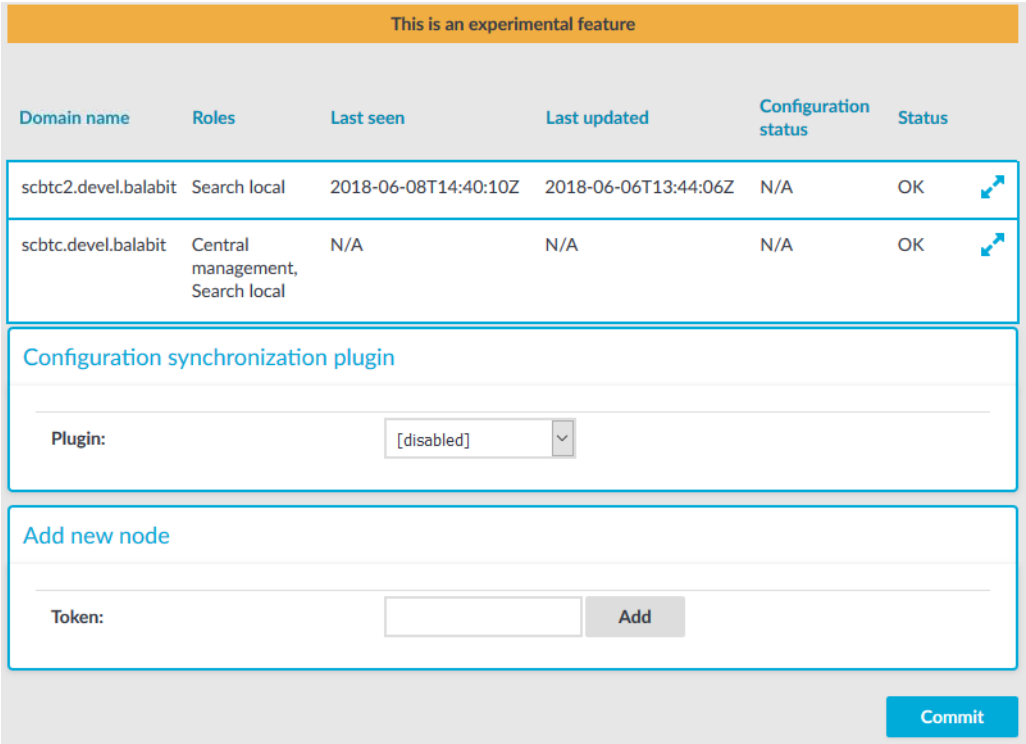


f. Paste the token in the **Token** field.

g. Click **Add**.

The node you have added is displayed in the list of nodes on the Central Management node.

Figure 93: Basic Settings > Cluster management — Nodes added to cluster



If you want to centrally manage the configuration of the node(s) you have joined to the cluster, assign the Managed Host role to them. For details on how to do that, see Assigning roles to nodes in your cluster on page 353.

You can join additional nodes to your cluster through the REST API, too. For details, see "Join node(s) to the cluster" in the REST API Reference Guide.

Assigning roles to nodes in your cluster

By default, nodes do not have any roles assigned to them. The only exception is the Central Management node, which you specifically promoted to fulfill that role. To assign a role to a node in the cluster, complete the following steps.

To assign roles to nodes in your cluster

1. On the web interface of your Central Management node, navigate to **Basic Settings > Cluster management**. This page displays all nodes of a cluster.
2. Click ⤢ next to the node that you want to assign a role to. The available roles are displayed.
3. Select the role that you want to assign to the node. For details on what each role means, see Cluster roles on page 346.

CAUTION:

Configuration options that you set on a node before joining it to the cluster will be overwritten by the configuration of the Central Management node. For example, policies and protocol-specific settings will be overwritten once you assign the Managed Host role to the node. Managed Host roles periodically fetch the configuration of the Central Management node and merge it into their own. This is called configuration synchronization.

To avoid the loss of policies and settings that are specific to your Managed Host node, use a configuration synchronization plugin. Such plugins enable you to limit the scope of configuration synchronization.

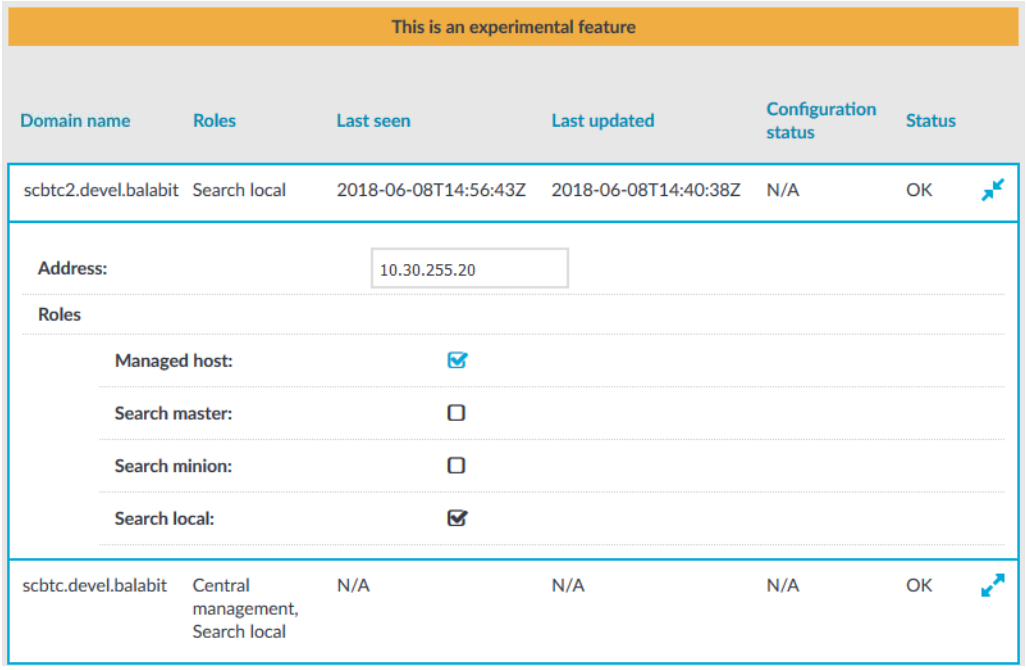
For more information, see Configuration synchronization across nodes in a cluster on page 355.

NOTE:

Regarding search roles:

- Ensure that each node has a search role and only one search role.
- You must assign the Search Master role before you can assign Search Minion roles.

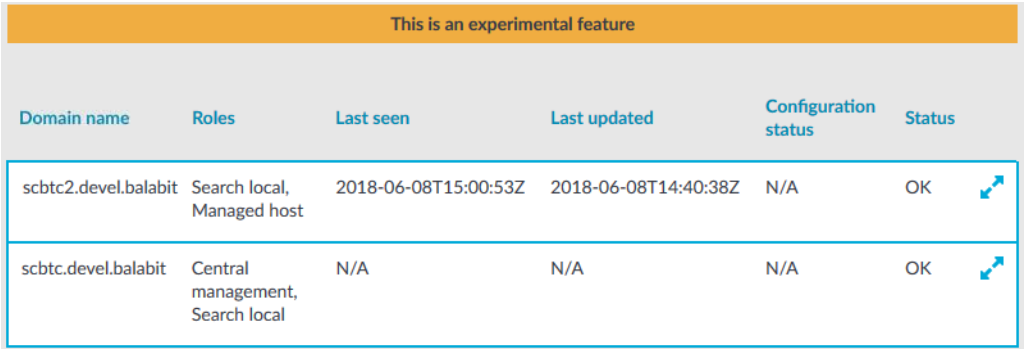
Figure 94: Basic Settings > Cluster management — Assign role to node



4. Click Commit.

The role you assigned is now displayed next to the node, under **Roles**.

Figure 95: Basic Settings > Cluster management — Role is assigned to node



You can assign roles to your nodes through the REST API, too. For details, see "Assign a role to a node" in the REST API Reference Guide.

Configuration synchronization across nodes in a cluster

Nodes fetch their configuration from the Central Management node, and merge it into their own configuration. Depending on their role, nodes may merge the whole configuration into their own (Managed Host nodes), or only the cluster-specific parts (nodes with no roles assigned). Whenever a configuration change is made on the Central Management node and the change is committed, it is synchronized to all nodes in the cluster as soon as the nodes fetch the latest configuration from the Central Management node.

Configuration synchronization has some implications for the SSH keys (if any) that have been recorded on your nodes before they were joined to the cluster. For details, see Configuration synchronization and SSH keys on page 356.

In some cases, you may want to keep certain parts of the configuration on your nodes outside the scope of configuration synchronization. In that case, use a configuration synchronization plugin. For more information, see Using a configuration synchronization plugin on page 356.

The following configuration settings are never overwritten by configuration synchronization, even when not using a configuration synchronization plugin:

- settings related to networking (**Basic Settings > Network**)
- settings related to local services (**Basic Settings > Local Services**)
- settings related to the management of SPS (**Basic Settings > Management**)
- settings related to the license of SPS (**Basic Settings > System > License**)

Configuration synchronization and SSH keys

The only SSH keys present on Managed Host nodes will always be the ones that have been recorded by the Central Management node. This is because the SSH keys stored on the Central Management node get synced to the Managed Host nodes during configuration synchronization. This means that the SSH keys recorded on the Managed Host nodes before they were joined to the cluster are overwritten by the keys stored on the Central Management node.

The Central Management records new SSH keys in the following cases:

- The Central Management node is configured to **Accept key for the first time** and a new key is automatically recorded when the Central Management node interacts with a server for the first time.
- A new key is recorded on the Central Management node on the **SSH Control > Server Host Keys** page and this change is committed.

These are the keys that get synced to your Managed Host nodes.

Using a configuration synchronization plugin

When synchronizing the central configuration across nodes, you may want to:

- Keep certain parts in the configuration of individual nodes as-is.
- Tailor certain parts of the central configuration to specific needs of individual nodes in the cluster (for example, your nodes may access external services at different network addresses).

You can achieve all of these by using a configuration synchronization plugin that contains transformations for the problematic parts. The plugin only runs on nodes that have the Managed Host role.

Customizing certain parts or features of a node using a configuration synchronization plugin has the same limitations as configuring SPS through the REST API. In other words, whatever you can configure through the REST API, you can configure the exact same settings using the plugin. One notable difference between the REST API and the plugin is that using the REST API, you can only read certain types of data (such as keys and passwords), while using the configuration synchronization plugin, you can write these types of data as well.

For details on how to configure SPS using the REST API, see REST API Reference Guide.

Data structures in the plugin are represented as nested JSON objects. For object references, the plugin uses keys.

The plugin works with the following key parameters:

- `local_config`: The current configuration of a Managed Host node (those parts that can be configured through the REST API).
- `merged_config`: The configuration of the Central Management node that is about to be synced to the Managed Host node (those parts that can be configured through the

REST API), with settings related to networking, local services, management, and the license of SPS whitelisted. These settings are never overwritten by configuration synchronization.

- `node_id`: The unique ID of the Managed Host node in the cluster (you can retrieve this identifier by querying the `/api/cluster/nodes` endpoint through the REST API).
- `plugin_config`: The configuration of the plugin provided as free-form text. Specifying the configuration of the plugin is optional. It enables you to run configuration synchronization on each cluster with different parameters if you have multiple clusters.

Example: Customizing an IP address in a connection policy

For example, an RDP connection policy on a Managed Host node specifies the following client and target addresses:

```
$ curl ... https://<url-of-Central-Management-node>/api/configuration/rdp/connections/<id-of-the-connection-policy>
```

```
{
       "body": {
              "network": {
                    "clients": [
                          "0.0.0.0/0"
                          ],
                    "ports": [
                          3389
                          ],
                   "targets": [
                         "10.30.255.28/24"
                         ]
              },
       },
       ...
 }
```

Let's suppose that on the Central Management node, an RDP connection policy is configured with these details:

```
$ curl ... https://<url-of-Managed-Node>/api/configuration/rdp/connections/<id-of-the-connection-policy>
```

```
{
       "body": {
              "network": {
                     "clients": [
                            "0.0.0.0/0"
                            ],
                     "ports": [
                            3389
                            ],
                   "targets": [
                          "10.30.255.8/24"
                          ]
              },
       },
       ...
}
```

To ensure that the details of the connection policy on the Managed Host node are kept as-is after configuration synchronization, add the following lines to the plugin `main.py` file:

```
$ cat main.py
def merge(local_config: dict, merged_config: dict, node_id: str, plugin_
config: str, **kwargs):
    merged_config['rdp']['connections'][<id-of-the-connection-policy>]
['network']['targets'][0] = "10.30.255.8/24"
    return merged_config
```

Due to possible new (as yet undefined) parameters, it is good practice to close the parameter list of the merge function with `**kwargs`.

In case you need assistance with writing customized transformations, contact our Professional Services Team, and a One Identity Service Delivery Engineer will be able to help you.

NOTE:

Configuration settings related to networking (**Basic Settings > Network**), local services (**Basic Settings > Local Services**), and the license of SPS (**Basic Settings > System > License**) are not overwritten on the nodes by configuration synchronization even when not using a plugin.

For the management of SPS (**Basic Settings > Management**), the following configuration settings are not overwritten:

- Mail settings
- System backup
- SSL certificates
- Change root password

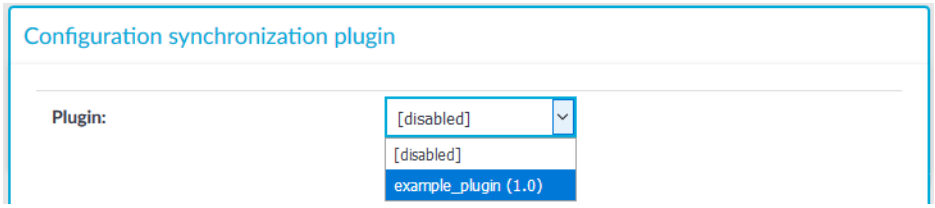
To use a configuration synchronization plugin

1. Upload a configuration synchronization plugin:
   a. Navigate to **Basic Settings > Plugins**.
   b. Browse for the file, and click **Upload**.

      NOTE:

      It is not possible to upload or delete plugins if SPS is in sealed mode.
2. Enable the plugin:
   a. Navigate to **Basic Settings > Cluster management > Configuration synchronization plugin**.
   b. Select the plugin you have uploaded.

      Figure 96: Basic Settings > Cluster management — Select configuration synchronization plugin

3. *Optional:* Enter the configuration of the plugin in the **Configuration** free-form text field. Specifying the configuration of the plugin enables you to run configuration synchronization on each cluster with different parameters if you have multiple clusters.
4. Click Commit.

You can also upload and enable the configuration synchronization plugin through REST. For details, see "Upload and enable a configuration synchronization plugin" in the REST API Reference Guide.

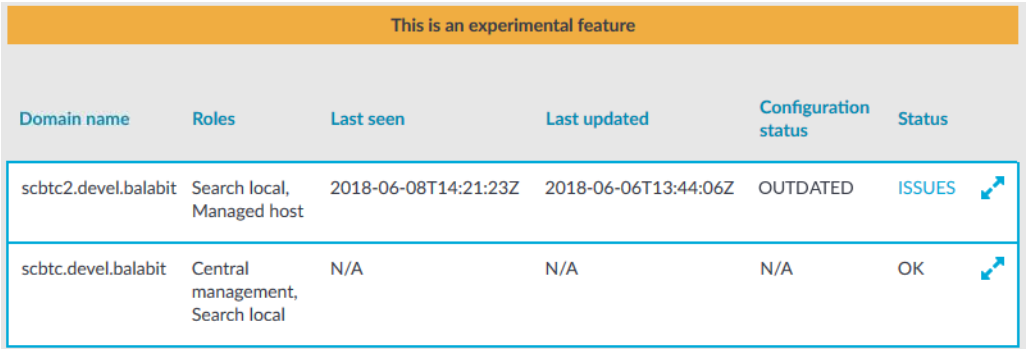
Monitoring the status of nodes in your cluster

The following describes how to monitor the status of nodes in your cluster.

To monitor the status of nodes in your cluster

1. On the web interface of your Central Management node, navigate to **Basic Settings > Cluster management**. This page displays all nodes of a cluster.

Figure 97: Basic Settings > Cluster management — Monitor status of nodes



The following status information is displayed for each node:

- **Last seen**: The last time the node sent status information to the Central Management node, in ISO 8601 format.
- **Last updated**: The last time the node's configuration was synchronized, in ISO 8601 format.
- **Configuration status**: Indicates the status of configuration synchronization. It has the following values:
  - **UP-TO-DATE**: The node has fetched the latest configuration from the Central Management node, and has applied it. It is in sync with the Central Management node.
  - **PENDING**: There has been a configuration change on the Central Management node, and the change has not been synchronized yet to the node.
  - **OUTDATED**: There has been some error on the node and therefore it is running an old configuration.
  - **NOT FETCHED**: The node has not fetched any configuration yet.
  - **N/A**: The node is the Central Management node, so it is not fetching its configuration from any other node.
- **Status**: Indicates whether any issues occurred during configuration synchronization. It has the following values:

- **OK**: Configuration synchronization was successful, no issues occurred.
- **ISSUES**: While synchronizing configuration, some issue(s) occurred. Click **ISSUES** to find out the details.
- **OFFLINE**: Indicates that status information was sent by the node longer than 60 seconds ago.

You can monitor the status of your nodes through the REST API, too. For details, see "Query the status of all nodes in the cluster" in the REST API Reference Guide and "Query one particular node" in the REST API Reference Guide.

Updating the IP address of a node in a cluster

When you have joined a node to a cluster, you can still change the IP address of the node if it is a Managed Host node.

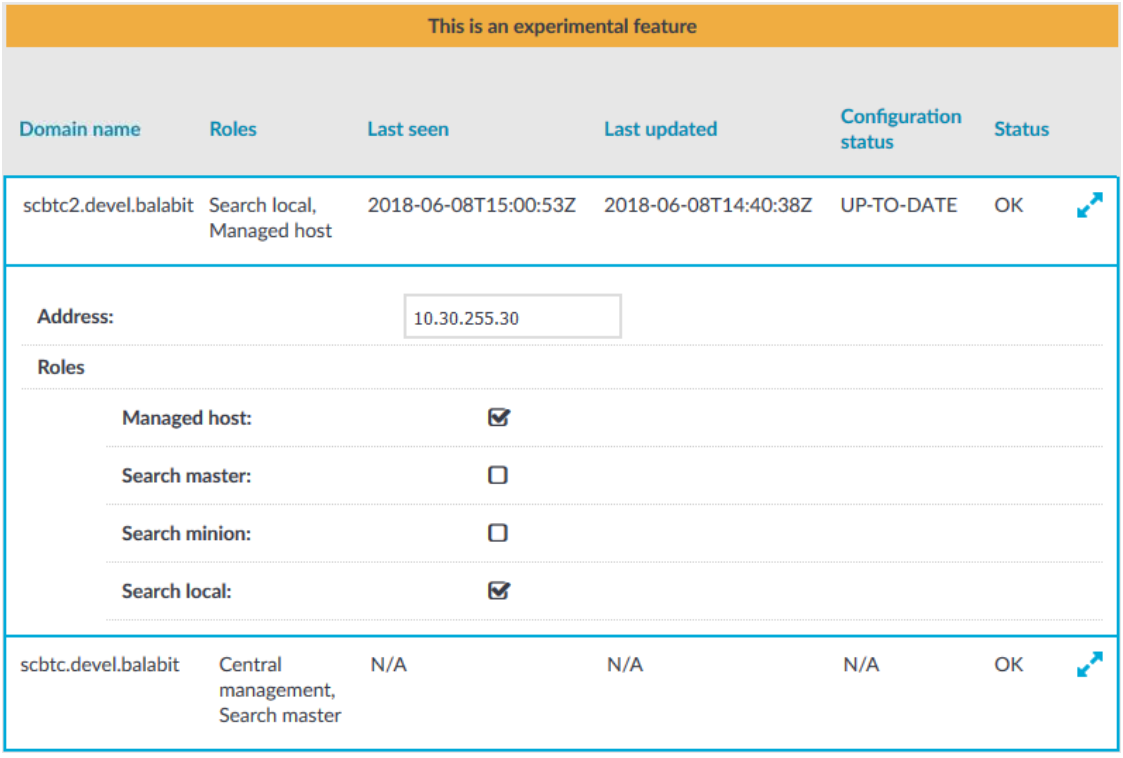
To update the IP address of a Managed Host node that is already the member of a cluster

1. On the web interface of your Central Management node, navigate to **Basic Settings > Cluster management**.
2. Click next to the Managed Host node that you want to update.
3. In the **Address** field, update the IP address of the node.

   CAUTION:

   Ensure that you are making the change for the Managed Host node. Do not change the IP address of the Central Management node.

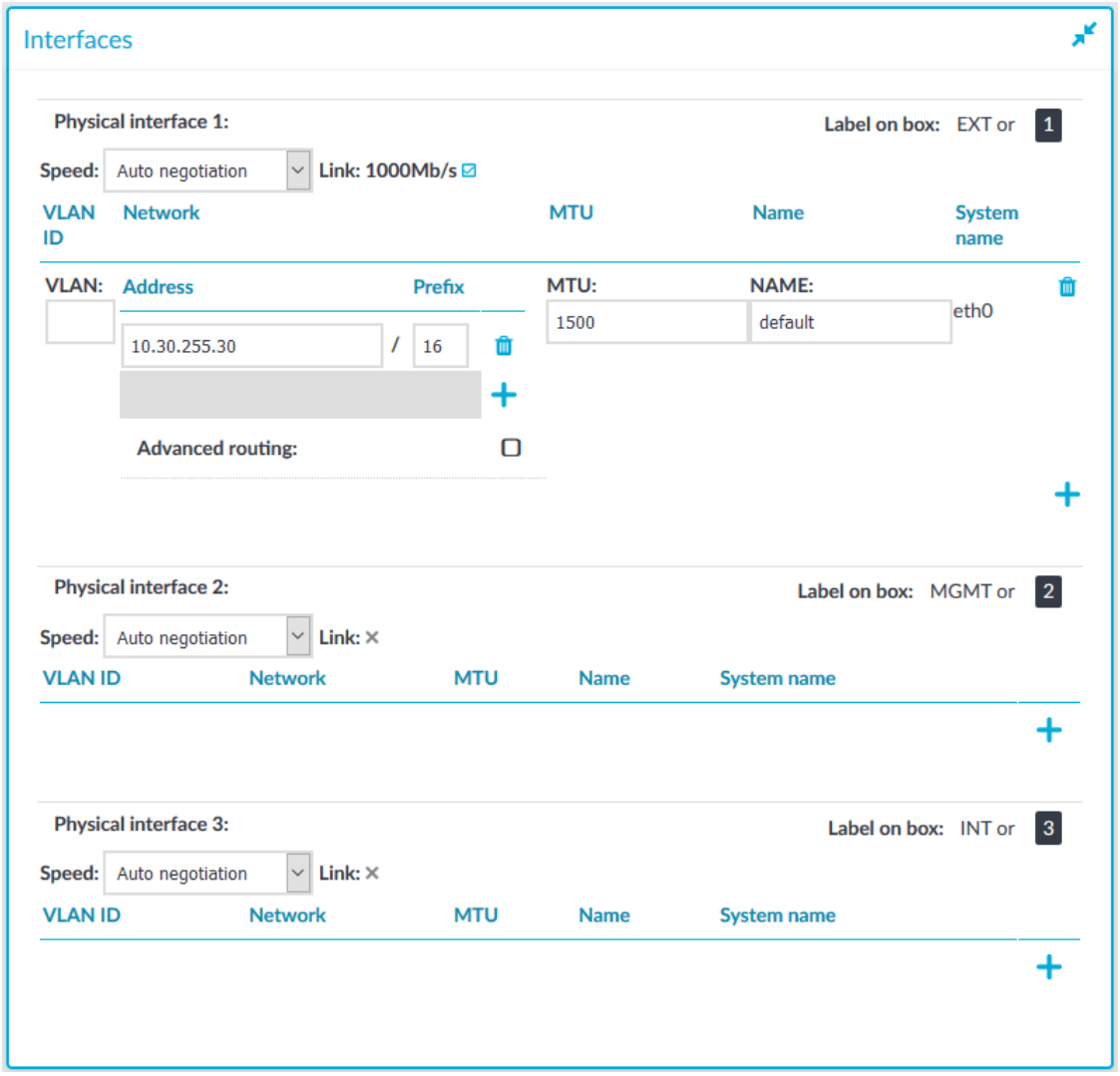
Figure 98: Basic Settings > Cluster management — Update IP address of node



4. Click Commit.
5. On the web interface of the node whose IP address you want to update, navigate to **Basic Settings > Network > Interfaces**.

6. In the **Address** field, update the IP address of the node.

Figure 99: Basic Settings > Network > Interfaces — Update IP address of node



7. Click Commit.

Managing a cluster with configuration synchronization without central search

You can configure your SPS cluster in the following ways:

- Configuration synchronization without a central search.

  It allows you to perform your configuration settings on your Central Management node. Managed Host nodes periodically fetch and merge the settings into their own (configuration synchronization). Central search is not configured and you can search for sessions on each node, including the Central Management node.

  For more information, see Configuration synchronization without a central search.

- Central search with configuration synchronization.

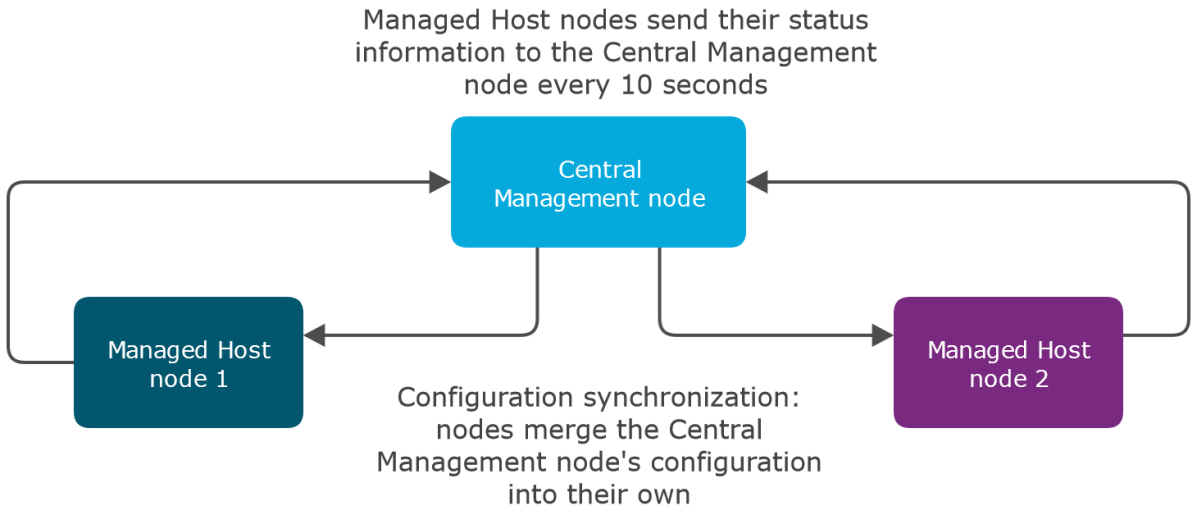
  IMPORTANT: One Identity does not recommend having a central search configuration without configuration synchronization.

  It allows you to use a Central Management node with a Search Master role to view session data recorded by the minion nodes of your cluster, as well as manage all the nodes in the cluster from one central location.

  For more information, see Central search with configuration synchronization.

The following figure shows a cluster with configuration synchronization without central search.

Figure 100: Configuration synchronization without central search



The figure above is an example of an SPS cluster configured as follows:

- There is a Central Management node.
- There are two Managed Host nodes (Managed Host node 1 and 2).
- The Central Management node is connected to the two Managed Host nodes.
- The Managed Host nodes fetch their configuration from the Central Management node, and merge it into their own configuration.
- The Managed Host nodes send their status information to the Central Management node every 10 seconds.

The Central Management node and the connected Managed Host nodes require different configuration settings as described in the table below:

Table 6: Managing a configuration synchronization without a central search

Role	Use and configuration settings
Central Management node	• Use it as a node with a central configuration, which is synchronized to the other nodes of the cluster. • Perform your configuration settings on this node. Managed Host nodes periodically fetch and merge these configuration settings into their own (configuration synchronization). • For backup and archive, configure a backup and archive server on your minion node, as well as on your Central Management node. • Ensure that you configure high availability (HA) for each node (for both your Central Management node and the Managed Host nodes). Also ensure that the Central Management node has a system backup configured. • You can search for all the sessions recorded on this node.
Managed Host node	• Use it to record sessions and send status information to the Central Management node. • Do not perform configuration settings on the minion. These are overwritten during configuration synchronization. NOTE: All configuration settings that you make on the minions are overwritten during configuration synchronization except the node specific configuration. • Set external and internal indexers. • For backup and archive, configure a backup and archive server on your minion node, as well as on your Central Management node. • Ensure that you configure high availability (HA) for each node (for both your Central Management node and the Managed Host nodes). Also ensure that the Central Management node has a system backup configured. • You can search for all the sessions recorded on this node.

For more information on each role, see Cluster roles.

Managing a cluster with central search configuration and configuration synchronization

You can configure your SPS cluster in the following ways:

- Configuration synchronization without a central search.

  It allows you to perform your configuration settings on your Central Management node. Managed Host nodes periodically fetch and merge the settings into their own (configuration synchronization). Central search is not configured and you can search for sessions on each node, including the Central Management node.

  For more information, see Configuration synchronization without a central search.

- Central search with configuration synchronization.

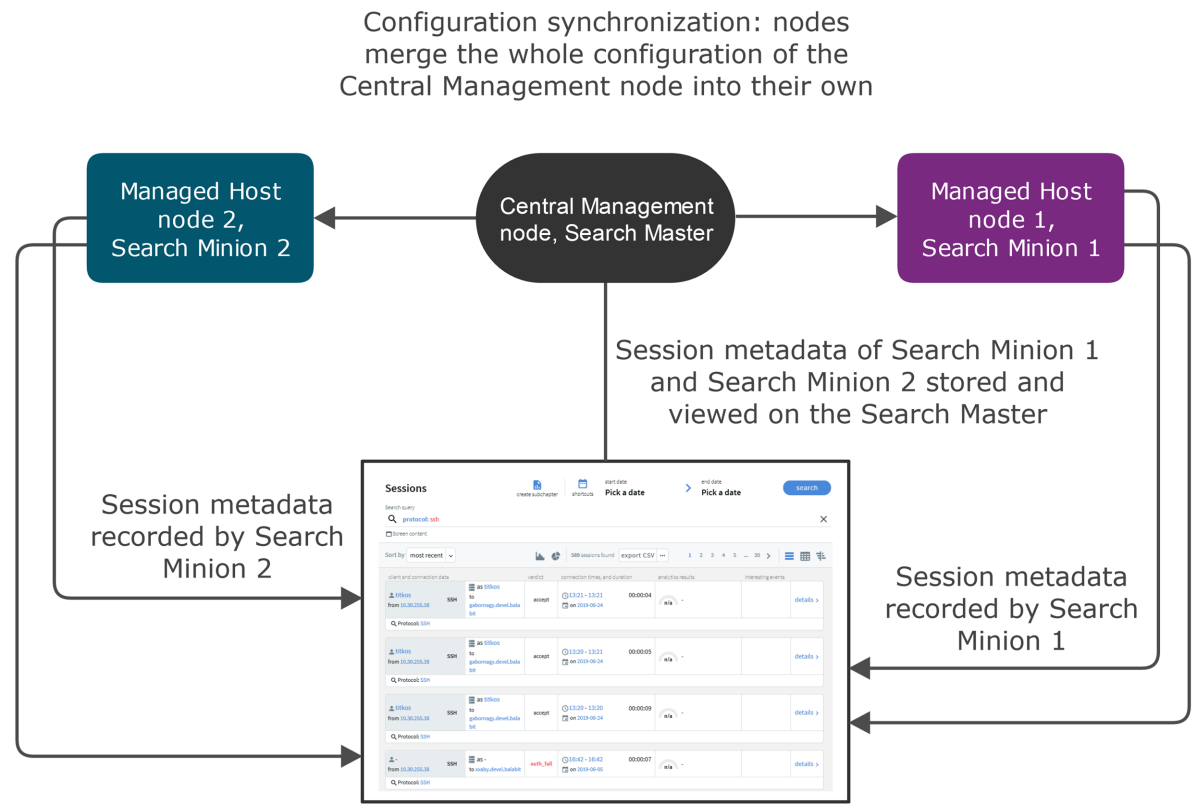
  > IMPORTANT: One Identity does not recommend having a central search configuration without configuration synchronization.

  It allows you to use a Central Management node with a Search Master role to view session data recorded by the minion nodes of your cluster, as well as manage all the nodes in the cluster from one central location.

  For more information, see Central search with configuration synchronization.

The following figure shows a cluster configured for central search with configuration synchronization.

Figure 101: Central search with configuration synchronization



The figure above is an example of an SPS cluster configured as follows:

- There is a Central Management node, which has a Search Master role.
- There are two Managed Host nodes (Managed Host node 1 and 2), each configured with a Search Minion role.
- The Central Management node is connected to the two minion nodes.
- The minion nodes record sessions, which are displayed on the Search interface of the Central Management node.
- The minion nodes fetch their configuration from the Central Management node, and merge it into their own configuration.

The Central Management node with a Search Master role and the connected Managed Host nodes with Search Minion roles require different configuration settings as described in the table below:

Table 7: Managing a central search configuration

Role	Use and configuration settings
Central Management node, Search Master	• Use it for viewing session data recorded by minions as well as managing all the nodes in the cluster.

Role	Use and configuration settings
	• Perform your configuration settings on this node. Managed Host nodes periodically fetch and merge these configuration settings into their own (configuration synchronization). • For backup and archive, configure a backup server on your Central Management node and an archive server on your minion node. • Ensure that you configure high availability (HA) for each node (for both your Central Management node and the Managed Host nodes). Also ensure that the Central Management node has a system backup configured. • This node cannot be used to record sessions.
Managed Host node, Search Minion	• Use it to record sessions and store audit trail files. • Do not perform configuration settings on the minion. These are overwritten during configuration synchronization. NOTE: All configuration settings that you make on the minions are overwritten during configuration synchronization except the node specific configuration. • Set external and internal indexers. • For backup and archive, configure an archive server on your minion node, and a backup server on your Central Management node. • Ensure that you configure high availability (HA) for each node (for both your Central Management node and the Managed Host nodes). Also ensure that the Central Management node has a system backup configured.

For more information on each role, see Cluster roles.

Managing a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster

High availability (HA) clusters can stretch across long distances, such as nodes across buildings, cities or even continents. The goal of HA clusters is to support enterprise business continuity by providing location-independent failover and recovery.

To set up a high availability cluster, connect two One Identity Safeguard for Privileged Sessions (SPS) units with identical configurations in high availability mode. This creates a primary-secondary (that is, active-backup) node pair. Should the primary node stop

functioning, the secondary node takes over the IP addresses of the primary node's interfaces. Gratuitous ARP requests are sent to inform hosts on the local network that the MAC addresses behind these IP addresses have changed.

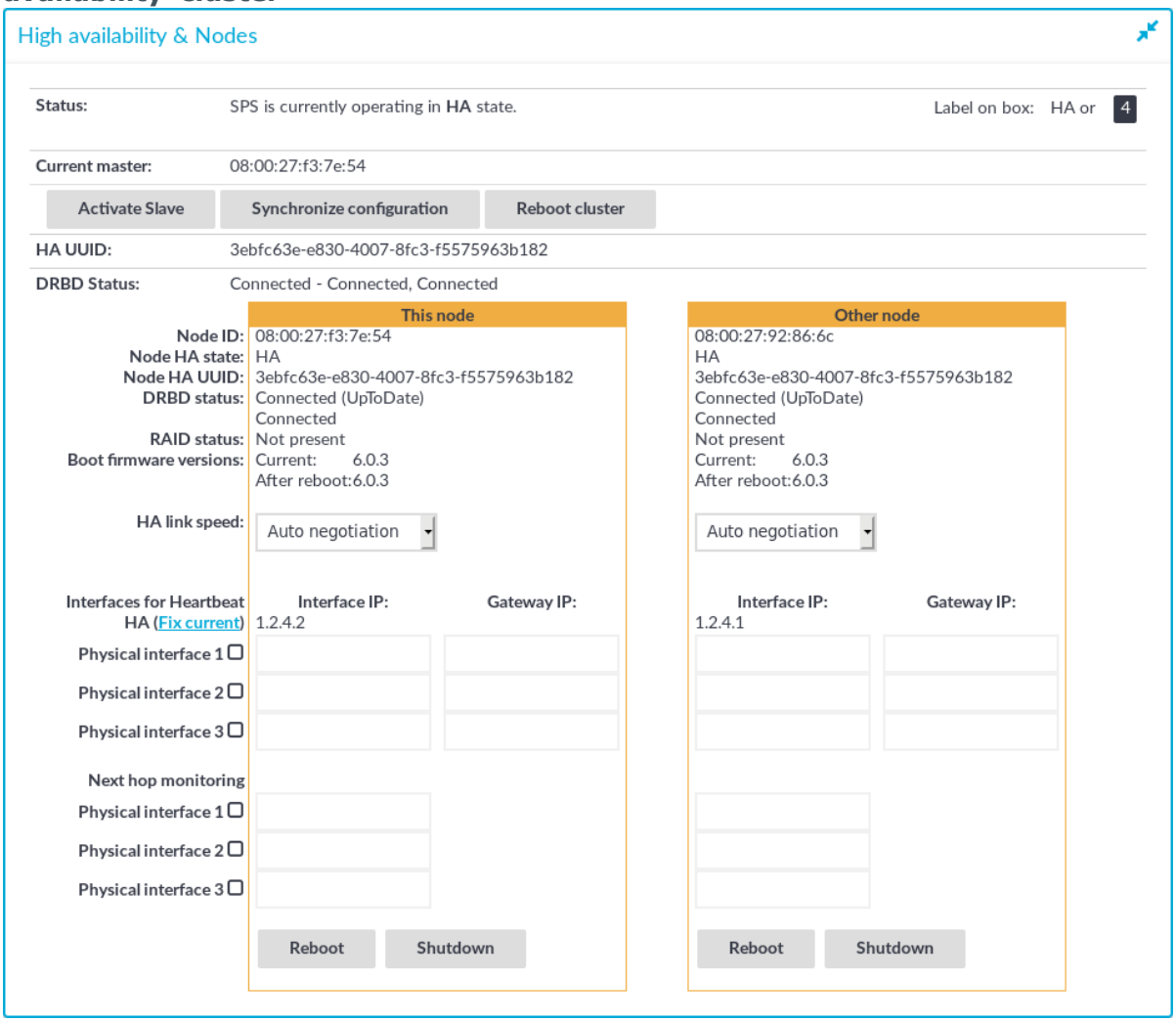
The primary node shares all data with the secondary node using the HA network interface (labeled as `4` or `HA` on the SPS appliance). The disks of the primary and the secondary node must be synchronized for the HA support to operate correctly. Interrupting the connection between running nodes (unplugging the Ethernet cables, rebooting a switch or a router between the nodes, or disabling the HA interface) disables data synchronization and forces the secondary node to become active. This might result in data loss. You can find instructions to resolve such problems and recover a SPS cluster in Troubleshooting a One Identity Safeguard for Privileged Sessions (SPS) cluster on page 873.

NOTE:

HA functionality was designed for physical SPS units. If SPS is used in a virtual environment, use the fallback functionalities provided by the virtualization service instead.

The **Basic Settings > High Availability** page provides information about the status of the HA cluster and its nodes.

Figure 102: Basic Settings > High Availability — Managing a high availability cluster



The following information is available about the cluster:

- **Status**: Indicates whether the SPS nodes recognize each other properly and whether those are configured to operate in high availability mode. For details, see Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses on page 873.
- **Current master**: The MAC address of the high availability interface (4 or HA) of the primary node. This address is also printed on a label on the top cover of the SPS unit.
- **HA UUID**: A unique identifier of the HA cluster. Only available in High Availability mode.
- **DRBD status**: Indicates whether the SPS nodes recognize each other properly and whether those are configured to operate in high availability mode. For details, see Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses on page 873.

- **DRBD sync rate limit**: The maximum allowed synchronization speed between the primary and the secondary node. For details, see Adjusting the synchronization speed on page 373.

The active (that is, primary) SPS node is labeled as **This node**. This unit inspects the SSH traffic and provides the web interface. The SPS unit labeled as **Other node** is the secondary node that is activated if the primary node becomes unavailable.

The following information is available about each node:

- **Node ID**: The MAC address of the `HA` interface of the node. This address is also printed on a label on the top cover of the SPS unit.

  For SPS clusters, the IDs of both nodes are included in the internal log messages of SPS. Note that if the central log server is a syslog-ng server, the `keep-hostname` option should be enabled on the syslog-ng server.
- **Node HA state**: Indicates whether the SPS nodes recognize each other properly and whether those are configured to operate in high availability mode. For details, see Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses on page 873.
- **Node HA UUID**: A unique identifier of the cluster. Only available in High Availability mode.
- **DRBD status**: The status of data synchronization between the nodes. For details, see Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses on page 873.
- **RAID status**: The status of the RAID device of the node. If it is not **Optimal**, there is a problem with the RAID device. For details, see Understanding One Identity Safeguard for Privileged Sessions (SPS) RAID status on page 882.
- **Boot firmware version**: Version number of the boot firmware.

  The boot firmware boots up SPS, provides high availability support, and starts the core firmware. The core firmware, in turn, handles everything else: provides the web interface, manages the connections, and so on.
- **HA link speed**: The maximum allowed speed between the primary node and the secondary node. The HA link's speed must exceed the **DRBD sync rate limit**, else the web UI might become unresponsive and data loss can occur.
- **Interfaces for Heartbeat**: Virtual interface used only to detect that the other node is still available. This interface is not used to synchronize data between the nodes (only heartbeat messages are transferred).

  You can find more information about configuring redundant heartbeat interfaces in Redundant heartbeat interfaces on page 374.
- **Next hop monitoring**: IP addresses (usually next hop routers) to continuously monitor both the primary node and the secondary node by using ICMP echo (ping) messages. If any of the monitored addresses becomes unreachable from the primary node while being reachable from the secondary node (in other words, more monitored addresses are accessible from the secondary node), then it is assumed that the primary node is unreachable and a forced takeover occurs – even if the

primary node is otherwise functional. For details, see Next-hop router monitoring on page 376.

HA cluster configuration and management options

This section is about the available configuration and management options for HA clusters.

Setting up a high availability cluster

For detailed instructions about setting up a HA cluster, see "Installing two SPS units in HA mode" in the Installation Guide.

Adjust the DRBD (primary-secondary) synchronization speed

You can change the limit of the DRBD synchronization rate. Note that this does not change the speed of normal data replication. For details, see Adjusting the synchronization speed on page 373.

Configure redundant heartbeat interfaces

You can configure virtual interfaces for each HA node to monitor the availability of the other node. For details, see Redundant heartbeat interfaces on page 374.

Configure next-hop monitoring

You can provide IP addresses (usually next hop routers) to continuously monitor both the primary node and the secondary node by using ICMP echo (ping) messages. If any of the monitored addresses becomes unreachable from the primary node while being reachable from the secondary node (in other words, more monitored addresses are accessible from the secondary node), then it is assumed that the primary node is unreachable and a forced takeover occurs – even if the primary node is otherwise functional. For details, see Next-hop router monitoring on page 376.

Reboot the HA cluster

To reboot both nodes, click **Reboot Cluster**. To prevent takeover, a token is placed on the secondary node. While this token persists, the secondary node halts its boot process to make sure that the primary node boots first. Following reboot, the primary node removes this token from the secondary node, allowing it to continue with the boot process.

If the token still persists on the secondary node following reboot, the **Unblock Slave Node** button is displayed. Clicking the button removes the token, and reboots the secondary node.

Reboot a node

This option reboots the selected node.

When rebooting the nodes of a cluster, reboot the other node (that is, the secondary node) first to avoid unnecessary takeovers.

Shutdown a node

This option forces the selected node to shut down.

When shutting down the nodes of a cluster, shut down the other node (that is, the secondary node) first. When powering on the nodes, start the primary node first to avoid unnecessary takeovers.

Manual takeover

To activate the other node (that is, the secondary node) and disable the currently active node, click **Activate slave**.

Activating the secondary node terminates all connections of One Identity Safeguard for Privileged Sessions (SPS) and might result in data loss. The secondary node becomes active after about 60 seconds, during which the protected servers cannot be accessed.

Adjusting the synchronization speed

One Identity Safeguard for Privileged Sessions (SPS) synchronizes the content of the hard disk of the primary node (previously also referred to as master node) and the secondary node (previously also referred to as slave node) in the following cases.

- When you configure two SPS units to operate in High Availability mode (converting a single node to a high availability cluster),
- when you replace a node from a cluster, or
- when recovering from a split-brain situation.
- Normal data replication (copying incoming data, for example, audit trails from the primary node to the secondary node is not synchronization.

Since this synchronization can take up significant system-resources, the maximal speed of the synchronization is limited, by default, to `10 Mbps`. However, this means that synchronizing large amount of data can take very long time, so it is useful to increase the synchronization speed in certain situations —.

To change the limit of the DRBD synchronization rate, navigate to **Basic Settings > High Availability > DRBD sync rate limit**, and select the desired value. Note the following points before changing the **DRBD sync rate limit** option.

- The **Basic Settings > High Availability > DRBD sync rate limit** option is visible only when synchronization is in progress, or when you have clicked **Convert to Cluster** but have not rebooted the cluster yet.
- Changing this option does not change the limit of the data replication speed.
- Set the sync rate carefully. A high value is not recommended if the load of SPS is very high, as increasing the resources used by the synchronization process may degrade the general performance of SPS. On the other hand, the HA link's speed must exceed

the speed of the incoming data, else the web UI might become unresponsive and data loss can occur.

The **Basic Settings > High Availability > DRBD status** field indicates whether the latest data (including SPS configuration, audit trails, log files, and so on) is available on both SPS nodes. For a description of each possible status, see Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses on page 873.

Redundant heartbeat interfaces

To avoid unnecessary takeovers and to minimize the chance of split-brain situations, you can configure additional heartbeat interfaces in One Identity Safeguard for Privileged Sessions (SPS). These interfaces are used only to detect that the other node is still available, they are not used to synchronize data between the nodes (only heartbeat messages are transferred). For example, if the main HA interface breaks down, or is accidentally unplugged and the nodes can still access each other on the redundant HA interface, no takeover occurs, but no data is synchronized to the secondary node until the main HA link is restored. Similarly, if connection on the redundant heartbeat interface is lost, but the main HA connection is available, no takeover occurs.

If a redundant heartbeat interface is configured, its status is displayed in the **Basic Settings > High Availability > Redundant Heartbeat status** field, and also in the **HA > Redundant** field of the System monitor. For a description of each possible status, see Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses on page 873.

The redundant heartbeat interface is a virtual interface with a virtual MAC address that uses an existing interface of SPS. The MAC address of the virtual redundant heartbeat interface is displayed as **HA MAC**. The MAC address of the redundant heartbeat interface is generated in a way that it cannot interfere with the MAC addresses of physical interfaces. Similarly, the HA traffic on the redundant heartbeat interface cannot interfere with any other traffic on the interface used.

If the nodes lose connection on the main HA interface, and after a time the connection is lost on the redundant heartbeat interfaces as well, the secondary node becomes active. However, as the primary node was active for a time when no data synchronization was possible between the nodes, this results in a split-brain situation, which must be resolved before the HA functionality can be restored. For details, see Recovering from a split brain situation on page 876.

NOTE:

Even if redundant HA links are configured, if the dedicated HA link fails, the secondary node will not be visible on the High Availability page anymore.

SPS nodes use UDP port 694 to send each other heartbeat signals.

To configure a redundant heartbeat interface

1. Navigate to **Basic Settings > High Availability > Interfaces for Heartbeat**.
2. Select the interface you want to use as redundant heartbeat interface (for example `Physical interface 1`). Using an interface as a redundant heartbeat interface does not affect the original traffic of the interface.

   Figure 103: Basic Settings > High Availability — Configuring redundant heartbeat interfaces

3. Enter an IP address into the **This node > Interface IP** field of the selected interface. Note the following:
   - The two nodes must have different **Interface IP**.
   - If you do not use next hop monitoring on the redundant interface, you can use any **Interface IP** (even if otherwise it does not exist on that network).
   - If you use next hop monitoring on the redundant interface, the **Interface IP** address must be a real IP address that is visible from the other node.
   - If you use next hop monitoring on the redundant interface, the **Interface IP** must be accessible from the next-hop address, and vice-versa. For details on next hop monitoring, see Next-hop router monitoring on page 376.

   Use an IPv4 address.
4. If the two nodes are in a different subnetwork, enter the IP address of the local gateway into the **This node > Gateway IP** field. The **Interface IP** address of the node must be accessible from the **Gateway IP** address.

   Use an IPv4 address.
5. Enter an IP address into the **Other node > Interface IP** field of the selected interface. Note the following:
   - The two nodes must have different **Interface IP**.
   - If you do not use next hop monitoring on the redundant interface, you can use any **Interface IP** (even if otherwise it does not exist on that network).
   - If you use next hop monitoring on the redundant interface, the **Interface IP** address must be a real IP address that is visible from the other node.
   - If you use next hop monitoring on the redundant interface, the **Interface IP** must be accessible from the next-hop address, and vice-versa. For details on next hop monitoring, see Next-hop router monitoring on page 376.

   Use an IPv4 address.

6. If the two nodes are in a different subnetwork, enter the IP address of the local gateway into the **Other node > Gateway IP** field. The **Interface IP** address of the node must be accessible from the **Gateway IP** address.

   Use an IPv4 address.

7. Repeat the previous steps to add additional redundant heartbeat interfaces if needed.
8. Click 

.
9. Restart the nodes for the changes to take effect: click **Reboot Cluster**.

Next-hop router monitoring

By default, HA takeover occurs only if the primary node stops working or becomes unreachable from the secondary node. However, this does not cover the scenario when the primary node becomes unaccessible to the outside world while the secondary node would be still accessible (for example because it is connected to a different router).

To address such situations, you can specify IP addresses (usually next hop routers) to continuously monitor both the primary node and the secondary node by using ICMP echo (ping) messages. One such address can be set up for every interface.

When setting up next hop monitoring, you have to make sure that the primary and secondary nodes can ping the specified address directly. You can either:

- Choose the addresses of the redundant-HA One Identity Safeguard for Privileged Sessions (SPS) interfaces so that they are on the same subnet as the next-hop address
- Configure the next-hop device with an additional IP-address that is on the same subnet as the redundant-HA SPS interfaces facing it

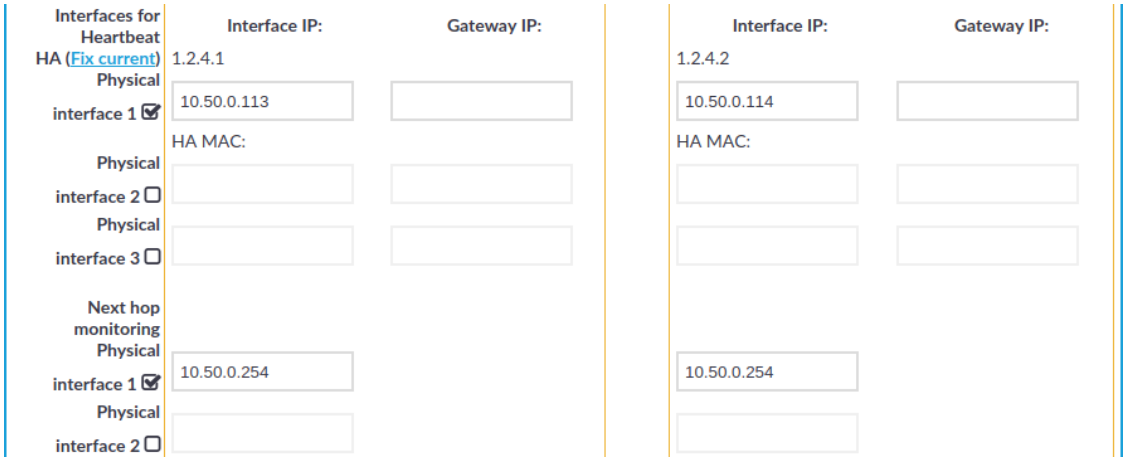
If any of the monitored addresses becomes unreachable from the primary node while being reachable from the secondary node (in other words, more monitored addresses are accessible from the secondary node) then it is assumed that the primary node is unreachable and a forced takeover occurs — even if the primary node is otherwise functional.

Naturally, if the secondary node is not capable of taking over the primary node (for example, because there is data not yet synchronized from the current primary node), no takeover is performed.

To configure next hop monitoring

1. Navigate to **Basic Settings > High Availability > Next hop monitoring**.
2. Select the interface to use for monitoring its next-hop router.

Figure 104: Basic Settings > High Availability — Configuring next hop monitoring



3. Enter the IP address to monitor from the current primary node (for example, the IP address of the router or the switch connected to the interface) into the **This node > Next hop IP** field of the selected interface. This IP address must be a real IP address that is visible from the interface, and must be on the same local network segment.

   Use an IPv4 address.

4. Enter the IP address to monitor from the current secondary node (for example the IP address of the router or the switch connected to the interface) into the **Other node > Next hop IP** field of the selected interface. This IP address must be a real IP address that is visible from the interface, and must be on the same local network segment.

   Use an IPv4 address.

5. Repeat the previous steps to add IP addresses to be monitored from the other interfaces if needed.

6. Click Commit.

   CAUTION:

   For the changes to take effect, you have to restart both nodes. To restart both nodes, click Reboot Cluster.

Upgrading One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) appliances are preinstalled with a Long Term Support (LTS) release. One Identity recommends that you upgrade to the latest LTS maintenance release as soon as possible. Each LTS release is supported for 3 years

after original publication date, and for 1 year after the succeeding LTS release is published (whichever date is later). You are encouraged to upgrade to succeeding LTS releases.

Feature Releases provide additional features which are not yet consolidated to an LTS release. To gain access to these features, you may install a supported Feature Release on the appliance, with the following conditions:

- You cannot roll back to an LTS release from a Feature Release.
- Feature Releases are released and supported in a timeline of 6 months. You have to keep upgrading SPS to the latest Feature Release to ensure that your appliance is supported.

For both LTS and Feature Releases, One Identity regularly incorporates security patches and bugfixes, and issues updated Revisions of the released product. One Identity strongly recommends always installing the latest Revision of the used software Release.

CAUTION:

Downgrading from the latest Feature Release, even to an LTS release, voids support for SPS.

The following sections describe how to keep SPS up to date, and how to install a new license:

- Prerequisites: Upgrade checklist on page 378.
- Upgrading a single node: Upgrading One Identity Safeguard for Privileged Sessions (SPS) (single node) on page 380. To upgrade SPS without using the web interface, see Firmware update using SSH on page 395.
- Upgrading a high availability cluster: Upgrading a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster on page 381.
- Troubleshooting: Troubleshooting on page 384.
- Renewing the SPS license: Updating the SPS license on page 388.
- Exporting the configuration of SPS: Exporting the configuration of One Identity Safeguard for Privileged Sessions (SPS) on page 384.
- Importing the configuration of SPS: Importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) on page 386.

Upgrade checklist

The following list applies to all configurations:

- You have created a configuration backup of One Identity Safeguard for Privileged Sessions (SPS).

  For detailed instructions, refer to Exporting the configuration of One Identity Safeguard for Privileged Sessions (SPS) on page 384.
- You have a valid support portal account.

To download the required firmware file and license, you need a valid support portal account. Note that registration is not automatic, and might require up to two working days to process.

- You have downloaded the latest SPS firmware from the Downloads page.
- You have read the Release Notes of the firmware before updating. The Release Notes might include additional instructions specific to the firmware version.

  The Release Notes are available at the Downloads page.
- You have verified that SPS is in good condition (no issues are displayed on the System Monitor).
- Optional: You have exported core dump files, if necessary for debugging, from **Basic Settings > Troubleshooting > Core files**. These files are removed during upgrade.

If you have a high availability cluster:

- You have IPMI access to the secondary node. You can find detailed information on using the IPMI in the following documents:

  For Safeguard Sessions Appliance 3000 and 3500, see the X9 SMT IPMI User's Guide.
- You have verified on the **Basic Settings > High Availability** page that the HA status is not degraded.

If you are upgrading SPS in a virtual environment:

- You have created a snapshot of the virtual machine before starting the upgrade process.
- You have configured and enabled console redirection (if the virtual environment allows it).

During the upgrade, SPS displays information about the progress of the upgrade and any possible problems in the following places:

- On the web interface of SPS, at any of the **Listening addresses** configured at **Basic settings > Local Services > Web login (admin and user)**. (After booting, you are directed to the login screen of SPS.)

  > NOTE:
  >
  > If you are upgrading to version 6.0.13 from version 5.0.x, this feature is enabled after the first boot to version 6.0.13. So during the upgrade to version 6.0.13, you will not be able to see any upgrade logs on the web interface.
- On the console, which you can monitor with IPMI (ILOM) or console access.

The information displayed in the browser and on the console is the same.

One Identity strongly recommends that you test the upgrade process in a non-production (virtual, and so on) environment first.

Upgrading SPS requires a reboot. We strongly suggest that you perform the upgrade on the production appliance during maintenance hours only, to avoid any potential data loss.

Upgrading One Identity Safeguard for Privileged Sessions (SPS) (single node)

The following describes how to upgrade One Identity Safeguard for Privileged Sessions (SPS) to a newer firmware version. To upgrade SPS without using the web interface, see Firmware update using SSH on page 395. One Identity recommends that you always use the latest maintenance release available.

CAUTION:

When upgrading to a new major release (that is, to a new Feature Release or a new Long-Term Supported release), always follow the instructions of the *How to upgrade to One Identity Safeguard for Privileged Sessions* guide for that release, as it contains more detailed instructions (available at the Safeguard for Privileged Sessions Documentation page).

CAUTION:

Physical SPS appliances based on Pyramid hardware are not supported in 5 F1 and later releases. Do not upgrade to 5 F1 or later on a Pyramid-based hardware. The last supported release for this hardware is 5 LTS, which is a long-term supported release.

If you have purchased SPS before August, 2014 and have not received a replacement hardware since then, you have Pyramid hardware, so do not upgrade to SPS 5 F1 or later. If you have purchased SPS after August 2014, you can upgrade to 5 F1.

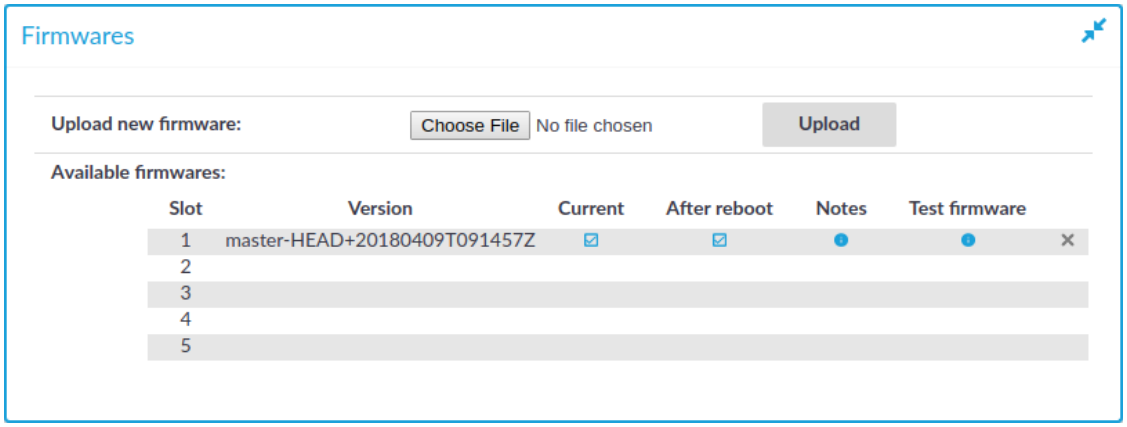
If you do not know the type of your hardware or when it was purchased, complete the following steps:

1. **Login to SPS.**
2. **Navigate to Basic Settings > Troubleshooting > Create support bundle, click Create support bundle, and save the file.**
3. **Open a ticket at https://support.oneidentity.com/create-service-request/.**
4. **Upload the file you downloaded from SPS in Step 1.**
5. **We will check the type of your hardware and notify you.**

To upgrade SPS to a newer firmware version

1. Navigate to **Basic Settings > System > Firmwares**.

   Figure 105: Basic Settings > System > Firmwares — Managing the firmwares

2. Upload the new firmware: **Browse** for the firmware `.iso` file and then click **Upload**.
3. To read the Upgrade Notes of the uploaded firmware, click on the icon. The Upgrade Notes are displayed in a pop-up window.
4. Click **Test** for the new firmware to check if your configuration can be upgraded to version 6.0.13. If the test returns any errors, correct them before continuing the upgrade process. If you encounter any problems, contact our Support Team.
5. **CAUTION:**

   Proceed only if the upgrade test is successful.

   Activate the firmware, but do not reboot SPS yet.
6. Navigate to **Basic Settings > System > Traffic Control > This node**, and choose **Reboot**.

   SPS attempts to boot with the new firmware. Wait for the process to complete.
7. Login to the SPS web interface to verify that the upgrade was successful.

   Navigate to **Basic Settings > System > Version details**, or check the system log for the version numbers SPS reports on boot. In case you encounter problems, you can find common troubleshooting steps in Troubleshooting on page 384.

Upgrading a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster

The following describes how to upgrade a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster to a newer firmware version. One Identity recommends that you always use the latest maintenance release available.

CAUTION:

If you have nodes with the Search Minion role configured, see Upgrading a high availability One Identity Safeguard for Privileged Sessions (SPS) cluster to avoid a critical error.

CAUTION:

When upgrading to a new major release (that is, to a new Feature Release or a new Long-Term Supported release), always follow the instructions of the *How to upgrade to One Identity Safeguard for Privileged Sessions* guide for that release, as it contains more detailed instructions (available at the Safeguard for Privileged Sessions Documentation page).

CAUTION:

Physical SPS appliances based on Pyramid hardware are not supported in 5 F1 and later releases. Do not upgrade to 5 F1 or later on a Pyramid-based hardware. The last supported release for this hardware is 5 LTS, which is a long-term supported release.

If you have purchased SPS before August, 2014 and have not received a replacement hardware since then, you have Pyramid hardware, so do not upgrade to SPS 5 F1 or later. If you have purchased SPS after August 2014, you can upgrade to 5 F1.

If you do not know the type of your hardware or when it was purchased, complete the following steps:

1. **Login to SPS.**
2. **Navigate to Basic Settings > Troubleshooting > Create support bundle, click Create support bundle, and save the file.**
3. **Open a ticket at https://support.oneidentity.com/create-service-request/.**
4. **Upload the file you downloaded from SPS in Step 1.**
5. **We will check the type of your hardware and notify you.**

To upgrade a high availability SPS cluster to a newer firmware version

1. Navigate to **Basic Settings > System > Firmwares**.
2. Upload the new firmware: **Browse** for the firmware `.iso` file and then click **Upload**.
3. To read the Upgrade Notes of the uploaded firmware, click on the icon. The Upgrade Notes are displayed in a pop-up window.
4. Click **Test** for the new firmware to check if your configuration can be upgraded to version 6.0.13. If the test returns any errors, correct them before continuing the

upgrade process. If you encounter any problems, contact our Support Team.

5. **CAUTION:**

   Proceed only if the upgrade test is successful.

   Activate the firmware, but do not reboot SPS yet.

6. Navigate to **Basic Settings > High availability**, and verify that the new firmware is active on the secondary node. This might take a few minutes.
7. In **Basic Settings > High availability > Other node**, click **Shutdown**.
8. Restart the primary node: click **This node > Reboot**.

   SPS attempts to boot with the new firmware. Wait for the process to complete.
9. Login to the SPS web interface to verify that the primary node upgrade was successful.

   Navigate to **Basic Settings > System > Version details**, or check the system log for the version numbers SPS reports on boot. In case you encounter problems, you can find common troubleshooting steps in Troubleshooting on page 384.
10. Use the IPMI to power on the secondary node.

    The secondary node attempts to boot with the new firmware, and reconnects to the primary node to sync data. During the sync process, certain services (including Heartbeat) are not available. Wait for the process to finish, and the secondary node to boot fully.
11. Navigate to **Basic Settings > High availability** and verify that the secondary node is connected, and has the same firmware versions as the primary node.

CAUTION:

If you have nodes with the Search Minion role configured, the Search Minion nodes must be upgraded first during high availability cluster upgrade. If you upgrade the Search Master node first, it is possible that a Search Minion node will create a legacy Elasticsearch index before the init script on the Search Master creates a new one. In this case, the Elasticsearch index will contain invalid schema mapping data, therefore as the high availability cluster's schema changes, the Search Minion nodes cannot push their documents into Elasticsearch, resulting in a critical error.

To avoid the critical error mentioned above, follow the method below to upgrade a high availability SPS cluster with Search Minion nodes to a newer firmware version

1. Set the Search Master node for upgrade so that it uses the newer firmware version for reboot. To do this, complete steps 1-11 of To upgrade a high availability SPS cluster to a newer firmware version.
2. Click **Shutdown**.
3. Upgrade your Search Minion nodes one after the other, using the method in steps 1-2 above.

4. Reboot the Search Master node, which will now boot with the newer firmware version.

With this method you detach the Search Minion nodes from the Search Master node and upgrade them separately before any other nodes. As a result, the whole high availability cluster will use the newer firmware version after reboot.

If you have accidentally upgraded the Search Master node first and encounter this critical error, contact our Support Team.

Troubleshooting

If you experience any strange behavior of the web interface, first try to reload the page by holding the SHIFT key while clicking the Reload button of your browser to remove any cached version of the page.

In the unlikely case that One Identity Safeguard for Privileged Sessions (SPS) encounters a problem during the upgrade process and cannot revert to its original state, SPS performs the following actions:

- Initializes the network interfaces using the already configured IP addresses.
- Enables SSH-access to SPS, unless SPS is running in sealed mode. That way it is possible to access the logs of the upgrade process that helps the One Identity Support Team to diagnose and solve the problem. Note that SSH access will be enabled on every active interface, even if management access has not been enabled for the interface.

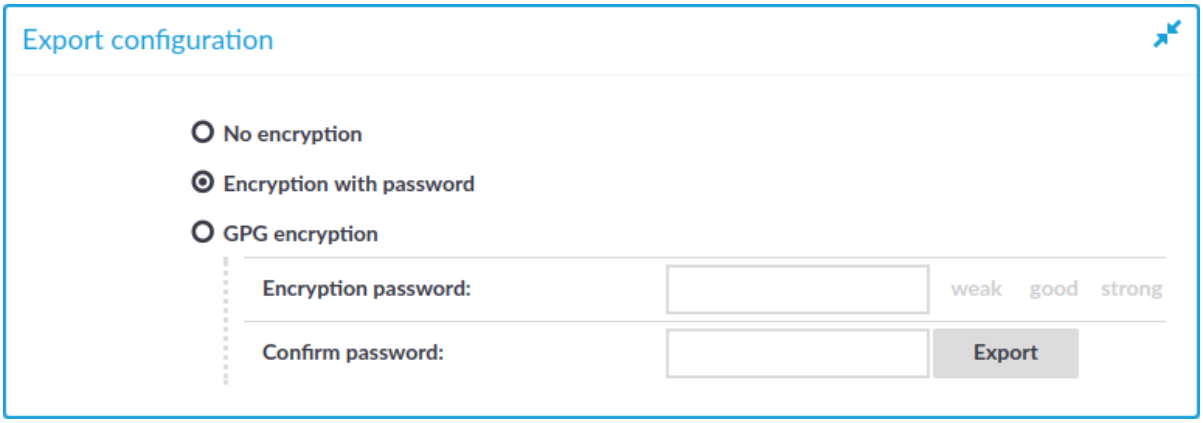
In case the web interface is not available within 30 minutes of rebooting SPS, check the information displayed on the local console and contact our Support Team.

Exporting the configuration of One Identity Safeguard for Privileged Sessions (SPS)

The configuration of One Identity Safeguard for Privileged Sessions (SPS) can be exported (for manual archiving, or to migrate it to another SPS unit) from the **Basic Settings > System** page. Use the respective action buttons to perform the desired operation.

You also have the option to export the configuration SPS into a local file using the console. For details, see Exporting and importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) using the console on page 396.

Figure 106: Basic Settings > System — Exporting the SPS configuration



To export the configuration of SPS

1. Navigate to **Basic Settings > System > Export configuration**.
2. Select how to encrypt the configuration:
   - To export the configuration file without encryption, select **No encryption**.

     CAUTION:

     One Identity does not recommend exporting the SPS configuration without encryption, as it contains sensitive information such as password hashes and private keys.
   - To encrypt the configuration file with a simple password, select **Encrypt with password** and enter the password into the **Encryption password** and **Confirm password** fields.

     NOTE:

     One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`
   - To encrypt the configuration file with GPG, select **GPG encryption**. Note that this option uses the same GPG key that is used to encrypt automatic system backups, and is only available if you have uploaded the public part of a GPG key to SPS at **Basic Settings > Management > System backup**. For details, see Encrypting configuration backups with GPG.
3. Click **Export**.

   NOTE:

   The exported file is a gzip-compressed archive. On Windows platforms, it can be decompressed with common archive managers such as the free 7-Zip tool.

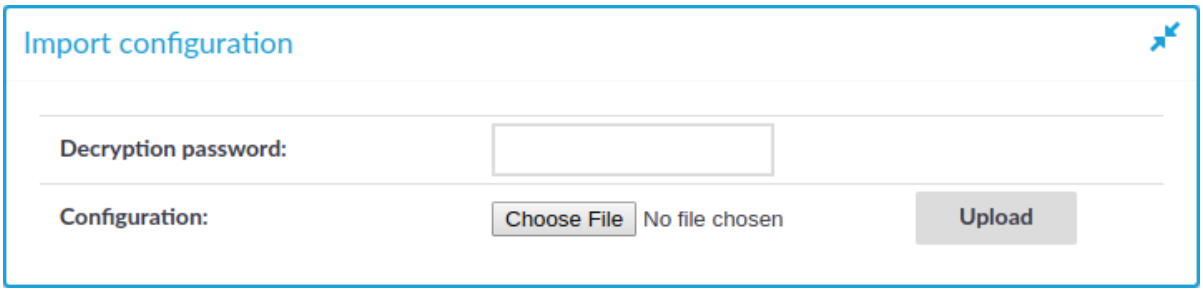
   The name of the exported file is `<hostname_of_SPS>-YYYYMMDDTHHMM.config`, the `-encrypted` or `-gpg` suffix is added for password-encrypted and GPG-encrypted files, respectively.

Importing the configuration of One Identity Safeguard for Privileged Sessions (SPS)

The configuration of One Identity Safeguard for Privileged Sessions (SPS) can be imported from the **Basic Settings > System** page. Use the respective action buttons to perform the desired operation.

You also have the option to import configuration of SPS from a local file using the console. For details, see Exporting and importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) using the console on page 396.

Figure 107: Basic Settings > System — Importing the SPS configuration



CAUTION:

It is not possible to import the configuration of an older major release (for example, 1.0) into a newer release (for example, 2.0).

To import the configuration of SPS

1. **CAUTION:**

   Do not export or import configuration between a physical SPS deployment and a virtual one. Because of the differences and limitations between physical and virtual appliances, configure the virtual appliance from scratch to ensure proper functionality. When you migrate a virtual SPS to another one, you can export and import the configuration.

   Navigate to **Basic Settings > System > Import configuration**.
2. Click **Browse** and select the configuration file to import.
3. Enter the password into the **Encryption password** field and click **Upload**.

   NOTE:

   One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

Managing the One Identity Safeguard for Privileged Sessions (SPS) license

Information of the current license of One Identity Safeguard for Privileged Sessions (SPS) is displayed on the **Basic Settings > System > License** page. The following information is displayed:

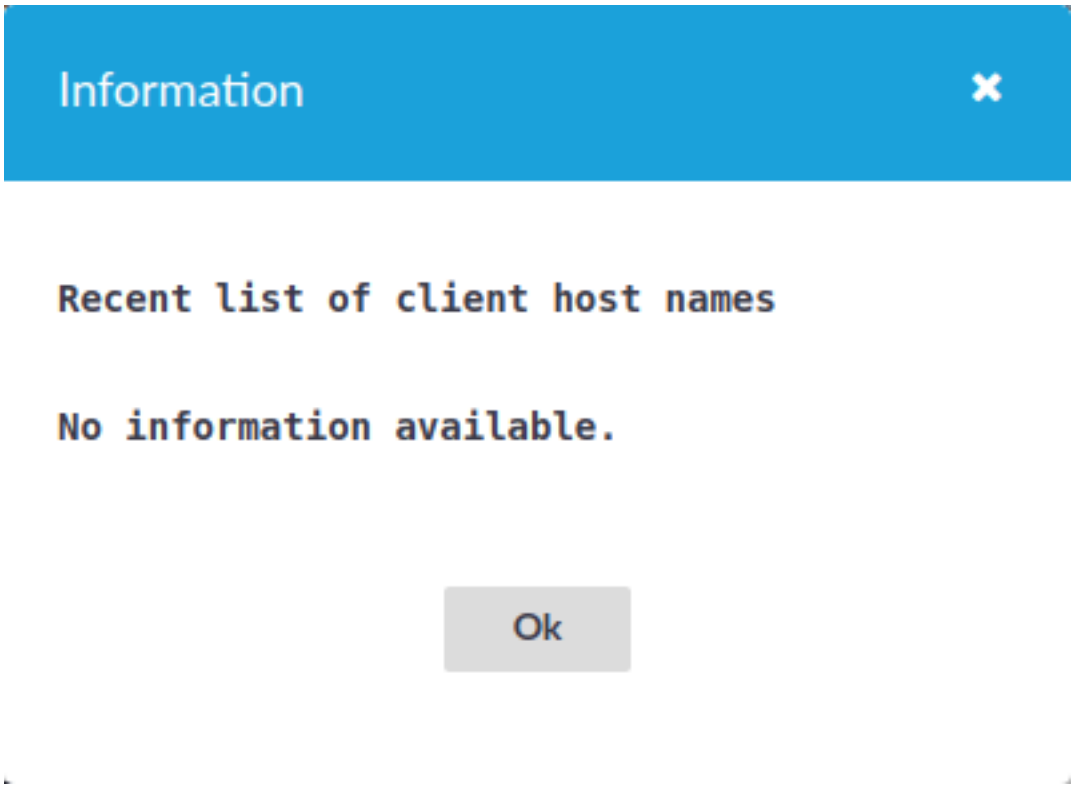
Figure 108: Basic Settings > System > License — Updating the license



- **Customer**: The company permitted to use the license (for example `Example Ltd.`).
- **Serial**: The unique serial number of the license.
- **Limit type**:
  - **Host**: Limits the number of servers (individual IP addresses) that can be connected through SPS.
  - **Session**: Limits the number of concurrent sessions (parallel connections) that

can pass through SPS at a time (for example 25). SPS will reject additional connection requests until an already established connection is closed.

- **Limit**: The actual value of the session or host limit. To list which hosts SPS counts against this limit, click the on the value of the limit.



- **Valid**: The period in which the license is valid. The dates are displayed in *YYYY/MM/DD* format.

The full text of the End User License Agreement is also displayed here.

SPS starts sending automatic alerts daily, 60 days before the license expires. An alert is sent also when the number of protected servers exceeds 90% of the limit set in the license.

Updating the SPS license

The SPS license must be updated before the existing license expires or when you purchase a new license.

CAUTION:

Before uploading a new license, One Identity recommends that you backup the configuration of SPS. For details, see Exporting the configuration of One Identity Safeguard for Privileged Sessions (SPS) on page 384.

To update the license

1. Navigate to **Basic Settings > System > License**.
2. Click **Browse** and select the new license file.

   NOTE:

   It is not required to manually decompress the license file. Compressed licenses (for example `.zip` archives) can also be uploaded.

3. Click **Upload**, then 

.
4. **CAUTION:**

   This step terminates all controlled connections going through SPS. Disconnect your clients from the protected servers before proceeding.

   To activate the new license, navigate to **Traffic control > All services** and click **Restart**.

Accessing the One Identity Safeguard for Privileged Sessions (SPS) console

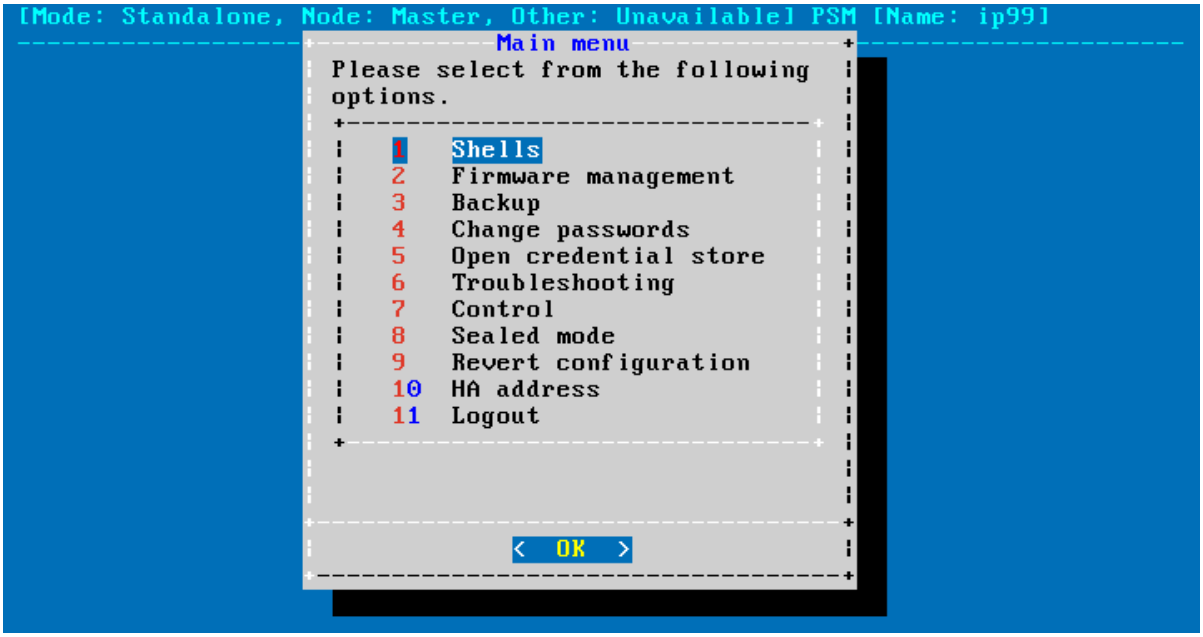
The following topics describe how to use the console menu of One Identity Safeguard for Privileged Sessions (SPS), how to enable remote SSH access to SPS, and how to change the root password from the web interface.

Using the console menu of One Identity Safeguard for Privileged Sessions (SPS)

Connecting to One Identity Safeguard for Privileged Sessions (SPS) locally or remotely using Secure Shell (SSH) allows you to access the console menu of SPS. The console menu provides access to the most basic configuration and management settings of SPS. It is mainly used for troubleshooting purposes, the primary interface of SPS is the web interface.

The console menu is accessible to the `root` user using the password set during completing the Welcome Wizard.

Figure 109: The console menu



The console menu provides allows you to perform the following actions.

Access the local core and boot shells.

This is usually not recommended and only required in certain troubleshooting situations. Select the boot/core shell's keyboard layout for the local console. This will not affect the keyboard layout if you have accessed the shell via SSH.

The boot firmware boots up SPS, provides high availability support, and starts the core firmware. The core firmware, in turn, handles everything else: provides the web interface, manages the connections, and so on.

Select the active firmware, and delete unneeded firmwares.

Accessing the firmware management is useful if after an update the new firmware does not operate properly and the web interface is not available to activate the previous firmware.

Start backup processes.

For more information about backup processes, see Data and configuration backups.

Change the passwords of the root and admin users.

For details, see Changing the root password of One Identity Safeguard for Privileged Sessions (SPS).

Access the network-troubleshooting functions and display the available log files.

If the web interface is inaccessible, it can be the result of an internal locking error. To resolve this issue, delete the lock files. After deletion, they are archived, and included in the support bundle if they are not older than 30 days. To create a support bundle, if the web interface is inaccessible, select **Create support bundle**.

NOTE:

If deleting the lock files did not resolve the issue, contact our Support Team.

Reboot and shutdown the system.

For details, see Controlling One Identity Safeguard for Privileged Sessions (SPS): reboot, shutdown.

Enable and disable sealed mode.

For details, see Sealed mode on page 397.

Set the IP address of the HA interface.

For more information about assigning an IP address to the HA interface of a node, see Resolving an IP conflict between cluster nodes.

NOTE:

Note that logging in to the console menu automatically locks the SPS interface, meaning that users cannot access the web interface while the console menu is used. The console menu can be accessed only if there are no users accessing the web interface. The connection of web-interface users can be terminated to force access to the console menu.

Enabling SSH access to the One Identity Safeguard for Privileged Sessions (SPS) host

Exclusively for troubleshooting purposes, you can access the One Identity Safeguard for Privileged Sessions (SPS) host using SSH.

Completing the Welcome Wizard automatically disables SSH access to SPS. Re-enabling it allows you to connect remotely to the SPS host and login using the `root` user. The password of the root user is the one you provided in the Welcome Wizard. For details on how to change the root password from the web interface, see Changing the root password of One Identity Safeguard for Privileged Sessions (SPS) on page 394.

CAUTION:

Accessing the One Identity Safeguard for Privileged Sessions (SPS) host directly using SSH is not recommended or supported, except for troubleshooting purposes. In such case, the One Identity Support Team will give you exact instructions on what to do to solve the problem.

For security reasons, disable SSH access to SPS when it is not needed. For details, see "Enabling SSH access to the One Identity Safeguard for Privileged Sessions (SPS) host" in the Administration Guide.

The following encryption algorithms are configured on the local SSH service of SPS:

- Key exchange (KEX) algorithms:

  ```
  diffie-hellman-group-exchange-sha256
  ```

- Ciphers:

  ```
  aes256-ctr,aes128-ctr
  ```

- Message authentication codes:

  ```
  hmac-sha2-512,hmac-sha2-256
  ```

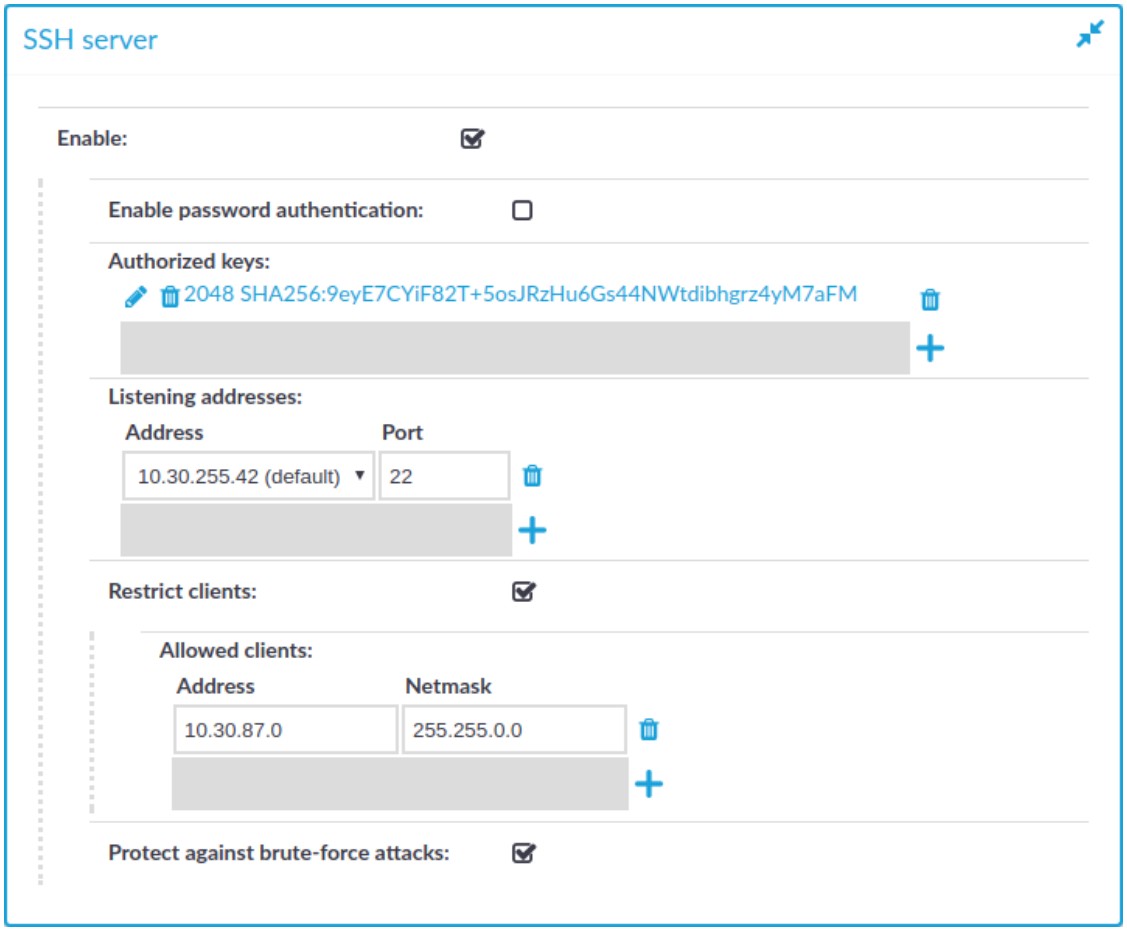
SSH access is, by default, protected against brute-force attacks: after 20 unsuccessful login attempts, the offending IP is blocked from accessing the SSH service for ten minutes.

You can turn off brute force protection by unselecting the **Protect against brute-force attacks** option for the SSH server.

To enable SSH access to the SPS host

1. Navigate to **Basic Settings > Local Services > SSH server**.

Figure 110: Basic Settings > Local Services > SSH server — Enabling remote SSH access to SPS



2. Select the **Enable** option.

   NOTE:

   Remote SSH access is automatically disabled if Sealed mode is enabled. For details, see Sealed mode on page 397.

3. Choose the authentication method for the remote SSH connections.
   - To enable password-based authentication, select the **Enable password authentication** option.
   - To enable public-key authentication, click + in the **Authorized keys** field, click and upload the public keys of the users who can access and manage SPS remotely via SSH.

     One Identity recommends using 2048-bit RSA keys (or stronger).

4. Choose an address and a port for the SSH server in the **Listening addresses** section.

   The available addresses correspond to the interface addresses configured in **Basic Settings > Network > Interfaces**. Only IPv4 addresses can be selected.

   To add multiple addresses, click +.

5. (Optional) To permit SSH acces only from selected subnets or IP addresses, select **Restrict clients**, click + and enter the IP address and netmask of the allowed clients.

   Use an IPv4 address.

   To add multiple addresses, click +.

6. Click Commit.

Changing the root password of One Identity Safeguard for Privileged Sessions (SPS)

The root password is required to access One Identity Safeguard for Privileged Sessions (SPS) locally, or remotely via an SSH connection. Note that the password of the `root` user can be changed from the console menu as well. For details, see Accessing the One Identity Safeguard for Privileged Sessions (SPS) console on page 389.

To change the root password of SPS

1. Navigate to **Basic Settings > Management > Change root password**.

   Figure 111: Basic Settings > Management > Change root password — Changing the root password of SPS

2. Enter the new password into the **New root password** and **Confirm password** fields.

   NOTE:

   One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

3. Click 

.

Firmware update using SSH

In some cases, uploading large files over HTTP is not possible. In such cases, you can update the firmware using SSH.

CAUTION:

The recommended way to update the firmware is using the One Identity Safeguard for Privileged Sessions (SPS) web interface (see Upgrading One Identity Safeguard for Privileged Sessions (SPS) on page 377). Update the SPS firmware using SSH is only if you cannot update the firmware using the web interface. Note that updating using SSH may be removed from later versions of SPS.

Prerequisites

- Remote SSH access to SPS must be enabled. For details, see Enabling SSH access to the One Identity Safeguard for Privileged Sessions (SPS) host on page 391.

To update the firmware using SSH

1. Download the firmware file to your computer.
2. Log in to SPS remotely using SSH, and select **Shells > Core shell** from the console menu.
3. Copy the firmware to the SPS host, for example, into the `/root/` directory.

   If you are copying the firmware to SPS using SCP and you issue the copy command on the client side and not within the core firmware, the root directory of the core firmware is: `/mnt/firmware/root`
4. Install the firmware: `/opt/scb/bin/firmwarectl install <path-to-firmware>`

   This command installs the firmware into the first empty slot, and returns the value of the slot where the firmware has been installed.
5. Check if you can upgrade to the new firmware, and resolve any errors before continuing: `/opt/scb/bin/firmwarectl precheck <slot-number-of-the-firmware>`

   In the returned values, `"exitcode": 0` means that the precheck has finished without any errors. `"exitcode": 1` means that errors have occurred, and the contents of `"output": []` gives you a clue as to what is causing the problem.
6. Activate the new firmware: `/opt/scb/bin/firmwarectl activate <slot-number-of-the-firmware>`

   Using the `/opt/scb/bin/firmwarectl list` command, you can check whether activation has been successful. In the returned values, look for your slot number and the value of `"active":`, it should say `true`. For example:

```
...
 "slot": 3,
 "precheck": true,
"active": true,
 "boot_link": "mnt/boot-firmware/slot3",
 "core_link": "mnt/firmware/slot3",
 "branch": "5.6",
 "version": "5.6.0a",
"current": false,
 ...
```

7. Reboot SPS: `xcbclient self xcb_do_reboot`
8. If the upgrade is successful, delete any unused firmware: `/opt/scb/bin/firmwarectl delete <slot-number-of-unused-firmware>`
9. Delete the firmware file you uploaded to SPS, it is not needed anymore: `rm -fv /root/<firmware-file-you-uploaded>`

Exporting and importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) using the console

For manual archiving, or to migrate it to another One Identity Safeguard for Privileged Sessions (SPS) unit, you can export/import the configuration of SPSfrom the console using the **/opt/scb/bin/configbundle.py** script.

NOTE:

You must run the **/opt/scb/bin/configbundle.py** script using the `root` user.

NOTE:

The configuration of your SPS may contain sensitive information. Make sure you delete any configuration export files that are not needed anymore.

To export/import the configuration of SPS from the console

1. Execute the following command to export the configuration of your SPS:

   ```
   /opt/scb/bin/configbundle.py create --bundle /<my destination folder>/bundle.tar.gz
   ```

   Where:

   - `/opt/scb/bin/configbundle.py`: The script you execute to export the configuration.
   - `create`: The option that lets you export a configuration.

- `--bundle`: The option used to specify the bundle file.
- `/<my destination folder>/bundle.tar.gz`: The path to the file where you wish to export the configuration.

  Replace `<my destination folder>` with the name of the folder where you wish to store the exported configuration.

2. Execute the following command to import the configuration of your SPS:

   ```
   /opt/scb/bin/configbundle.py import --bundle /<my destination folder>/bundle.tar.gz
   ```

   Where:

   - `/opt/scb/bin/configbundle.py`: The script you execute to import the configuration.
   - `import`: The option that lets you import a configuration.
   - `--bundle`: The option used to specify the bundle file.
   - `/<my destination folder>/bundle.tar.gz`: The path to the file from which you wish to import the configuration.

     Replace `<my destination folder>` with the name of the folder where your configuration export file is stored.

Sealed mode

When sealed mode is enabled, the following settings are automatically applied:

- One Identity Safeguard for Privileged Sessions (SPS) cannot be accessed remotely via SSH for maintenance.
- The root password of SPS cannot be changed in sealed mode.
- It is not possible to upload or delete plugins in sealed mode.
- Sealed mode can be disabled only from the local console. For details, see Disabling sealed mode on page 398.

To enable sealed mode use one of the following methods:

- Select the **Sealed mode** option during the Welcome Wizard.
- Select **Basic Settings > System > Sealed mode > Activate sealed mode** on the SPS web interface.
- Log in to SPS as root using SSH or the local console, and select **Sealed mode > Enable** from the console menu.

Disabling sealed mode

The event of disabling sealed mode is logged. The following describes how to disable sealed mode.

To disable sealed mode

1. Go to the One Identity Safeguard for Privileged Sessions (SPS) appliance and access the local console.
2. Log in as `root`.
3. From the console menu, select **Sealed mode > Disable**
4. Select **Back to Main menu > Logout**.

Out-of-band management of One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) 6.0.13 includes a dedicated out-of-band management interface conforming to the Intelligent Platform Management Interface (IPMI) v2.0 standards. The IPMI allows system administrators to monitor the system health of SPS and to manage the computer events remotely, independently of the operating system of SPS. SPS is accessible using the IPMI only if the IPMI is physically connected to the network.

Note that the IPMI supports only 100Mbps Full-Duplex speed.

- For details on connecting the IPMI, see "Installing the SPS hardware" in the Installation Guide.
- For details on configuring the IPMI, see Configuring the IPMI from the console on page 399.
- For details on using the IPMI to remotely monitor and manage SPS, see the following document:

  For Safeguard Sessions Appliance 3000 and 3500, see the X9 SMT IPMI User's Guide.

Basic information about the IPMI is available also on the SPS web interface on the **Basic Settings > High Availability** page. The following information is displayed:

Figure 112: Basic Settings > High Availability — Information about the IPMI SPS



- **Hardware serial number**: The unique serial number of the appliance.
- **IPMI IP address**: The IP address of the IPMI.
- **IPMI subnet mask**: The subnet mask of the IPMI.
- **IPMI default gateway**: The address of the default gateway configured for the IPMI.
- **IPMI IP address source**: Shows how the IPMI receives its IP address: dynamically from a DHCP server, or it uses a fixed static address.

Configuring the IPMI from the console

The following describes how to modify the network configuration of IPMI from the console of One Identity Safeguard for Privileged Sessions (SPS).

Prerequisites

SPS is accessible using the IPMI only if the IPMI is physically connected to the network. For details on connecting the IPMI, see "Installing the SPS hardware" in the Installation Guide.

CAUTION:

IPMI searches for available network interfaces during boot. Make sure that IPMI is connected to the network through the dedicated Ethernet interface before SPS is powered on.

CAUTION: SECURITY HAZARD!

The IPMI, like all out-of-band management interfaces, has known vulnerabilities that One Identity cannot fix or have an effect on. To avoid security hazards, One Identity recommends that you only connect the IPMI to well-protected, separated management networks with restricted accessibility. Failing to do so may result in an unauthorized access to all data stored on the SPS appliance. Data on the appliance can be unencrypted or encrypted, and can include sensitive information, for example, passwords, decryption keys, private keys, and so on.

For more information, see Best Practices for managing servers with IPMI features enabled in Datacenters.

NOTE:

The administrator of SPS must be authorized and able to access the IPMI for support and troubleshooting purposes in case vendor support is needed.

The following ports are used by the IPMI:

- Port 22 (TCP): SSH (configurable)
- Port 80 (TCP): Web (configurable)
- Port 161 (UDP, TCP): SNMP (configurable)

- Port 443 (TCP): Web SSL (configurable)
- Port 623 (UDP): Virtual Media (configurable)
- Port 5900 (TCP): IKVM Server (configurable)
- Port 5985 (TCP): Wsman (configurable)

The SSH encrypted connection (port 22) works with the following properties:

Supported:	Safeguard Sessions Appliance 3000	Safeguard Sessions Appliance 3500
Ciphers	aes128-ctr, aes256-ctr	3des-cbc, aes128-ctr, aes128-cbc, aes256-ctr, aes256-cbc
KEX algorithm	curve25519-sha256, ecdh-sha2-nistp256, curve25519-sha256@libssh.org, ecdh-sha2-nistp384, diffie-hellman-group1-sha1, ecdh-sha2-nistp521, diffie-hellman-group14-sha1	curve25519-sha256, ecdh-sha2-nistp256, curve25519-sha256@libssh.org, ecdh-sha2-nistp384, diffie-hellman-group1-sha1, ecdh-sha2-nistp521, diffie-hellman-group14-sha1
MACs	hmac-sha1, hmac-sha2-256, hmac-sha1-96, hmac-sha2-512	hmac-md5, hmac-sha2-256, hmac-sha1, hmac-sha2-512, hmac-sha1-96
HostKey algorithms	ssh-rsa, ssh-dss	ssh-rsa, ssh-dss
Compression	enabled	enabled

SSL encrypted connections work with the following properties:

Supported:	Safeguard Sessions Appliance 3000	Safeguard Sessions Appliance 3500
TLSv1.2	enabled	enabled
TLS Fallback SCSV	supported	supported
Heartbleed	not vulnerable	not vulnerable
Server Ciphers	Preferred TLSv1.2 256 bits ECDHE-RSA-AES256-GCM-SHA384 Curve P-256 DHE 256	Preferred TLSv1.2 256 bits ECDHE-RSA-AES256-GCM-SHA384 Curve P-256 DHE 256
	Accepted TLSv1.2 256 bits ECDHE-RSA-AES256-SHA384 Curve P-256 DHE 256	Accepted TLSv1.2 256 bits ECDHE-RSA-AES256-SHA384 Curve P-256 DHE 256
	Accepted TLSv1.2 256 bits DHE-RSA-AES256-GCM-SHA384 DHE	Accepted TLSv1.2 256 bits DHE-RSA-AES256-GCM-SHA384 DHE

Supported:	Safeguard Sessions Appliance 3000	Safeguard Sessions Appliance 3500
	2048 bits	2048 bits
	Accepted TLSv1.2 256 bits DHE-RSA-AES256-SHA256 DHE 2048 bits	Accepted TLSv1.2 256 bits DHE-RSA-AES256-SHA256 DHE 2048 bits
	Accepted TLSv1.2 256 bits AES256-GCM-SHA384	Accepted TLSv1.2 256 bits AES256-GCM-SHA384
	Accepted TLSv1.2 256 bits AES256-SHA256	Accepted TLSv1.2 256 bits AES256-SHA256
	Accepted TLSv1.2 128 bits ECDHE-RSA-AES128-GCM-SHA256 Curve P-256 DHE 256	Accepted TLSv1.2 128 bits ECDHE-RSA-AES128-GCM-SHA256 Curve P-256 DHE 256
	Accepted TLSv1.2 128 bits ECDHE-RSA-AES128-SHA256 Curve P-256 DHE 256	Accepted TLSv1.2 128 bits ECDHE-RSA-AES128-SHA256 Curve P-256 DHE 256
	Accepted TLSv1.2 128 bits DHE-RSA-AES128-GCM-SHA256 DHE 2048 bits	Accepted TLSv1.2 128 bits DHE-RSA-AES128-GCM-SHA256 DHE 2048 bits
	Accepted TLSv1.2 128 bits DHE-RSA-AES128-SHA256 DHE 2048 bits	Accepted TLSv1.2 128 bits DHE-RSA-AES128-SHA256 DHE 2048 bits
	Accepted TLSv1.2 128 bits AES128-GCM-SHA256	Accepted TLSv1.2 128 bits AES128-GCM-SHA256
	Accepted TLSv1.2 128 bits AES128-SHA256	Accepted TLSv1.2 128 bits AES128-SHA256
Server Key Exchange Groups	TLSv1.2 128 bits secp256r1 (NIST P-256)	TLSv1.2 128 bits secp256r1 (NIST P-256)
Server Signature Algorithms	TLSv1.2 Server accepts all signature algorithms.	TLSv1.2 Server accepts all signature algorithms.

To modify the network configuration of IPMI from the console of SPS

1. Use the local console (or SSH) to log in to SPS as root.
2. Choose **Shells > Boot shell**.
3. Check the network configuration of the interface:

   # ipmitool lan print

   This guide assumes that `channel 1` is used for LAN. If your setup differs, adjust the following commands accordingly.

4. Configure the interface. You can use DHCP or configure a static IP address manually.

   Use an IPv4 address.

   - To use DHCP, enter the following command:

     # ipmitool lan set 1 ipsrc dhcp
   - To use static IP, enter the following command:

     # ipmitool lan set 1 ipsrc static

     Set the IP address:

     # ipmitool lan set 1 ipaddr <IPMI-IP>

     Set the netmask:

     # ipmitool lan set 1 netmask <IPMI-netmask>

     Set the IP address of the default gateway:

     # ipmitool lan set 1 defgw ipaddr <gateway-IP>
5. Configure IPMI to use the dedicated Ethernet interface.
   - On the N1000, T1, T4, and T10 appliances, issue the following command:

     # ipmitool raw 0x30 0x70 0xc 1 0
   - On the 1000d and 10000 appliances, issue the following command:

     # ipmitool raw 0x30 0x70 0xc 1 1 0
6. Verify the network configuration of IPMI:

   # ipmitool lan print 1

   Use a browser to connect to the reported network address.
7. Change the default password:
   a. Log in to the IPMI web interface using the default login credentials (username: `ADMIN`, password: `ADMIN` or `changeme`, depending on your hardware).

      NOTE:

      The login credentials are case sensitive.

   b. Navigate to **Configure > Users**.

   c. Select **ADMIN**, and choose **Modify User**.

   d. Change the password, and save the changes with **Modify**.

Configuring the IPMI from the BIOS

To configure IPMI from the BIOS when configuring your One Identity Safeguard for Privileged Sessions (SPS) physical appliance for the first time, complete the following steps.

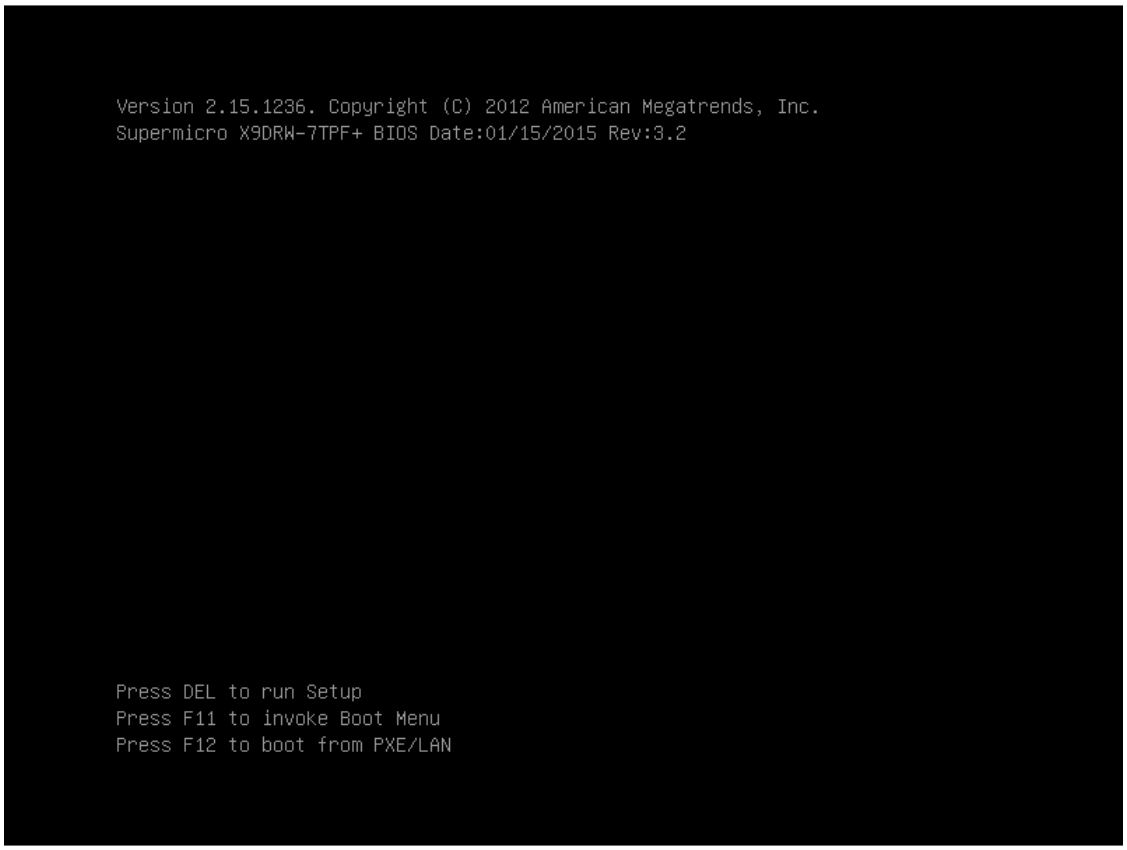
Prerequisites

To apply the procedure outlined here, you will need physical access to a monitor and keyboard.

To configure the IPMI from the BIOS

1. Press the DEL button when the POST screen comes up while the appliance is booting.

   Figure 113: POST screen during booting

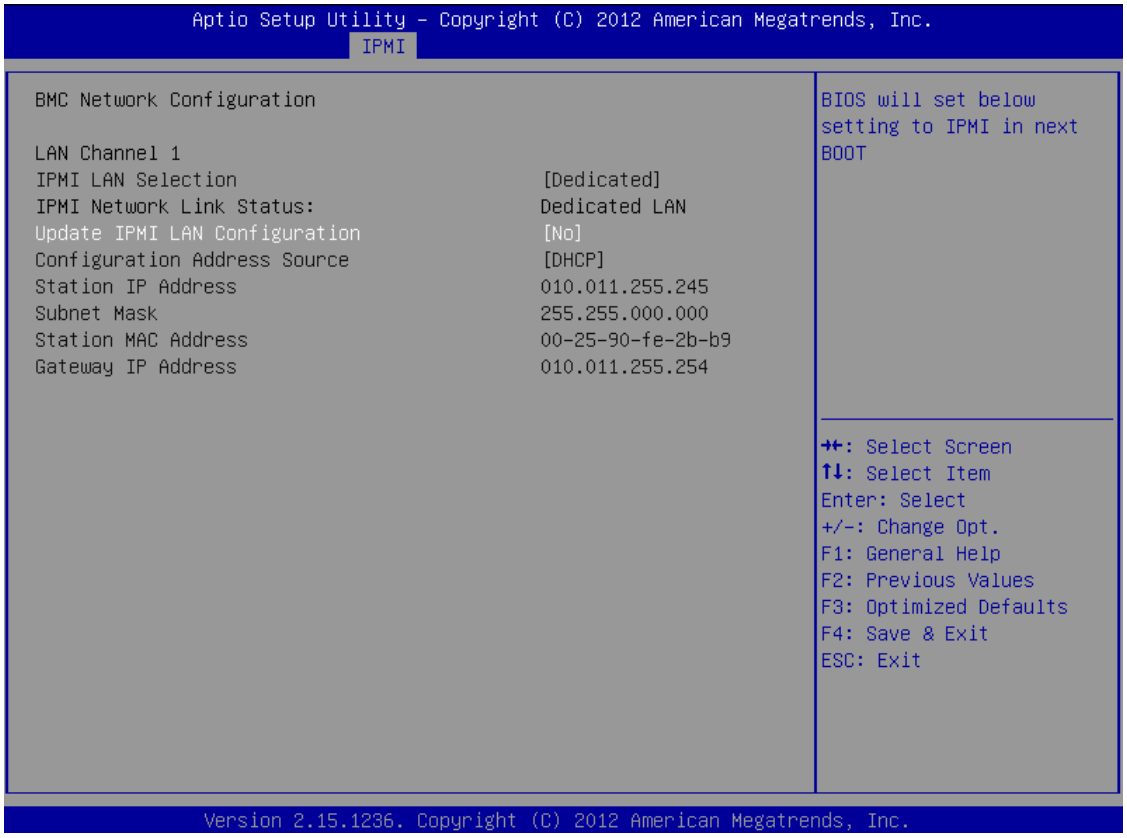
2. In the BIOS, navigate to the **IPMI** page.
3. On the **IPMI** page, select **BMC Network Configuration**, and press Enter.

Figure 114: IPMI page > BMC Network Configuration option



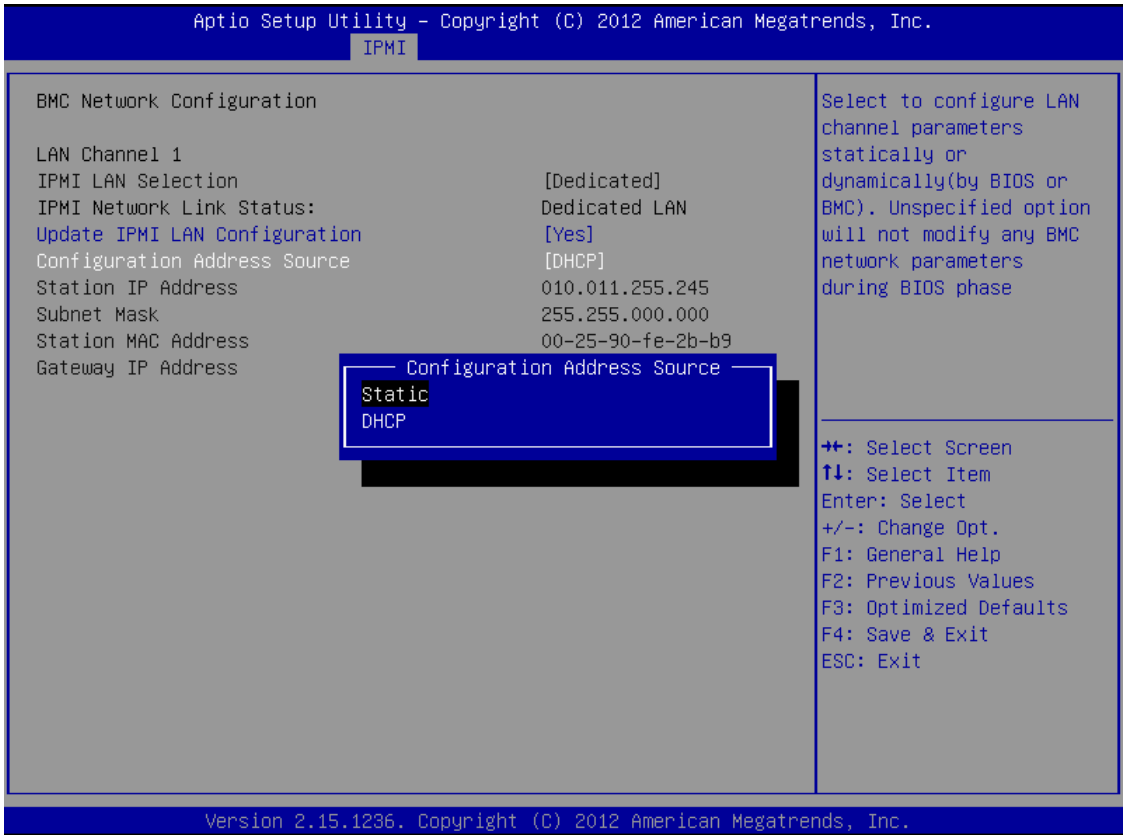
4. On the **BMC Network Configuration** page, select **Update IPMI LAN Configuration**, press `Enter`, and select **Yes**.

Figure 115: BMC Network Configuration page > Update IPMI LAN Configuration



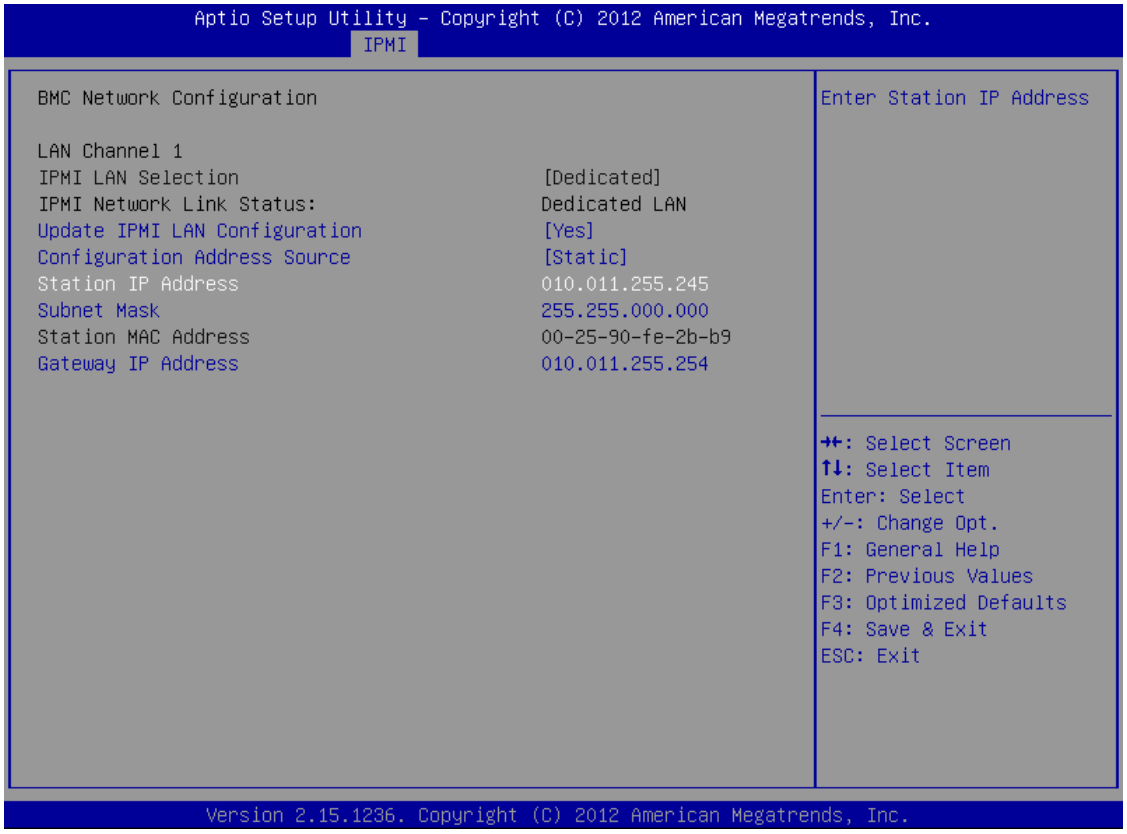
5. Stay on the **BMC Network Configuration** page, select **Configuration Address Source**, press **Enter**, and select **Static**.

Figure 116: BMC Network Configuration page > Configuration Address Source



6. Still on the **BMC Network Configuration** page, configure the **Station IP Address**, **Subnet Mask**, and **Gateway IP Address** individually.

Figure 117: BMC Network Configuration page > Station IP Address, Subnet Mask, Gateway IP Address



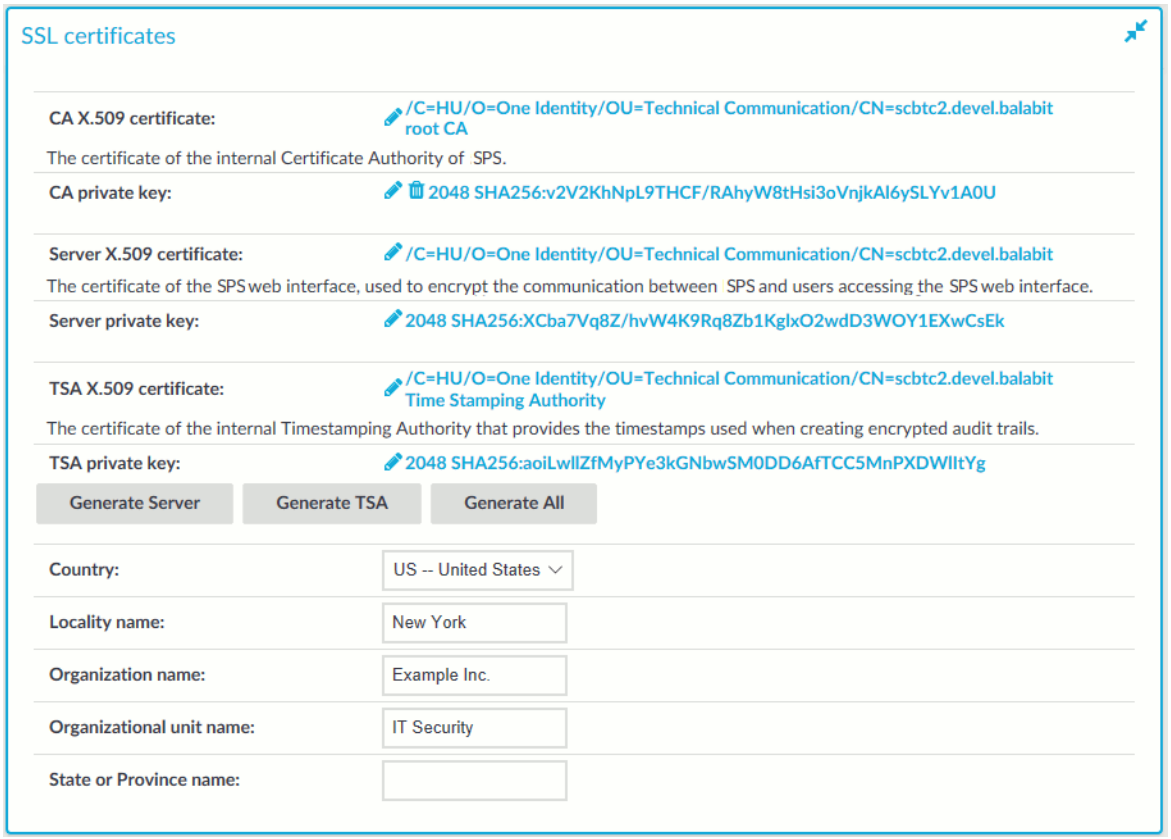
7. Press F4 to save the settings, and exit from the BIOS.

   About a minute later, you will be able to log in on the IPMI web interface.

Managing the certificates used on One Identity Safeguard for Privileged Sessions (SPS)

One Identity Safeguard for Privileged Sessions (SPS) uses a number of certificates for different tasks that can be managed from the **Basic Settings > Management > SSL certificates** menu.

Figure 118: Basic Settings > Management > SSL certificates — Changing the web certificate of SPS



The following certificates can be modified here:

- **CA certificate**: The certificate of the internal Certificate Authority of SPS.
- **Server certificate**: The certificate of the SPS web interface, used to encrypt the communication between SPS and the administrators.

  NOTE:

  If this certificate is changed, the browser of SPS users will display a warning stating that the certificate of the site has changed.

- **TSA certificate**: The certificate of the internal Timestamping Authority that provides the timestamps used when creating encrypted audit-trails.

NOTE:

SPS uses other certificates for different purposes that are not managed here, for example, to encrypt data stored on SPS. For details, see Encrypting audit trails on page 465.

Use every keypair or certificate only for one purpose. Do not reuse cryptographic keys or certificates (for example, do not use the certificate of the One Identity Safeguard for Privileged Sessions (SPS) webserver to encrypt audit trails, or the same keypair for signing and encrypting data).

For every certificate, the distinguished name (DN) of the X.509 certificate and the fingerprint of the private key is displayed. To display the entire certificate click on the DN. To display the public part of the private key, click on the fingerprint. It is not possible to download the private key itself from the SPS web interface, but the public part of the key can be downloaded in different formats (for example PEM, DER, or OpenSSH). Also, the X.509 certificate can be downloaded in PEM and DER formats.

During the initial configuration, SPS creates a self-signed CA certificate, and uses this CA to issue the certificate of the web interface (see **Server certificate**) and the internal Timestamping Authority (**TSA certificate**).

There are two methods to manage certificates of SPS:

- **Recommended**: Generate certificates using your own PKI solution and upload them to SPS.

  Generate a CA certificate and two other certificates signed with this CA using your PKI solution and upload them to SPS. For the Server and TSA certificates, upload the private key as well. One Identity recommends using 2048-bit RSA keys (or stronger), and to use certificates that have the appropriate `keyUsage` or `extendedKeyUsage` fields set (for example, `extendedKeyUsage=serverAuth` for the SPS web server certificate).

  For details on uploading certificates and keys created with an external PKI, complete Uploading external certificates to One Identity Safeguard for Privileged Sessions (SPS) on page 410.

  CAUTION:

  The Server and the TSA certificates must be issued by the same Certificate Authority.

- Use the certificates generated on SPS. In case you want to generate new certificates and keys for SPS using its self-signed CA certificate, or generate a new self-signed CA certificate, complete Generating certificates for One Identity Safeguard for Privileged Sessions (SPS) on page 409.

  NOTE:

  Generate certificates using your own PKI solution and upload them to SPS whenever possible. Certificates generated on SPS cannot be revoked, and can become a security risk if they are somehow compromised.

Generating certificates for One Identity Safeguard for Privileged Sessions (SPS)

Create a new certificate for the One Identity Safeguard for Privileged Sessions (SPS) webserver or the Timestamping Authority using the internal CA of SPS, or create a new, self-signed CA certificate for the internal Certificate Authority of SPS.

One Identity recommends using 2048-bit RSA keys (or stronger).

To create a new certificate for the SPS webserver

1. Navigate to **Basic Settings > Management > SSL certificates**.
2. Fill the fields of the new certificate:
    a. **Country**: Select the country where SPS is located (for example HU - Hungary).
    b. **Locality name**: The city where SPS is located (for example Budapest).
    c. **Organization name**: The company who owns SPS (for example Example Inc.).
    d. **Organization unit name**: The division of the company who owns SPS (for example IT Security Department).
    e. **State or Province name**: The state or province where SPS is located.
3. Select the certificate you want to generate.
    - To create a new certificate for the SPS web interface, select **Generate Server**.
    - To create a new certificate for the Timestamping Authority, select **Generate TSA**.
    - To create a new certificate for the internal Certificate Authority of SPS, select **Generate All**. Note that in this case new certificates are created automatically for the server and TSA certificates as well.

    NOTE:

    When generating new certificates, the server and TSA certificates are signed using the certificate of the CA. If you have uploaded an external CA certificate along with its private key, it will be used to create the new server and TSA certificates. If you have uploaded an external CA certificate without its private key, use your external PKI solution to generate certificates and upload them to SPS.

    CAUTION:

    Generating a new certificate automatically deletes the earlier certificate.

4. Click Commit.

Uploading external certificates to One Identity Safeguard for Privileged Sessions (SPS)

Upload a certificate generated by an external PKI system to One Identity Safeguard for Privileged Sessions (SPS).

Prerequisites

The certificate to upload. For the **TSA X.509 Certificate** and **Server X.509 Certificate**, the private key of the certificate is needed as well. The certificates must meet the following requirements:

- SPS accepts certificates in PEM format. The DER format is currently not supported.
- SPS accepts private keys in PEM (RSA), and PUTTY format. Password-protected private keys are also supported.

  NOTE:

  One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

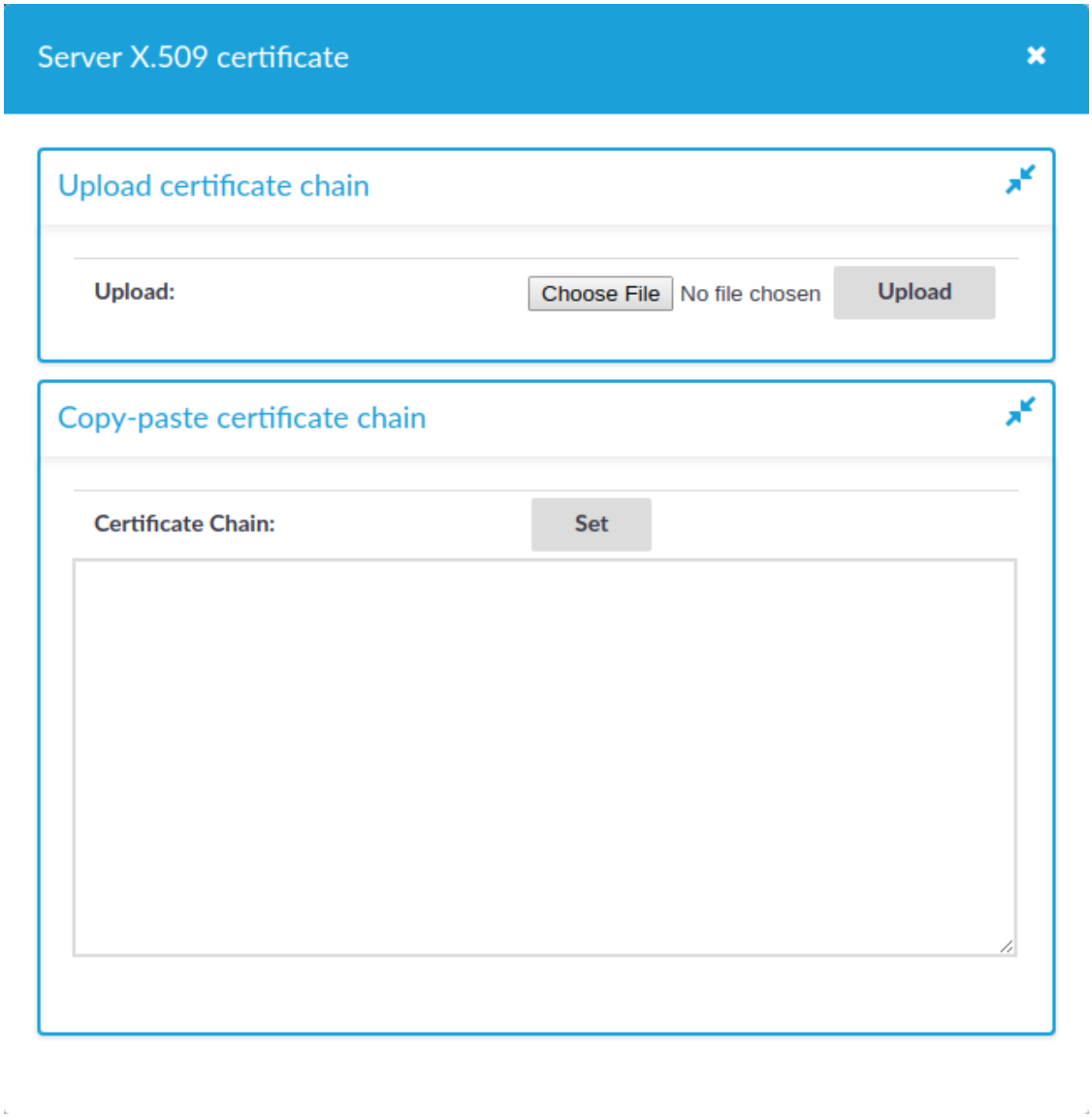
  For the internal CA certificate of SPS, uploading the private key is not required.
- For the TSA certificate, the `X509v3 Extended Key Usage` attribute must be enabled and set to `critical`. Also, its default value must be set to `Time Stamping`.
- For the Server certificate, the `X509v3 Extended Key Usage` attribute must be enabled and its default value set to `TLS Web Server Authentication`. Also, the Common Name of the certificate must contain the domain name or the IP address of the SPS host. If the web interface is accessible from multiple interfaces or IP addresses, list every IP address using the `Subject Alt Name` option.
- For the certificate used to sign audit trails, the `X509v3 Extended Key Usage` attribute must be enabled and its default value set to `Sign (downloadable) executable code`.

One Identity recommends using 2048-bit RSA keys (or stronger).

To upload a certificate generated by an external PKI system to SPS

1. Navigate to **Basic Settings > Management > SSL certificates**.
2. Click to upload the new certificate. A pop-up window is displayed.

Figure 119: Basic Settings > Management > SSL certificates — Uploading certificates



Select **Browse**, select the file containing the certificate, and click **Upload**.

For the Server X.509 Certificate

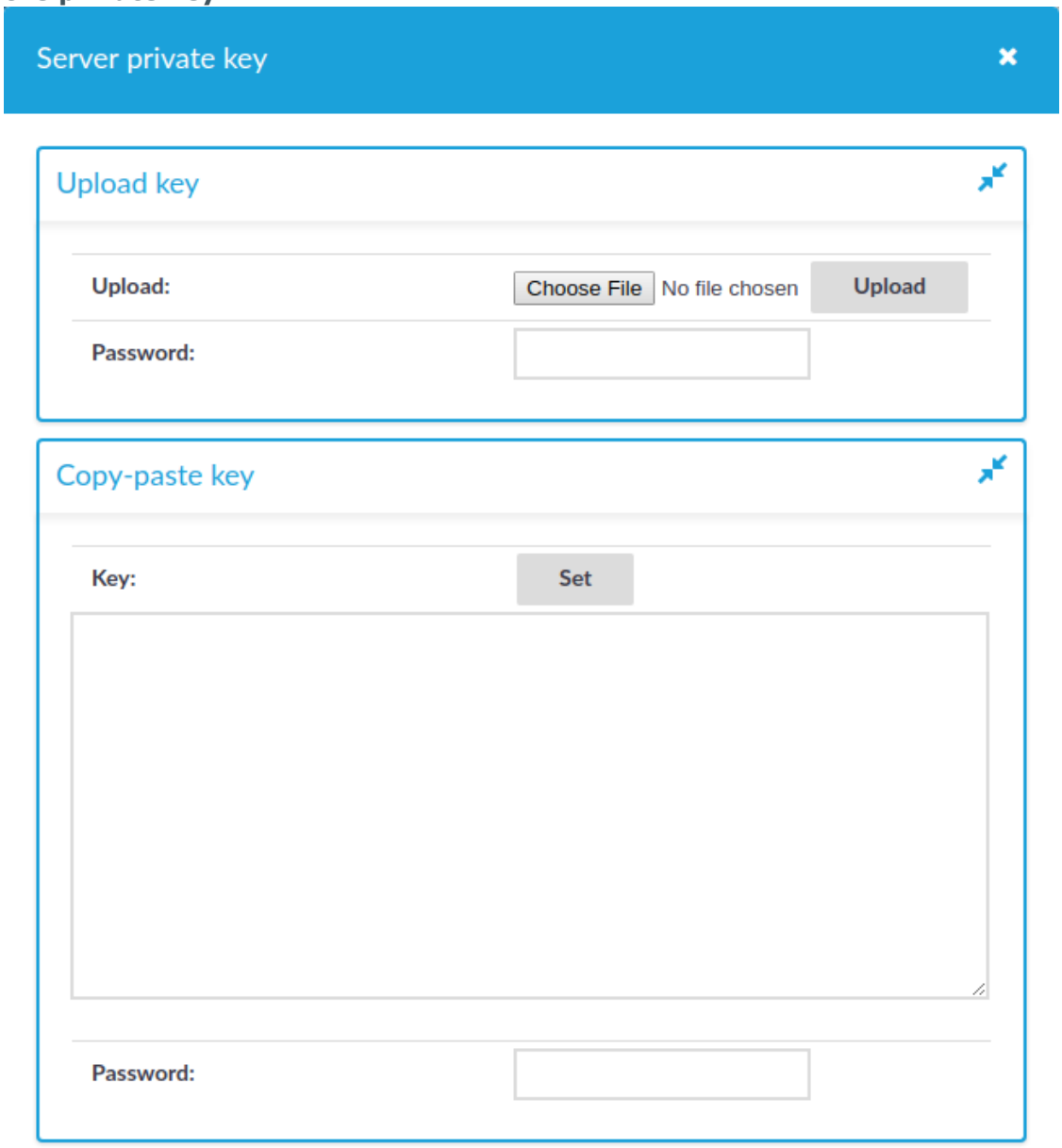
For the **Server X.509 Certificate**, you can also upload a certificate chain. For that, copy the certificates after each other in a single file. Alternatively, you can copy and paste the certificates one by one after each other into the **Certificate** field and click **Set**. The certificates do not have to be in order, SPS will order them and validate the chain: if a member of the chain is missing, an error message is displayed.

NOTE:

Certificate chains are supported only for the **Server X.509 Certificate**.

To upload the private key corresponding to the certificate, click icon. A pop-up window is displayed.

Figure 120: Basic Settings > Management > SSL certificates — Uploading the private key



3.

Select **Browse**, select the file containing the private key, provide the **Password** if the key is password-protected, and click **Upload**. Alternatively, you can also copy-paste the private key into the **Key** field, provide the **Password** there, and click **Set**.

In the case of a certificate chain, the private key has to be the same as the bottom level certificate.

Expected result

The new certificate is uploaded. If you receive the `Certificate issuer mismatch` error message after importing a certificate, you must import the CA certificate which signed the certificate as well (the private key of the CA certificate is not mandatory).

NOTE:

To download previously uploaded certificates, click on the certificate and either download the certificate (or certificate chain) in one single PEM or DER file, or you can download single certificate files separately (if it is a certificate chain).

Generating TSA certificate with Windows Certificate Authority on Windows Server 2008

To generate a TSA certificate with Windows Certificate Authority (CA) that works with One Identity Safeguard for Privileged Sessions (SPS), generate a CSR (certificate signing request) on a computer running OpenSSL and sign it with Windows CA, then import this certificate into SPS for timestamping.

Prerequisites

A valid configuration file for OpenSSL with the following extensions:

```
[ tsa_cert ]
extendedKeyUsage = critical,timeStamping
```

TIP:

You can copy `/etc/xcb/openssl-ca.cnf` from SPS to the computer that will be used for signing. Rename the file to `openssl-temp.cnf`.

The TSA certificate is considered valid, in terms of compatibility with SPS, if the following conditions are met:

- Must be a valid CA certificate (**CA** is `true`).
- **Key Usage**: `Time Stamping` is required. No other key usage is permitted.
- **Extended Key Usage**: Must be set to `critical`.

- **Optional Key Usage**: If **Key Usage** is present, it must be `digitalSignature` and/or `nonRepudiation`. Other values are not permitted. Make sure that in **Encryption**, **Allow key exchange without key encryption (key agreement)** is selected.

  CAUTION:

  In Encryption, do NOT select Allow key exchange only with key encryption (key encipherment), because it will result in errors.

The following X509v3 extensions are supported:

- Hard requirement:

  X509v3 Extended Key Usage must be `critical`, and must only contain `Time Stamping`.

- Optional:

  X509v3 Key Usage, if present, must be `digitalSignature` and/or `nonRepudiation`.

To generate TSA certificate with Windows Certificate Authority on Windows Server 2008

1. Create CSR using the new configuration file: **openssl req -set_serial 0 -config openssl-temp.cnf -reqexts tsa_cert -new -newkey rsa:2048 -keyout timestamp.key -out timestamp.csr -nodes**
2. Complete the required fields according to your environment:

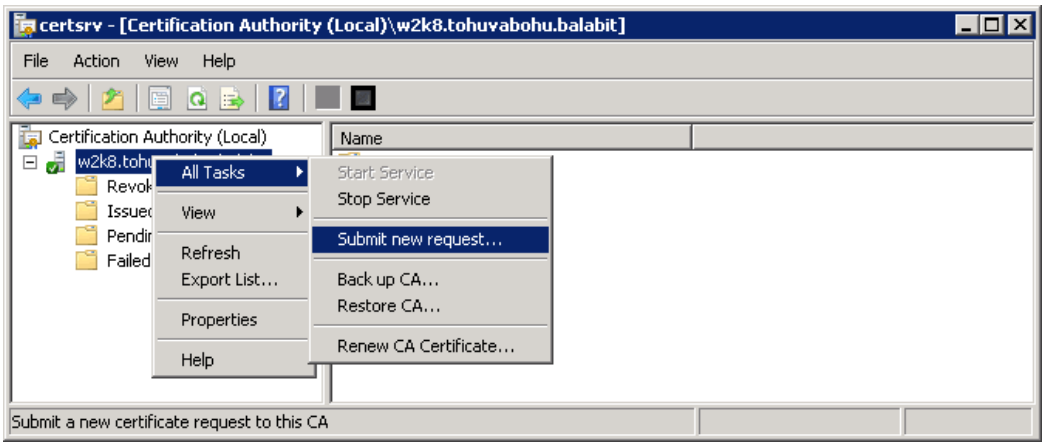
```
Generating a 2048 bit RSA private key
.....................+++
.....................................+++
writing new private key to 'timestamp.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) []:New York
Locality Name (eg, city) []:New York
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Examplecompany
IT Security
Organizational Unit Name (eg, section) []:Service Delivery
Common Name (eg, YOUR name) []:scb35-1-i1.tohuvabohu.examplecompany
Email Address []:vlad@examplecompany.com
```

3. Sign the generated CSR with your Windows CA. Make sure that the CSR file is accessible from your Windows CA server.

a. To issue and sign the new certificate request, open the Microsoft Certification Authority Management Console: **Start > Run** and run `certsrv.msc`.

b. Right-click on the server name and navigate to **All Tasks > Submit new request...**.

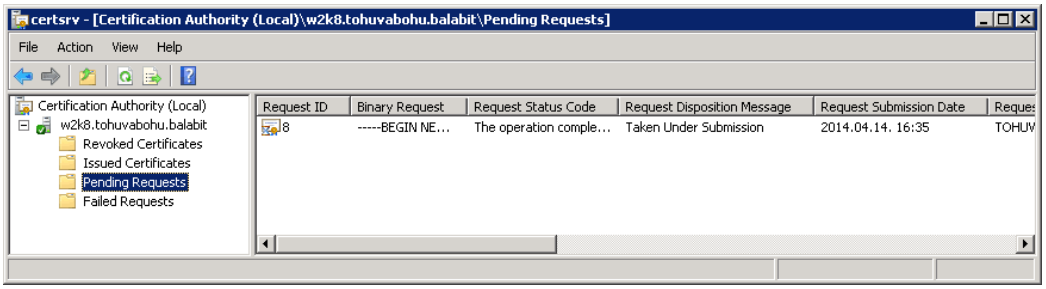
Figure 121: Submitting a new request



c. Select the CSR created in the second step.

d. On the left pane, click **Pending Requests**. The new certificate request is displayed in the right pane.

Figure 122: Issuing a new certificate

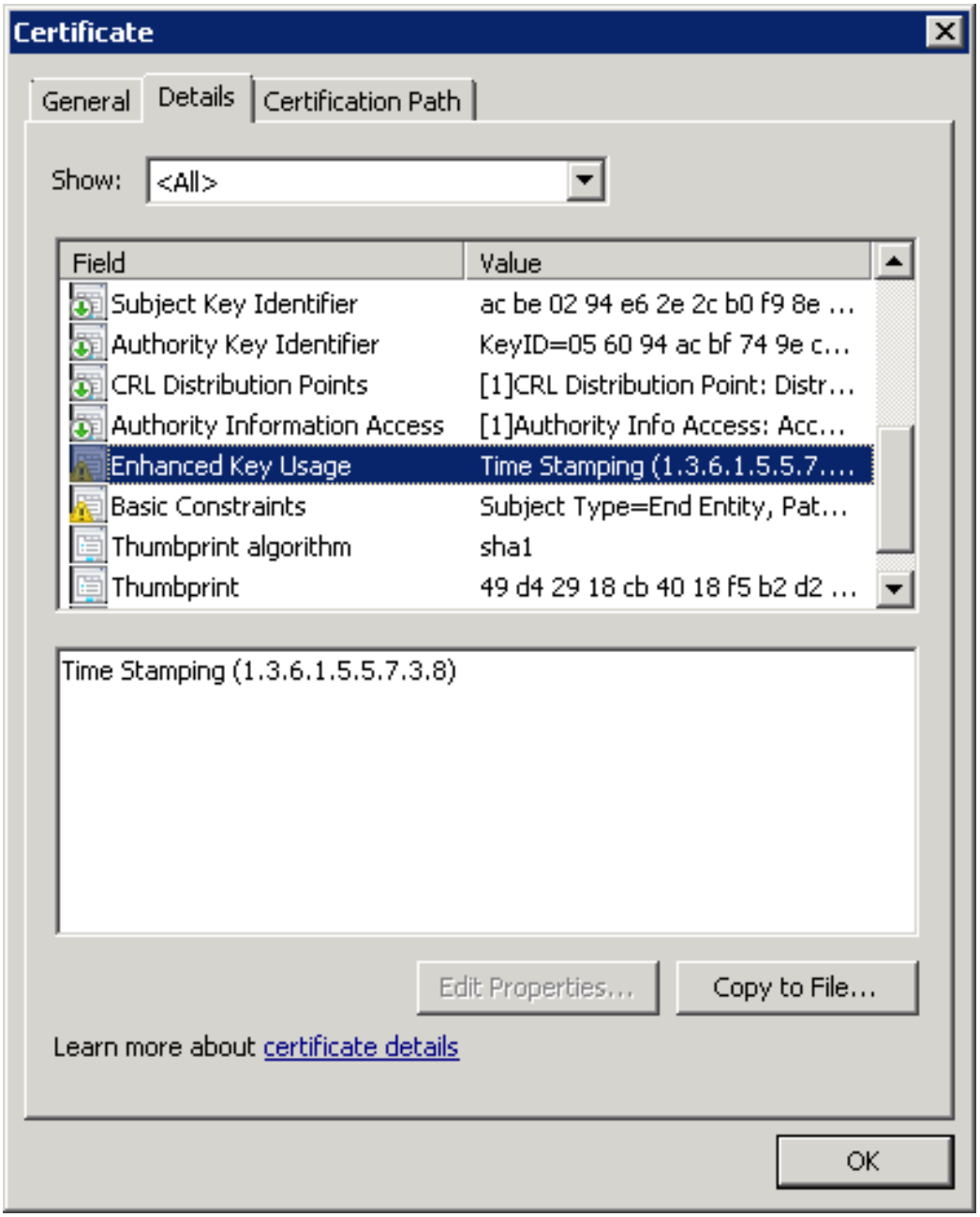


e. To issue the new SSL certificate, right-click on the pending certificate request, select "All Tasks" and click on "Issue".

f. Select "Issued Certificates" and double-click on the certificate issued in the previous step.

g. The CA Certificate window opens. Navigate to the **Details** tab. Ensure that the required **Enhanced Key Usage** field is visible and contains the `Time Stamping` value.

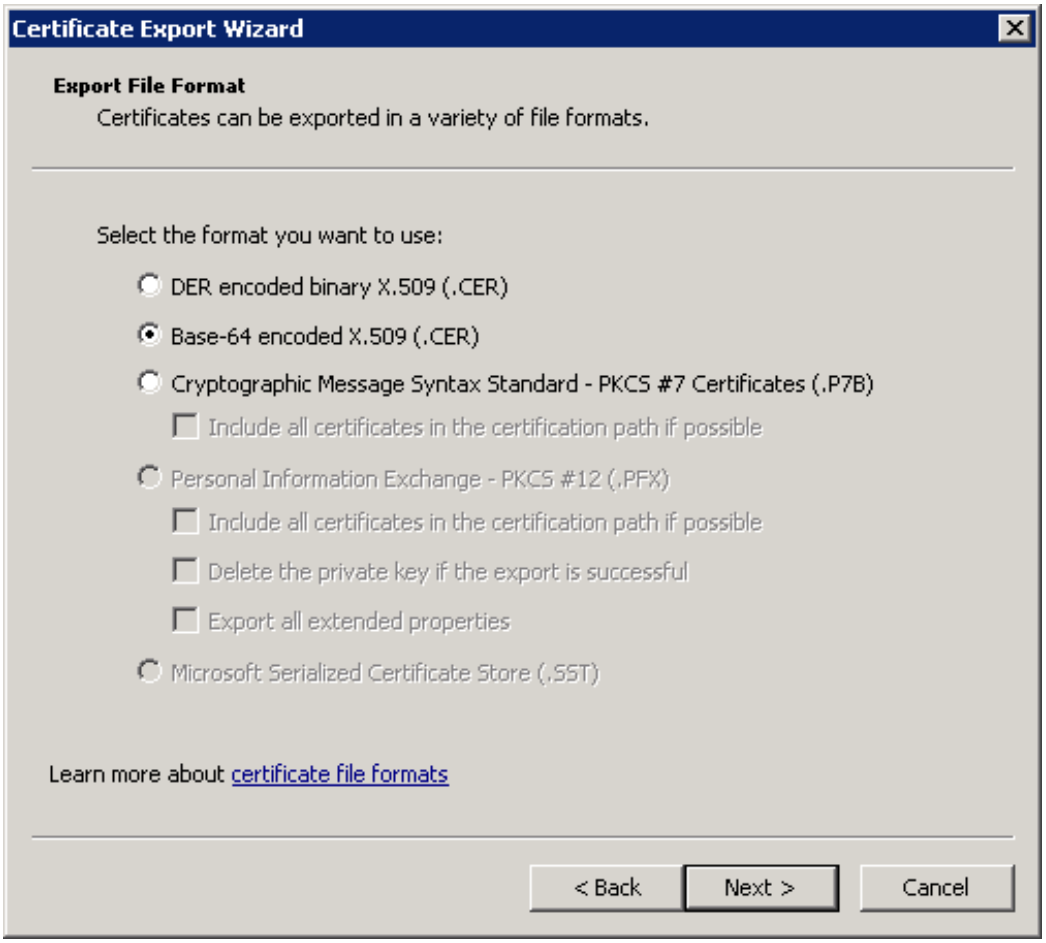
Figure 123: Verifying certificate details



h. Click **Copy to File**. The Certificate Export Wizard launches. Click **Next**.

i. Select the format of the certificate: **Base-64 encoded X.509 (.CER)**. Click **Next**.

Figure 124: Selecting certificate file format



j. Select location to save the certificate, and save it.

k. The **Completing the Certificate Export Wizard** screen is displayed. Click **Finish**.

4. In SPS, navigate to **Basic Settings > Management > SSL certificates**.
5. Click next to **TSA X.509 certificate**, browse for the previously generated certificate, and click **Upload**.
6. Click next to **TSA private key**, browse for the previously generated key, and click **Upload**.

NOTE:

If the root CA (the **CA X.509 certificate** field under **Basic Settings > Management > SSL certificates**) that is used for other certificates on SPS is different from the CA that was used to sign the TSA certificate, a warning is displayed. In this scenario, ignore this warning.

Generating TSA certificate with Windows Certificate Authority on Windows Server 2012

To generate a TSA certificate with Windows Certificate Authority (CA) that works with One Identity Safeguard for Privileged Sessions (SPS), generate a CSR (certificate signing request) on a computer running OpenSSL and sign it with Windows CA, then import this certificate into SPS for timestamping.

Prerequisites

A valid configuration file for OpenSSL with the following extensions:

```
[ tsa_cert ]
extendedKeyUsage = critical,timeStamping
```

> **TIP:**
>
> You can copy `/etc/xcb/openssl-ca.cnf` from SPS to the computer that will be used for signing. Rename the file to `openssl-temp.cnf`.

The TSA certificate is considered valid, in terms of compatibility with SPS, if the following conditions are met:

- Must be a valid CA certificate (**CA** is `true`).
- **Key Usage**: `Time Stamping` is required. No other key usage is permitted.
- **Extended Key Usage**: Must be set to `critical`.
- **Optional Key Usage**: If **Key Usage** is present, it must be `digitalSignature` and/or `nonRepudiation`. Other values are not permitted. Make sure that in **Encryption**, **Allow key exchange without key encryption (key agreement)** is selected.

  > **CAUTION:**
  >
  > **In Encryption, do NOT select Allow key exchange only with key encryption (key encipherment), because it will result in errors.**

The following X509v3 extensions are supported:

- Hard requirement:

  X509v3 Extended Key Usage must be `critical`, and must only contain `Time Stamping`.
- Optional:

  X509v3 Key Usage, if present, must be `digitalSignature` and/or `nonRepudiation`.

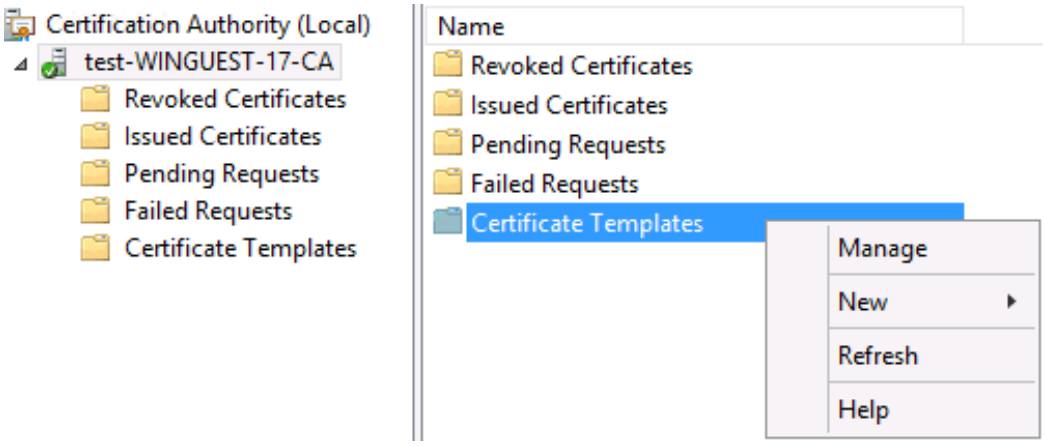
To generate TSA certificate with Windows Certificate Authority on Windows Server 2012

1. Create CSR using the new configuration file: **openssl req -set_serial 0 -config openssl-temp.cnf -reqexts tsa_cert -new -newkey rsa:2048 -keyout timestamp.key -out timestamp.csr -nodes**
2. Complete the required fields according to your environment:

```
Generating a 2048 bit RSA private key
.....................+++
.....................................+++
writing new private key to 'timestamp.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) []:New York
Locality Name (eg, city) []:New York
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Examplecompany
IT Security
Organizational Unit Name (eg, section) []:Service Delivery
Common Name (eg, YOUR name) []:scb35-1-i1.tohuvabohu.examplecompany
Email Address []:vlad@examplecompany.com
```

3. Create and configure a time stamping web server template in the Certificate Authority, and use that to generate the TSA certificate.
    a. Start the Certification Authority Microsoft Management Console, and select the CA server.
    b. Right-click on **Certificate Templates**, and choose **Manage**.

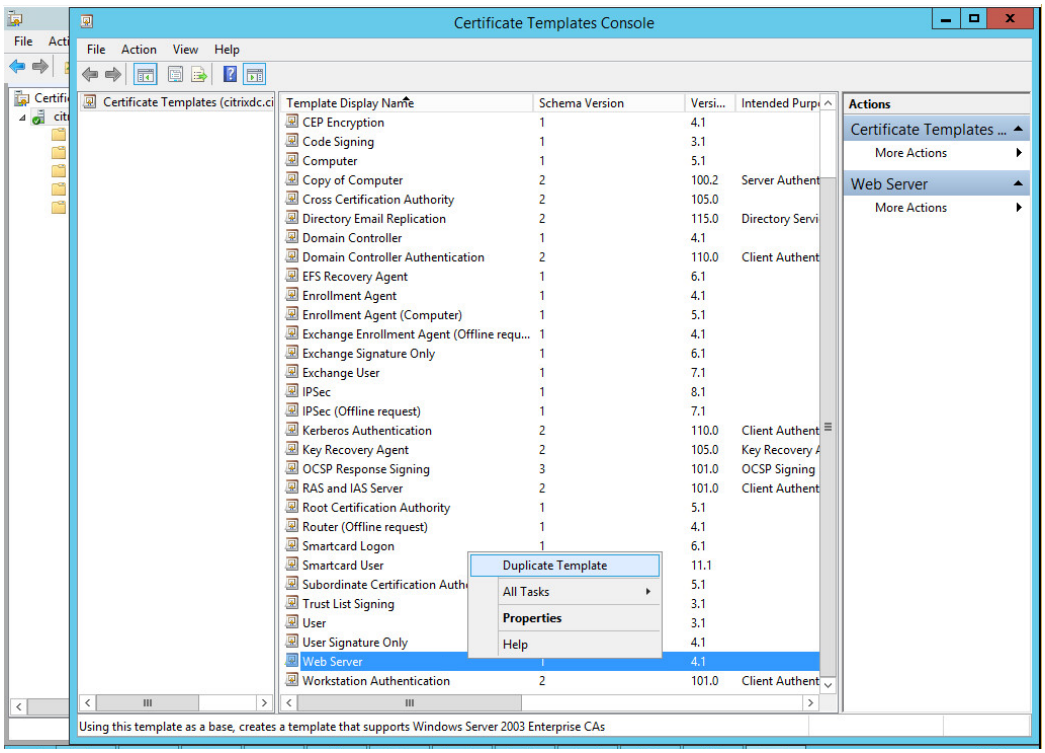
Figure 125: Managing certificate templates



The *Certificate Templates Console* opens.

c. Right-click the **Web Server** template, and choose **Duplicate Template**.

Figure 126: Duplicating a Template

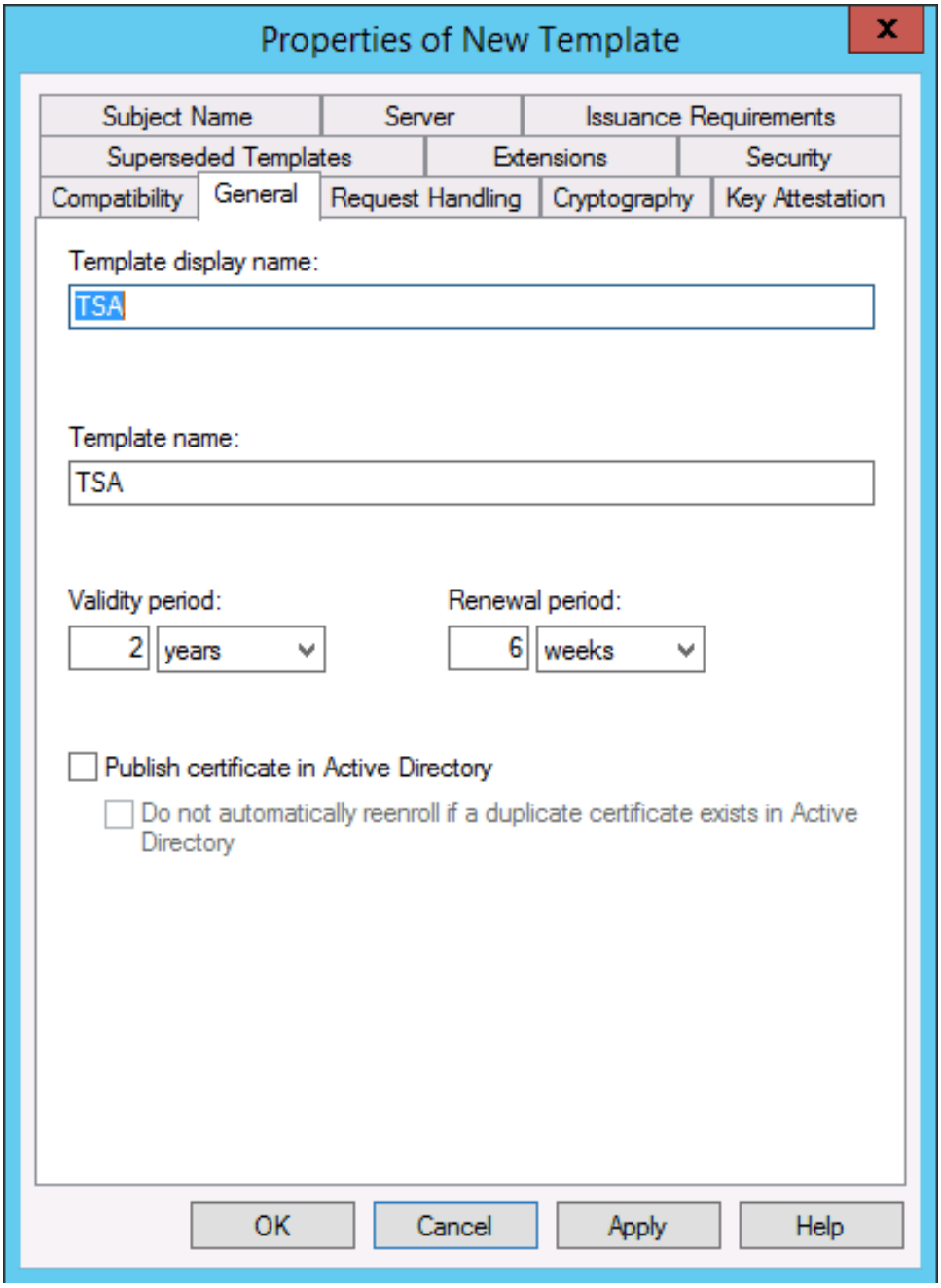


The *Properties of New Template* window is displayed.

d. Make the following changes to the new template:

- On the *General* tab, change the **Template display name** to TSA.

Figure 127: Creating the new template

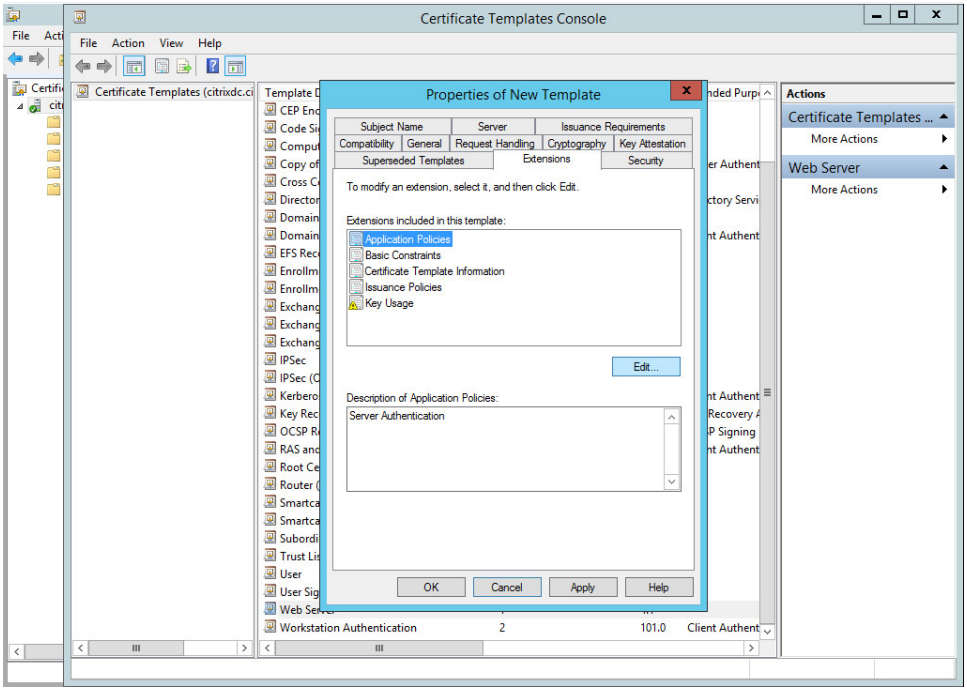


- On the *Request Handling* tab, enable the **Allow private key to be exported** option.
- On the *Extensions* tab, make the following changes:

Edit Application Policies

Select **Application Policies** and click **Edit** below the list of extensions.

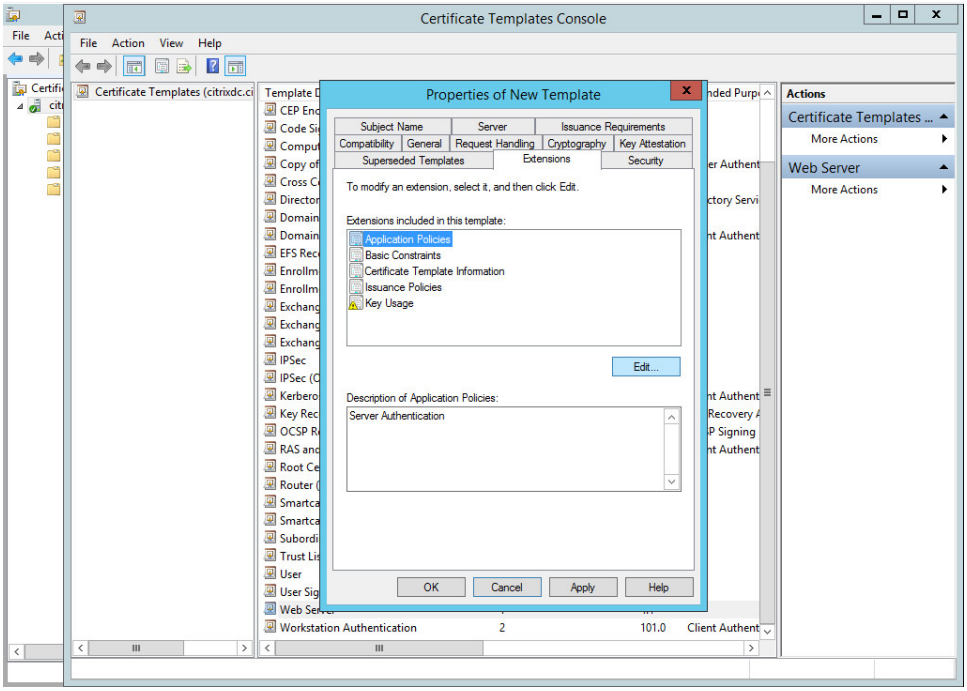
Figure 128: Editing Application Policies



Remove Server Authentication

Select **Server Authentication** and click **Remove**.

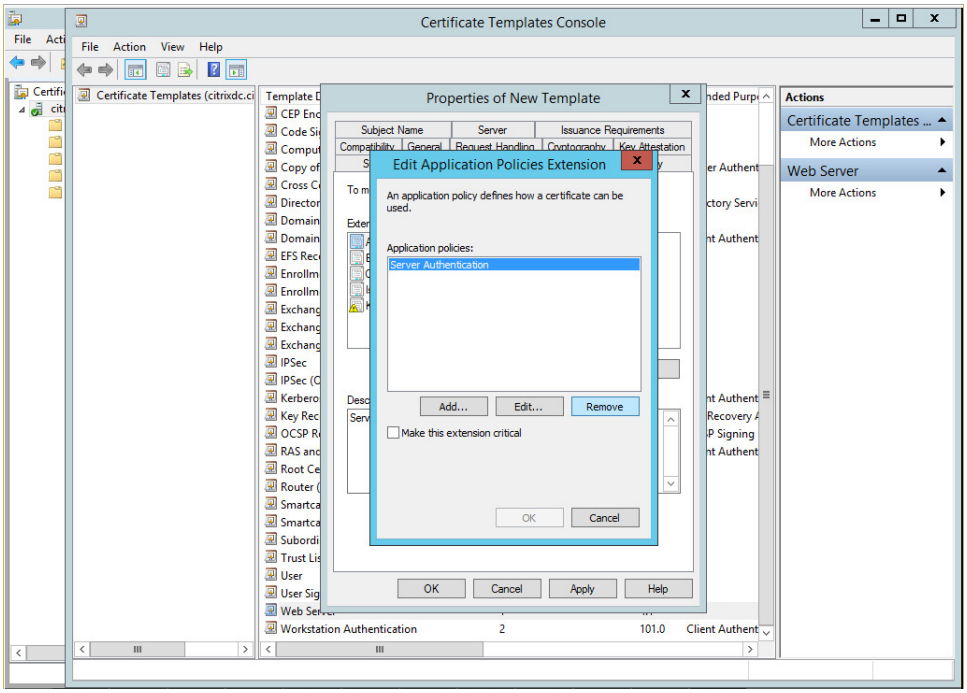
Figure 129: Removing Server Authentication



Add Time Stamping

Click **Add**, select **Time Stamping** and click **OK**.

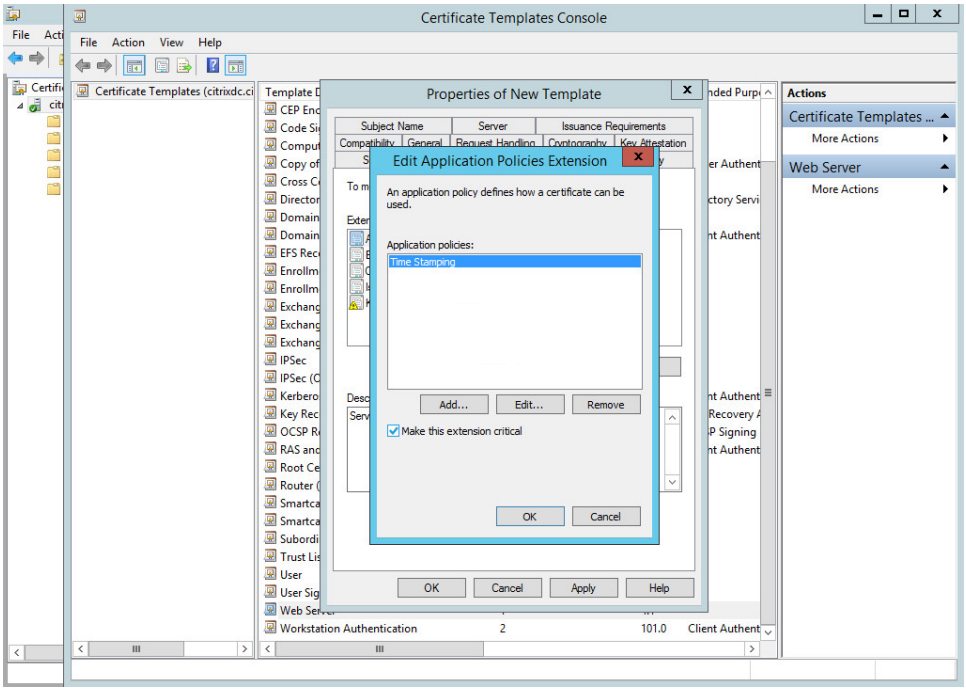
Figure 130: Adding Time Stamping



Make Time Stamping critical

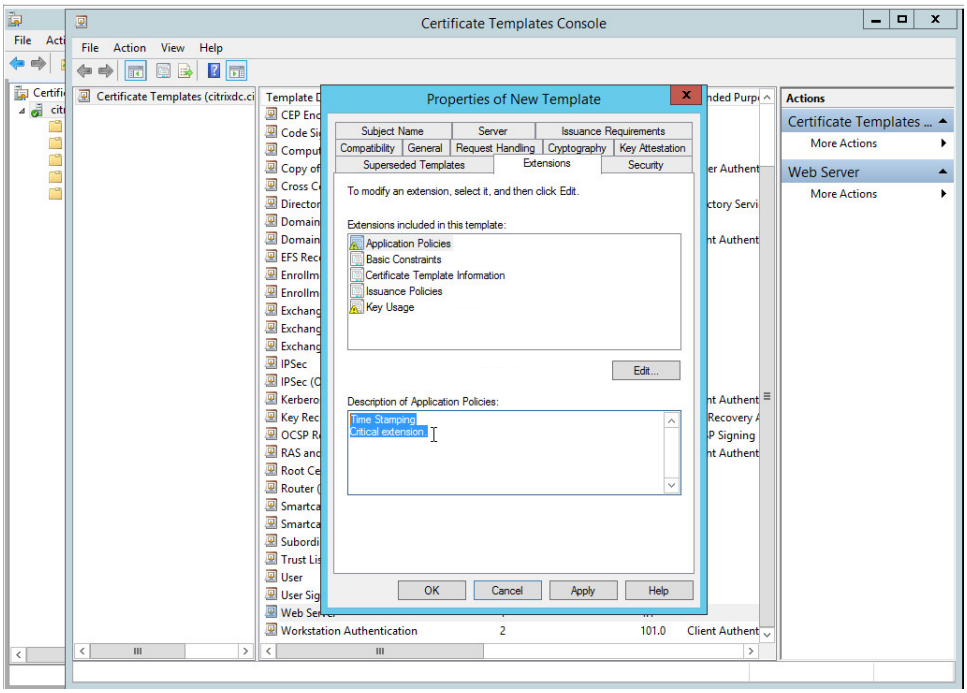
Select **Time Stamping** and enable the **Make this extension critical** option, then click **OK**.

Figure 131: Making Time Stamping critical



Time Stamping and **Critical extension** are listed in the **Description of Application Policies**.

Figure 132: Description of Application Policies



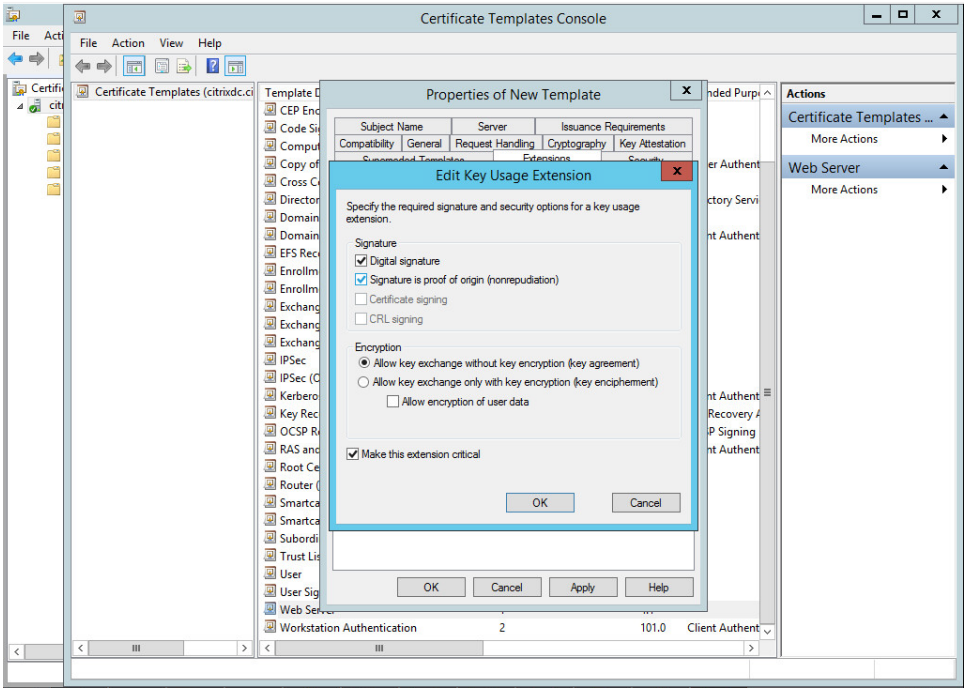
Edit Key Usage

Select **Key usage**, click **Edit**. Enable the **Signature is proof of origin (nonrepudiation)** option.

Select **Allow key exchange without key encryption (key agreement)**.

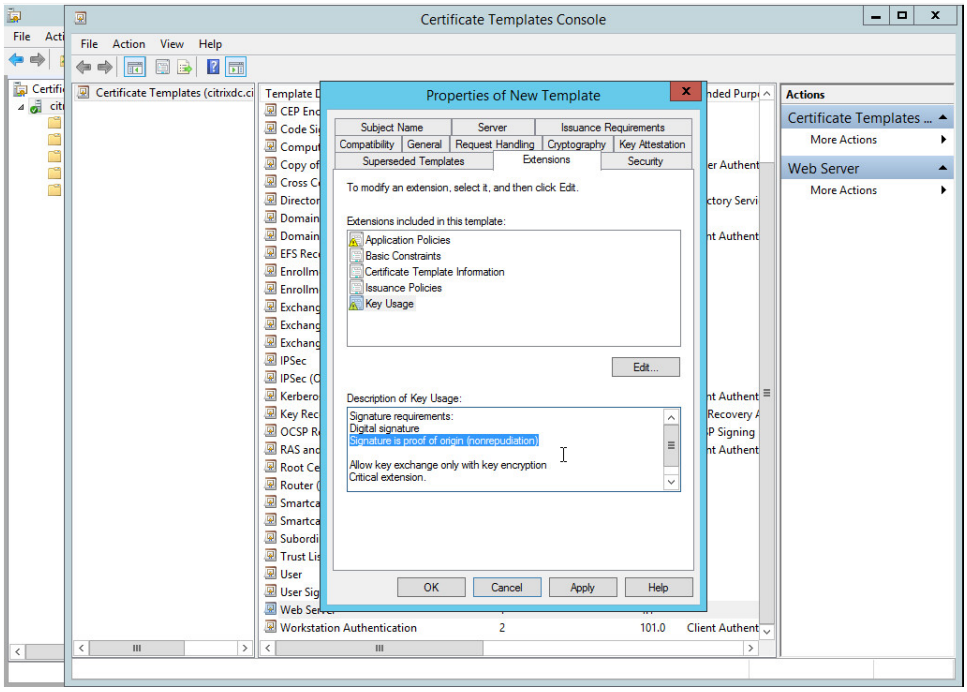
Click **OK**.

Figure 133: Editing Key Usage



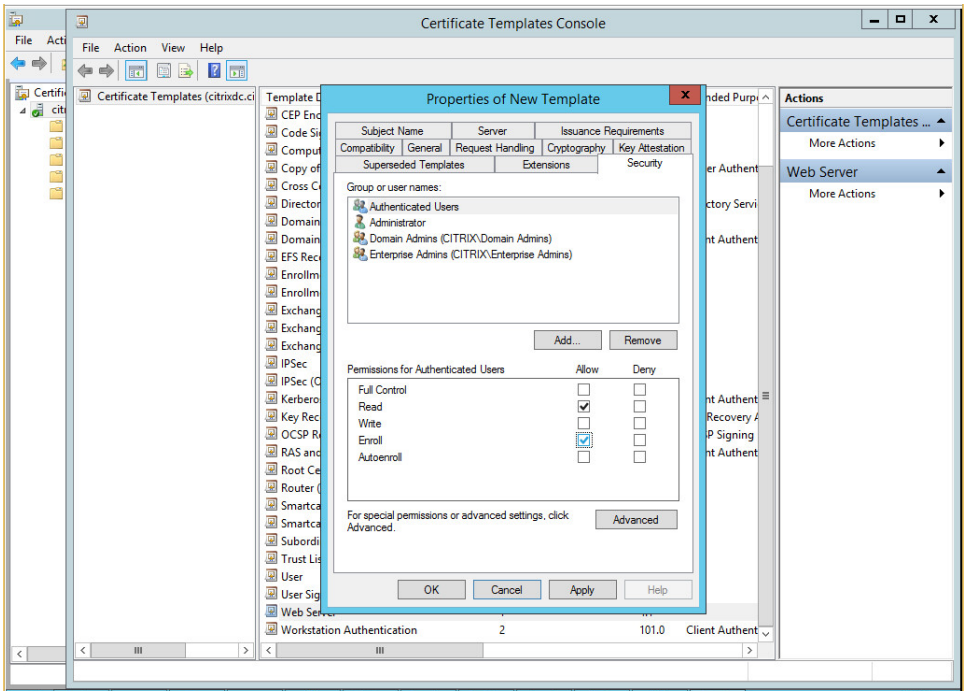
The following are listed in the **Description of Key Usage**.

Figure 134: Description of Key Usage



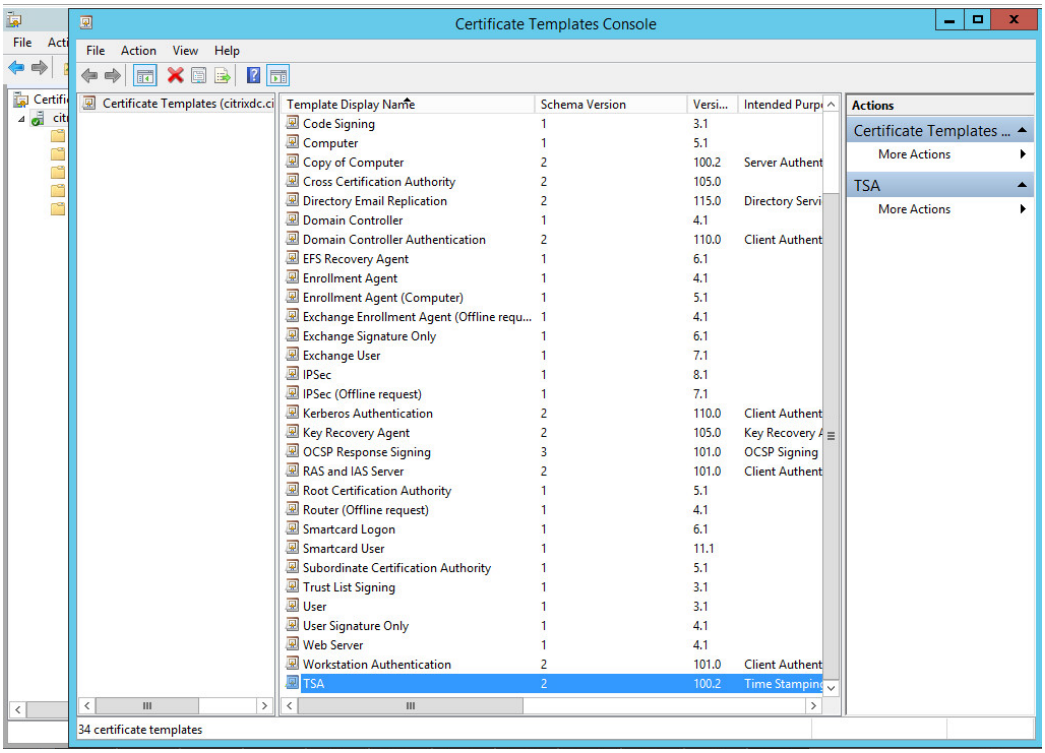
- On the *Security* tab, select **Authenticated Users**, and set **Enroll** to **Allow**.

Figure 135: Configuring permissions for the template



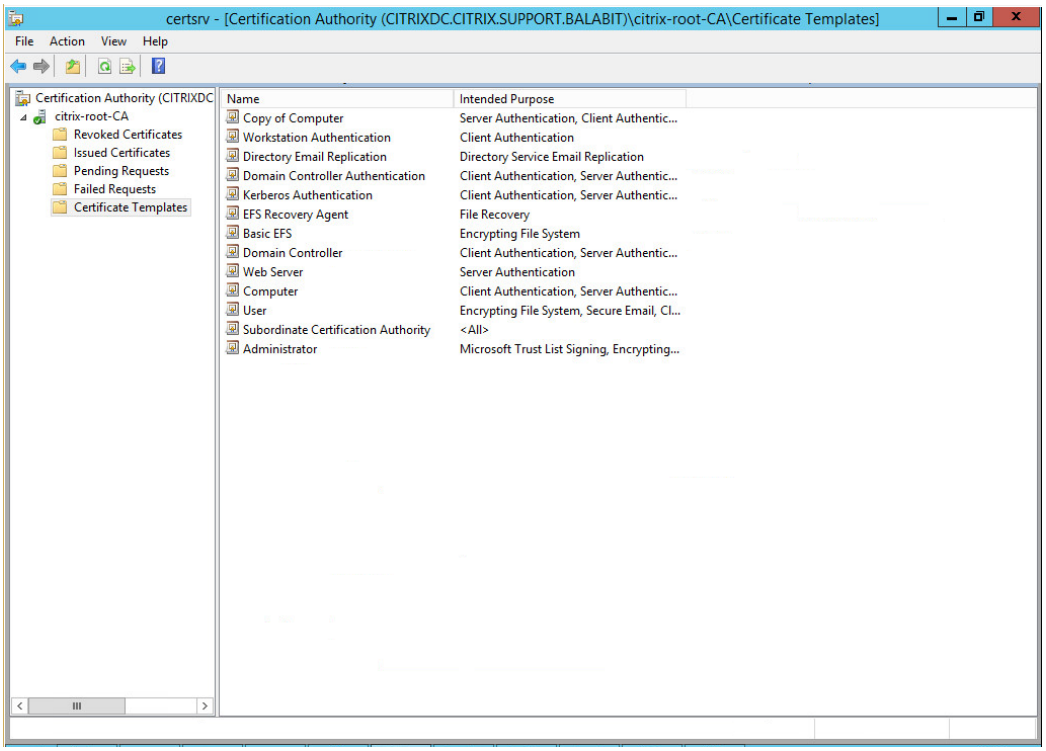
e. Click **Apply**. Click **OK**. The new TSA template is now displayed in the list of templates.

Figure 136: The new TSA template is now displayed in the list of templates



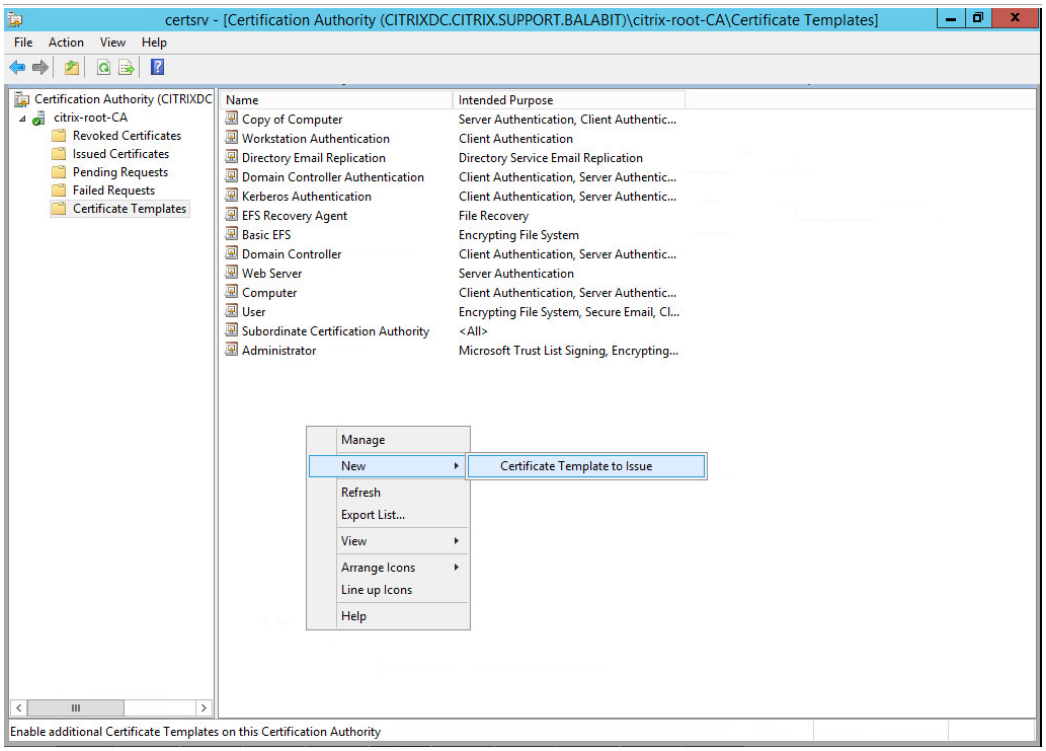
f. Close this window and return to the Certification Authority main screen, and select the **Certificate Templates** folder.

Figure 137: Certificate Templates



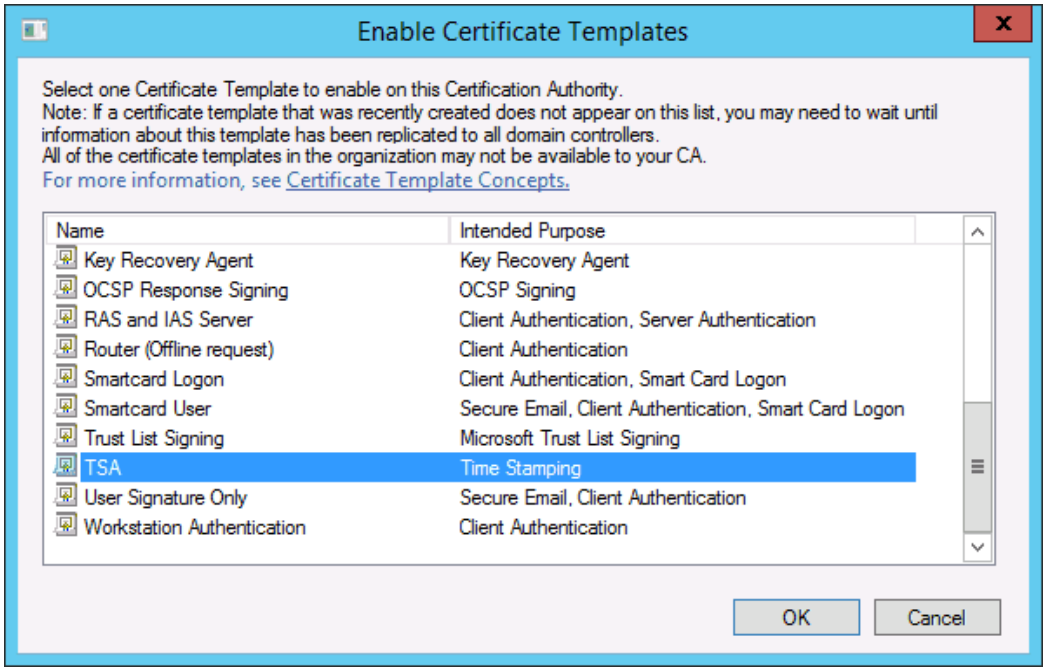
Right-click under the list, and choose **New > Certificate Template to Issue**.

Figure 138: Certificate Template to Issue



The *Enable Certificate Templates* window is displayed.

Figure 139: Enable the new template



g. Select the TSA certificate template, and choose **OK**. Close this window.

h. Open the command line, and issue the following command:

   certreq -submit -attrib "CertificateTemplate:TSA" <CSR>

   Replace <CSR> with the full path of the CSR created earlier (in the second step).

i. The *Certification Authority List* is displayed. Select the CA.

j. The *Save Certificate* window is displayed. Choose an output folder.

   The certificate is generated to the specified folder.

4. In SPS, navigate to **Basic Settings > Management > SSL certificates**.
5. Click next to **TSA X.509 certificate**, browse for the previously generated certificate, and click **Upload**.
6. Click next to **TSA private key**, browse for the previously generated key, and click **Upload**.

NOTE:

If the root CA (the **CA X.509 certificate** field under **Basic Settings > Management > SSL certificates**) that is used for other certificates on SPS is different from the CA that was used to sign the TSA certificate, a warning is displayed. In this scenario, ignore this warning.

General connection settings

Connections determine if a server can be accessed from a particular client. The policies used in the connection definition can restrict the availability of the *connection* based on the username, time, authentication method, and so on. Channel policies (see Creating and editing channel policies on page 448) determine if a particular channel can be used within an already established connection. The policies used in the channel policy can restrict the availability of the *channel* based on the server and the client IP address, username, and so on. The types of policies available in a connection depend on the protocol (SSH, RDP, and so on) enabled in the connection.

One Identity Safeguard for Privileged Sessions (SPS) compares the connection policies to the parameters of the connection request one-by-one, starting with the first policy in the policy list. The first connection policy completely matching the connection request is applied to the connection.

This section describes how to configure connections, and details the general configuration options and policies that apply to every type of connection that SPS can control: HTTP, ICA, RDP, SSH, Telnet, and VNC. For a detailed list of supported protocol versions, see Supported protocols and client applications on page 38.

Protocol-specific configuration options are described in their respective sections: HTTP-specific settings on page 490, ICA-specific settings on page 506, RDP-specific settings on page 515, SSH-specific settings on page 545, Telnet-specific settings on page 570, and VNC-specific settings on page 584.

Configuring connections

The following describes how to configure connections.

NOTE:

Avoid using the IP address configured for administrator or user login on One Identity Safeguard for Privileged Sessions (SPS) when configuring HTTP or SSH connections.

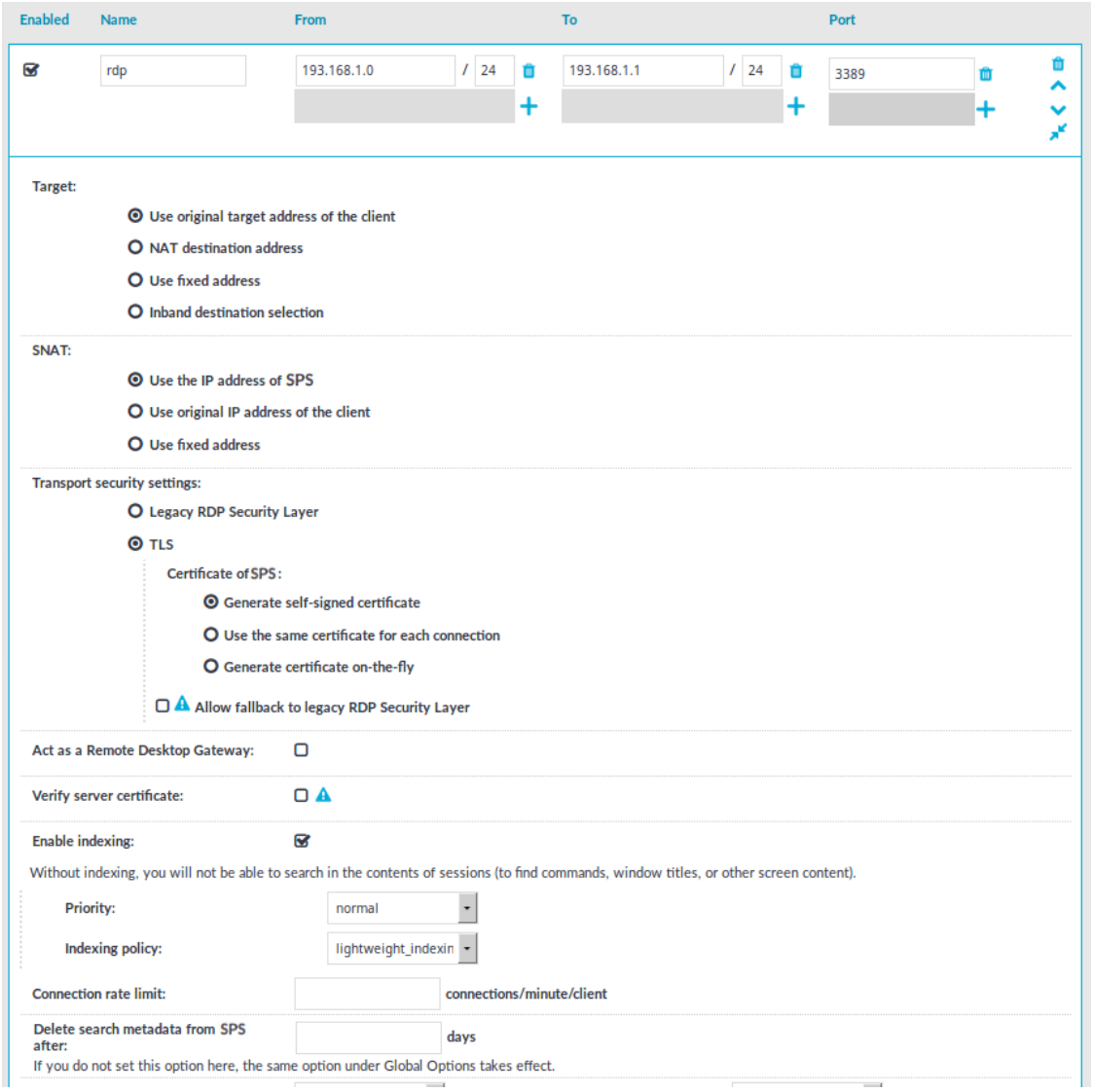
To configure connections

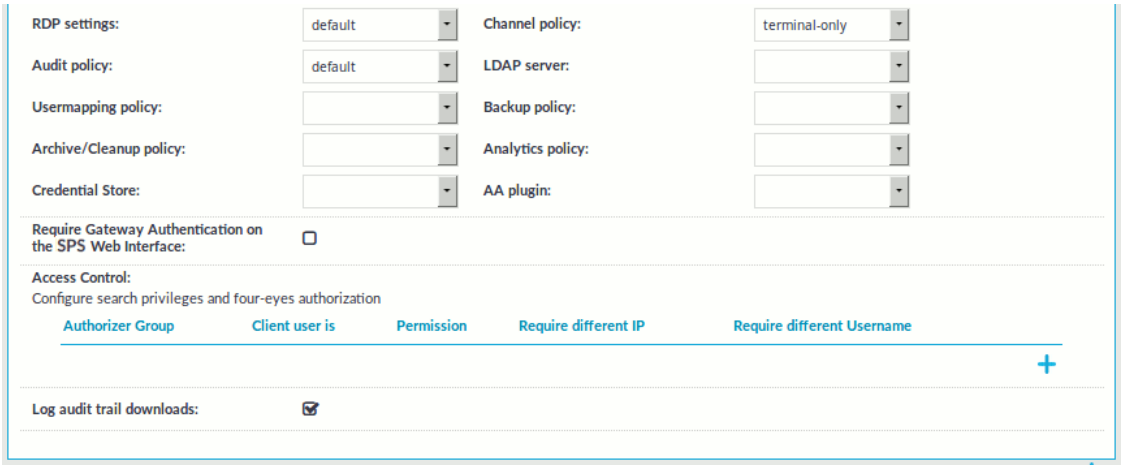
1. Select the type of connection from the main menu.
   - To configure a HTTP connection, select **HTTP Control > Connections**.
   - To configure an ICA connection, select **ICA Control > Connections**.
   - To configure a Remote Desktop connection, select **RDP Control > Connections**.
   - To configure a Secure Shell connection, select **SSH Control > Connections**.
   - To configure a Telnet connection, select **Telnet Control > Connections**.
   - To configure a VNC connection, select **VNC Control > Connections**.
2. Click + to define a new connection and enter a name that will identify the connection (for example `admin_mainserver`).

   TIP:

   It is recommended to use descriptive names that give information about the connection, for example refer to the name of the accessible server, the allowed clients, and so on.

Figure 140: <Protocol name> Control > Connections — Configuring connections





3. Enter the IP address of the client that will be permitted to access the server into the **From** field. Click + to list additional clients.

   You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).

   You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:

   - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.
   - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.

4. Enter the IP address that the clients will request into the **To** field.

   You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).

   You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:

   - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.
   - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.
   - In non-transparent mode, enter the IP address of a SPS logical interface.

     For more information on setting up logical network interfaces on SPS, see Managing logical interfaces on page 120.
   - In transparent mode, enter the IP address of the protected server.

   Click + to add additional IP addresses.

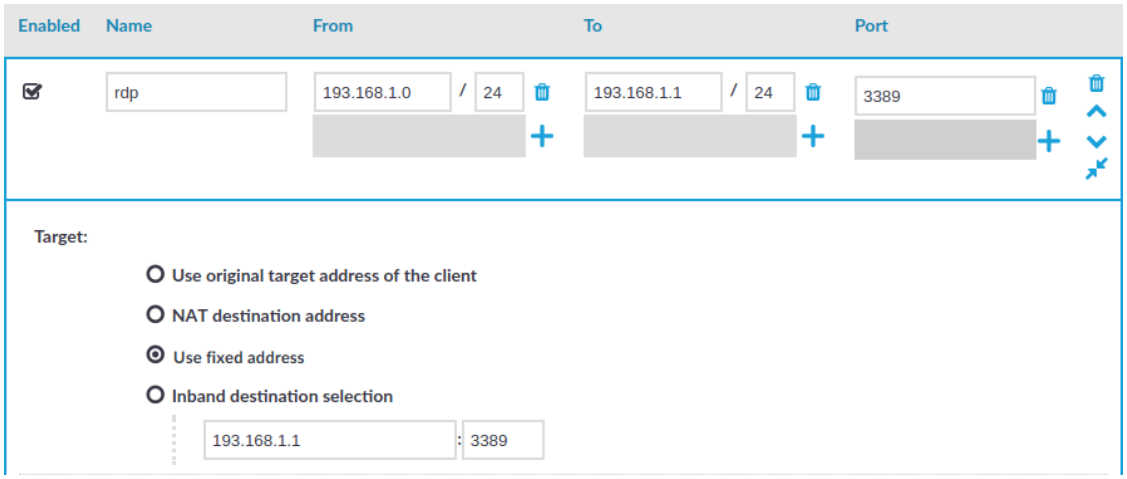
5. If the clients use a custom port to address the server instead of the default port used by the protocol, enter the port number that the clients will request into the **Port** field. Click + to list additional port numbers.

   NOTE:

   SPS can handle a maximum of 15 unique ports per connection policy. If you wish to specify more than 15 custom ports, create additional connection policies.

6. *Non-transparent mode*: Enter the IP address and port number of the target server into the **Target** field. SPS will connect all incoming client-side connections to this server. For details on organizing connections in non-transparent mode, see Organizing connections in non-transparent mode on page 835.

   Figure 141: <Protocol name> Control > Connections — Configuring non-transparent connections



7. Configure advanced settings if needed, like network address translation, channel policy, gateway authentication, various policies, or other settings.
8. Click Commit to save the connection.

   TIP:

   To temporarily disable a connection, deselect the checkbox before the name of the connection.

9. If needed, reorder the list of the connection policies. You can move connection policies by clicking the ^ and v buttons.

   One Identity Safeguard for Privileged Sessions (SPS) compares the connection policies to the parameters of the connection request one-by-one, starting with the first policy in the policy list. The first connection policy completely matching the connection request is applied to the connection.

10. Depending on your needs and environment, you may want to set further settings for your connections.
    - To modify the destination or source addresses of the connections, see Modifying the destination address on page 441 and Modifying the source address on page 446.
    - Select a **Backup Policy** and an **Archiving Policy** for the audit trails and indexes of the connection.

      You can find more information on creating backup and archive policies in Data and configuration backups on page 146 and Archiving and cleanup on page 159.

      If you have indexed trails, the index itself is also archived:

      When using the **Indexer service**: Every 30 days, unless the **Backup & Archive/Cleanup > Archive/Cleanup policies > Delete data from SPS after** is configured to occur less frequently (more than 30 days). For example, if the **Delete data from SPS after** is 60 days, the index will be archived every 60 days. The content of the archived index will be the content that was available X days before the archival date, where X is the number in the **Delete data from SPS after** field.

      CAUTION:

      Hazard of data loss Make sure you also backup your data besides archiving (for details, see Data and configuration backups on page 146). If a system crash occurs, you can lose up to 30 days of index, since the index is only archived in every 30 days.

      NOTE:

      The backup and archive policies set for the connection operate only on the audit trails and indexes of the connection. General data about the connections that is displayed on the **Search** page is archived and backed up as part of the system-backup process of SPS.
    - If you want to timestamp, encrypt, or sign the audit trails, configure an **Audit Policy** to suit your needs. For details, see Audit policies on page 465.

      CAUTION:

      In RDP connections, if the client uses the Windows login screen to authenticate on the server, the password of the client is visible in the audit trail. To avoid displaying the password when replaying the audit trail, you are recommended to encrypt the upstream traffic in the audit trail using a separate certificate from the downstream traffic. For details, see "Encrypting audit trails" in the Administration Guide.

- To require the users to authenticate themselves not only on the target server, but on SPS as well, see Configuring gateway authentication on page 742.
- To require four-eyes authorization on the connections, with the possibility of an auditor monitoring the connection in real-time, see Configuring four-eyes authorization on page 752.
- In the case of certain connections and scenarios (for example SSH authentication, gateway authentication, Network Level Authentication (NLA) connections), SPS can authenticate the user to an LDAP database, or retrieve the group memberships of the user. To use these features, select an **LDAP Server**. For details, see Authenticating users to an LDAP server on page 459.

  NOTE:

  To display the usergroups that can access a specific Connection Policy, open the Connection Policy, then select **Show connection permissions > Show** on the Connections page.

- To limit the number of new connection requests accepted from a single client IP address per minute, enter the maximal number of accepted connections into the **Connection rate limit** field.

NOTE:

Protocol-specific configuration options are described in their respective sections: HTTP-specific settings on page 490, ICA-specific settings on page 506, RDP-specific settings on page 515, SSH-specific settings on page 545, Telnet-specific settings on page 570, and VNC-specific settings on page 584.

11. If your clients and servers support it, configure the connection to use strong encryption.
    - For HTTP connections, see Enabling TLS encryption in HTTP on page 498.
    - For Citrix ICA connections, use the following scenario: Client - Broker - original secure gateway - Secure Ticket Authority (STA) - SPS - Server.
    - For RDP connections, see Enabling TLS-encryption for RDP connections on page 529.
    - For SSH connections, see Creating and editing protocol-level SSH settings on page 564.
    - For Telnet connections, see Enabling TLS-encryption for Telnet connections on page 571.
    - For VNC connections, see Enabling TLS-encryption for VNC connections on page 584.
12. For graphical connections, adjust the settings of your servers for optimal performance:
    - **CAUTION:**

      For optimal performance and text recognition in graphical

protocols, disable antialiasing on your servers. Antialiased text in the audit trails of RDP, VNC, and X11 connections is not recognized by the OCR engine of the Audit Player. The indexer service recognizes antialiased text, but its accuracy depends on the exact antialiasing settings. Disable antialiasing in order to properly index the trails of these connections.

Note that antialiasing is enabled by default on Windows Vista and newer. Antialiasing is also called font smoothing. ClearType is an antialiasing technology used on Microsoft Windows, and should be disabled for optimal performance.

- When processing RDP connections, SPS attempts to extract the username from the connection. To ensure that your users can access the target servers only when their username is recorded, see Usernames in RDP connections on page 539.

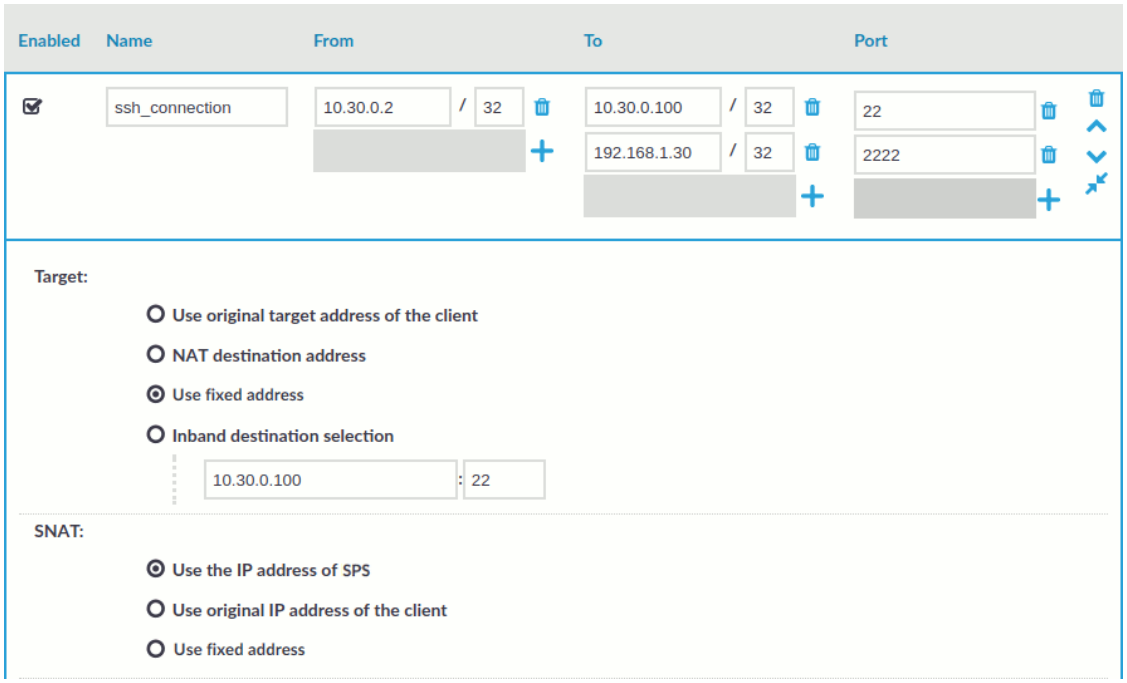
Modifying the destination address

The destination address is the address of the server where the clients finally connect to.

To modify the destination address of a connection

1. Navigate to the **Connections** tab storing the connection and click to display the details of the connection.

Figure 142: <Protocol name> Control > Connections — Configuring connections



2. The **Target** section allows you to configure Network Address Translation (NAT) on the server side of One Identity Safeguard for Privileged Sessions (SPS). Destination NAT determines the target IP address of the server-side connection. Set the destination address as required. The following options are available:

   NOTE:

   It is not possible to direct the traffic to the IP addresses belonging to SPS.

   - **Use the original target address of the client**: Connect to the IP address targeted by the client. This is the default behavior in transparent mode. This option is not available in non-transparent mode. For HTTP connections, you can use the **Use the original target address of the client** option only when the **Act as HTTP proxy** option is disabled.
   - **NAT destination address**: Perform a network address translation on the target address. Enter the target address in IP address/Prefix format.

     You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:

     - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.

- If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.

- **Use fixed address**: Enter the IP address and port number of the server. The connection will connect always to this address, redirecting the clients to the server.

  You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:

  - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.
  - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.

- **Inband destination selection**: Extract the address of the server from the username. Note that for HTTP connections, you can use the **Inband destination selection** option only when the **Act as HTTP proxy** option is enabled. For details, see Configuring inband destination selection on page 443.

3. Click Commit.

Configuring inband destination selection

With inband destination selection, you can create a single connection policy and allow users to access any server by including the name of the target server in their username (for example, `ssh username@targetserver@scb_address`, or `username%@targetserver%scb_address`).

Prerequisites

- Inband destination selection is not available for Virtual Networking (VNC).

  NOTE:

  When using inband destination selection and TN3270 pattern sets in a connection, only destinations that are consistent with the specified pattern set will work.

- To use inband destination selection in HTTP connections, you must enable the **Act as HTTP proxy** option. For details, see Enabling One Identity Safeguard for Privileged Sessions (SPS) to act as a HTTP proxy on page 495.
- To use inband destination selection with RDP connections, it is recommended to use SPS as a Remote Desktop Gateway (or RD Gateway). For details, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on

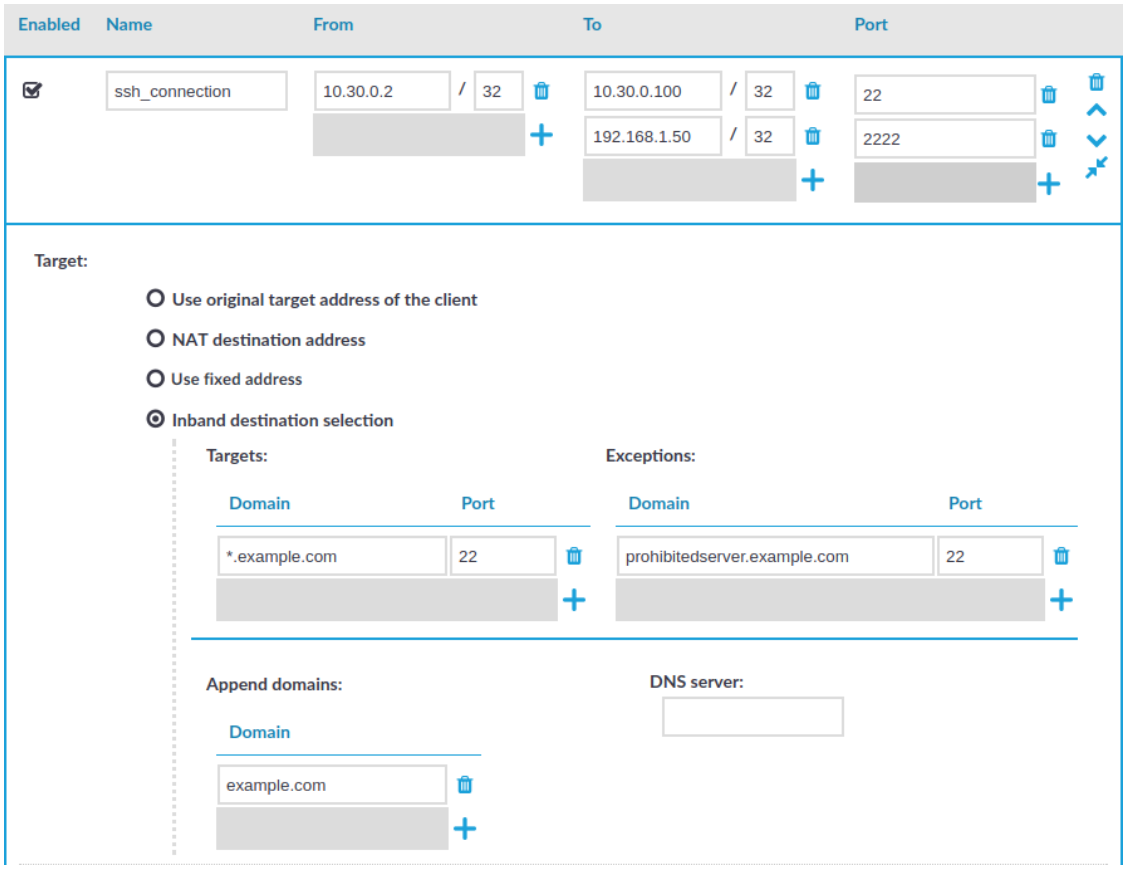
page 531.

- To use inband destination selection with RDP connections without using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway (or RD Gateway), you must use SSL-encrypted RDP connections (see Enabling TLS-encryption for RDP connections on page 529).
- For details on setting the clients to use inband destination selection in SSH connections, see Using inband destination selection in SSH connections on page 837.
- For details on setting the clients to use inband destination selection in Telnet connections, see Inband destination selection in Telnet connections on page 580.

To configure a Connection Policy to extract the address of the server from the username

1. Navigate to the Connection policy you want to modify, for example, to **SSH Control > Connections**.
2. Select **Inband destination selection**.

Figure 143: <Protocol name> Control > Connections — Configuring inband destination selection



3. *Optional Step*: Enter the IP address or the hostname of the domain name server used to resolve the address of the target server into the **DNS Server** field.

   If you do not set the **DNS Server** field, SPS will use the global DNS server (set on the **Basic Settings > Networking** page) to resolve the hostnames in this connection.

4. *Optional Step*: Configure domain names and CNAME records.

   If the clients do not include the domain name when addressing the server (for example they use `username@server` instead of `username@server.example.com`, or `username%server` for RDP connections), SPS can automatically add domain information (for example `example.com`). Enter the domain name to add into the **Append domain** field.

   SPS can also resolve CNAME records.

   To enter more domain names (for example because connections extend through subnets), click +. In case of more domain names in the **Append domain** field, SPS appends the first domain name in the list that the target can be resolved with.

5. Enter the addresses of the servers that the users are permitted to access into the **Targets** field. Note the following points:

   - Use the IP address/prefix (for example `192.168.2.16/32`, or `10.10.0.0/16`) format. Alternatively, you can use the FQDN of the server. To permit access to any server, enter `*`.
   - For FQDN, you can use the `*` and `?` wildcard characters.

     CAUTION:

     If only the hostname of the server is listed and the client targets the server using its IP address, SPS refuses the connection.

   - If the clients target the server using its IP address, include the IP address of the server in the **Targets > Domain** list. This is required because SPS resolves the hostnames to IP addresses, but does not reverse-resolve IP addresses to hostnames.
   - If the clients target the server using its hostname, then the `hostname-from-the-client-request + the-value-of-the-Append-domain-option` must appear in the **Targets > Domain** list. Alternatively, you must include the IP address of the `hostname-from-the-client-request + the-value-of-the-Append-domain-option` host.

Example: Hostnames and inband destination selection

For example, you have set **Append domain** to `example.com`, and your clients use the `username%servername` request, then you must include either the `servername.example.com` host or its IP address in the **Targets > Domain** list.

6. If the clients can access only a specified port on the server, enter it into the **Port** field. If the **Port** is not set, the clients may access any port on the server.
7. If there are any servers that the users cannot target using inband destination selection, add them to the **Exceptions** field.
8. To use inband destination selection with RDP connections without using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway (or RD Gateway), you must use SSL-encrypted RDP connections (see Enabling TLS-encryption for RDP connections on page 529).
9. Click 

.

Expected result

The connection policy will extract the address of the destination server from the protocol information.

NOTE:

For examples on using inband destination selection to establish an SSH connection, including scenarios where non-standard ports or gateway authentication is used, see Using inband destination selection in SSH connections on page 837.

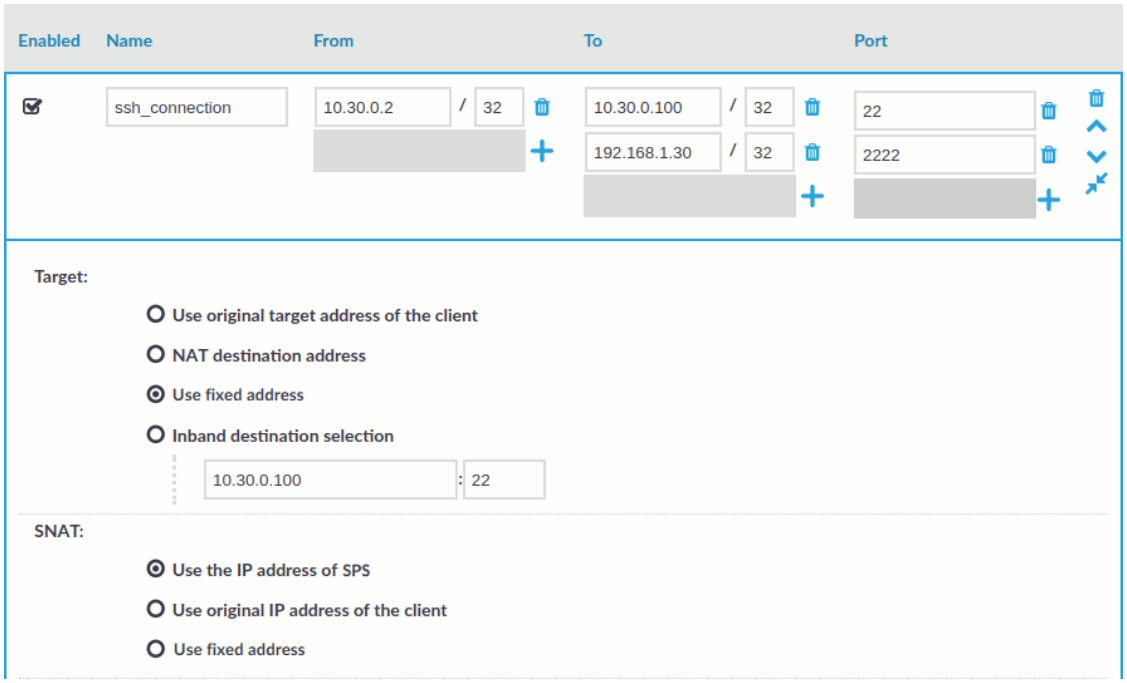
Modifying the source address

The source address is the address that One Identity Safeguard for Privileged Sessions (SPS) uses to connect the server. The server sees this address as the source of the connection.

To modify the source address of a connection

1. Navigate to the **Connections** tab storing the connection and click ⤢ to display the details of the connection.

Figure 144: <Protocol name> Control > Connections — Configuring connections



2. The **SNAT** section allows you to configure Source Network Address Translation (SNAT) on the server side of SPS. SNAT determines the IP address SPS uses in the server-side connection. The target server will see the connection coming from this address. The following options are available:
   - **Use the IP address of a SPS logical interface**: Server-side connections will originate from SPS's logical network interface. This is the default behavior of the connection.
   - **Use the original IP address of the client**: Server-side connections will originate from the client's IP address, as seen by SPS.
   - **Use fixed address**: Enter the IP address that will be used as the source address in server-side connections.

     You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:
     - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.
     - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.

CAUTION:

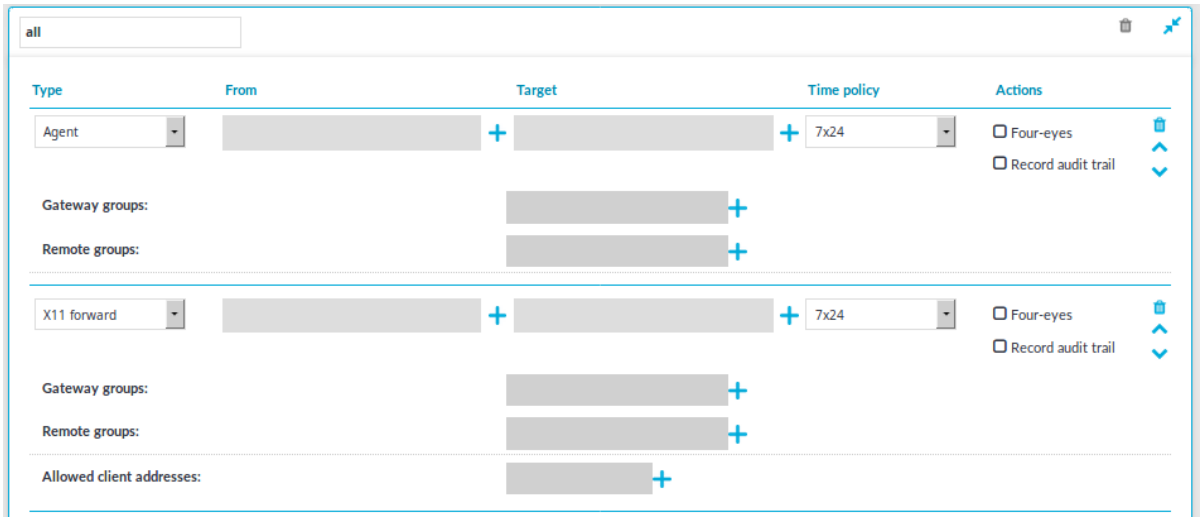
Do not forget to properly configure routers and other network devices when using the Use fixed address option: messages sent by the server to this address must reach SPS.

3. Click Commit.

Creating and editing channel policies

The Channel policy lists the channels (for example, terminal session and SCP in SSH, or Drawing and Clipboard in RDP) that can be used in the connection, and also determines if the channel is audited or not. The Channel policy can also restrict access to each channel based on the IP address of the client or the server, a user list, user group, or a time policy. For example, all clients may access the servers defined in a connection via SSH terminal, but the channel policy may restrict SCP access only to a single client. The policies set in the channel policy are checked when the user attempts to open a particular channel type in the connection.

Figure 145: <Protocol name> Control > Channel Policies — Configuring channel policies



To create a new channel policy or edit an existing one

1. Channel policies are configured individually for every protocol. Navigate to the **Channel Policies** tab of the respective protocol (for example, **SSH Control > Channel Policies**) and click + to create a new channel policy. Enter a name for the policy (for example, `shell_and_backup`).
2. Click + to add a new channel.

3. Select the channel to be enabled in the connection from the **Type** field. All restrictions set in the following steps will be effective on this channel type. The available channels are different for every protocol. For their descriptions, see the following sections:
   - The HTTP protocol has only one channel type with no special configuration options.
   - Supported ICA channel types on page 507 for the Independent Computing Architecture protocol.
   - Supported RDP channel types on page 516 for the Remote Desktop Protocol.
   - Supported SSH channel types on page 547 for the Secure Shell protocol
   - The Telnet protocol has only one channel type with no special configuration options.
   - The VNC protocol has only one channel type with no special configuration options.
4. To restrict the availability of the channel only to certain clients, click + in the **From** field and enter the IP address of the client allowed to use this type of the channel. Repeat this step until all required client IPs are listed.

   You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:
   - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.
   - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.
5. To restrict the availability of the channel only to certain servers, click + in the **Target** field and enter the IP address of the server allowed to use this type of the channel. Repeat this step until all required server IPs are listed.

   > NOTE:
   >
   > Use the real IP address of the server, which may be different from the one addressed by the clients, specified in the **Target** field of the connection policy.

   You can also enter a hostname instead of the IP address, and One Identity Safeguard for Privileged Sessions (SPS) automatically resolves the hostname to IP address. Note the following limitations:
   - SPS uses the Domain Name Servers set **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server** fields to resolve the hostnames.
   - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.

6. To restrict the availability of the channel only to certain users, click + in the **Remote Group** field and enter the name of the user group allowed to use this type of the channel. Repeat this step until all permitted groups are listed.

   To restrict the availability of the channel when using gateway authentication, click + in the **Gateway Group** field and enter the name of the user group allowed to use this type of the channel. Repeat this step until all permitted groups are listed.

   You may list local user lists as defined in Creating and editing user lists on page 457, or LDAP groups (for details on accessing LDAP servers from SPS, see Authenticating users to an LDAP server on page 459). Note the following behavior of SPS:

   - If you list multiple groups, members of any of the groups can access the channel.

     NOTE:

     When listing both a whitelist and blacklist in the **Remote Group** section and a username appears on both lists, the user will be able to access the channel.

   - If you do not list any groups, anyone can access the channel.

     NOTE:

     When the channel opens, there are certain cases when the remote group is not known yet. For example, in case of an RDP or ICA login screen, the drawing channel has to be opened first to properly display the logon screen. Only those channel rules will apply, where the **Remote group** field is empty. In case of network level authentication, all required information is present already so this limitation does not apply.

   - If a local user list and an LDAP group has the same name and the LDAP server is configured in the connection that uses this channel policy, both the members of the LDAP group and the members of the local user list can access the channel.

   NOTE:

   User lists and LDAP support is currently available only for the SSH and RDP protocols. For other protocols, see Configuring gateway authentication on page 742.

7. Select a time policy to narrow the availability of the channel. If the time policy of the channel policy is set to 7x24, the channel is always available. For details, see Configuring time policies on page 457.
8. Some channel types require additional parameters, for example port forwarding in SSH needs the IP addresses and ports of the source and destination machines. Click + in the **Details** field and enter the required parameters. For a list of parameters used by the different channels, see Supported SSH channel types on page 547 and Supported RDP channel types on page 516.
9. Select the **Record audit trail** option to record the activities of the channel into audit trails. Typically large file-transfers (for example system backups, SFTP channels) are

not audited because they result in very large audit trails. Check regularly the free hard disk space available on SPS if you do audit such channels. You can also receive alerts about disk space fill-up if you set these. For details, see Preventing disk space fill-up on page 139 and System related traps on page 140.

10. Select the **4 eyes** option to require four-eyes authorization to access the channel. For details, see Configuring four-eyes authorization on page 752.
11. Repeat Steps 2-10 to add other channels to the policy.

    NOTE:

    The order of the rules matters. The first matching rule will be applied to the connection. Also, note that you can add the same channel type more than once, to fine-tune the policy.

12. Click 

to save the list.

Real-time content monitoring with Content Policies

You can monitor the traffic of certain connections in real time, and execute various actions if a certain pattern (for example, a particular command or text) appears in the command line or on the screen, or if a window with a particular title appears in a graphical protocol. Since content-monitoring is performed real-time, One Identity Safeguard for Privileged Sessions (SPS) can prevent harmful commands from being executed on your servers. SPS can also detect numbers that might be credit card numbers. The patterns to find can be defined as regular expressions. In case of ICA, RDP, and VNC connections, SPS can detect window title content.

The following actions can be performed:

- Log the event in the system logs.
- Immediately terminate the connection.
- Send an e-mail or SNMP alerts about the event.
- Store the event in the connection database of SPS.

SPS currently supports content monitoring in SSH session-shell connections, Telnet connections, RDP and Citrix ICA Drawing channels, and in VNC connections.

NOTE:

Command, credit card and window detection algorithms use heuristics. In certain (rare) situations, they might not match the configured content. In such cases, contact our Support Team to help analyze the problem.

Real-time content monitoring in graphical protocols is not supported for Arabic and CJK languages.

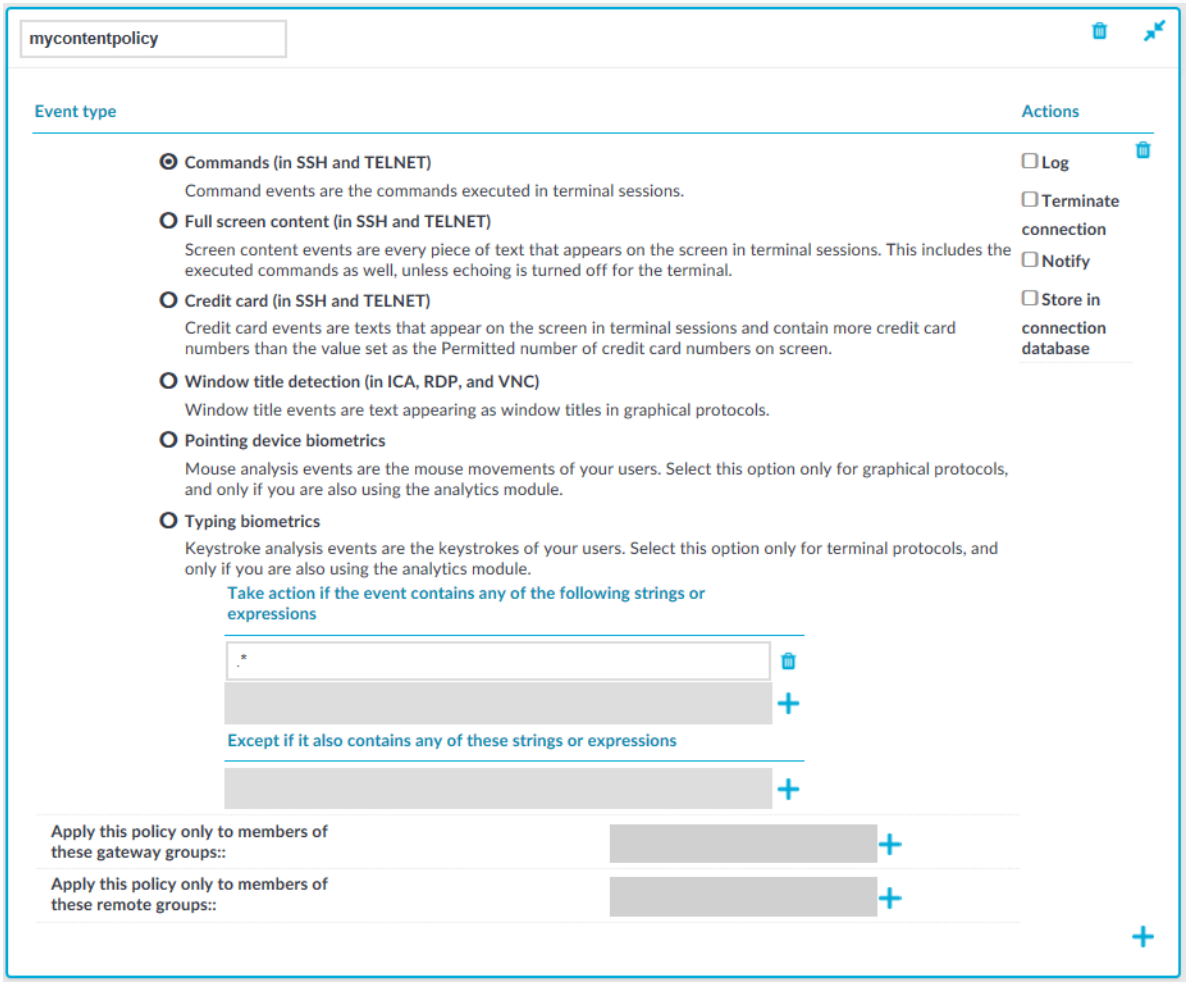
Creating a new content policy

The following describes how to create a new content policy that performs an action if a predefined content appears in a connection.

NOTE:

Using content policies significantly slows down connections (approximately 5 times slower), and can also cause performance problems when using the indexer service.

Figure 146: Policies > Content Policies — Content policies



To create a new content policy that performs an action if a predefined content appears in a connection

1. Navigate to **Policies > Content Policies**, click + and enter a name for the policy.
2. Select the type of event that you want to monitor:

- **Commands**: The commands executed in the `session-shell` channel of SSH connections, or in Telnet connections.

  CAUTION:

  During indexing, if a separate certificate is used to encrypt the upstream traffic, command detection works only if the upstream key is accessible on the machine running the indexer.

  NOTE:

  Command detection is case-insensitive.

- **Screen content**: Every text that appears on the screen. For example, every text that is displayed in the terminal of SSH or Telnet connections. This includes the executed commands as well, unless echoing is turned off for the terminal.
- **Credit card**: Process every text that appears on the screen and attempt to detect credit card numbers in SSH or Telnet connections. One Identity Safeguard for Privileged Sessions (SPS) performs an action if the number of detected credit card numbers exceeds the value set as **Permitted number of credit card numbers**.

  Credit card number detection is based on the Luhn algorithm and lists of known credit card number prefixes.
- **Window title detection**: Text appearing as window titles in case of RDP, Citrix ICA, and VNC connections. Note the following points.

  ### Supported themes

  - Windows Classic at 96 and 120 DPI
  - Windows 2012 at 96 and 120 DPI (metro)
  - Windows 7 Normal at 96 and 120 DPI (not aero)
  - Windows 8 at 96 and 120 DPI (metro)

  ### Limitations

  - Windows 10, Windows 2016 Server, and Windows 2019 Server themes are not supported
  - Windows Aero themes are not supported.
  - Windows that do not have an X (close window) button in the top-right corner (or it is not visible) are not detected.
  - Use window title detection for sessions that use a single monitor. The feature works in multi-monitor environments as well, but becomes very slow, therefore it is not recommended.

  The configuration JSON file contains the most common window title color schemes.

NOTE:

Window title detection is case-insensitive.

NOTE:

Do not adjust or modify the following settings unless you know exactly what you are doing. Misconfiguring them will severely decrease the performance of SPS.

- If a special color is used, open `/opt/scb/etc/window-title-default` on the server, and add the color scheme in RGB. In case of a single color, enter `"to": null`. After adding a new color, temporarily disable all traffic going through SPS. Navigate to **Basic Settings > System > Traffic control** and click **Stop** in the **All services** field. Login to SPS as `root` locally (or remotely using SSH) to access the Console menu. Select **Shells > Core Shell**, and issue the **systemctl restart zorp-core.service** command.
- The minimum and maximum height and the minimum width of the window title are determined in pixels, as `"minheight"`, `"maxheight"` and `"minwidth"`.

- **Pointing device biometrics**: Select this option only for graphical protocols, and only if you are also using One Identity Safeguard for Privileged Analytics (SPA). SPA can analyze mouse movement patterns of your users as a biometric identity verification method to protect against account theft. For details, see the One Identity Safeguard for Privileged Analytics website.
- **Typing biometrics**: Select this option only for terminal-based protocols, and only if you are also using One Identity Safeguard for Privileged Analytics (SPA). SPA can analyze the typing patterns of your users as a biometric identity verification method to protect against account theft. For details, see the One Identity Safeguard for Privileged Analytics website.

3. Select **Take action if the event contains any of the following strings or expressions**, click + and enter a string or regular expression. SPS will perform an action if this expression is found in the connection, unless it is listed in the **Except if it also contains any of these strings or expressions** list. For example, SPS can terminate the connection if the user issues the **rm -rf *** in an SSH connection. Repeat this step to add further expressions if needed.
   - Use Perl Compatible Regular Expressions (PCRE).
   - The following characters must be escaped using a backslash character: ' (single-quote). For example, instead of `.*'` use `.*\'`
   - SPS uses substring search to find the expression in the content. That is, SPS finds the expression even if there is more content before or after the matching part. For example, the `conf` pattern will match the following texts: `conf`, `configure`, `reconfigure`, `arcconf`, and so on.

- Using complicated regular expressions or using many regular expressions will affect the performance of SPS.
- If the multiple expressions are set, SPS processes them one after the other, and stops processing the content if the first match is found, even if other expressions would also match the content. Therefore, when using multiple expressions, start with the most specific one, and add general expressions afterward.

Example: Sample regular expressions for content policies

The following simple regular expressions are samples to demonstrate what kinds of events that can be detected using content policies.

- The `enable` command on Cisco devices: the user enters privileges mode.
- The `conf term` command on Cisco devices: the user configures the networking parameters of the device.
- The `sudo` and `su -` commands: the user enters privileged mode Linux and other UNIX platforms.

4. To add an exception to the **Take action if the event contains any of the following strings or expressions** rule, select **Except if it also contains any of these strings or expressions**, click + and enter a string or regular expression. SPS will not perform any action if this expression is found in the connection. For example, to permit the users to delete only the `/tmp` directory in an SSH connection, enter `rm -rf /tmp`. Repeat this step to add further expressions if needed.

Example: Sample content policies using Ignore rules

The following expressions can be used to perform an action if any SQL command is used in MySQL, except for the `select` and `help` commands:

- Into the **Take action if the event contains any of the following strings or expressions** expression, enter `mysql>.*`
- Add two **Except if it also contains any of these strings or Except if it also contains any of these strings or Except if it also contains any of these strings or expressions** expressions: `mysql> select.*` and `mysql> help.*`

5. Select the action to perform.
   - **Log**: Send a log message into the system logs. The log message includes the expression that matched the content. On log level 6, the message includes the matching content as well.

- **Terminate connection**: Immediately terminate the connection. When using the **Terminate connection** action for the **Command** event type, and a command matches an expression, the connection is terminated before the command is executed. When using the **Terminate connection** action, note the following points.
  - Select the **Log** or **Notify** action as well so that it is easy to find out why a connection was terminated.
  - If the connection is terminated by a content policy, the **Verdict** of the connection becomes `ACCEPT-TERMINATED`.
- **Notify**: Send an e-mail or SNMP alert about the event. To configure the alerts, navigate to **Basic Settings > Alerting & Monitoring** and set the required alerts for the **Real time audit event detected** (`scbAuditRealTime`) event.
- **Store in connection database**: Add the event to the SPS connection database. These events are displayed in the **Alerts** column of the **Search** page. If the column is not visible, click **Customize columns...**.

6. To apply the content policy only for users belonging to specific groups, select **Apply this policy only to members of these gateway groups** or **Apply this policy only to members of these remote groups**, + and specify the usergroups as needed. If **Apply this policy only to members of these gateway groups** or **Apply this policy only to members of these remote groups** is set, the content policy is applied only to connections of these usergroups.
7. To add a new rule to the policy, click + and repeat Steps 2-6.

   Note that if you have more than one rules in a policy, SPS evaluates them as follows.

   a. SPS evaluates the first (top) rule.

   b. If the rule contains **Apply this policy only to members of these gateway groups** or **Apply this policy only to members of these remote groups** restrictions, SPS checks if the current user belongs to any of the specified groups. If the groups do not match, SPS skips the rule.

   c. If the content matches any entry of the **Except if it also contains any of these strings or expressions** list, SPS skips the rule.

   d. If the content matches any entry of the **Take action if the event contains any of the following strings or expressions** list, SPS performs the action configured for the rule. Otherwise, SPS skips the rule.

   e. If the current rule did not match the content, SPS evaluates the next rule of the policy (if any).

8. Click Commit. A new content policy is created.
9. To use the content policy created in the previous steps, select the policy in the channel policy that is used to control the connections.

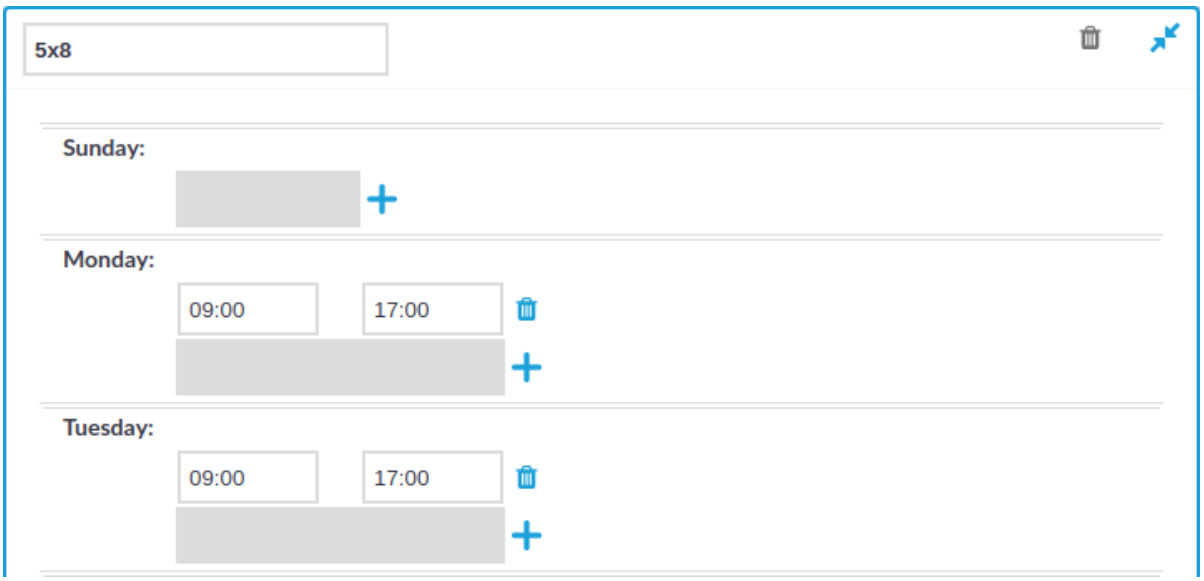
   NOTE:

   It is not required to enable auditing to use content policies.

Configuring time policies

The time policy determines the timeframe when the users are permitted to access a particular channel. By default, there is no time-based restriction, all channels are available 7x24.

Figure 147: Policies > Time Policies — Configuring time policies



To create a time policy or edit an existing one

1. Navigate to the **Time Policies** tab of the **Policies** menu item and click + to create a new time policy. Enter a name for the policy (for example `workhoursonly`).
2. Click ⤢ to display the days of the week and the allowed intervals.
3. Enter the intervals for each day when the users are allowed to access the connection. Use the `hh:mm` format (for example from 08:00 to 16:00).
4. To add multiple intervals for a day, click +.
5. Click Commit.
6. To actually restrict access to a connection or a channel based on the policy created in the previous steps:
   - Select this policy in the **Time Policy** field of the channel policy.
   - Click Commit.

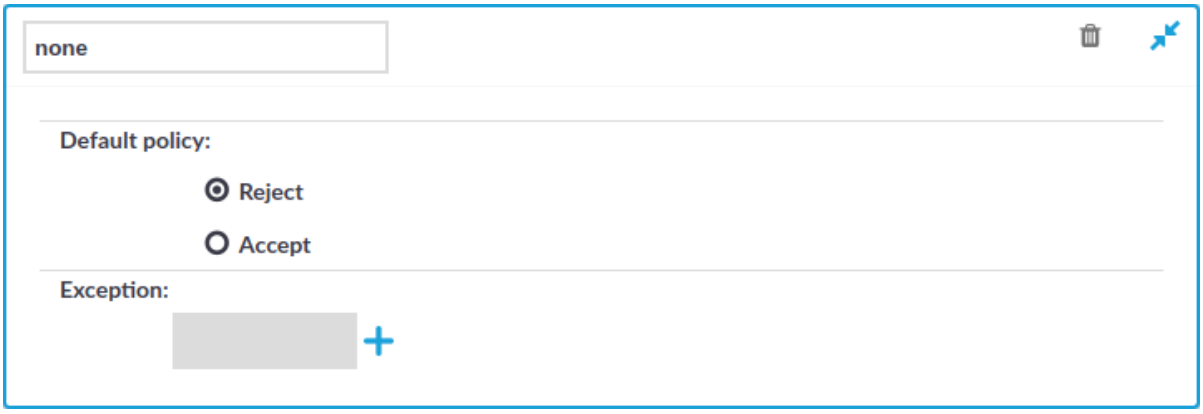
Creating and editing user lists

User lists are white- or blacklists of usernames that allow fine-control over who can access a connection or a channel.

CAUTION:

User Lists are white- or blacklists of usernames that determine who can access the server remotely. However, this cannot prevent a user from accessing the server from a local terminal.

Figure 148: Policies > User Lists — Configuring user lists



To create a new user list or edit an existing one

1. Navigate to the **User Lists** tab of the **Policies** menu and click + to create a new user list. Enter a name for the list **User List** field (for example `serveradmins`).

   CAUTION:

   Usernames, the names of user lists, and the names of usergroups are case sensitive.

2. Click ⤢ to display the list of users.
3. Select the default policy of the user list. Select **Reject** for a whitelist, that is, to allow access only to the members of the list. Select **Accept** for a blacklist, that is, to allow access to everyone except the members of the list.
4. Click + and enter a username into the displayed field. Repeat this step until all required usernames are listed.

   CAUTION:

   Usernames, the names of user lists, and the names of usergroups are case sensitive.

5. Click Commit to save the list.
6. To actually restrict access to a channel based on the user list created in the previous steps:

- Navigate to the **Channel Policies** tab of the type of connection you want to control and click ⤢ to display the details of the policy.
- Click + in the **Group** section to add a new group to the policy and enter the name of the group. Repeat this step to add other groups.

  CAUTION:

  Usernames, the names of user lists, and the names of usergroups are case sensitive.

  NOTE:

  When listing more groups, users of any of the listed groups can access the channel. For details, see Creating and editing channel policies on page 448.

  When listing both a whitelist and blacklist in the **Group** section and a username appears on both lists, the user will be able to access the channel.

- Click Commit.

Authenticating users to an LDAP server

You can use the LDAP policy to set the details of the LDAP server you wish to use to:

- authenticate gateway users (available in SSH and Telnet as Authentication Policy)
- query gateway groups (available for RDP, Telnet, SSH, and ICA)
- query remote groups (available for RDP, Telnet, SSH, ICA, and HTTP)

NOTE: This feature is not available for Virtual Network Computing (VNC) connections.

Prerequisites

Make sure that the response timeout of the LDAP/Active Directory server is at least 120 seconds.

NOTE:

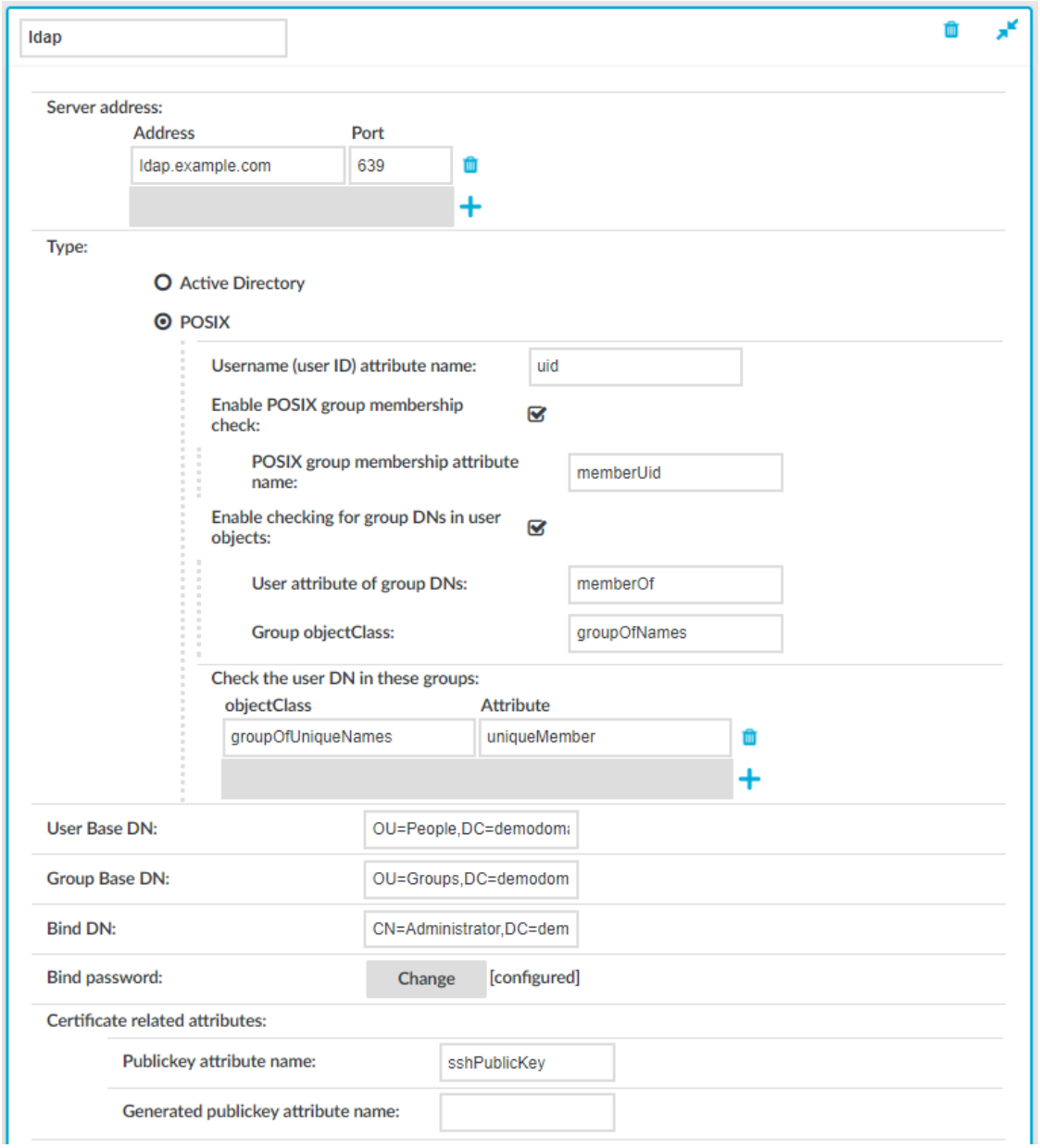
- In RDP (including RDG) connections, you can use the LDAP policy for group membership check only, you cannot use it as the authentication backend. However, you can use a trusted AD domain for authentication and LDAP for group membership check.

  In this case, LDAP will only use the username without the domain name to verify the group membership.
- One Identity Safeguard for Privileged Sessions (SPS) treats user and group names in a case insensitive manner if the matching rule for the attribute in question is case insensitive in the LDAP database.

To configure an LDAP policy for a connection

1. Navigate to **Policies > LDAP Servers** and click + to create a new LDAP policy.

Figure 149: Policies > LDAP Servers — Configuring LDAP Server policies



2. Enter a name for the policy (for example `ldapservers`).
3. Enter the IP address or hostname and port of the LDAP server into the **Server Address** field. If you want to encrypt the communication between SPS and the LDAP server, in case of TLS, enter 636 as the port number, or in case of STARTTLS, enter 389 as the port number.

   Use an IPv4 address.

To add multiple servers, click + and enter the address of the next server. If a server is unreachable, SPS will try to connect to the next server in the list in failover fashion.

> **CAUTION:**
>
> **If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example `ldap.example.com`) in the Server Address field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.**

4. Select the type of your LDAP server in the **Type** field. Select:

   - **Active Directory** to connect to Microsoft Active Directory servers.

     You can enable nested groups. Select **Enable AD group membership check**, then **Enable nested groups**.

     > **CAUTION:**
     >
     > **Nested groups can slow down the query and cause the connection to timeout if the LDAP tree is very large. In this case, disable the Enable nested groups option.**

     To also check group membership based on group DNs in a user attribute, select **Enable checking for group DNs in user objects** and enter the name of the user attribute, for example, `memberOf` in the **User attribute of group DNs** field.

     > **CAUTION:**
     >
     > **Using this option significantly slows down log on to the SPS web interface if you have too many groups.**
     >
     > **Only use this option if you have an LDAP schema where the user groups can only be determined from a user attribute that contains the group DNs.**

     To check for group membership based on user DNs in group attributes, use the **Check the user DN in these groups** options.

     For more information, see Active Directory LDAP backend¶.

   - **POSIX** to connect to servers that use the POSIX LDAP scheme.

     If your LDAP server uses a custom POSIX LDAP scheme, you might need to set which LDAP attributes store the username, or the attributes that set group memberships. For example, if your LDAP scheme does not use the `uid` attribute to store the usernames, set the **Username (user ID) attribute name** option.

     In addition to the primary group membership checking, you can allow checking for supplementary group memberships by selecting the **Enable POSIX group membership check** and specifying the **POSIX group membership attribute name** field.

To also check group membership based on group DNs in a user attribute, select **Enable checking for group DNs in user objects** and enter the name of the user attribute, for example, `memberOf` in the **User attribute of group DNs** field and objectClass, for example, `groupOfNames` in the **Group objectClass** field.

> **CAUTION:**
>
> **Using this option significantly slows down log on to the SPS web interface if you have too many groups.**
>
> **Only use this option if you have an LDAP schema where the user groups can only be determined from a user attribute that contains the group DNs.**

To check for group membership based on user DNs in group attributes, use the **Check the user DN in these groups** options.

For more information, see Posix LDAP backend¶.

For an overview about LDAP user and group resolution in SPS, see Overview.

5. In the **User Base DN** field, enter the name of the DN to be used as the base of queries regarding users (for example: `OU=People,DC=demodomain,DC=exampleinc`).

   > NOTE:
   >
   > You must fill in this field. It is OK to use the same value for **User Base DN** and **Group Base DN**.
   >
   > However, note that specifying a sufficiently narrow base for the LDAP subtrees where users and groups are stored can speed up LDAP operations.

6. In the **Group Base DN** field, enter the name of the DN to be used as the base of queries regarding groups (for example: `OU=Groups,DC=demodomain,DC=exampleinc`).

   > NOTE:
   >
   > You must fill in this field. It is OK to use the same value for **User Base DN** and **Group Base DN**.
   >
   > However, note that specifying a sufficiently narrow base for the LDAP subtrees where users and groups are stored can speed up LDAP operations.

7. In the **Bind DN** field, enter the Distinguished Name that SPS should use to bind to the LDAP directory (for example: `CN=Administrator,DC=demodomain,DC=exampleinc`).

   > NOTE:
   >
   > SPS accepts both pre-win2000-style and Win2003-style account names (User Principal Names), for example `administrator@example.com` is also accepted.

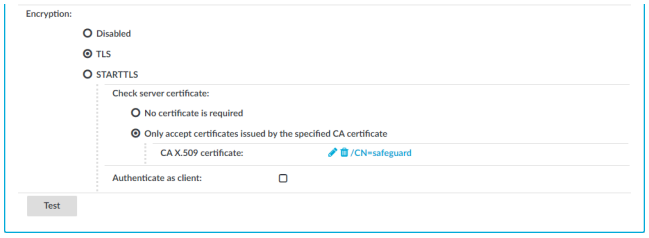
8. To configure or change the password to use when binding to the LDAP server, click **Change** and enter the password. Click **Update**. Click Commit.

NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: !"#$%&'()*+,-./:;<=>?@[\]^-`{|}

9. *Skip this step if you use passwords to authenticate the users.*
   - If you use public-key authentication and receive the public key of the users from the LDAP database, enter the name of the LDAP attribute that stores the public keys of the users into the **Publickey attribute name** field. For details on using public-key authentication with the LDAP database, see Configuring public-key authentication on One Identity Safeguard for Privileged Sessions (SPS) on page 831.
   - If you use X.509 certificate for authentication and receive the certificates of the users from the LDAP database, enter the name of the LDAP attribute that stores the certificates of the users into the **Certificate attribute name** field.
10. *Skip this step if you use passwords to authenticate the users.*
    - If you use public-key authentication and want SPS to generate server-side encryption keys on-the-fly and store them in a separate attribute on the LDAP server, enter the name of the attribute into the **Generated publickey attribute name** field.
    - If you use certificate authentication and want SPS to generate server-side certificates on-the-fly and store them in a separate attribute on the LDAP server, enter the name of the attribute into the **Generated certificate attribute name** field.
11. If you want to encrypt the communication between SPS and the LDAP server, in **Encryption**, select the **TLS** or the **STARTTLS** option and complete the following steps:

Figure 150: Policies > LDAP Servers — Configuring encryption



NOTE:

TLS-encrypted connection to Microsoft Active Directory is supported only on Windows 2003 Server and newer platforms. Windows 2000 Server is not supported.

- If you want SPS to verify the certificate of the server, select **Only accept certificates issued by the specified CA certificate** and click the icon in

the **CA X.509 certificate** field. A pop-up window is displayed.

Click **Browse**, select the certificate of the Certificate Authority (CA) that issued the certificate of the LDAP server, then click **Upload**. Alternatively, you can paste the certificate into the **Copy-paste** field and click **Set**.

SPS will use this CA certificate to verify the certificate of the server, and reject the connections if the verification fails.

CAUTION:

According to recent cryptographic research, SHA-1 algorithm cannot be trusted as secure anymore, because signatures can be forged with reasonable costs. As a result, SHA-1 algorithm is not supported in SPS for X.509 certificate chains. Starting from SPS versions 6.0.4 and 6.5.0, certificates with SHA1-based signatures are no longer trusted for Active Directory or LDAP authentication, and future versions might refuse to validate SHA-1 signatures altogether.

Note that Root CA certificates may still contain SHA-1 signatures, because the signature is not validated for self-signed certificates. It is expected that other software such as clients and servers connected to SPS might reject SHA-1 signatures in a similar fashion.

Signing CAs in SPS generate certificates with SHA-256 since versions 4.3.4 and 5.0.0.

CAUTION:

If you will use a TLS-encrypted with certificate verification to connect to the LDAP server, use the full domain name (for example `ldap.example.com`) in the Server Address field, otherwise the certificate verification might fail. The name of the LDAP server must appear in the Common Name of the certificate.

- If the LDAP server requires mutual authentication, that is, it expects a certificate from SPS, enable **Authenticate as client**. Generate and sign a certificate for SPS, then click in the **Client X.509 certificate** field to upload the certificate. After that, click in the **Client key** field and upload the private key corresponding to the certificate.

One Identity recommends using 2048-bit RSA keys (or stronger).

12. To commit the changes, click Commit.
13. Click **Test** to test the connection.

NOTE:

Testing TLS and STARTTLS-encrypted connections is not supported.

Audit policies

An audit trail is a file storing the recorded activities of the administrators. Audit trails are not created automatically for every connection: auditing must be enabled manually in the channel policy used in the connection. The available default channel policies enable auditing for the most common channels. Audit trails are automatically compressed, and can be encrypted, timestamped, and signed as well. Audit trails can be replayed using the Safeguard Desktop Player application (for details, see Safeguard Desktop Player User Guide), or directly in your browser (for details, see Replaying audit trails in your browser on page 721).

TIP:

By default, every connection uses the built-in **default** audit policy. Unless you use a custom audit policy, modifying the **default** audit policy will affect every audited channel of the connections passing through One Identity Safeguard for Privileged Sessions (SPS).

CAUTION:

In RDP connections, if the client uses the Windows login screen to authenticate on the server, the password of the client is visible in the audit trail. To avoid displaying the password when replaying the audit trail, you are recommended to encrypt the upstream traffic in the audit trail using a separate certificate from the downstream traffic. For details, see "Encrypting audit trails" in the Administration Guide.

- For details on how to configure audit trail encryption, see Encrypting audit trails on page 465.
- For details on how to configure timestamping, see Timestamping audit trails with built-in timestamping service on page 469 and Timestamping audit trails with external timestamping service on page 472.
- For details on how to configure audit trail signing, see Digitally signing audit trails on page 474.

Encrypting audit trails

To prevent unauthorized access to the audit trail files, One Identity Safeguard for Privileged Sessions (SPS) can encrypt:

- The entire trail.
- The entire trail, and the upstream part with an additional (set of) certificate(s).
- Only the upstream part.

With upstream encryption, the passwords are visible only with the private key of the certificate used for encrypting the upstream traffic.

NOTE:

Even if the upstream traffic is encrypted with a separate certificate, only one audit trail file is created for a session.

CAUTION:

One Identity Safeguard for Privileged Sessions (SPS) 5 F4 and later versions use a new encryption algorithm to encrypt the recorded audit trails (AES128-GCM). This change has the following effects:

- **If you are using external indexers to index your audit trails, you must upgrade them to the latest version. Earlier versions will not be able to index encrypted audit trails recorded with SPS 5 F4 and later.**
- **To replay an encrypted audit trail recorded with SPS 5 F4 or later, you can use the latest version of the Safeguard Desktop Player application, or the browser-based player of SPS. You cannot replay such audit trails using earlier versions of Safeguard Desktop Player, nor any version of the Audit Player application.**

Encrypting the upstream part has the following limitations:

- During indexing, command detection does not work without the upstream encryption keys.

TIP:

For more information on uploading certificates for indexing and replaying audit trails, see:

- Configuring the internal indexer on page 593 and Replaying encrypted audit trails in your browser on page 724 for uploading certificates for the indexer service.
- Replaying encrypted audit trails in your browser on page 724 for uploading certificates to a user's private keystore.
- "Replay encrypted audit trails" in the Safeguard Desktop Player User Guide for uploading certificates to the Safeguard Desktop Player application.

Encrypting audit trails requires one or more X.509 certificate in PEM format that uses an RSA key, depending on the configuration.

NOTE:

Certificates are used as a container and delivery mechanism. For encryption and decryption, only the keys are used.

One Identity recommends using 2048-bit RSA keys (or stronger).

Use every keypair or certificate only for one purpose. Do not reuse cryptographic keys or certificates (for example, do not use the certificate of the One Identity Safeguard for Privileged Sessions (SPS) webserver to encrypt audit trails, or the same keypair for signing and encrypting data).

The following encryption options are available:

- *Encrypt with a single certificate.* This is the most simple approach: SPS uses one certificate to encrypt the audit trails, and anyone who has the private key of that

certificate can replay the audit trails. If that key is lost, there is no way to open the audit trails.

- *Encrypt separately with multiple certificates.* SPS uses two or more certificates separately to encrypt the audit trails, and anyone who has the private key of one of the encryption certificates can replay the audit trails.
- *Encrypt jointly with two certificates.* SPS uses two certificates together (a certificate-pair) to encrypt the audit trails. The private keys of both encryption certificates are needed to replay the audit trails. This is a kind of "four-eyes in auditing".

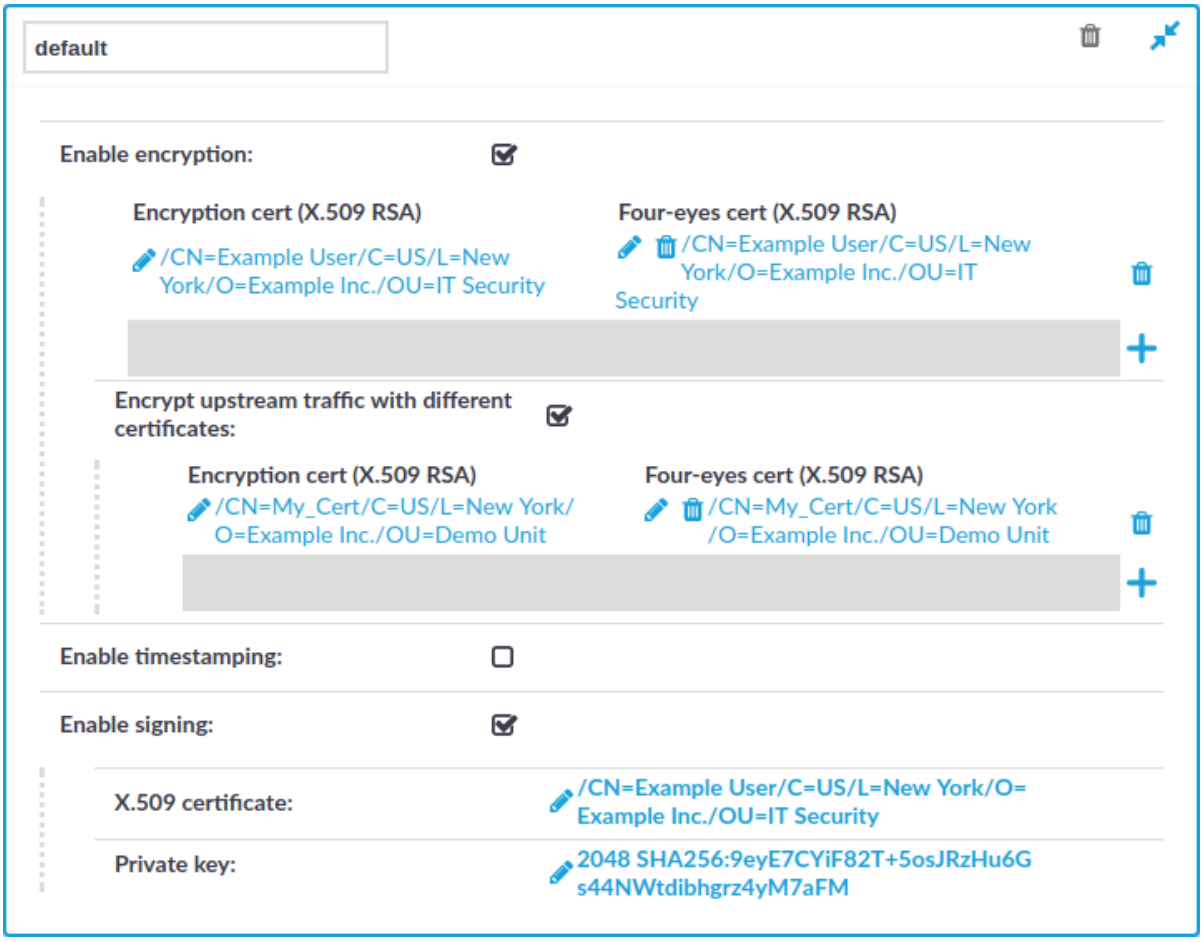
You can combine the different encryption methods. For example, you can encrypt the audit trails with multiple certificate-pairs, and replay the trails only if the private keys of a certificate-pair are available. This is true for encrypting the upstream traffic as well. At the extreme, you will need four private keys to fully replay an audit trail: two to open the normal traffic, and two more to display the upstream traffic.

Note that SPS itself cannot create the certificates used to encrypt the audit trails.

TIP:

If two certificates are displayed in a row, they are a certificate-pair and you need the private key of both to replay the audit trails. If two certificates are displayed in separate rows, you need the one of the private keys to replay the audit trails. If there are multiple rows containing two certificates, you need the private keys of the certificate(s) listed in any of the rows.

Figure 151: Policies > Audit Policies — Encrypting audit trails: joint encryption with a certificate pair



Each audit policy can have up to 8 lines of certificate pairs.

To encrypt audit trails

1. Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

   TIP:

   By default, every connection uses the built-in **default** audit policy. Unless you use a custom audit policy, modifying the **default** audit policy will affect every audited channel of the connections passing through One Identity Safeguard for Privileged Sessions (SPS).

2. Select the **Enable encryption** option.
3. To upload a certificate for encrypting the entire trail:
   a. Click the + icon under the **Encryption cert (X.509 RSA) 4-eyes cert (X.509 RSA)** row.

b. Click on the left icon and upload a certificate to SPS. This certificate will be used to encrypt the audit trails, and it must not include the private key.

> NOTE:
>
> To replay the audit trails, you need the private key of the certificate on the computer running the Safeguard Desktop Player application.

c. (Optional) To encrypt the audit trails jointly with another certificate, click on the right icon and upload a certificate to SPS. Note that the private key of both certificates will be required to replay the audit trails.

d. Repeat these steps to encrypt the audit trails separately with additional certificates.

4. To upload a certificate for encrypting the upstream traffic:

a. Select **Encrypt upstream traffic with different certificates**.

b. Click the + icon under the **Encryption cert (X.509 RSA) 4-eyes cert (X.509 RSA)** row.

c. Click on the left icon and upload a certificate to SPS. This certificate will be used to encrypt the audit trails, and it must not include the private key.

> NOTE:
>
> To replay the upstream part of the audit trails, you need the private key of the certificate on the computer running the Safeguard Desktop Player application.

d. (Optional) To encrypt the audit trails jointly with another certificate, click on the right icon and upload a certificate to SPS. Note that the private key of both certificates will be required to replay the audit trails.

e. Repeat these steps to encrypt the upstream separately with additional certificates.

5. Click Commit.

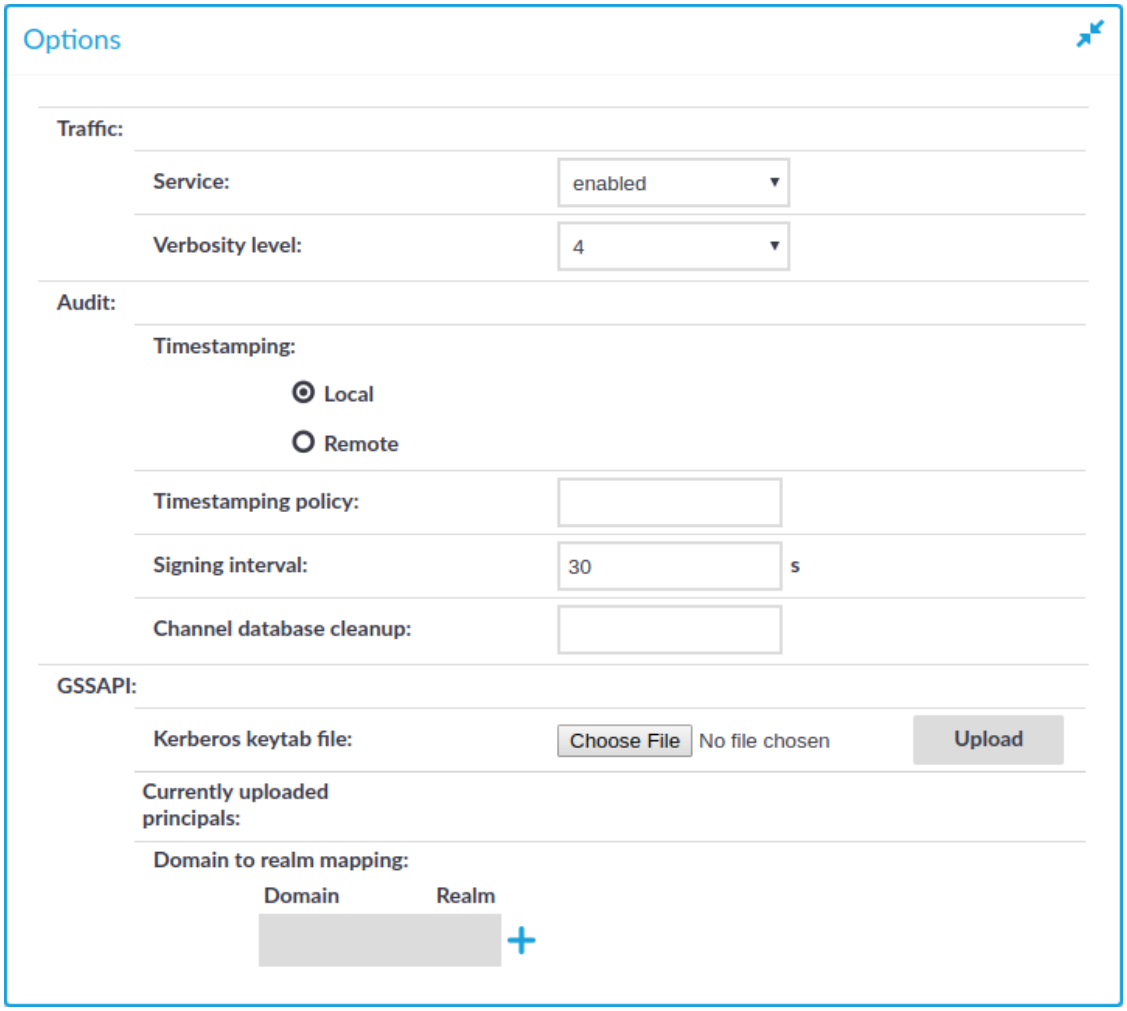
Timestamping audit trails with built-in timestamping service

The following describes how to add timestamps to the audit trails by using the built-in timestamping service of One Identity Safeguard for Privileged Sessions (SPS).

To add timestamps to the audit trails by using the built-in timestamping service of SPS

1. Configure the timestamping interval. You have to repeat these steps for each protocol (HTTP, ICA, RDP, SSH, Telnet, and VNC) you want to configure:

Figure 152: <Protocol name> Control > Global Options —Configuring local timestamping



a. In the protocol control settings, navigate to **Global Options > Timestamping** (for example, **SSH Control > Global Options > Timestamping**).

b. Select **Local**.

> NOTE:
>
> Make sure that you leave the **Timestamping policy** field empty. **Timestamping policy** has relevance only when **Timestamping** is set to **Remote**.

c. Set the **Signing interval**. You can choose any value between 10 and 100

000 seconds.

> NOTE:
>
> The same interval setting applies to timestamping and signing.

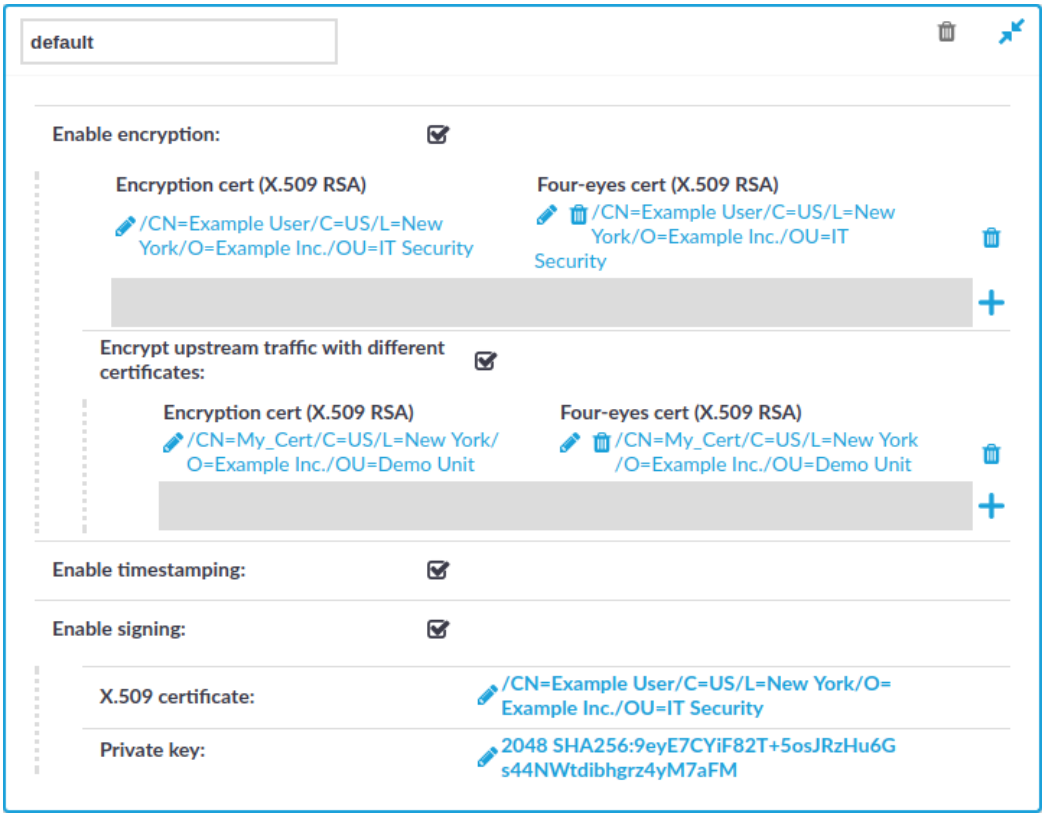
d. Click Commit.

2. Configure audit policies to use timestamping. You have to repeat these steps for each audit policy you want to configure:
   a. Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

> TIP:
>
> By default, every connection uses the built-in **default** audit policy. Unless you use a custom audit policy, modifying the **default** audit policy will affect every audited channel of the connections passing through One Identity Safeguard for Privileged Sessions (SPS).

   b. Select the **Enable timestamping** option.

Figure 153: Policies > Audit Policies — Timestamping audit trails



c. Click Commit. SPS will automatically add timestamps to the audit trails of every connection that is audited and uses this audit policy.

NOTE:

For details on how to change the certificate used for timestamping, see Managing the certificates used on One Identity Safeguard for Privileged Sessions (SPS) on page 407.

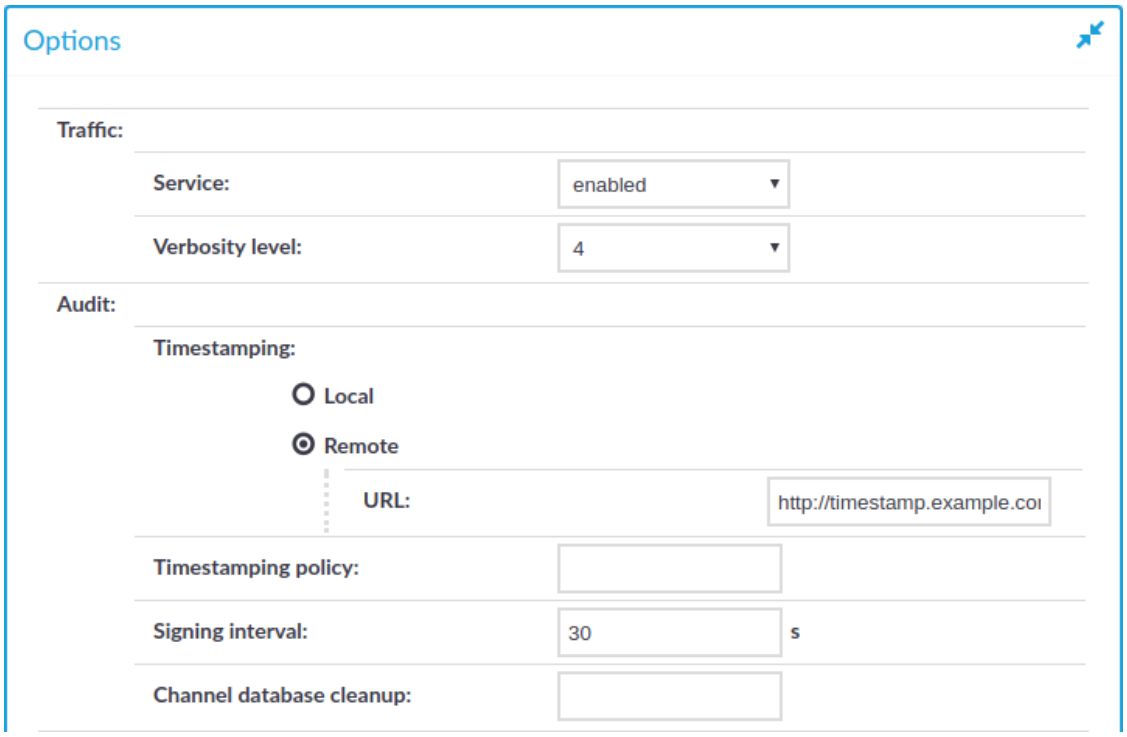
Timestamping audit trails with external timestamping service

The following describes how to request timestamps from a remote Timestamping Authority (TSA).

To request timestamps from a remote TSA

1. Configure the remote TSA, and the timestamping interval. You have to repeat these steps for each protocol (HTTP, ICA, RDP, SSH, Telnet, and VNC) you want to configure:

Figure 154: <Protocol name> Control > Global Options — Configuring a remote TSA



a. In the protocol control settings, navigate to **Global Options > Timestamping** (for example, **SSH Control > Global Options > Timestamping**).

b. Select **Remote**.

c. Enter the address of the timestamping server into the **URL** field. Note that currently only plain HTTP services are supported, password-protected and HTTPS services are not supported.

d.

   If the Timestamping Server has timestamping policies configured, enter the OID of the policy to use into the **Timestamping policy** field. One Identity Safeguard for Privileged Sessions (SPS) will include this ID in the timestamping requests sent to the TSA.

e. Set the **Signing interval**. You can choose any value between 10 and 100 000 seconds.

   NOTE:

   The same interval setting applies to timestamping and signing.

f. Click Commit.

2. Configure audit policies to use timestamping. You have to repeat these steps for each audit policy you want to configure:

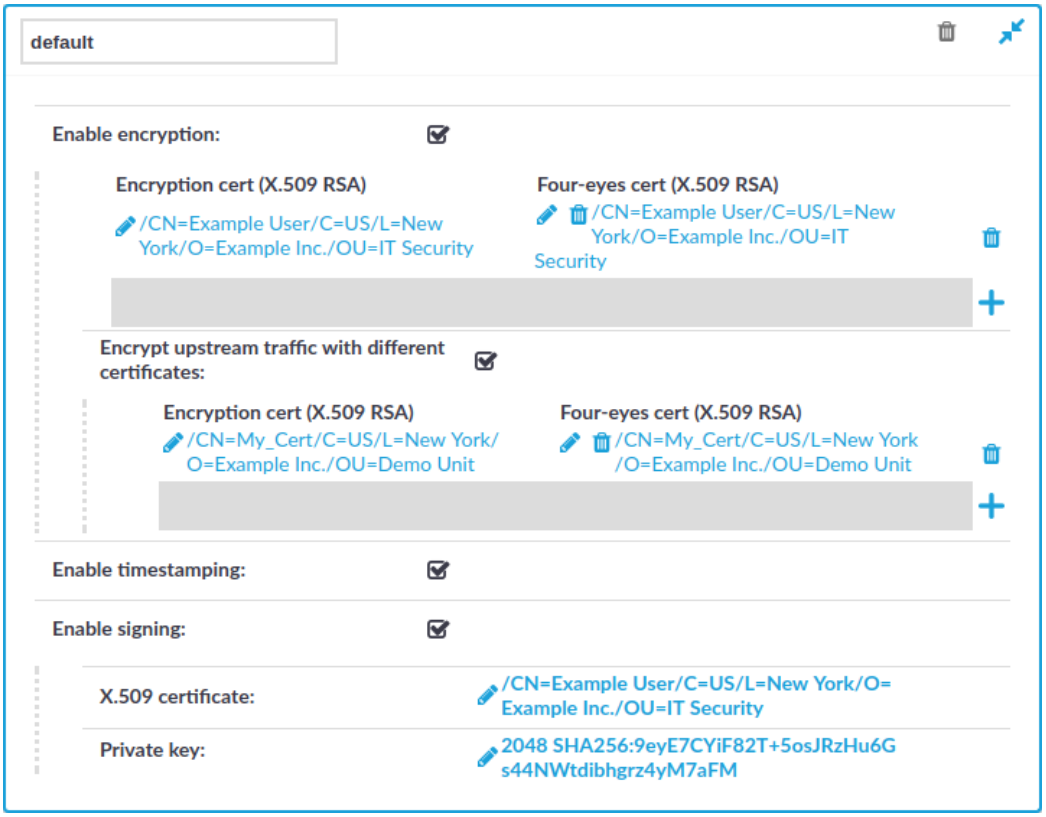
   a. Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

      TIP:

      By default, every connection uses the built-in **default** audit policy. Unless you use a custom audit policy, modifying the **default** audit policy will affect every audited channel of the connections passing through One Identity Safeguard for Privileged Sessions (SPS).

   b. Select the **Enable timestamping** option.

Figure 155: Policies > Audit Policies — Timestamping audit trails



c. Click Commit. SPS will automatically add timestamps to the audit trails of every connection that is audited and uses this audit policy.

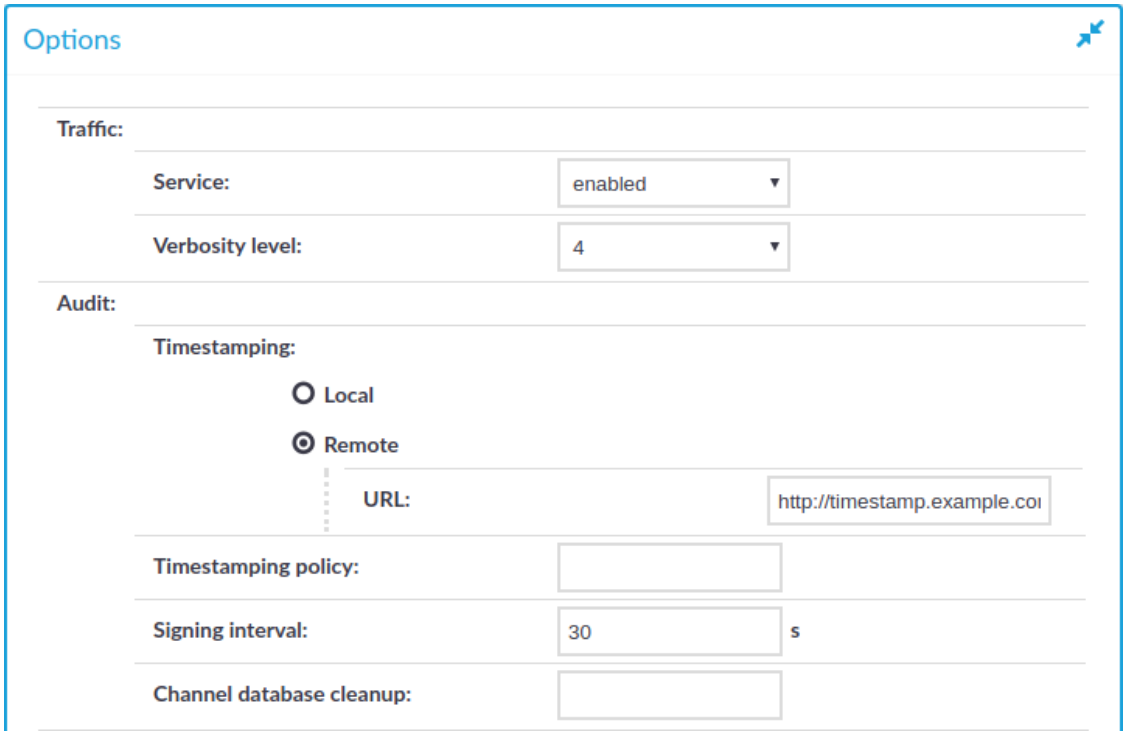
Digitally signing audit trails

One Identity Safeguard for Privileged Sessions (SPS) can digitally sign the audit trails to prevent the manipulation of the audit trail files. This requires an X.509 certificate and also the private key of the certificate. Note that SPS can generate a private key that you can use to create a certificate, but SPS itself cannot create the certificate used to sign the audit trails.

To enable the digital signing of the audit trails

1. Configure the signing interval. You have to repeat these steps for each protocol (HTTP, ICA, RDP, SSH, Telnet, and VNC) you want to configure:

Figure 156: <Protocol name> Control > Global Options — Configuring the signing interval



a. In the protocol control settings, navigate to **Global Options > Timestamping** (for example, **SSH Control > Global Options > Timestamping**).

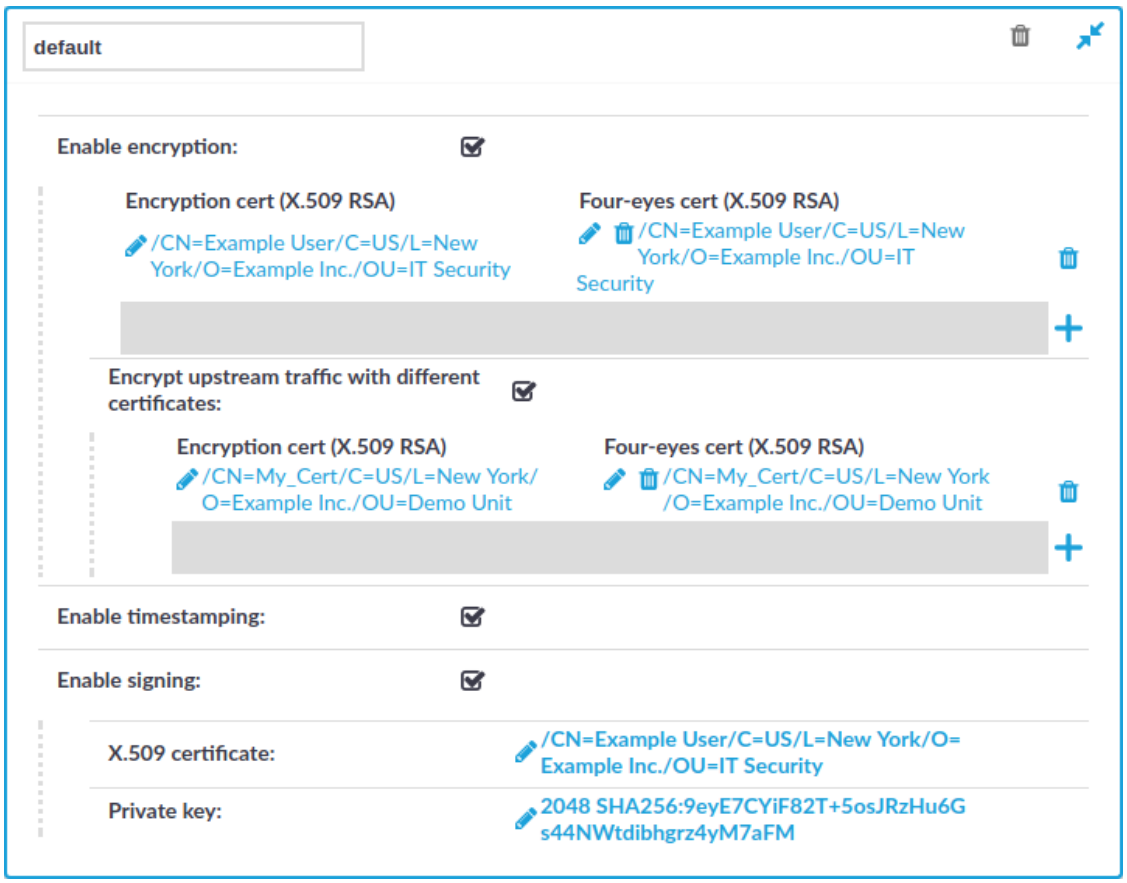
b. Set the **Signing interval**. You can choose any value between 10 and 100 000 seconds.

> NOTE:
>
> The same interval setting applies to timestamping and signing.

c. Click Commit.

2. Navigate to **Policies > Audit Policies** and select the audit policy you will use in your connections.

Figure 157: Policies > Audit Policies — Signing audit trails



> **TIP:**
>
> By default, every connection uses the built-in **default** audit policy. Unless you use a custom audit policy, modifying the **default** audit policy will affect every audited channel of the connections passing through One Identity Safeguard for Privileged Sessions (SPS).

3. Select the **Enable signing** option.
4. Upload a certificate and the corresponding private key to SPS.
5. Click Commit. SPS will automatically sign the audit trails of every connection that is audited and uses this audit policy.
6. Repeat the above steps for other audit policies if needed.

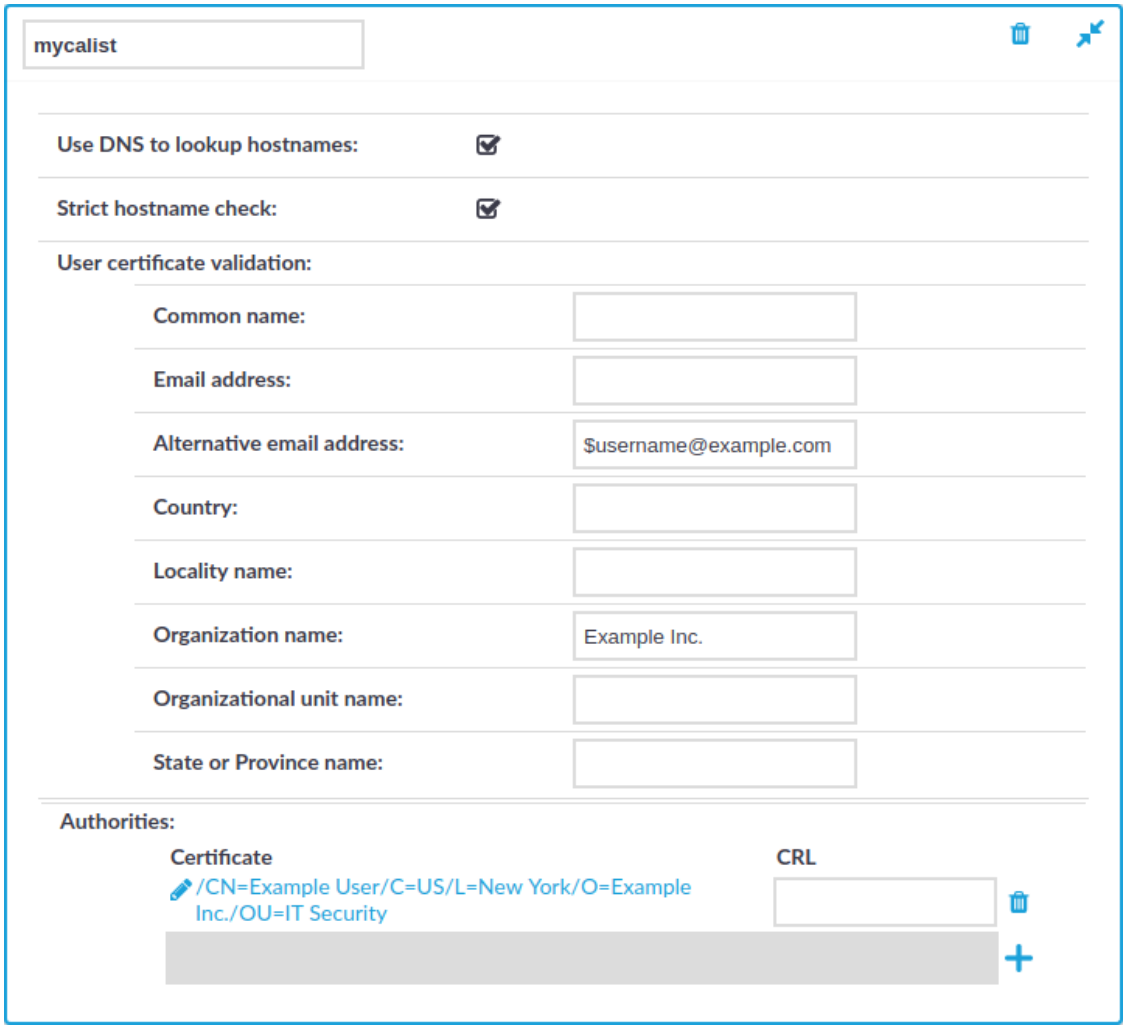
Verifying certificates with Certificate Authorities

One Identity Safeguard for Privileged Sessions (SPS) can check the validity of certificates using the certificates and certificate-revocation lists of the certificate authorities that issued the certificates. This can be used for example to verify the certificates of the servers in SSH/RDP connections.

To create a list of CA certificates to use during the certificate validation

1. Navigate to **Policies > Trusted CA Lists** and click + to create a new list.

Figure 158: Policies > Trusted CA Lists — Creating Trusted CA lists



2. Enter a name for the CA list into the topmost field.
3. Click in the **Certificate** field, and upload the certificate of the Certificate Authority (CA) that will be used to validate the certificates.
4. Enter the URL of the Certificate Revocation List of the CA into the **CRL** field. Certificates appearing on the CRL list will be automatically rejected.

NOTE:

Note that only `.pem` format CRLs are accepted. CRLs that are in PKCS7 format (`.crl`) are not accepted.

5. To further limit which certificates are accepted, you may use the following options:
   - **Strict hostname check**: Select this option to accept only certificates when the Common Name of the certificate contains the hostname or the IP address of the host showing the certificate.
   - **Use DNS to lookup hostnames**: Select this option to use the domain name server set on **Basic Settings > Network > Naming** to resolve the hostnames and IP addresses for certificate validation. If you have enabled the **Strict hostname check** option, you probably want to enable this option as well.
   - To restrict the accepted certificates based on the content of the certificate, enter the required value into the appropriate field of the **User certificate validation** section. For example, to accept only certificates that contain `Example Inc.` in their Organization Name field, enter `Example Inc.` in to the **Organization Name** field. In the Common name, E-mail address, and Alternative e-mail address fields you can use the `$username` macro to refer to the username used in the connection. This macro makes it possible to check that the user is using his own certificate.
6. Click Commit.

Signing certificates on-the-fly

At a number of places, One Identity Safeguard for Privileged Sessions (SPS) can generate the server certificates on the fly. This technique is used for example in SSL-encrypted RDP sessions, RDP sessions that use Network Level Authentication (CredSSP), or SSH connections that use X.509-based authentication.

NOTE:

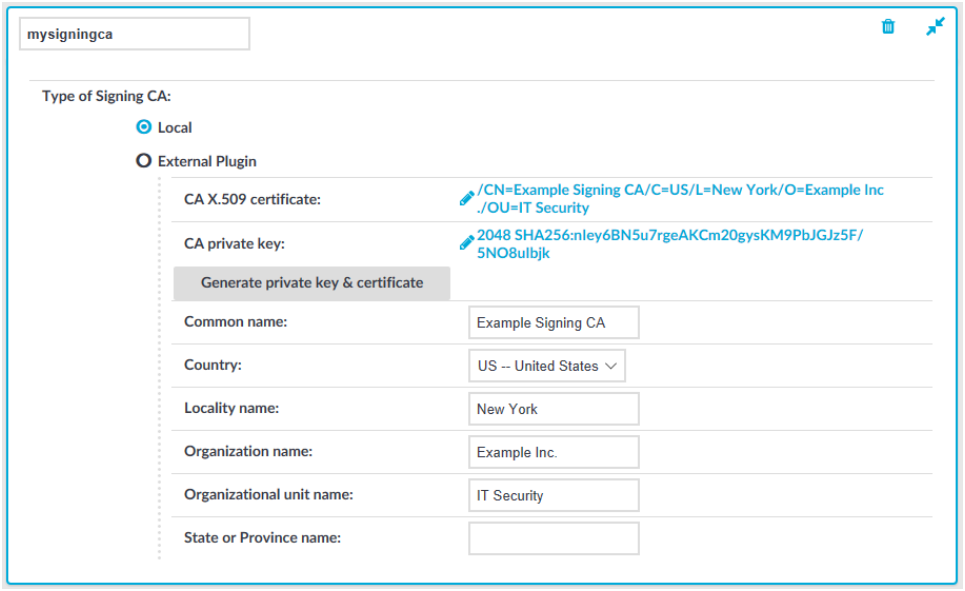
Note the following points about using signing CAs:

- Signing CAs require a CA certificate permitted to sign certificates, and also the corresponding private key.
- These CAs cannot be used to sign audit trails. For details on how to configure the certificates used to sign audit trails, see Digitally signing audit trails on page 474.
- The version of the generated certificates will be the same as the version of the signing CA.
- SPS ignores the CRL (from the crlDistributionPoints extension) of the signing CA when generating certificates. If you want to include a CRL in the generated certificates, you must set it manually. See the following steps for details.

To create a signing CA

1. Navigate to **Policies > Signing CAs** and click +.
2. Select:
   - **Local** to use the built-in signing CA solution, and complete the following steps:
     1. Enter a name for the CA into the topmost field.

        Figure 159: Policies > Signing CAs — Creating Signing CAs - Local

     2. To upload a CA certificate and its private key, complete the following steps. Skip this step if you want to generate a CA on SPS.
        a. Click **Edit** in the **CA X.509 certificate** field and upload the certificate of the certificate authority. Alternatively, you can upload a certificate chain, where one member of the chain is the CA that will sign the certificates.
        b. Click **Edit** in the **CA private key** field and upload the private key of the certificate authority that will sign the certificates.
        c. (Optional) Enter the URL of the Certificate Revocation List (CRL) that you generated using your Certificate Authority in your Public Key Infrastructure (PKI) solution. The URL pointing to this CRL will be included in the certificate. This is the CRL information that will be shown to clients connecting to SPS.

           Note that the CRL list is not generated by the internal CA of SPS. The list must come from your own PKI solution.
        d. Click Commit.
     3. To generate a CA certificate on SPS, complete the following steps:

a. Enter the Common Name for the CA certificate into the **Common Name** field. This name will be visible in the `Issued By` field of the certificates signed by this CA.

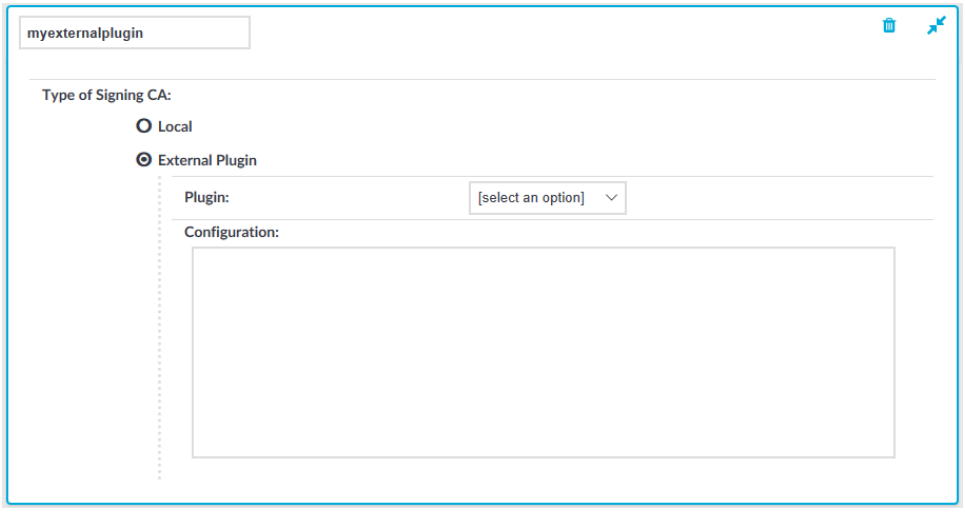
b. Fill the other fields as required, then click **Generate private key and certificate**.

c. Click Commit.

- **External Plugin** to use an external signing CA plugin, and complete the following steps:

  1. Enter a name for the CA into the topmost field.

     Figure 160: Policies > Signing CAs — Creating Signing CAs - External Plugin

  2. From the **Plugin** field, select an uploaded external plugin using the drop-down menu.

     To be able to select from the drop-down menu, you must have an external plugin uploaded in **Basic Settings > Plugins > Signing CAs**.

     For more information about how to create an external Signing CA plugin, see Creating an external Signing CA.

  3. Optionally, fill the **Configuration** field as required by the uploaded plugin.

     The input you enter in the **Configuration** field is passed down to the plugin.

Creating an external Signing CA

Overview

The External Signing CA plugin's purpose is to generate certificate and private key pairs signed by a Certificate Authority. By using this type of plugin the certificate signing can be tailored to fit any custom requirement.

Details

The plugin is a ZIP file containing a *MANIFEST* and a *main.py* file.

The MANIFEST file

The MANIFEST file is a YAML file, and should conform to version 1.2 of the YAML specification. It should contain the following information about the plugin:

```
api: 1.0
type: signingca
name: MySigningCaPlugin
version: 1.0
description: My custom Signing CA
```

The type of the plugin must be `signingca`.

The main.py file

A `Plugin` class containing the following methods must be defined in the *main.py* file:

- `generate_for_addresses`: generates a key/certificate pair for the given addresses (IP/DNS)
- `generate_for_username`: generates a key/certificate pair for the given username
- `generate_for_subject`: generates a key/certificate pair for the given subject values

Method arguments

Each method must take the following arguments:

- `generate_for_addresses`:
  - `addresses`: *{list of str}* - contains either IP or DNS addresses for which the certificate shall be issued
  - `keytype`: *{str}* - contains a fixed value of 'RSA' or 'DSS' indicating the requested key type for the certificate

- `generate_for_username`:
  - `username` *{str}* - contains the username for which the certificate shall be issued
  - `keytype`: *{str}* - contains a fixed value of 'RSA' or 'DSS' indicating the requested key type for the certificate
- `generate_for_subject`:
  - `subject` *{list of (str, str)}* - contains the certificate subject as type-value pairs (tuples). Valid types are the following:
    - `'C'` country name
    - `'ST'` state or province name
    - `'L'` locality name
    - `'O'` orangization name
    - `'OU'` organizational unit
    - `'CN'` common name
    - `'emailAddress'` email address
  - `keytype`: *{str}* - contains a fixed value of 'RSA' or 'DSS' indicating the requested key type for the certificate

Method return values

Each method returns a *{dict}* that must have the following keys:

- `key`: *{str}* the generated key
- `chain`: *{list* of str} a PEM encoded certificate chain containing the generated certificate as the first element

Example

The code below demonstrates a simple plugin that can sign certificates with a built in CA. By default it uses a pre-generated CA certificate and key to complete signing requests. To use a custom certificate, provide a certificate and a key in a python dict format in the configuration field.

Note: If you wish to try this sample code, you will need to provide a MANIFEST file (see below) and the following package dependencies in the .zip file alongside the plugin:

- `asn1crypto`
- `certbuilder`
- `oscrypto`

main.py:

```
#!/usr/bin/env pluginwrapper3

#
# Copyright (c) One Identity
# All Rights Reserved.
#

from ast import literal_eval
from certbuilder import CertificateBuilder, pem_armor_certificate
from ipaddress import ip_address
from oscrypto import asymmetric

class Plugin(object):

    plugin_root_ca = """-----BEGIN CERTIFICATE-----
MIIDhjCCAm6gAwIBAgIIW+mOlk1Cu4swDQYJKoZIhvcNAQELBQAwYTELMAkGA1UE
BhMCSFUxETAPBgNVBAgMCEJ1ZGFwZXN0MREwDwYDVQQHDAhCdWRhcGVzdDEQMA4G
A1UECgwHQmFsYWJpdDEaMBgGA1UEAwwRQmFsYWJpdCBQbHVnaW4gQ0EwHhcNMTgx
MTEyMTQzMDQ2WhcNMTkxMTEyMTQzMDQ2WjBhMQswCQYDVQQGEwJIVTERMA8GA1UE
CAwIQnVkYXBlc3QxETAPBgNVBAcMCEJ1ZGFwZXN0MRAwDgYDVQQKDAdCYWxhYml0
MRowGAYDVQQDDBFCYWxhYml0IFBsdWdpbiBDQTCCASIwDQYJKoZIhvcNAQEBBQAD
ggEPADCCAQoCggEBAO73W0ONVwIaBJas+qUe0VBZ4rtk6PtzRNenZcBkTCkITuuF
DAQ3T1qLUsCyQ4uHMo+yKZUqR3HxbWGxS2l4IaHP6Hbna2kNEyYEsg16mGVUz6tc
D6bxFu3EpB7eU/OXh8RS8URIfZbLNrql1sKe7k1hpXUDS74Ra/avUIYKIpZ5sCjs
F6MBZWz5u3tNUa53xVmqgpnQ6pozN+OQ6k74DjK4xqWqJgTWcN6rxZ9k2voQYE3s
H66jl53q+Zl0D4w/AEW5W3OYNHJtx3tsc36sD2i0doqBCAAvflcSDEs7TXhfXSkC
qCBKyx8ics5EL9h49MDPGwDTehzwvXusz8LlxeMCAwEAAaNCMEAwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQUyFWUMJli0q5CtJOp25IqK2M70oAwDgYDVR0PAQH/
BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCWoQCJPqfM4Sjg0R2O42yrE2GtQsXf
Qb3Dur+CefWLcvjI28t1xuj31khDgpNTwk4IVYrvarNX33C3tjYKgcimwWRMijbA
p8kZzFajOZSWC32CQtkWL79LLkJCTJB7b/4E41oNQPHtOoNCqFY+uQogP90qZ1w1
xlFX8ie/W3cuqhfzW6+/M3iCIwdjhBquvOo6mE3t2/lUcGXE20GayFsKnEmgpDJa
nxoG1+m+s5zCwDuukX8Lr1O7maTMwNVhm5P5QWeEPbGRN7yw+CfzcvPIbFYwnZ5x
XeC9Vtoj2Jbom8RV9uus8R5LfYBJ+HZh74wbGhIC2Kf9LrJTK7r92uVA
-----END CERTIFICATE-----"""

    plugin_root_ca_key = """-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDu91tDjVcCGgSW
rPqlHtFQWeK7ZOj7c0TXp2XAZEwpCE7rhQwEN09ai1LAskOLhzKPsimVKkdx8W1h
sUtpeCGhz+h252tpDRMmBLINephlVM+rXA+m8RbtxKQe3lPzl4fEUvFESH2Wyza6
pdbCnu5NYaV1A0u+EWv2r1CGCiKWebAo7BejAWVs+bt7TVGud8VZqoKZ0OqaMzfj
kOpO+A4yuMalqiYE1nDeq8WfZNr6EGBN7B+uo5ed6vmZdA+MPwBFuVtzmDRybcd7
bHN+rA9otHaKgQgAL35XEgxLO014X10pAqggSssfInLORC/YePTAzxsA03oc8L17
rM/C5cXjAgMBAAECggEBAIucxpw76naW3tFtNG7eB2pLaZUUSq4F1VWtPlxd/MUI
Tpt5OuEHs3vx5CIixCWzkk2zyGmWrvEaHU6zN5ziC7wu7ODzKaTRd7uBiMkpM/oX
x9CU06w0NLIrbbt/J0ss36xKzRyYwY8lIM+Bbmx8UDuzbehkSY89PHd+S6xUJYsF
YmOVM00wx1N6yZGKHUV9GLRnysHb+DBbjGIcjDqmlsdyuAzlB7/DAeToLFNkZvzx
Qzza6whMILXS9Qp39dzn7nJuJywfoOAX2q5LgOrPise2QY1FuAy0GTfqvBDR1eGd
```

```
NwFW5YtH89l347AIDPgklKvKaii2iIw1ZEMf1AlX7mECgYEA+0JM9sToMgLhHJDj
cUsznVn3xzjDT/4O95LdAq1fVrn10wh/SwGCBBPMHQkV1nS70d//1aGctZLWk5+F
K3aPGV9Eas70mOGNcXdU1ITpFNuVfbKK8uH5NF/oEuoD4zRabunrj/zEk0Qu/D9v
pN4qEwJoV1SD/9HpfpaUG/xuBLECgYEA83mtE34zYTY2TLBr4HvoiWrFYoEpPldN
64oD+w1/D0Wd9hxCyzO3y2SmrBmmbzoawTckxD/VKndeRdV5dLlEnAV4F35bPsQl
dRJJEAAQPqqc1z4x6c2my27WPSbm4mIcvfTc65UFu4ovm/koywc96fwvpTX6JIN1
X8zHZ/tQaNMCgYEAl3yk3I9hk22K/ecZSiBWEUPCETpW/66kpX3FhKy085wQ61iP
LtDM69pn0QW+RduBtgsAu3PCAPN0LfManlbP9jMrE96NOHOdDNEusycjRHET04JH
JiM6VeqRCH5RM7ZH4+FjJh/3APc2AN3aWSOdaHKmKCkLoLyVs73jtG/ggTECgYBp
reCf22E1yrAa7WCFmYK/UqbGMMXUF1Ts7YT4zUzfNhpwHqgnRxV5pQBrJt8E3DWM
tACzZfmCazlyGkyTi27qQb10hRXZ0o1nmT45Qa3LZYaaLpa/otHI7xzyghYpIOjU
0pmpb4+DbWFo0+cO6N/I1ftgPGOMwbqKkHnk+kJWnQKBgQDQwITXna8OVjn86AQP
m36JHXi0RNVO/x4+b8T7nurU6XCPIzE0PxfVVSXXsbKTWlq48GIw9ZNpPKPSTCQy
fnYC+Pcu+4A+bfUwFk21khnN/fP5vyFlFhTrGneZeKWhUxv5iOqASEaizfOePmtj
et/4B9LUf9KQFstlhuIR4AP2OA==
-----END PRIVATE KEY-----"""

    def __init__(self, configuration=""):
        self.cert_x509name = {
            'country_name': 'HU',
            'state_or_province_name': 'Budapest',
            'locality_name': 'Budapest',
            'organization_name': 'One Identity',
            'common_name': '',
        }
        self.cert_alt_domains = []
        self.cert_alt_ips = []

        if configuration is not "":
            try:
                configdict = literal_eval(configuration)
                self.plugin_root_ca = configdict['ca']
                self.plugin_root_ca_key = configdict['key']
            except Exception:
                pass

        self.root_ca_certificate = asymmetric.load_certificate(bytes
(self.plugin_root_ca, 'utf-8'))
        self.root_ca_key = asymmetric.load_private_key(bytes(self.plugin_root_
ca_key, 'utf-8'))

    def _sign_certificate(self):
        end_entity_public_key, end_entity_private_key = asymmetric.generate_pair
(self.cert_keytype, bit_size=2048)
        end_entity_private_key = asymmetric.dump_private_key(end_entity_private_
key, None)
        end_entity_subject = self.cert_x509name
```

```
        builder = CertificateBuilder(
            end_entity_subject,
            end_entity_public_key
        )

        builder.subject_alt_domains = self.cert_alt_domains
        builder.subject_alt_ips = self.cert_alt_ips
        builder.issuer = self.root_ca_certificate
        end_entity_certificate = builder.build(self.root_ca_key)
        end_entity_certificate = pem_armor_certificate(end_entity_certificate)
        return end_entity_certificate, end_entity_private_key

    _subject_type_mapping = {
        'C': 'country_name',
        'ST': 'state_or_province_name',
        'L': 'locality_name',
        'O': 'organization_name',
        'OU': 'organizational_unit_name',
        'CN': 'common_name',
        'emailAddress': 'email_address',
    }

    def _set_certificate_x509_name(self, x509name):
        for (t, v) in x509name:
            self.cert_x509name[self._subject_type_mapping[t]] = v

    def _set_certificate_keytype(self, keytype):
        # Certbuilder lists DSS key as DSA so we have to translate the string
here
        if keytype == "dss": keytype = "dsa"
        if keytype not in ['dsa', 'rsa']:
            raise ValueError('Certificate type should be either \'rsa\' or
\'dss\'')
        else:
            self.cert_keytype = keytype

    def _set_certificate_cn(self, cn):
        self.cert_x509name['common_name'] = cn

    def _set_certificate_addresses(self, addresses):
        for address in addresses:
            try:
                ip_address(address)
                self.cert_alt_ips.append(address)
            except ValueError:
                self.cert_alt_domains.append(address)

    def _build_response(self, cert, key):
```

```
        return {'key': key.decode('ascii'), 'chain': [cert.decode('ascii'),
self.plugin_root_ca]}

    def generate_for_addresses(self, addresses: list, keytype: str):
        self._set_certificate_keytype(keytype)
        self._set_certificate_addresses(addresses)
        self._set_certificate_cn(addresses[0])
        return self._build_response(
            *self._sign_certificate()
        )

    def generate_for_username(self, username: str, keytype: str):
        self._set_certificate_keytype(keytype)
        self._set_certificate_cn(username)
        return self._build_response(
            *self._sign_certificate()
        )

    def generate_for_subject(self, x509name: list, keytype: str):
        self._set_certificate_keytype(keytype)
        self._set_certificate_x509_name(x509name)
        return self._build_response(
            *self._sign_certificate()
        )
```

Use the following snippet as the *MANIFEST* file:

```
# Name of the plugin, may contain [a-zA-Z0-9]
name: HelloSigningCaPlugin

# Version of the plugin, only for display purposes
version: 0.1

# Type of the plugin - this is a signingca plugin
type: signingca

# API version of the SCB the plugin was written for, in major.minor format
api: 1.0

# Free form description.
description: This is an example plugin used for testing.

# Entry point for the plugin (also for running with python3)
entry_point: main.py
```

Creating a Local User Database

Local User Databases are available for HTTP, RDP, SSH and Telnet protocols, and can be used to authenticate the clients to credentials that are locally available on One Identity Safeguard for Privileged Sessions (SPS). Such credentials include passwords and public keys. **Local User Databases** are most commonly used in inband gateway authentication scenarios.

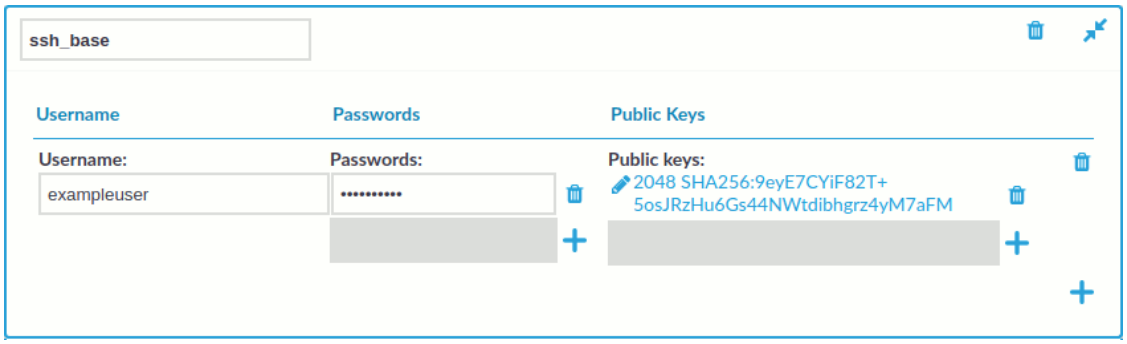
NOTE:

To store credentials on SPS and use them to authenticate on the server, use a local Credential Store. For details, see Using credential stores for server-side authentication on page 757.

To create a Local User Database

1. Navigate to **Policies > Local User Databases** and click +.
2. Enter the name of the **Local User Database**.
3. Click + to add entries.

Figure 161: Policies > Local User Databases — Mapping keys



4. Enter the name of the user into the **Username** field.

   NOTE:

   If you also use Usermapping policies, enter the username that the client will use on the server side. If you also use gateway authentication, the gateway username can be used as well.

5. If you use public-key based authentication on the client side, click the + icon in the **Public Keys** field, and upload the public key of the client.

   SPS will verify that a client trying to use the username set in Step 3 is authenticating itself with the private key that corresponds to the uploaded public key or certificate.

   One Identity recommends using 2048-bit RSA keys (or stronger).

6. Repeat the above steps to add other users as required.
7. Click Commit.
8. Navigate to the **Authentication Policies** tab of the respective protocol and select the **Local User Database** there.

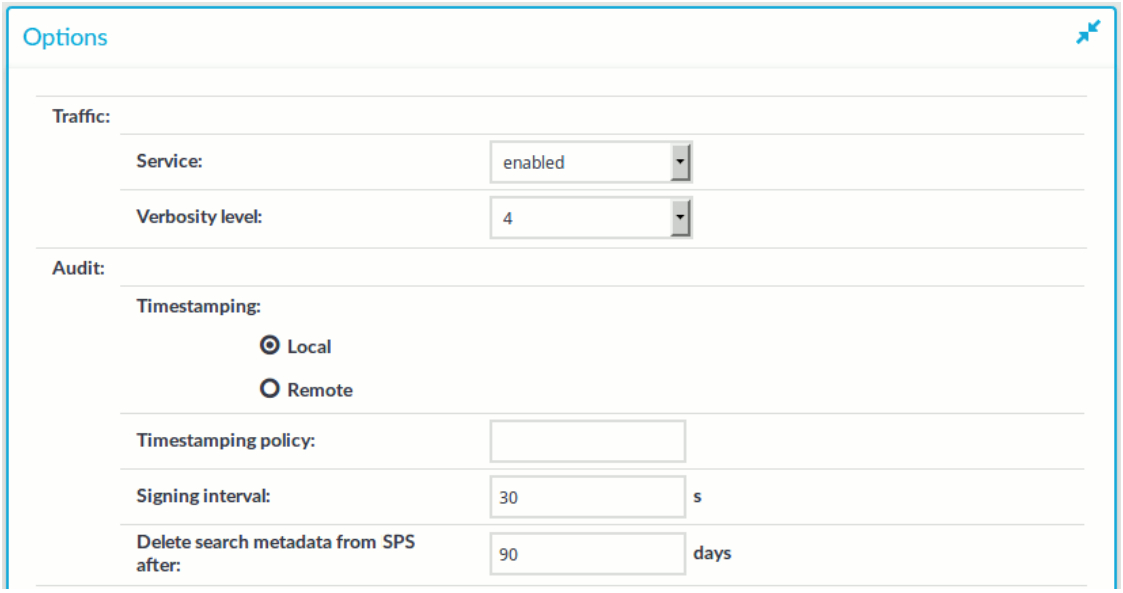
Configuring cleanup for the One Identity Safeguard for Privileged Sessions (SPS) connection database

One Identity Safeguard for Privileged Sessions (SPS) can automatically archive audit trails older than a specified retention time. However, the metadata of the corresponding connections is not deleted from the SPS connection database. Deleting the stored data about old connections decreases the size of the database, making searches faster, and might be also required by certain policies or regulations. The period after metadata is deleted can be specified individually for the different protocols, (for example, data about SSH connections can be stored longer than other connections) and also for every connection policy.

To configure SPS to delete the metadata of old connections for a particular protocol

1. Navigate to the **Global Options** page of the respective protocol, for example, to `SSH Control > Global Options`.

2. **Figure 162: <Protocol name> Control > Global Options — Configuring connection database cleanup for a protocol**



   Enter how long SPS (in days) should keep the metadata into the **Delete search metadata from SPS after** field. For example, if you specify `365`, SPS will delete the data of connections older than a year. Enter zero (`0`) to keep the data indefinitely (this is also the default behavior of SPS).

NOTE:

The database cleanup occurs once a day at 22:01 PM.

The time you specify in the **Delete search metadata from SPS after** field cannot be shorter than the **Delete data from SPS after** field set for the Archive policies used in the connections of this protocol. Note that since the database cleanup happens once a day at 22:01 PM, if you specify the same retention time, for example, 1 day in the **Delete data from SPS after** field, ensure that the archiving or cleanup is set to start before 22:01 PM.

The time you specify in the **Delete search metadata from SPS after** field cannot be shorter than the **Delete search metadata from SPS after** field set in the individual connection policies of this protocol.

3. Click Commit and repeat the previous step for other protocols if needed.

4. **Figure 163: <Protocol name> Control > Connections — Configuring connection database cleanup for a connection**



To delete the metadata of certain connections earlier than the time set in the **Global Options > Delete search metadata from SPS after** field of the protocol, navigate to the particular connection policy, and enter how long SPS (in days) should keep the metadata of the sessions of this connection policy into the **Delete search metadata from SPS after** field. Enter zero (`0`) to use the settings of the protocol (this is also the default behavior of SPS).

NOTE:

The time you specify in the **Delete search metadata from SPS after** field cannot be shorter than the **Delete data from SPS after** field set for the Archive policies used in the connections of this protocol. Note that since the database cleanup happens once a day at 22:01 PM, if you specify the same retention time, for example, 1 day in the **Delete data from SPS after** field, ensure that the archiving or cleanup is set to start before 22:01 PM.

5. Click Commit and repeat the previous step for other connections if needed.

Expected outcome:

Every day SPS deletes the metadata of connections older than the given cleanup time from the connection database.

HTTP-specific settings

The following sections describe configuration settings available only for the HTTP protocol. Use the following policies to control who, when, and how can access the HTTP connection. For details on configuring Channel Policies, see Creating and editing channel policies on page 448. For a list of supported client applications, see Supported protocols and client applications on page 38.

Auditing HTTP and HTTPS connections is possible in both transparent and non-transparent modes. SPS can also be used as an HTTP/HTTPS proxy to simplify client configuration and integration into your network environment, or it can forward HTTP traffic, behaving as a HTTP tunnel.

- *Channel Policy*: The HTTP protocol has only one channel type with no special configuration options. The available channel policy options are the following: **From**, **Target**, **Time policy**, **Record audit trail**, and **Remote groups**. Note that the **Remote groups** option is used only if the user performs inband authentication using one of the supported HTTP authentication methods (see Authentication in HTTP and HTTPS on page 491). To retrieve the groups of an authenticated user from an LDAP database, you must also set an **LDAP Server** in the Connection Policy (for HTTP/HTTPS connections, One Identity Safeguard for Privileged Sessions (SPS) uses this server only to retrieve the group membership of authenticated users, you cannot authenticate the users to LDAP from SPS). For details on configuring these options, see Creating and editing channel policies on page 448.

  When setting **Target**, note the following:

  - If the connection uses DNAT (NAT destination address), the target address of the original client will be compared to the **Target** parameter of the Channel policy, that is not necessarily equivalent with the server's address.
  - If the connection is redirected to a Fix address, the redirected address will be compared to the **Target** parameter of the Channel policy.
- *HTTP connections*: For details, see Setting up HTTP connections on page 493.
- *HTTP sessions*: HTTP settings determine the parameters of the connection on the protocol level, including timeout value, and so on. For details, see Session-handling in HTTP on page 501.
- *HTTP settings*: HTTP settings determine the parameters of the connection on the protocol level, including timeout value, and so on. For details, see Creating and editing protocol-level HTTP settings on page 502.

Limitations in handling HTTP connections

Avoid using the IP address configured for administrator or user login on One Identity Safeguard for Privileged Sessions (SPS) when configuring HTTP or SSH connections.

The current version of SPS does not support the following features that are available for other protocols:

- Four-eyes authorization

Forwarding HTTP connections to an HTTP proxy is not supported. If your clients use an HTTP proxy to access the target servers, place SPS behind the proxy: *Clients - HTTP Proxy - SPS*.

CAUTION:

The *Clients - SPS - HTTP Proxy* scenario is NOT supported.

Authentication in HTTP and HTTPS

For the audited HTTP and HTTPS connections, One Identity Safeguard for Privileged Sessions (SPS) supports the following inband authentication methods for the HTTP protocol. These authentication methods are automatically supported for every Connection policy, without further configuration.

- Basic Access Authentication (according to RFC2617)
- The NTLM authentication method commonly used by Microsoft browsers, proxies, and servers

SPS records the username used in the authentication process into the **Username** and **Remote username** fields of the connection database.

For authenticated sessions, SPS can perform group-based user authorization that allows you to finetune access to your servers and services: you can set the required group membership in the Channel policy of the HTTP connection. Note that group-based authorization in HTTP works only for authenticated sessions (for HTTP/HTTPS connections, SPS uses this server only to retrieve the group membership of authenticated users, you cannot authenticate the users to LDAP from SPS). If a username is not available for the session, SPS will permit the connection even if the **Remote groups** field is set.

SPS does not store failed HTTP authentication attempts in the connection database. This means that the **Verdict** field of the Search page will never contain `CONN-AUTH-FAIL` values for HTTP connections.

Note that authentication also affects the way SPS handles HTTP sessions. For details, see Session-handling in HTTP on page 501.

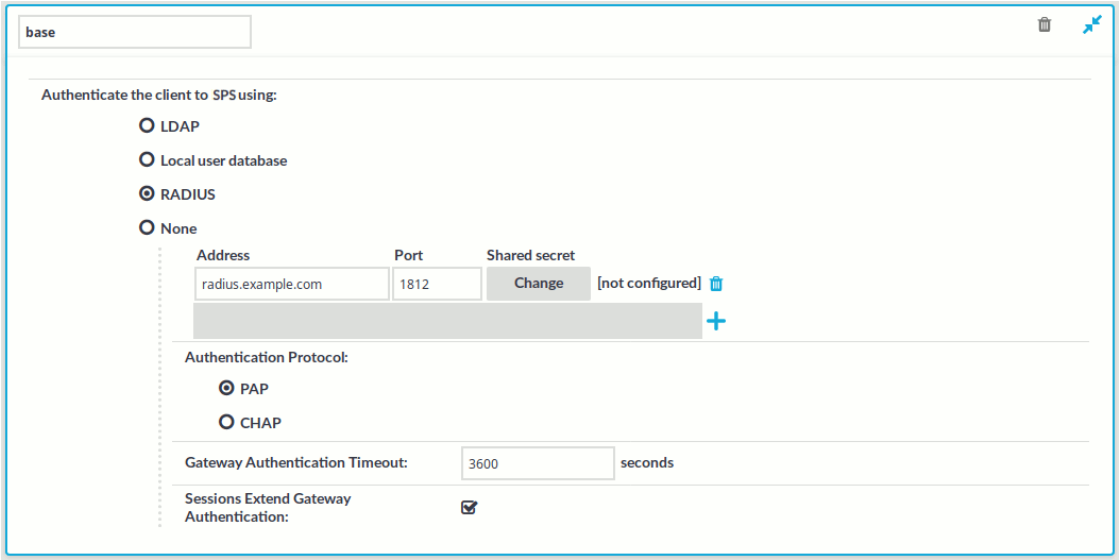
Creating a new HTTP authentication policy

An authentication policy is a list of authentication methods that can be used in a connection. Connection definitions refer to an authentication policy to determine how the client can authenticate to One Identity Safeguard for Privileged Sessions (SPS).

To create a new authentication policy

1. Navigate to **HTTP Control > Authentication Policies**, and click +.

   Figure 164: HTTP Control > Authentication Policies — Configuring HTTP authentication policies

2. Enter a name for the policy into the **Name** field.
3. Select the authentication method used on the client-side in the **Authenticate the client to SPS using** field. For the client-side connection, SPS can authenticate the client inband (within the HTTP protocol) using the following authentication methods:
   - **LDAP**: SPS will authenticate the client to the LDAP database set in the **LDAP Server** of the connection policy. To use LDAP authentication on the client side, select **Authenticate the client to SPS using > LDAP**.

     NOTE:

     SPS will authenticate the client-side connection to the LDAP server configured in the connection policy. This is not necessarily the same as the LDAP server used to authenticate the users accessing the SPS web interface.
   - **Local user database**: Authenticate the client locally on the SPS gateway using a **Local user database**. Select the database to use in the **Local user**

database field. For details on creating a **Local User Database**, see Creating a Local User Database on page 486.

- **RADIUS**: SPS will authenticate the client to the specified RADIUS server. Select **Authenticate the client to SPS using > RADIUS**, enter the IP address (use an IPv4 address) or hostname of the RADIUS server into the **Address** field, the port number of the RADIUS server into the **Port** field, and the shared secret of the RADIUS server into the **Shared secret** field. Only password-authentication is supported (including one-time passwords), challenge-response based authentication is not.

  To add more RADIUS servers, click + and fill in the respective fields.
- **None**: Do not perform client-side authentication, the client will authenticate only on the target server.

4. Specify the time remaining until a successful gateway authentication times out into the **Gateway Authentication Timeout** field.

   To avoid interruptions for active HTTP sessions, select the **Sessions Extend Gateway Authentication** checkbox. When enabled, active HTTP sessions can extend the gateway authentication beyond the configured timeout.

5. Click Commit.

   NOTE:

   The client-side authentication settings apply for authenticating the user inband (that is, within the HTTP protocol) to the SPS gateway.

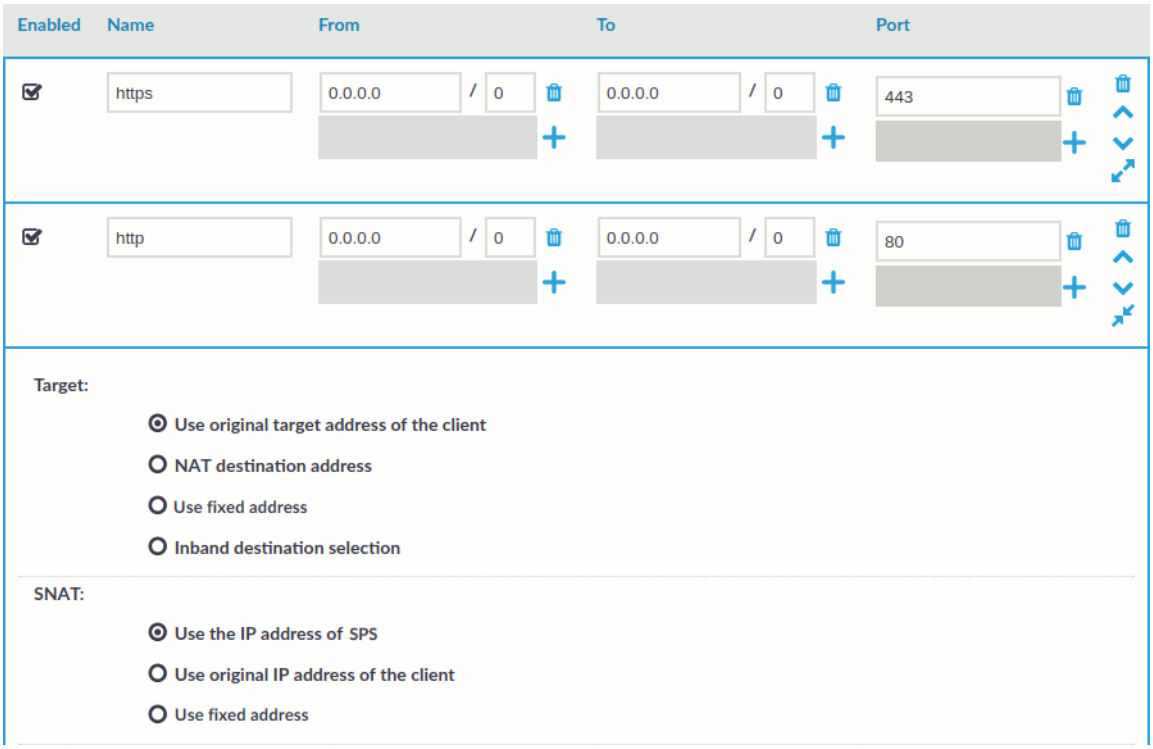
Setting up HTTP connections

This section focuses on describing the HTTP-specific details of connection configuration. For a detailed description on configuring connections, see General connection settings on page 434.

Setting up a transparent HTTP connection

The following describes how to set up a transparent HTTP connection. To audit HTTP connections in non-transparent mode, see Enabling One Identity Safeguard for Privileged Sessions (SPS) to act as a HTTP proxy on page 495.

Figure 165: HTTP Control > Connections — Transparent HTTP connection



To set up a transparent HTTP connection

1. In the **Name** field, enter the name of the connection that will identify the connection policy.
2. In the **From** field, enter the IP address and prefix of the client that will be permitted to access the server.

   You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).
3. In the **To** field, enter the IP address and prefix that the clients will target.

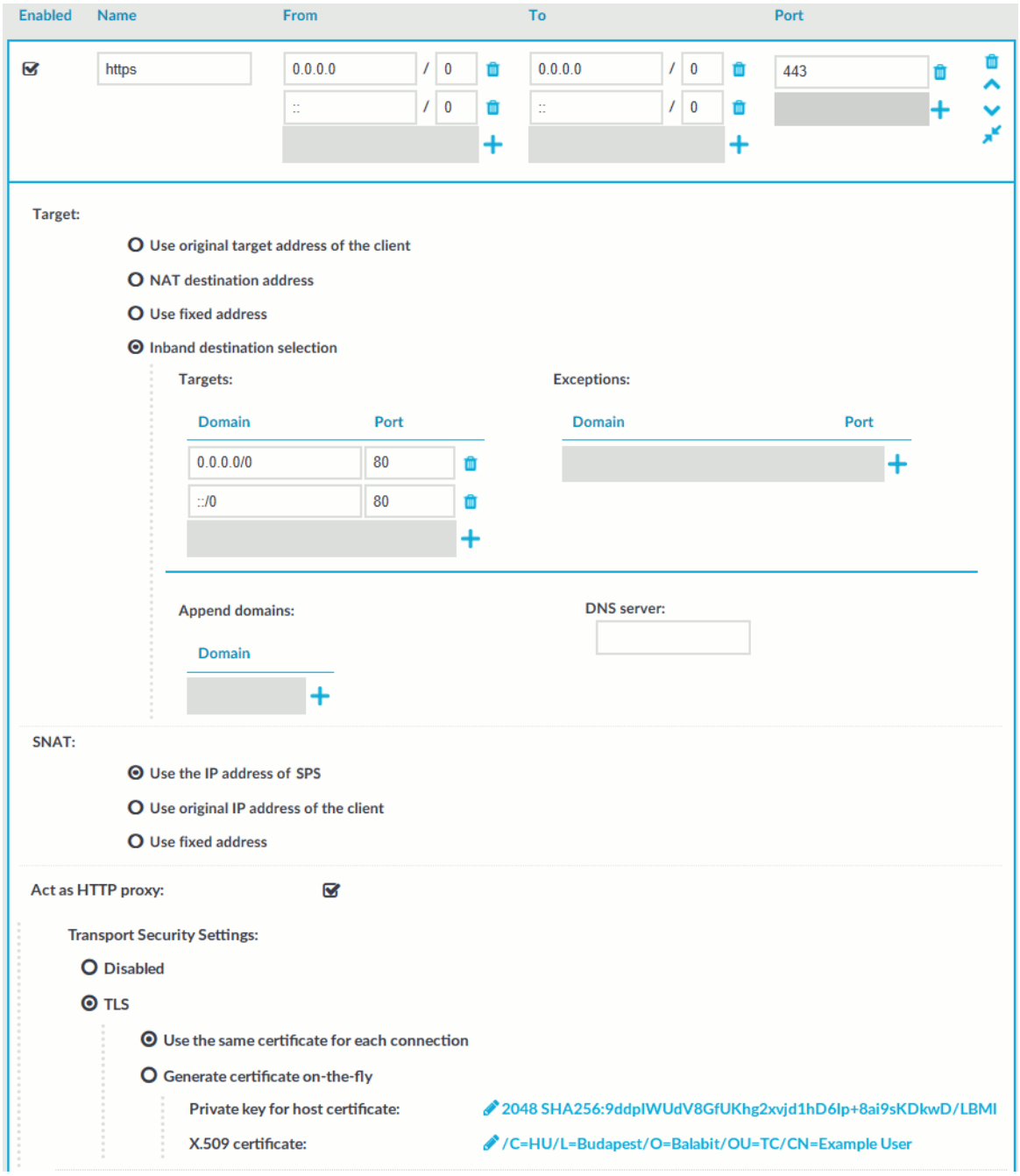
   You can use an IPv4 or an IPv6 address. To limit the IP range to the specified address, set the prefix to 32 (IPv4) or 128 (IPv6).
4. In the **Target** section, select **Use original target address of the client**.
5. In the **SNAT** section, select **Use the original IP address of One Identity Safeguard for Privileged Sessions (SPS)**.
6. Since SPS cannot automatically decide whether the incoming sessions are encrypted or not, it is required to setup another identical connection policy for the same sessions, for HTTPS. As a result, HTTP and HTTPS sessions will be saved into separate trails.
   a. Setup a new connection policy with the same settings as above.
   b. Set the **Port** to 443.

c. Enable TLS encryption. For details, see Enabling TLS encryption in HTTP on page 498.

Enabling One Identity Safeguard for Privileged Sessions (SPS) to act as a HTTP proxy

The following describes how to enable One Identity Safeguard for Privileged Sessions (SPS) to act as a HTTP proxy.

Figure 166: HTTP Control > Connections — Act as HTTP proxy



To enable SPS to act as a HTTP proxy

1. Enable **Act as HTTP proxy** to configure the client to use SPS as a HTTP proxy.

   You can use SPS as a HTTP proxy through TLS. All traffic between the browser and SPS is tunneled through TLS.

To use this feature, ensure that the client software can establish secure web proxy connections and supports client software configuration, such as, proxy auto-configuration files.

For information about making browser specific settings for Chromium, see Secure Web Proxy.

2. Select whether you want encrypted web proxy connection between the HTTP client and SPS.

   Since there is now a secure channel between the web browser and SPS, you can also enable proxy authentication. This makes it possible for the web browser to do an inband gateway authentication to SPS before being able to issue HTTP requests through SPS.

   - To disable encryption between the HTTP client and SPS, select **Disabled**.

     NOTE:

     Since the forwarded data may contain sensitive information, One Identity recommends using encryption between the HTTP client and SPS.

   - To use encryption between the HTTP client and SPS, select one of the following options:
     - To use a fix certificate, select **Use the same certificate for each connection** and copy or upload the certificate.
     - To generate a certificate on-the-fly, signed by a provided Signing CA, select **Generate certificate on-the-fly**. It uses the parameters of the signing CA, excluding the CN field, which is filled with the name of the target host name.

       NOTE:

       When **Generate certificate on-the-fly** is selected and the connection is in transparent setup, the **CN** field is filled in using Server Name Indication (SNI). If the client does not support SNI, the **CN** field will contain the target IP, which may cause certificate verification warning on the client browser.

3. Select **Inband destination selection** as **Target**.
4. To permit access to any HTTP servers, enter `0.0.0.0/0` into the **Domain** field. Alternatively, enter the IP address or subnet of the HTTP address you want permit access to. For IPv6 addresses, add **::/0** as well.
5. To permit HTTP access to the destination servers on any port, leave the **Domain > Port** field empty. Otherwise, clients will be permitted only to access the specified port.
6. Enter the port where SPS should accept HTTP connections into the **To > Port** field. The default port number when using the **Act as HTTP proxy** setting is `3128`. This value should be the same as the proxy port setting on your clients.
7. Ensure that you have set SPS as proxy on the clients.

CAUTION:

To perform gateway authentication on SPS, the client browsers must be configured to use a Proxy Auto-Configuration (PAC) script.

To perform gateway authentication in a TLS-encrypted channel, the script must return an HTTPS address. Note that currently the Safari browsers do not support TLS-encryption in gateway authentication. For example:

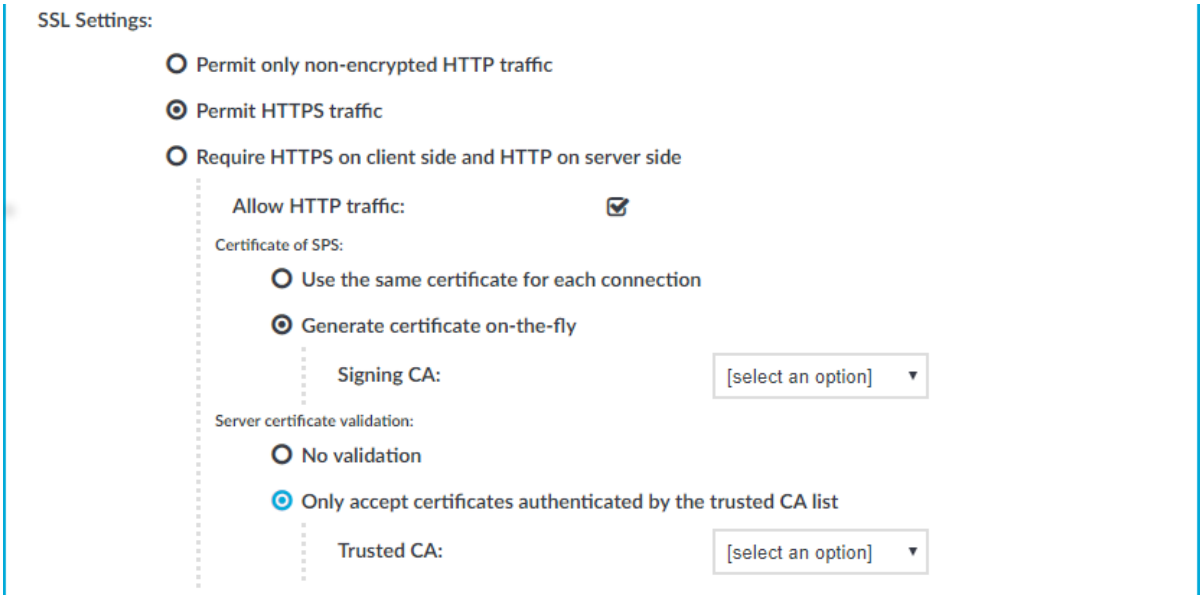
```
if (dnsDomainIs(host, "example-domain.com") || dnsDomainIs(host,
"www.example-domain.com"))
return "HTTPS 192.168.11.121:3128";
```

The client browsers might require the certificate of SPS to contain the `Subject Alternate Name` field. Certificates generated on SPS using the Generate certificate on-the-fly option automatically contain this field. If you Use the same certificate for each connection, make sure this field is present and properly set.

Enabling TLS encryption in HTTP

This setting either enforces TLS encryption or accepts both HTTP and HTTPS requests.

Figure 167: HTTP Control > Connections> SSL Settings — Enabling SSL encryption in HTTP



To enable SSL encryption

1. In **SSL Settings**, select **Permit HTTPS traffic**. To control plain HTTP traffic with the same connection policy, enable **Allow HTTP traffic**.
2. Select the certificate to show to the clients.
   - To use the same certificate for every peer, complete the following steps.
     1. Generate and sign a certificate for One Identity Safeguard for Privileged Sessions (SPS) in your PKI system, then export the certificate and its private key.
     2. Select **Use the same certificate for each connection**.
     3. Select **Private key for host certificate**, click and upload the private key.
     4. Select **X.509 host certificate**, click and upload the certificate.

   NOTE:

   When using the **Use the same certificate for each connection** option and the connection policy that allows access to multiple servers using HTTPS, the client browsers will display a warning because the certificate used in the connection will be invalid (namely, the Common Name of the certificate will not match the hostname or IP address of the server).

   - To use a separate certificate for every connection, complete the following steps.
     1. Create a certificate authority that will be used to sign the certificates that SPS shows to the peer. For details, see Signing certificates on-the-fly on page 478.
     2. Select **Generate certificate on-the-fly**.
     3. Select the certificate authority to use in the **Signing CA** field.

   Limitations

   NOTE:

   When **Generate certificate on-the-fly** is selected and the connection is in transparent setup, the **CN** field is filled in using Server Name Indication (SNI). If the client does not support SNI, the **CN** field will contain the target IP, which may cause certificate verification warning on the client browser.

   NOTE:

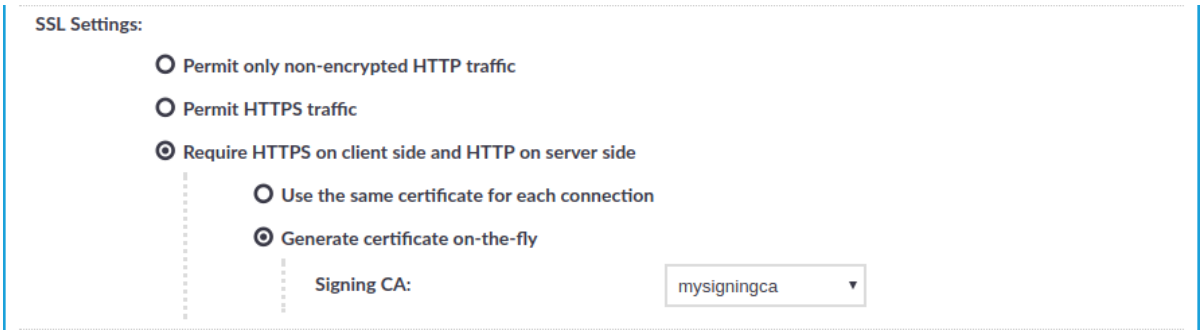
   Import the certificate of the signing Certificate Authority to your clients. Otherwise, the client browsers will display a warning due to the unknown Certificate Authority.
3. Select how SPS should authenticate the server.

- To permit connections to servers without requesting a certificate, select **No validation**.
- To permit connections only to servers with a valid certificate that was signed by a specific CA, complete the following steps.
    1. Create a list of trusted Certificate Authorities that will be used to validate the certificates of the servers. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities on page 476.
    2. Select **Only accept certificates authenticated by the trusted CA list**.
    3. In the **Trusted CA** field, select the certificate authority list to use.

Configuring half-sided SSL encryption in HTTP

The following steps describe how to enable half-sided SSL encryption (which requires HTTPS on client side, and HTTP on server side).

Figure 168: HTTP Control > Connections > SSL Settings — Enabling half-sided TLS encryption in HTTP



To enable half-sided TLS encryption, require HTTPS on client side, and HTTP on server side

1. In **SSL Settings**, select **Require HTTPS on client side and HTTP on server side**.

    NOTE:

    If the connection is configured at **Target** to **Use fixed address** and the port number is set to `443`, One Identity Safeguard for Privileged Sessions (SPS) will still automatically use port `80` to connect to the server, when **Require HTTPS on client side and HTTP on server side** is selected.

2. Select the certificate to show to the clients.

- To use the same certificate for every peer, complete the following steps.
  1. Generate and sign a certificate for SPS in your PKI system, and export the certificate and its private key.
  2. Select **Use the same certificate for each connection**.
  3. Select **Private key for host certificate**, click and upload the private key.
  4. Select **X.509 host certificate**, click and upload the certificate.

  NOTE:

  When using the **Use the same certificate for each connection** option and the connection policy that allows access to multiple servers using HTTPS, the client browsers will display a warning because the certificate used in the connection will be invalid (namely, the Common Name of the certificate will not match the hostname or IP address of the server).
- To use a separate certificate for every connection, complete the following steps.
  1. Create a certificate authority that will be used to sign the certificates that SPS shows to the peer. For details, see Signing certificates on-the-fly on page 478.
  2. Select **Generate certificate on-the-fly**.
  3. Select the certificate authority to use in the **Signing CA** field.

Limitations

NOTE:

When **Generate certificate on-the-fly** is selected and the connection is in transparent setup, the **CN** field is filled in using Server Name Indication (SNI). If the client does not support SNI, the **CN** field will contain the target IP, which may cause certificate verification warning on the client browser.

NOTE:

Import the certificate of the signing Certificate Authority to your clients. Otherwise, the client browsers will display a warning due to the unknown Certificate Authority.

Session-handling in HTTP

Communication over HTTP consists of client requests and server responses (also called exchanges). Unlike in other protocols, for example SSH, these request-response pairs do not form a well-defined, continuous connection. Therefore, One Identity Safeguard for Privileged Sessions (SPS) assumes that an HTTP request-response pair belongs to a specific session if the following points are true:

- The IP address of the client is the same
- The hostname of the target server (not the IP address) is the same
- The username is the same (if the user has performed inband authentication)
- The time elapsed since the last request-response pair between the same client and server is less then the session timeout value (15 minutes by default).
- The first session cookie SPS finds within the request is the same. Note that the cookie must be listed in the **Session Cookie Settings** option. For details, see Creating and editing protocol-level HTTP settings.

SPS creates a separate audit trail and records the accessed URLs for every session. These are displayed on the **Search** page. If any of the columns is not visible, click **Customize columns...**.

For technical reasons, in authenticated sessions the login page where the user provides the credentials is not part of the session associated with the username. This means that even if the login page is the first that the user visits, SPS will record two sessions: the first does not include a username, the second one does. These two sessions are visible on the **Active Connections** page (until the unauthenticated session times out).

Creating and editing protocol-level HTTP settings

HTTP settings determine the parameters of the connection on the protocol level, including timeout value, and so on.

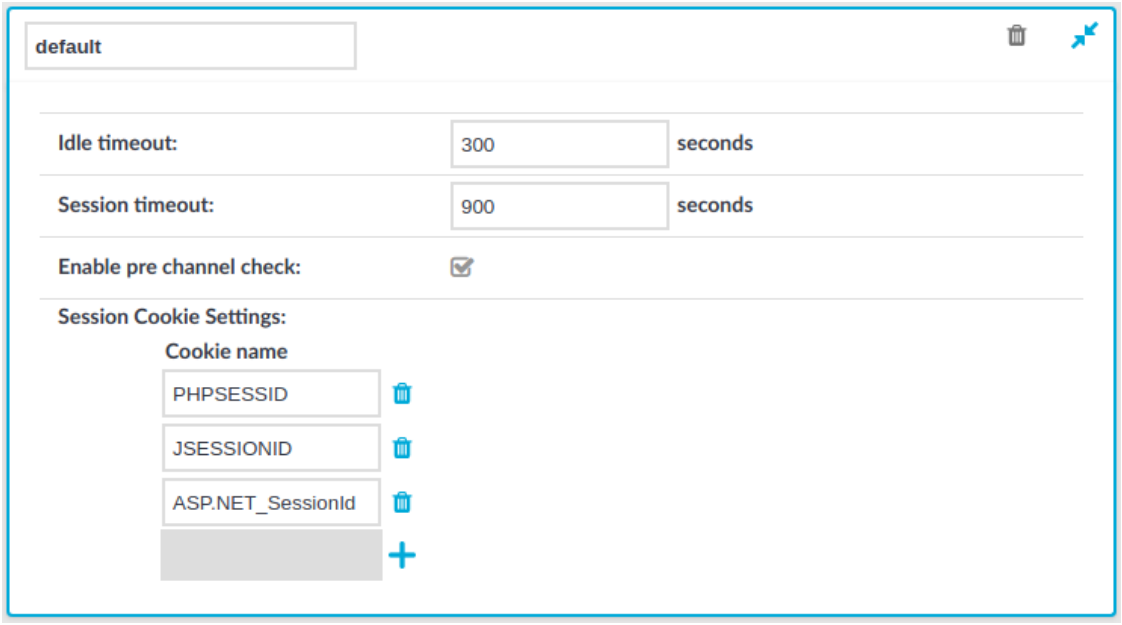
CAUTION:

Modifying timeout settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

To create a new HTTP settings profile or edit an existing one

1. Navigate to the **Settings** tab of the **HTTP Control** menu item and click + to create a HTTP setting profile. Enter a name for the profile (for example `http_special`).

Figure 169: HTTP Control > Settings — Creating and editing protocol-level HTTP settings



2. Click to display the parameters of the profile.
3. Modify the parameters as needed. The following parameters are available:
   - **Idle timeout**: Timeout value for the session in seconds. To avoid early timeout, set it to a larger value, for example a week (604800 seconds).

     CAUTION:

     Determining if a connection is idle is based on the network traffic generated by the connection, not the activity of the user. For example, if an application or the taskbar of a graphical desktop displays the time which is updated every minute, it generates network traffic every minute, negating the effects of timeout values greater than one minute and preventing One Identity Safeguard for Privileged Sessions (SPS) from closing the connection.
   - **Session timeout**: Timeout value for the session in seconds.
   - **Enable pre channel check**: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all, SPS does not establish the server-side connection.

     NOTE:

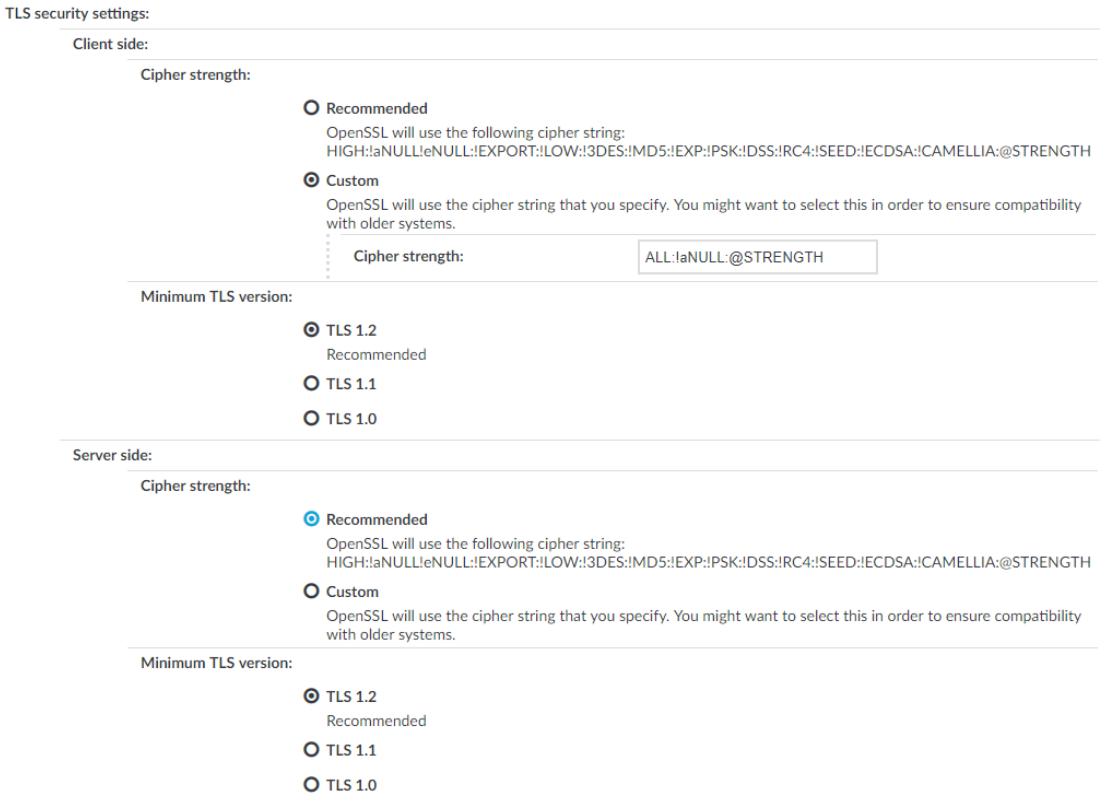
     This option cannot be disabled.

4. To distinguish the audited HTTP requests and responses based on the session cookies of web applications, click **Session Cookie Settings** > +, and enter the name of the session cookie, for example, `PHPSESSID`, `JSESSIONID`, or `ASP.NET_SessionId`. Note that the names of session cookies are case sensitive.

   Repeat this step to add multiple cookie names. Note that if you list multiple cookie names, SPS will use the first one it finds to assign the requests to a session.

5. To configure TLS security settings on both the **Client side** and the **Server side**, proceed to **TLS security settings**.

Figure 170: <Protocol> Control > Settings > TLS security settings - configuring TLS security settings



- **Cipher strength** specifies the cipher string OpenSSL will use. The following settings options are possible:
  - **Recommended**: this setting only uses ciphers with adequate security level.
  - **Custom**: this setting allows you to specify the list of ciphers you want to permit SPS to use in the connection. This setting is only recommended in order to ensure compatibility with older systems. For more details on customizing this list, check the 'openssl-ciphers' manual page on your

SPS appliance.

For example: `ALL:!aNULL:@STRENGTH`

- **Minimum TLS version** specifies the minimal TLS version SPS will offer during negotiation. The following settings options are possible:
  - **TLS 1.2**: this setting will only offer TLS version 1.2 during negotiation. This is the recommended setting.
  - **TLS 1.1**: this setting will offer TLS version 1.1 and later versions during negotiation.
  - **TLS 1.0**: this setting will offer TLS version 1.0 and later versions during negotiation.

NOTE:

Note that SPS only permits TLS-encrypted connections. SSLv3 is not supported.

6. Click 

.
7. Select this settings profile in the **HTTP settings** field of your connections.

ICA-specific settings

The following sections describe configuration settings available only for the ICA protocol. Use the following policies to control who, when, and how can access the ICA connection.

NOTE:

As an experimental feature, IPv6 addresses can be configured for ICA connections.

- *ICA connections*: For details, see Setting up ICA connections on page 506.
- *Channel Policy*: The channel policy determines which ICA channels (for example clipboard, file-sharing, and so on) can be used in the connection, and whether they are audited or not. The different channels may be available only under certain restrictions, as set in the channel policy. For details, see Creating and editing channel policies on page 448.
- *ICA settings*: ICA settings determine the parameters of the connection on the protocol level, including timeout value and display parameters. For details, see Creating and editing protocol-level ICA settings on page 509.
- *Deployment scenarios*: These describe the available One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a Citrix environment. For details, see One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a Citrix environment on page 510

Setting up ICA connections

This section focuses on describing the ICA-specific details of connection configuration. For a detailed description on configuring connections, see General connection settings on page 434.

CAUTION:

If the clients are accessing a remote application or desktop that is shared for Anonymous users (that is, the Users properties of the application is set to `Allow anonymous users` in the Citrix Delivery Services Console), the actual remote session will be running under an Anonymous account name (for example, `Anon001`, `Anon002`, and so on), not under the username used to access the remote server. Therefore, you need to enable usermapping to the `Anon*` usernames.

To accomplish this, create a usermapping policy and set the Username on the server option to `Anon*`, and the Groups option to `*`, then use this usermapping policy in your ICA connections. For details on using usermapping policies, see Configuring usermapping policies on page 740.

Reliable connection is also known as Common Gateway Protocol (CGP). It attempts to reconnect to the server in case of a network failure. To use this feature, enable **Reliable** and enter the default port in the **Port** field in the upper right corner.

Enable **Act as SOCKS proxy** to configure the client to use One Identity Safeguard for Privileged Sessions (SPS) as a SOCKS proxy. If you have enabled this option, you can select **Inband destination selection** as **Target**. Enter the IP address or the IP address/Prefix of the brokers (Citrix XML Brokers) used by the client in this connection policy into the **Address** field. It is also recommended to enable access to the brokers on port 443, as the clients usually try to access the broker using this port first. Disabling port 443 will cause a denied connection to appear on the SPS Search interface for every connection attempt (but the clients will be able to connect the server).

CAUTION:

SPS does not audit or monitor the traffic between the brokers and the clients in any way, and are not listed on the SPS search interface. Only the connections between the clients and the actual servers are audited.

CAUTION:

If SPS is acting as a SOCKS proxy and a client attempts to access a server that it is not permitted to access according to the configuration of SPS, SPS will deny the connection. However, the Citrix client application will automatically attempt to connect the server directly without using a proxy and will succeed if the server is directly accessible from the client. Ensure that your firewalls are configured properly to prevent such connections, as these direct connections cannot be audited by SPS.

NOTE:

When enabling **Reliable connection** or **Act as SOCKS proxy** the first time, a warning is displayed suggesting the default port to be used based on the specific settings. Also, read the tooltips on these options as they contain up-to-date information about the default port numbers.

Supported ICA channel types

The available ICA channel types and their functionalities are described below. For a list of supported client applications, see Supported protocols and client applications on page 38.

- **Drawing (Thinwire)**: Enables access to the server's desktop (screen). This channel is for remoting graphics and user input (keyboard, mouse). This channel must be enabled for ICA to work.
- **Audio Mapping**: Enable access to the sound device of the server.
- **Drive Mapping**: Enable access to the client's hard drives on the server.
- **Clipboard**: Enable access to the server's clipboard: the clipboard of the remote desktop can be pasted into local applications (and vice-versa). Note that One Identity Safeguard for Privileged Sessions (SPS) can audit the clipboard channel, but the Safeguard Desktop Player currently cannot search or display its contents.
- **Smartcard**: Enable using client side installed smartcards in server-side applications.
- **Printer (COM1)**: Enable access to the serial port COM1.
- **Printer (COM2)**: Enable access to the serial port COM2.
- **Printer (LPT1)**: Enable access to the parallel port LPT1.
- **Printer (LPT2)**: Enable access to the parallel port LPT2.
- **Printer Spooler**: Enable access to the client's printer from the remote desktops and applications.
- **HDX Mediastream**: Some user widgets (for example Flash player) will not run on the server but on the client. These widgets are controlled from the server side using this channel. This is not supported by Safeguard Desktop Player and it is disabled by default.
- **USB**: Enable using client side installed USB devices in server-side applications.
- **Seamless**: Enable seamless channels that run a single application on the ICA server, instead of accessing the entire desktop. When disabled, the application window will be accessed along with an empty desktop.
- **Speedbrowse**: Speeds up web browsing. Not currently supported by Safeguard Desktop Player, should be disabled by default.
- **Custom**: Applications can open custom channels to the clients connecting remotely to the server. Enabling the **Custom** channel allows the clients to access all of these custom channels. To permit only specific channels, enter the unique names of the channel into the **Details** field.

NOTE:

When the channel opens, there are certain cases when the remote group is not known yet. For example, in case of an RDP or ICA login screen, the drawing channel has to be opened first to properly display the logon screen. Only those channel rules will apply, where the **Remote group** field is empty. In case of network level authentication, all required information is present already so this limitation does not apply.

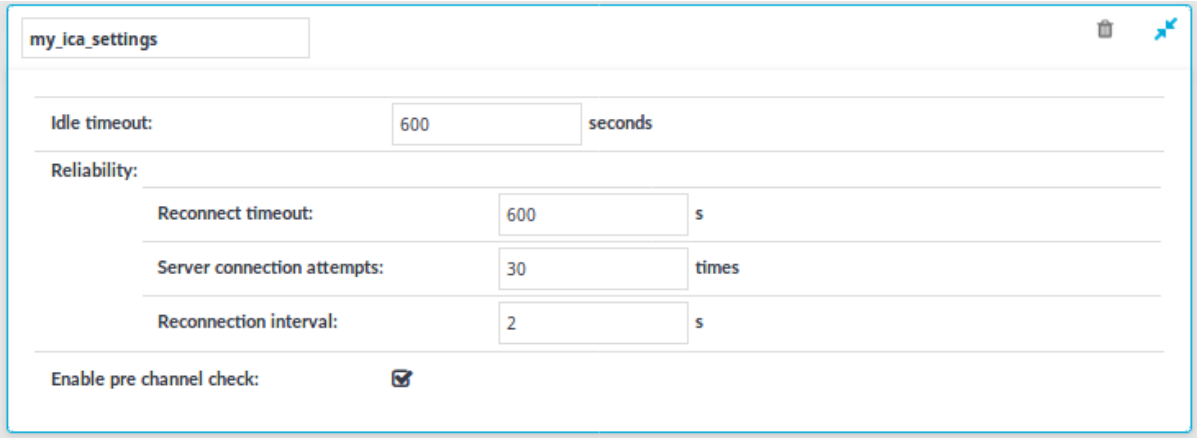
NOTE:

Multi-stream ICA is not supported in SPS 6.0.13.

Creating and editing protocol-level ICA settings

ICA settings determine the parameters of the connection on the protocol level, including timeout value, and so on.

Figure 171: ICA Control > Settings — ICA settings



CAUTION:

Modifying the ICA settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

To create a new ICA settings profile or edit an existing one

1. Navigate to the **Settings** tab of the **ICA Control** menu item and click + to create an ICA setting profile. Enter a name for the profile (for example `ica_special`).
2. Click ⤢ to display the parameters of the ICA connection.
3. Modify the parameters as needed. The following parameters are available:
    - **Idle timeout**: Connection timeout value in milliseconds. To avoid early timeout, set it to a larger value, for example a week (604800 seconds).

      CAUTION:

      Determining if a connection is idle is based on the network traffic generated by the connection, not the activity of the user. For example, if an application or the taskbar of a graphical desktop displays the time which is updated every minute, it generates network traffic every minute, negating the effects of timeout values greater than one minute and preventing One Identity Safeguard for Privileged Sessions (SPS) from closing the connection.

- **Reconnect timeout**: How many seconds SPS waits for reconnections when reliable connections are used. Reliable connections use the Common Gateway Protocol (CGP).
- **Server connection attempts**: How many times SPS tries to connect to the target server.
- **Reconnection intervals**: How many seconds SPS waits between two connection attempts on the server side.
- **Enable pre channel check**: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all, SPS does not establish the server-side connection.

NOTE:

Reliability settings only apply if you have enabled **Reliable connection** in **ICA Control > Connections**.

4. Click Commit.
5. Select this settings profile in the **ICA settings** field of your connections.

One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a Citrix environment

This section enlists the available One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a Citrix environment. The text on the arrows are formatted in (<step number>) <target port> format. The target ports define the protocols used in the communication:

- *80*: Web service, HTTP: the list of available resources fetched in an XML format from the broker (v12 and v11 with Citrix Virtual Apps (formerly known as Citrix XenApp) only). The broker sends all the necessary information, including secure gateway and server addresses to the client.
- *8080*: XML service, HTTP+XML: application discovery, load balancing (v12 and v11 with Citrix Virtual Apps (formerly known as Citrix XenApp) only), used to fetch target to the application/desktop by the client from the broker (used for load balancing, and so on).
- *443*: XML service access or SOCKS/ICA or CGP/ICA wrapped in TLS. The client communicates with the secure gateway on this port for everything.
- *1080*: SOCKS. The client can be configured to access the target server and the broker using a SOCKS proxy.
- *1494*: Plain ICA.
- *2598*: CGP/ICA (reliable mode enabled).

CAUTION:

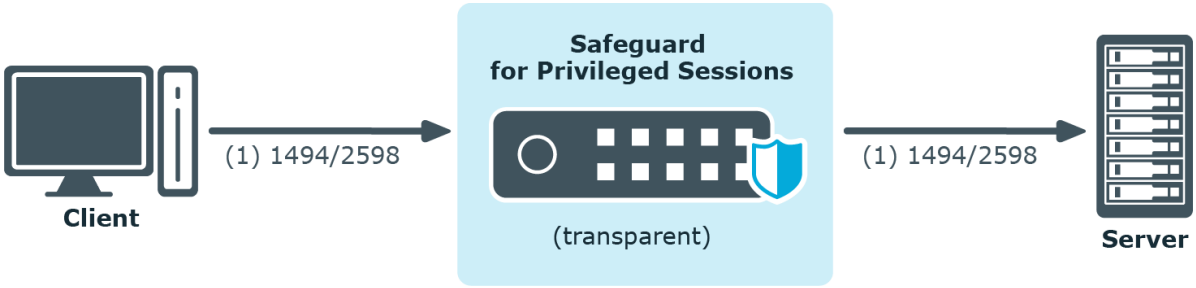
Accessing Citrix Virtual Desktops (formerly known as Citrix XenDesktop) is supported only in the following scenarios. Only reliable connections (CGP) are supported.

- **Client - Broker - SPS - Server (Transparent mode) on page 511**
- **Client - Broker - SPS as socks proxy - Server on page 513**

Client - SPS - Server (Transparent mode)

The SPS is deployed between the client and the server and the clients use predefined connection files or Program Neighbourhood, without a broker or secure gateway. The clients try to connect to their original ICA/CGP server.

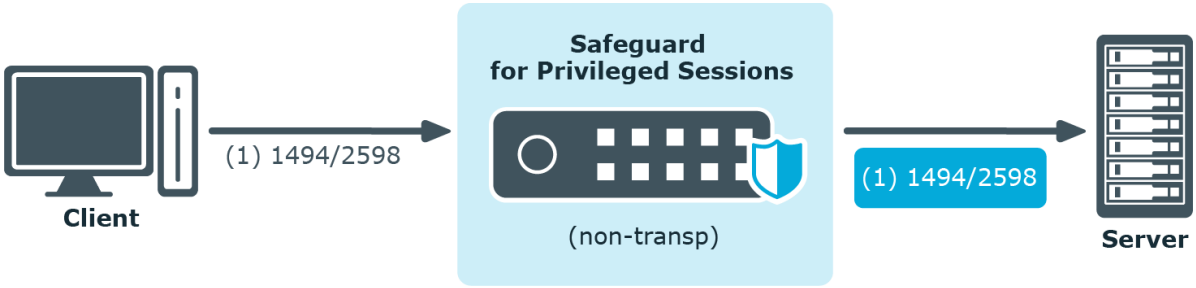
Figure 172: Client - SPS - Server (Transparent mode)



Client - SPS - Server (Non-transparent mode)

The SPS is deployed between the client and the server and the clients use predefined connection files or Program Neighbourhood, without a broker or secure gateway. The clients try to connect to SPS, which can distinguish between the potential targets for example by source IP, or by having multiple IP addresses itself.

Figure 173: Client - SPS - Server (Non-transparent mode)

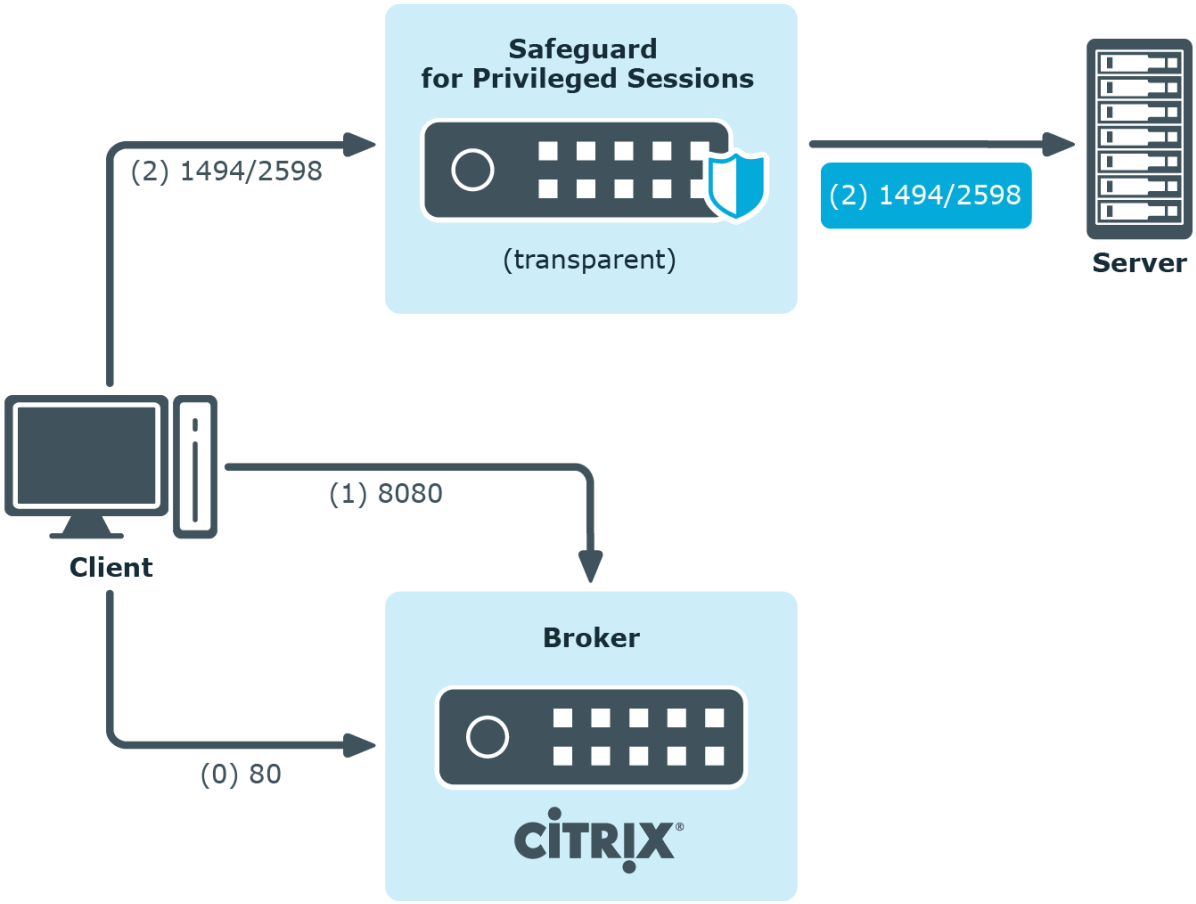


Client - Broker - SPS - Server (Transparent mode)

The clients are using a farm broker which gives them a list of the available applications and servers, but they do not use a secure gateway in the network. The SPS is placed between

the clients and the servers in transparent mode, and it catches the connections when the clients try to connect to the server IP addresses they got from the broker.

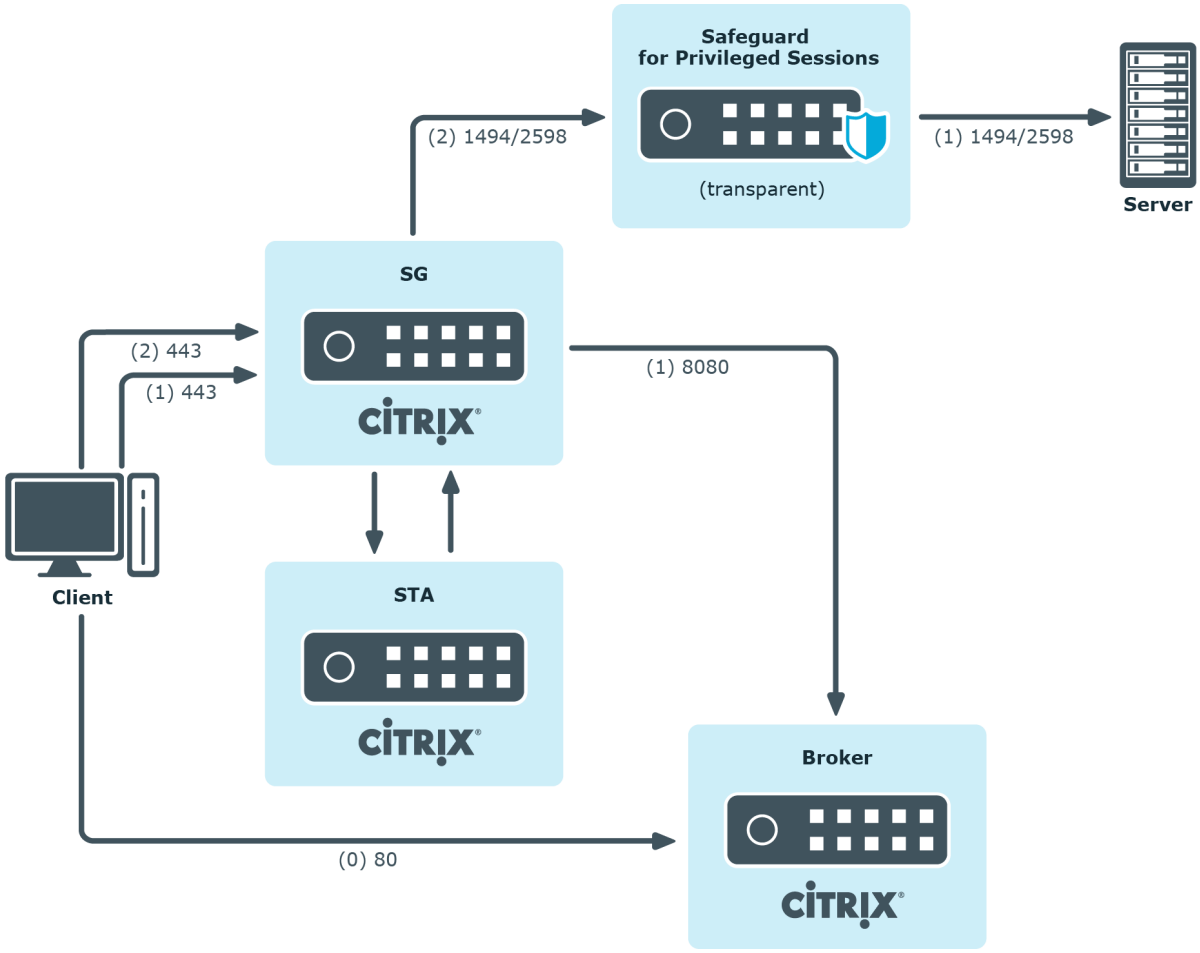
Figure 174: Client - Broker - SPS - Server (Transparent mode)



Client - Broker - original secure gateway - Secure Ticket Authority (STA) - SPS - Server

In this setup, a secure gateway is used in the network and the SPS is placed between this gateway and the servers in transparent mode. The clients connect to the broker for the list of available applications/servers and then make their further connections through the original secure gateway. That gateway forwards the connections either to the broker or to the CGP/ICA servers, which latter the SPS intercepts and audits/controls.

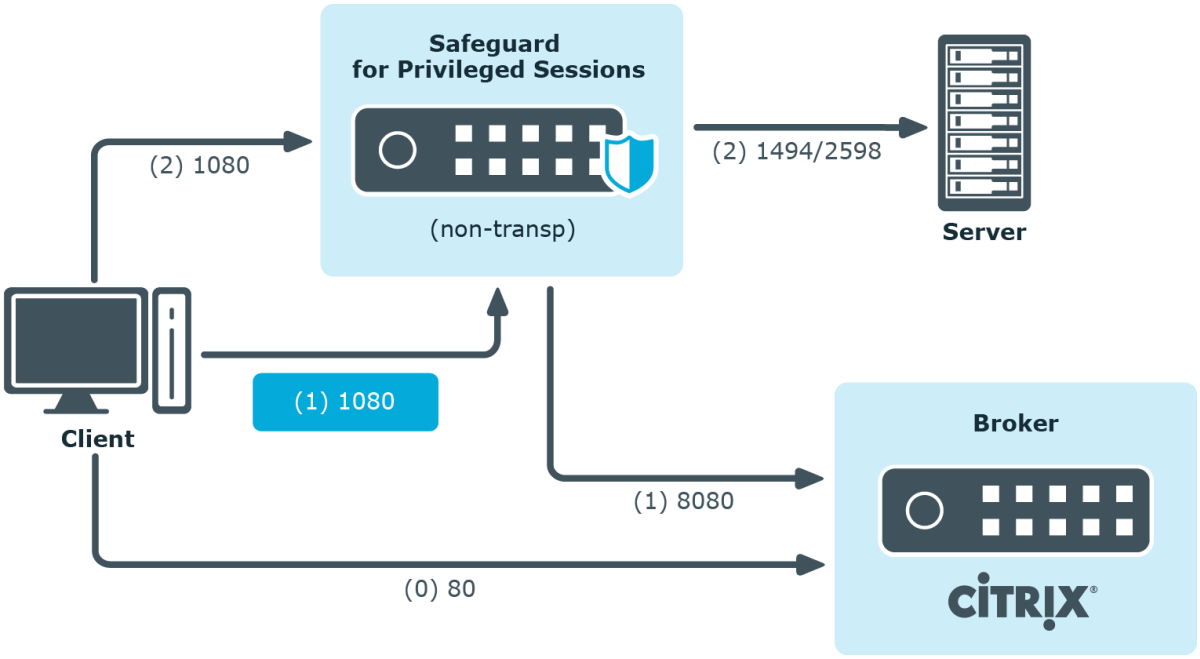
Figure 175: Client - Broker - original secure gateway - Secure Ticket Authority (STA) - SPS - Server



Client - Broker - SPS as socks proxy - Server

In this setup, the SPS acts as a SOCKS proxy for the client. It can be set either manually or specified by the broker. The client then makes all its connections to the broker or to the server using SPS as a proxy and hence it can audit/control these connections.

Figure 176: Client - Broker - SPS as socks proxy - Server



To configure such a scenario, you must set the ICA Connection Policy as follows:

- Enter the IP address of SPS into the **To** field. This must be the public IP address that the clients will target.
- Select **Inband destination selection**, and list the IP addresses or networks of target servers in the **Targets** field. (For details, see Configuring inband destination selection on page 443.)
- Select **Act as a SOCKS proxy**.
- Add the IP addresses of your brokers to the **Brokers** field.

Troubleshooting Citrix-related problems

Accessing Citrix servers using the Remote Desktop Protocol

Accessing Citrix servers using the Remote Desktop Protocol may fail in certain situations, and the connection is terminated with the `ERROR: error while decompressing packet` error message on the client, or with the `Event56, TermDD, The Terminal Server security layer detected an error in the protocol stream and has disconnected the client.` message on the server.

To overcome this problem, modify the settings of the network card of the server, and disable the **Large Send Offload** option.

NOTE:

The problem is not related to using One Identity Safeguard for Privileged Sessions (SPS) in your environment.

11

RDP-specific settings

The following sections describe configuration settings available only for the RDP protocol. Use the following policies to control who, when, and how can access the RDP connection.

- *Channel Policy*: The channel policy determines which RDP channels (for example clipboard, file-sharing, and so on) can be used in the connection, and whether they are audited or not. The different channels may be available only under certain restrictions, as set in the channel policy. For details, see Creating and editing channel policies on page 448.
- *RDP settings*: RDP settings determine the parameters of the connection on the protocol level, including timeout value, display parameters, and the version of RDP permitted. For details, see Creating and editing protocol-level RDP settings on page 519.
- *Domain membership*: When using Network Level Authentication (CredSSP) One Identity Safeguard for Privileged Sessions (SPS) must be a member of the domain. For details, see Network Level Authentication (NLA) with domain membership on page 524.
- *TLS-encrypted connections*: For details on how to setup TLS-encrypted RDP connections, see Enabling TLS-encryption for RDP connections on page 529 and Verifying the certificate of the RDP server in encrypted connections on page 527.
- *SPS as a Remote Desktop Gateway*: For details on how to configure SPS to accept connections using the Remote Desktop Gateway Server Protocol, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on page 531.
- *Content Policy*: Content policies allow you to inspect the content of the connections for various text patterns, and perform an action if the pattern is found. For example, SPS can send an e-mail alert if a specific window title appears in RDP and VNC connections. For details, see Creating a new content policy on page 452.
- *Authentication and Authorization plugin*:

  One Identity Safeguard for Privileged Sessions (SPS) provides a plugin framework to integrate SPS to external systems to authenticate or authorize the user before authenticating on the target server. Such plugins can also be used to request additional information from the users, for example, to perform multi-factor authentication.

  For details, see Integrating external authentication and authorization systems on page 768.

Using multiple monitors (Multimon) is supported. To enable Multimon, use one of the following three methods:

- enable **Display > Use all my monitors for the remote session** option in the Remote Desktop Client (`mstsc.exe`) window of the client machine
- use the `/multimon` switch on the `mstsc.exe` command line
- add the `use multimon:i:1` row to the RDP file

NOTE:

The **Maximum display width** and **Maximum display height** options should be high enough to cover the combined resolution of the client monitor setup. Connections that exceed these limits will automatically fail. Make sure to adjust these settings if your clients use multiple monitors. For example, if your clients use two monitors that have a resolution of 1920x1080 pixels each, set **Maximum display width** to 4000, and **Maximum display height** to 2200.

Limitations

The RDP connection fails due to the following Windows-side settings:

- If the target user is a member of the Protected Users security group.
- If Remote Credential Guard is enabled and used.
- If Restricted Admin mode is enabled and used. For more information on how to enable or disable the Restricted Admin mode on Windows, see Remote Desktop Services: Enable Restricted Admin mode.

Supported RDP channel types

The available RDP channel types and their functionalities are described below. For details on configuring Channel Policies, see Creating and editing channel policies on page 448. For a list of supported client applications, see Supported protocols and client applications on page 38.

- **Drawing**: Enables access to the server's graphical desktop (screen). This channel must be enabled for RDP to work.

  NOTE:

  In case the Drawing channel is disabled and the load of One Identity Safeguard for Privileged Sessions (SPS) is high, or the connection requires four-eyes authorization and the Authorizer is slow to accept the connection, the client might receive the following error message:

  ```
  The Remote Desktop Gateway server administrator has ended the
  connection.
  Try reconnecting later or contact your network administrator for
  assistance
  ```

- **Clipboard**: Enables access to the server's clipboard: the clipboard of the remote desktop can be pasted into local applications (and vice-versa). Note that SPS can audit the clipboard channel, and that files transferred via the clipboard can be audited Configuring SPS to enable exporting files from audit trails after RDP file transfer via clipboard.

  If the **Clipboard** channel is enabled, it implicitly enables copying files as well, as the user can simply copy-paste the file. Copy-pasted files will not be visible in the logs or the **File operations** column of the **Search** page. To ensure that SPS records file transfer events, you must disable the **Clipboard** channel.
- **Redirects**: Enables access to every device redirection available in RDP, like file-sharing, printer sharing, device (for example, CD-ROM) sharing, and so on.
  - To make the list of file operations available in the **File operations** column of the **Search** page, navigate to the Channel Policies page of the protocol, and enable the **Log file transfers to database** option. This option is disabled by default.
  - To send the file operations into the system log, enable the **Log file transfers to syslog** option. This option is disabled by default.

    NOTE:

    Turning logging on might result in a slight performance penalty. If traffic load slows processes down, disable the option.

  To enable only specific types of redirections, use the following channels:
  - **Serial redirect**: Enables access to serial-port redirections.
  - **Parallel redirect**: Enables access to parallel-port redirections.
  - **Printer redirect**: Enables access to shared printers.

    When enabling printer redirection, you may need to use TSVCTKT and XPSRD channels — these enable XPS printing.

    Note that these channels are dynamic virtual channels and you have to be enable them using the **Custom** channel type.

    For more information on TSVCTKT and XPSRD channels, see *section 2.1 Transport* in Microsoft Technical Document [MS-RDPEXPS].

    Before consulting the cited Microsoft Technical Document, it is recommended to start by reading [MS-RDSOD]: Remote Desktop Services Protocols Overview.
  - **Disk redirect**: Enables access to shared disk drives.
    - To make the list of file operations available in the **File operations** column of the **Search** page, navigate to the Channel Policies page of the protocol, and enable the **Log file transfers to database** option. This option is disabled by default.

- To send the file operations into the system log, enable the **Log file transfers to syslog** option. This option is disabled by default.

  NOTE:

  Turning logging on might result in a slight performance penalty. If traffic load slows processes down, disable the option.

  If the **Clipboard** channel is enabled, it implicitly enables copying files as well, as the user can simply copy-paste the file. Copy-pasted files will not be visible in the logs or the **File operations** column of the **Search** page. To ensure that SPS records file transfer events, you must disable the **Clipboard** channel.

- **SCard redirect**: Enables access to shared SCard devices.

To permit only specific redirections, enter the unique name of the redirection into the **Details** field. For example, if you want to enable access only to the shared disk drive C:, enable the **Disk redirect** channel and enter `C:` into the **Permitted devices** field. Note that the name of the device comes from the device itself, so it is case sensitive, and may not always be reliable from a security point of view.

- **Sound**: Enables access to the sound device of the server.
- **Custom**: Applications can open custom channels to the clients connecting remotely to the server. Enabling the **Custom** channel allows the clients to access all of these custom channels. To permit only specific channels, enter the unique names of the channel into the **Permitted devices** field.

  For example, to monitor RemoteApp connections, you need to configure custom channels. For more information, see Configuring RemoteApps on page 542.
- **Seamless**: Enables seamless channels that run a single application on the RDP server, instead of accessing the entire desktop.
- **Dynamic virtual channel**: Enables the server to open channels back to the client dynamically. To restrict which dynamic channels are permitted, select **Channel details**, click + and enter the name of the permitted channel.

  Additionally, you may need to use one or more of the following:

  - PNPDR and FileRedirectorChannel channels: Enable Plug and Play devices.

    For more information, see *section 2.1 Transport* in Microsoft Technical Document [MS-RDPEPNP].

  - URBDRC channels: Enable USB redirection.

    For more information, see *section 2.1 Transport* in Microsoft Technical Document [MS-RDPEUSB].

  Before consulting any of the listed Microsoft Technical Documents, it is recommended to start by reading [MS-RDSOD]: Remote Desktop Services Protocols Overview.

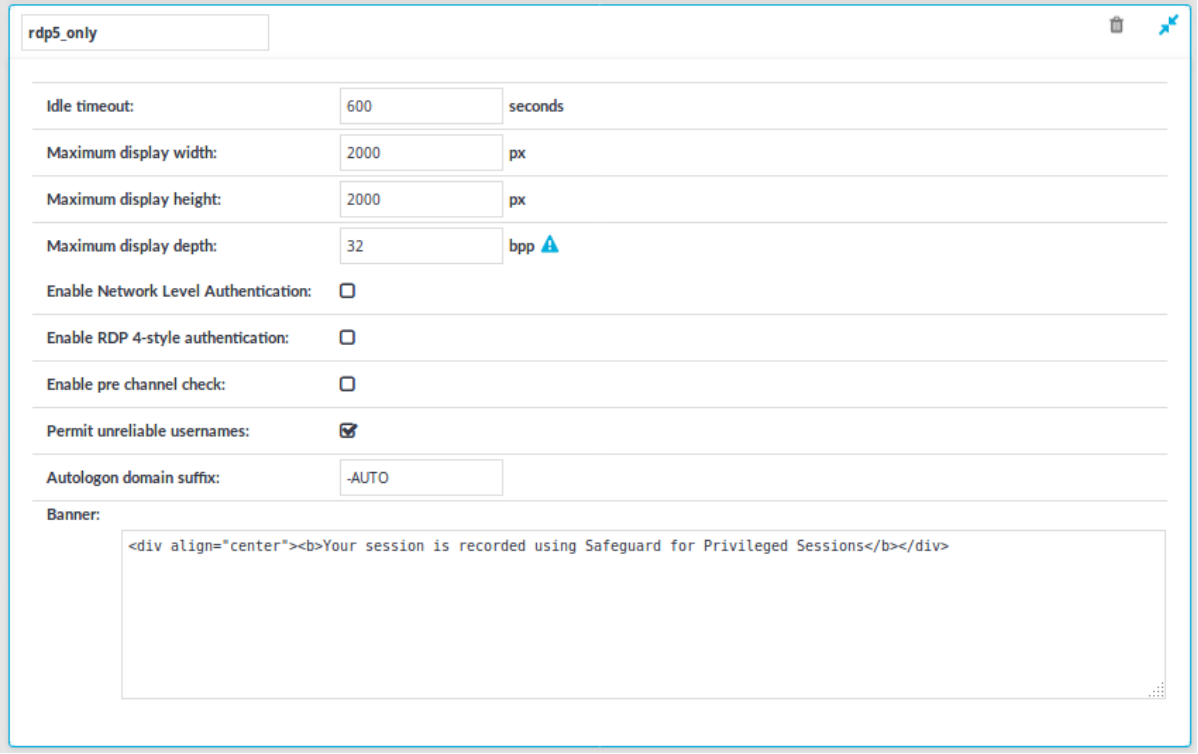
NOTE:

When the channel opens, there are certain cases when the remote group is not known yet. For example, in case of an RDP or ICA login screen, the drawing channel has to be opened first to properly display the logon screen. Only those channel rules will apply, where the **Remote group** field is empty. In case of network level authentication, all required information is present already so this limitation does not apply.

Creating and editing protocol-level RDP settings

RDP settings determine the parameters of the connection on the protocol level. For example, timeout value, the version of RDP permitted in the connection, and display parameters.

Figure 177: RDP Control > Settings — RDP settings



CAUTION:

Modifying the RDP settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

To create a new RDP settings profile or edit an existing one

1. Navigate to **RDP Control > Settings** and click ➕ to create an RDP setting profile. Enter a name for the profile (for example, `rdp5only`).
2. Click ⤢ to display the parameters of the RDP connection.
3. Modify the parameters as needed. The following parameters are available:
   - **Idle timeout**: Timeout value for the connection in seconds. To avoid early timeout, set it to a larger value, for example a week (604800 seconds).

     ⚠ **CAUTION:**

     Determining if a connection is idle is based on the network traffic generated by the connection, not the activity of the user. For example, if an application or the taskbar of a graphical desktop displays the time which is updated every minute, it generates network traffic every minute, negating the effects of timeout values greater than one minute and preventing One Identity Safeguard for Privileged Sessions (SPS) from closing the connection.

     ⚠ **CAUTION:**

     If the value is set below `31` seconds, MSTSC can fail and prevent new connections if Act as a Remote Desktop Gateway is enabled in RDP Control > Connections. To prevent this, set the Idle timeout value to at least `31` seconds.

   - **Maximum display width**: The maximum allowed width of the remote desktop in pixels (for example `1024`).

     ℹ NOTE:

     The **Maximum display width** and **Maximum display height** options should be high enough to cover the combined resolution of the client monitor setup. Connections that exceed these limits will automatically fail. Make sure to adjust these settings if your clients use multiple monitors. For example, if your clients use two monitors that have a resolution of 1920x1080 pixels each, set **Maximum display width** to 4000, and **Maximum display height** to 2200.

   - **Maximum display height**: The maximum allowed height of the remote desktop in pixels (for example `768`).

     ℹ NOTE:

     The **Maximum display width** and **Maximum display height** options should be high enough to cover the combined resolution of the client monitor setup. Connections that exceed these limits will automatically fail. Make sure to adjust these settings if your clients use multiple monitors. For example, if your clients use two monitors that have a resolution of 1920x1080 pixels each, set **Maximum display width** to 4000, and **Maximum display height** to 2200.

- **Maximum display depth**: The maximum allowed color depth the remote desktop in bits (for example 24). The following values are valid: 8, 15, 16, 24.

  CAUTION:

  - **Using 32-bit color depth is currently not supported: client connections requesting 32-bit color depth automatically revert to 24-bit.**
  - **Certain Windows versions do not support 24-bit color depth. In this case, those versions can only be displayed in 16-bit color depth. SPS automatically changes its settings to 16-bit.**

- **Enable Network Level Authentication**: Select this option to enable the use of Network Level Authentication (NLA, also called Credential Security Service Provider or CredSSP).

  - If you enable this option, you have to configure SPS to join your domain. If you use a domain, then the target servers and SPS must be in the same domain, or you must establish trust between the domains that contain the target servers and SPS. For details on the type of trust required, see Using One Identity Safeguard for Privileged Sessions (SPS) across multiple domains on page 526. For details on configuring SPS to join a domain, see Network Level Authentication (NLA) with domain membership on page 524.
  - If you cannot or do not want to join SPS to the domain, see Network Level Authentication without domain membership on page 527.

  Note the following points:

  - SSL-encrypted connections do not require this option, it is only needed for Network Level Authentication (NLA).
  - Smartcard authentication cannot be used when the **Enable Network Level Authentication** option is enabled.

  CAUTION:

  To access hosts running Windows 2008 Server R2 using Network Level Authentication (NLA), select the *Enable RDP4 style authentication* option as well.

- **Enable RDP4 style authentication**: Select this option to enable RDP4 authentication within the RDP5 protocol. This might be needed for compatibility reasons with certain client applications.
- **Enable pre channel check**: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all, SPS does not establish the server-side connection.
- **Permit unreliable usernames**: SPS automatically terminates RDP connections if it cannot reliably extract the username from the RDP connection.

Enable this option to permit connections with unreliable usernames. For details on ensuring that the usernames in RDP connections are reliable, see Usernames in RDP connections on page 539.

Known issue

When accessing a Windows Server 2003 R2 host, the **Permit unreliable usernames** option is disabled, and the username is unreliable, SPS terminates the connection, but only after the user logs in. As a result, the session is not closed on the server-side.

- **Autologon domain suffix**: Enter the suffix that the client will append to the domain when using autologon in conjunction with Network Level Authentication (CredSSP).

4. To display a banner message to the clients before authentication, enter the message into the Banner field. For example, this banner can inform the users that the connection is audited. SPS displays this banner in a graphical window that has only an OK button. Note the following points:
   - You can write a plain-text or a basic HTML-formatted banner.

     CAUTION:

     If the banner is overly complex HTML using deeply embedded structures, displaying the banner will fail, causing the RDP connections to time out.

   - When using HTML markup, the entire banner must be a single HTML object (for example, a div).

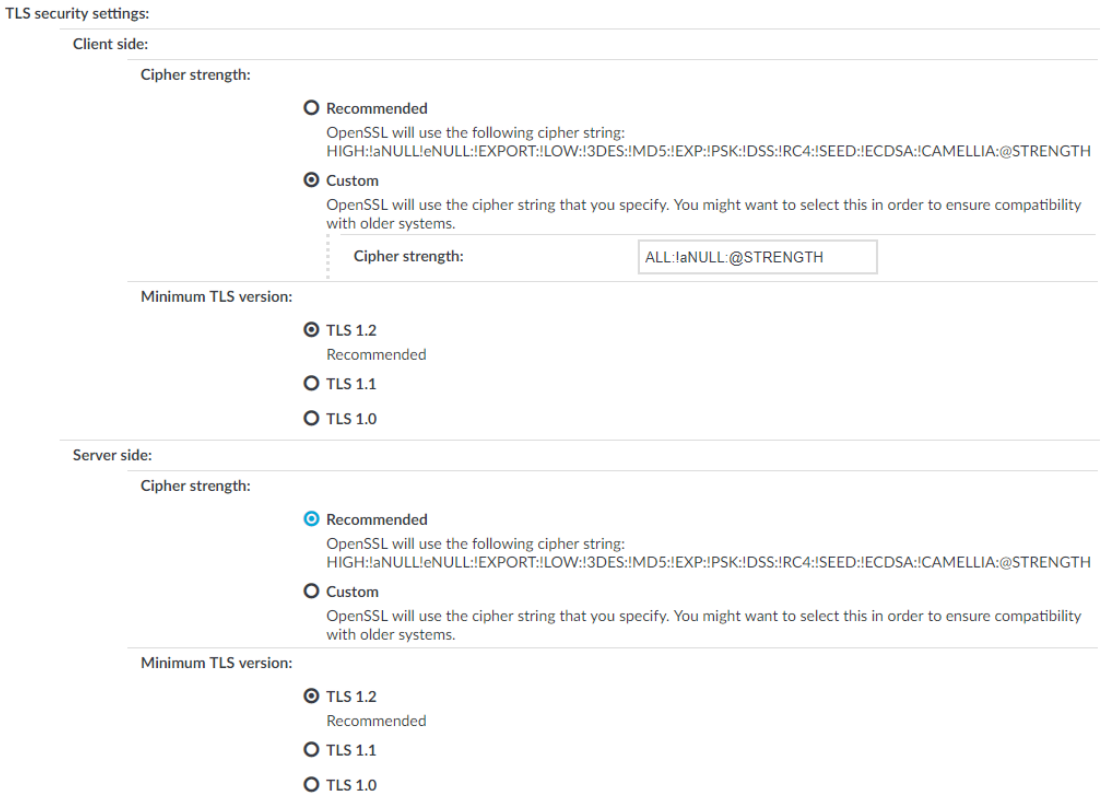
     ```
     <div align="center"><b>Your session is recorded using Privileged
     Session Monitoring</b></div>
     ```

   - In HTML, you can embed images (for example, a company logo) as data URLs in an img tag:

     ```
     To include a logo or other image, use a base64-encoded data url
     within an, like this: <img alt="Embedded Image"
     src="..." />.
     ```

   - Note that while you can include links in the text, your users cannot click or copy them.
5. To configure TLS security settings on both the **Client side** and the **Server side**, proceed to **TLS security settings**.

Figure 178: <Protocol> Control > Settings > TLS security settings - configuring TLS security settings



- **Cipher strength** specifies the cipher string OpenSSL will use. The following settings options are possible:
  - **Recommended**: this setting only uses ciphers with adequate security level.
  - **Custom**: this setting allows you to specify the list of ciphers you want to permit SPS to use in the connection. This setting is only recommended in order to ensure compatibility with older systems. For more details on customizing this list, check the 'openssl-ciphers' manual page on your SPS appliance.

    For example: `ALL:!aNULL:@STRENGTH`
- **Minimum TLS version** specifies the minimal TLS version SPS will offer during negotiation. The following settings options are possible:
  - **TLS 1.2**: this setting will only offer TLS version 1.2 during negotiation. This is the recommended setting.
  - **TLS 1.1**: this setting will offer TLS version 1.1 and later versions during negotiation.

- **TLS 1.0**: this setting will offer TLS version 1.0 and later versions during negotiation.

NOTE:

Note that SPS only permits TLS-encrypted connections. SSLv3 is not supported.

NOTE:

TLS 1.1 and 1.2 support for Windows 7 Service Pack 1 (SP1) and for Windows Server 2008 R2 Service Pack 2 (SP2) is not available by default. For more information about the requirements and process of enabling this feature, click here or contact our Support Team.

6. Click 

.
7. Select this settings profile in the **RDP settings** field of your connections.

Network Level Authentication (NLA) with One Identity Safeguard for Privileged Sessions (SPS)

You can configure One Identity Safeguard for Privileged Sessions (SPS) to use Network Level Authentication (NLA) in two different scenarios.

Network Level Authentication (NLA) with domain membership

To use Credential Security Service Provider (CredSSP, also called Network Level Authentication or NLA) when One Identity Safeguard for Privileged Sessions (SPS) is member of the domain. If you cannot or do not want to join SPS to the domain, see "Network Level Authentication without domain membership" in the Administration Guide.

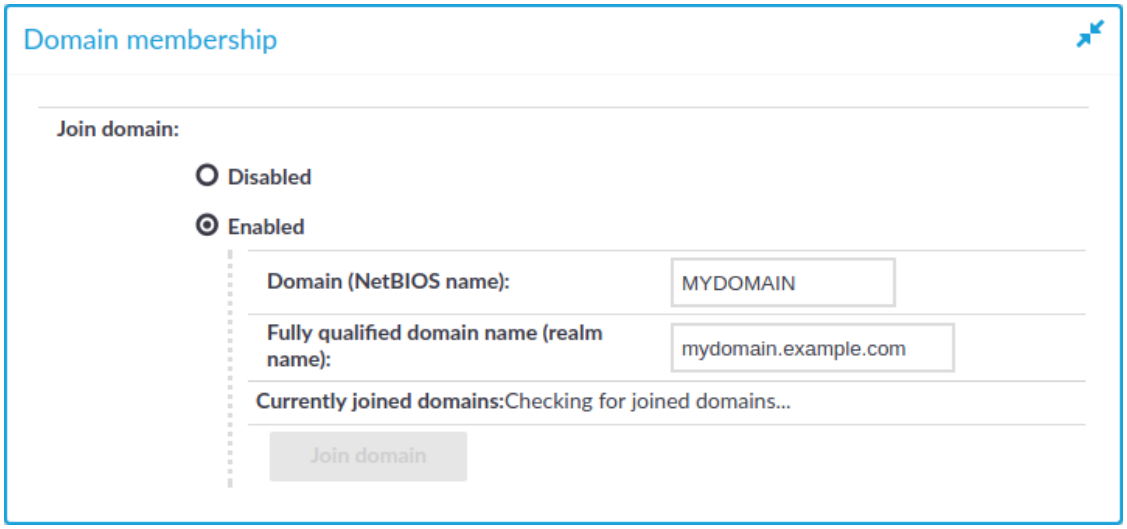
Prerequisites

The target servers and SPS must be in the same domain, or you must establish trust between the domains that contain the target servers and SPS. For details on the type of trust required, see "Using One Identity Safeguard for Privileged Sessions (SPS) across multiple domains" in the Administration Guide.

To use NLA with domain membership

1. Navigate to **RDP Control > Settings**, and select the **Enable Network Level Authentication** option. (If you will have connections that will not use Network Level Authentication, create a separate RDP Settings policy).
2. Navigate to **RDP Control > Domain membership**.
3. Enter the name of the domain (for example `mydomain`) into the **Domain** field.

   Figure 179: RDP Control > Domain membership — Joining a domain

4. Enter the name of the realm (for example `mydomain.example.com`) into the **Full domain name** field.

   NOTE:

   Ensure that your DNS settings are correct and that the full domain name can be resolved from SPS. To check this, navigate to **Basic Settings > Troubleshooting > Ping**, enter the full domain name into the **Hostname** field, and select **Ping host**.

5. Click Commit.
6. Click **Join domain**. A pop-up window is displayed.
7. SPS requires an account to your domain to be able to join the domain. Enter the following information:
   - The name of the user into the **Username** field.
   - The password into the **Password** field.

     NOTE:

     One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

- The name of your domain controller into the **Domain controller** field. If you leave this field blank, SPS tries to find the domain controller automatically.

  NOTE:

  Ensure that your DNS settings are correct and that the hostname of the domain controller can be resolved from SPS. To check this, navigate to **Basic Settings > Troubleshooting > Ping**, enter the name of the domain controller into the **Hostname** field, and select **Ping host**.

- The organizational unit (OU) into the **Organization unit** field.

  The OU string reads from top to bottom without RDNs, and is delimited by a '/'. Note that '\' is used for escape by both the shell and ldap, so it may need to be doubled or quadrupled to pass through, and it is not used as a delimiter.

8. Click **Join domain**.
9. If successful, SPS displays the name of the domain it joined.

   NOTE:

   If you need SPS to leave the domain for some reason, click **Leave domain**.

Using One Identity Safeguard for Privileged Sessions (SPS) across multiple domains

If your users are in a domain (`EXAMPLE-DOMAIN`), One Identity Safeguard for Privileged Sessions (SPS) is also in that domain (`EXAMPLE-DOMAIN`), but your users need to access servers that are in a different domain (`OTHER-DOMAIN`), you must establish a level of trust between the domains. This is summarized in the following table.

Domain username of the client	Domain of the target server	Result
EXAMPLE-DOMAIN\my-username	EXAMPLE-DOMAIN	Connection is established
EXAMPLE-DOMAIN\my-username	OTHER-DOMAIN	If OTHER-DOMAIN trusts EXAMPLE-DOMAIN, the connection is established
OTHER-DOMAIN\my-username	OTHER-DOMAIN	If two-way trust is established between OTHER-DOMAIN and EXAMPLE-DOMAIN, the connection is established
OTHER-DOMAIN\my-username	EXAMPLE-DOMAIN	If two-way trust is established between OTHER-DOMAIN and EXAMPLE-DOMAIN, the connection is established

NOTE:

If you use an LDAP database when using SPS accross multiple domains, LDAP will only use the username without the domain name to verify the group membership.

Network Level Authentication without domain membership

There are scenarios when you want to use One Identity Safeguard for Privileged Sessions (SPS) to monitor RDP access to servers that accept only Network Level Authentication (NLA, also called CredSSP), but the client, SPS, and the server are not in the same domain (there is no trust between their domains), or any of them is not in a domain at all. For example, you cannot add SPS to the domain for some reason, or the RDP server is a standalone server that is not part of a domain. The following table shows such a scenario.

User	Client domain membership	SPS domain membership	Server domain membership
local or any domain	any domain	not a domain member, or other than <server-domain>	<server-domain>

Limitations

- Server-side redirection may not work.

To use NLA without domain membership

1. Navigate to **RDP Control > Settings**, and select the RDP settings policy that you use in your connection policies.
2. Clear the **Enable Network Level Authentication > Require domain membership** option.
3. Click Commit.

Verifying the certificate of the RDP server in encrypted connections

By default, One Identity Safeguard for Privileged Sessions (SPS) accepts any certificate shown by the server. The following describes how to accept only verified certificates.

To accept only verified certificates

1. Create a list of trusted CA certificates that will be used to verify the certificate of the server. For details, see Verifying certificates with Certificate Authorities on page 476.
2. Navigate to **RDP Control > Connections** and select the connection policy to modify.
3. Select **Verify server certificate**.

NOTE:

This setting has no effect if the session uses Network Level Authentication, because in such cases SPS uses a different method to validate the server certificate.

When using Network Level Authentication (NLA, also called CredSSP), there is no verification performed in the TLS layer due to the TLS session-binding. For more information on TLS session-binding, see section *[MS-CSSP]: Credential Security Support Provider (CredSSP) Protocol* in the Microsoft documentation.

Figure 180: RDP Control > Connections — Using SSL-encryption in RDP connections



4. Select the CA list to use for verifying the certificate of the server from the **Trusted CA list** field.
5. Click Commit.
6. (Optional) Configure your Windows servers to display a certificate signed with the above Certificate Authority for incoming RDP connections. To accomplish this, complete the following steps:
    a. Generate a certificate that contains the IP address or the hostname of the target server in its Common Name (CN) field and sign it with the Certificate Authority whose certificate you added to the **Trusted CA list** of SPS.
    b. Convert the signed certificate of the target server to PKCS12 format that includes the private key.
    c. Start the Microsoft Management Console (MMC) on the target server and select **Add Snap-in > Certificates > Computer Account**.
    d. Right-click the **Personal** store, then select **All Tasks > Import**, and select the certificate created for the server.
    e. Complete the Certificate Import Wizard, but do not select the **Extended certificate properties** option.
    f. Select **Start > Administrative tools > Remote Desktop > Remote Desktop Session Host Configuration**.
    g. Right-click the connection you want to configure and select **Properties > General**.
    h. Set the **Security layer** to **SSL**.
    i. Click **Certificate > Select** and select the imported certificate. The server will use this certificate to verify its identity for the incoming RDP connections.

Enabling TLS-encryption for RDP connections

To enable TLS-encryption in an RDP connection policy, you have two options:

- Enable Network Level Authentication (NLA, also called CredSSP). To enable NLA in RDP connections, see Network Level Authentication (NLA) with One Identity Safeguard for Privileged Sessions (SPS) on page 524. Note that Network Level Authentication uses SSL-encryption with self-signed certificates, so you do not have to configure a signing CA.
- Complete the following steps to configure TLS-encryption.

Prerequisites

Depending on your requirements, one or more of the following might be needed:

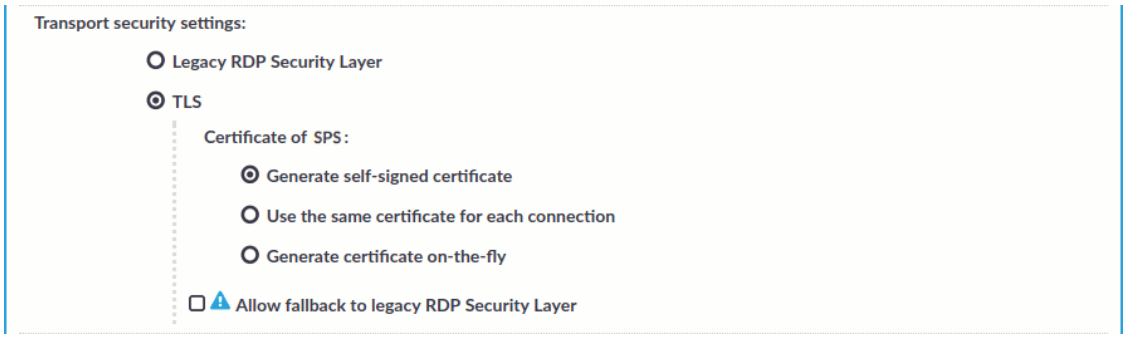
- To use the same certificate for each session, an X.509 certificate and its private key are required. One Identity Safeguard for Privileged Sessions (SPS) can display this certificate to the peers on the client side. Use your own PKI system to generate these certificates, as they cannot be created on SPS. Note that the Common Name of the certificate must contain the domain name or the IP address of target machine. otherwise the clients might reject the certificate.
- To generate certificates on-the-fly for a connection, a signing certificate authority is required. For details on creating a signing CA, see Signing certificates on-the-fly on page 478.

One Identity recommends using 2048-bit RSA keys (or stronger).

To configure TLS-encryption

1. Navigate to **RDP Control > Connections** and select the connection policy in which you want to enable TLS.

   Figure 181: RDP Control > Connections — Enabling TLS-encryption for RDP connections

2. Set the encryption settings used between the client/server and SPS in the **Transport security settings** section.

   To require encryption, select **TLS**. When the connection is encrypted, SPS has to show a certificate to the peer.

3. Select the certificate to show to the peers.

   - If you want to enable TLS-encryption, but you do not have a certificate that is generated by an external CA, or a signing CA, select **Generate self-signed certificate**. This option is selected by default.
   - To use the same certificate for every peer, complete the following steps.
     1. Generate and sign a certificate for SPS in your PKI system, and export the certificate and its private key.
     2. Select **Use the same certificate for each connection**.
     3. Select **Private key for host certificate**, click and upload the private key.
     4. Select **X.509 host certificate**, click and upload the certificate.
   - If you want to use your own Signing CA, complete the following steps.
     1. Create a certificate authority that will be used to sign the certificates that SPS shows to the peer. For details, see Signing certificates on-the-fly on page 478.
     2. Select **Generate certificate on-the-fly**.
     3. In the **Signing CA** field, select the certificate authority to use.

     NOTE:

     Import the certificate of the signing Certificate Authority to your clients. Otherwise, the client applications will display a warning because of the unknown Certificate Authority.
   - To disable TLS-encryption for RDP connections completely, select **Legacy RDP Security Layer** (also known as: Standard RDP Security). You might want to do this if you were using legacy RDP encryption, and you are experiencing compatibility issues. For example, you might experience compatibility issue when you attempt to connect to a very old Windows machine (for example, Windows Server 2003 or older).

4. (Optional) Even if you choose TLS-encryption , you have the option to choose using legacy RDP encryption as well. If you experience compatibility issues (for example, when you attempt to connect to a very old Windows machine, such as Windows Server 2003 or older) and want to allow using legacy RDP encryption if TLS-encryption is not possible, select **Allow fallback to legacy RDP Security Layer** (also known as: Standard RDP Security).

CAUTION: SECURITY HAZARD!

Selecting the Legacy RDP Security Layer or the Allow fallback to legacy RDP Security Layer options can significantly reduce the strength of the encryption used.

Selecting these options is only recommended if you cannot overcome compatibility issues in any other way.

To avoid security hazard, we recommend using TLS encryption.

5. Click Commit.

Expected result

The encryption settings are applied to the connection policy.

Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway

Remote Desktop Gateway is a Remote Desktop Services server role. This role allows authorized remote users to connect to resources located on an internal or private network from any Internet-connected device. The accessible resources can be terminal servers, remote applications, remote desktops, and so on.

The Remote Desktop Gateway Server Protocol is a remote procedure call (RPC) protocol using HTTPS as the transport mechanism, used primarily for tunneling client to server traffic across firewalls. The One Identity Safeguard for Privileged Sessions (SPS) can act as a Remote Desktop Gateway, receiving connections using the Remote Desktop Gateway Server Protocol and transferring them to the target servers using the RDP protocol.

The Remote Desktop Gateway Server Protocol enables inband destination selection, meaning that SPS can extract the address of the target server from the client connections. This greatly simplifies managing connections on SPS without having to encode the name of the target server in the username, which was problematic as the length of the username is limited on many platforms — especially in non-transparent mode.

Prerequisites

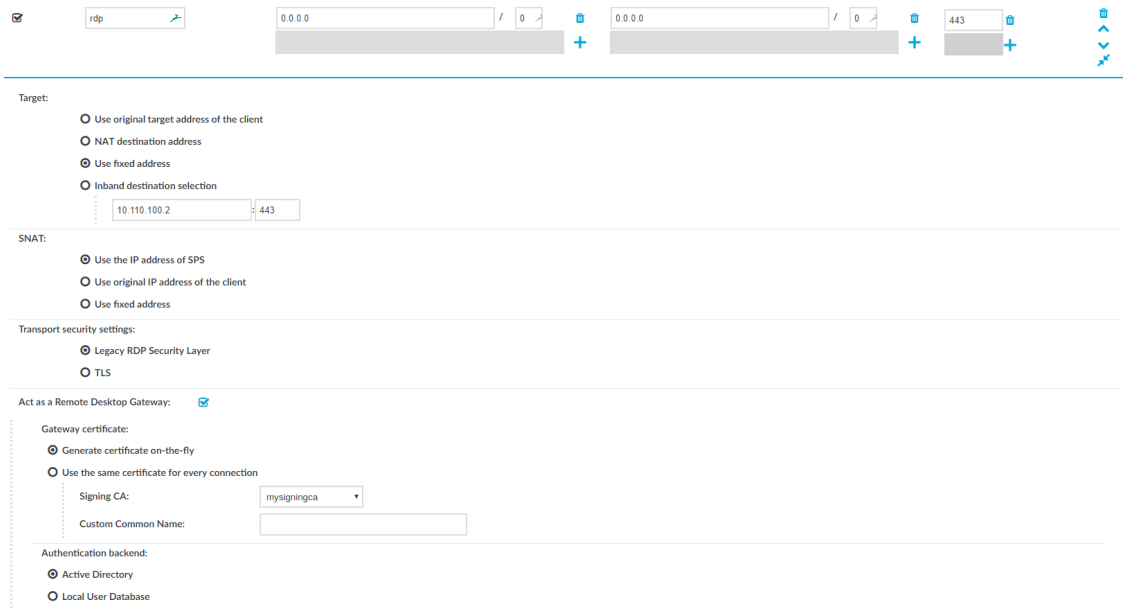
- To access remote servers using a Remote Desktop Gateway, the clients must use version 6.1 or newer of the Remote Desktop application. Note that officially only version 6.0 is available for the Windows 2003 Server operating system, though it is possible to install a newer version. However, this is a problem only when initiating RDP connections from the Windows 2003 Server host, not when the Windows 2003 Server is the target of the connection.

- SPS must be a member of a Windows Domain (for details on joining a domain, see Network Level Authentication (NLA) with domain membership on page 524), or you must use a **Local User Database** (for details, see Creating a Local User Database on page 486).
- Ensure that the system times of the Domain Controller, the target servers, the clients, and SPS are synchronized.
- Gateway authentication on the SPS web interface cannot be used for connection policies that use SPS as a Remote Desktop Gateway. However, the Remote Desktop applications of the clients can be configured to perform two separate authentications, one on the Remote Desktop Gateway (that is, on SPS), and one on the target server. For details on configuring the Remote Desktop applications of the clients to perform gateway authentications, see Configuring Remote Desktop clients for gateway authentication on page 534.
- The Remote Desktop Gateway Server Protocol supports various authentication methods. SPS acting as a Remote Desktop Gateway supports only NTLM authentication.
- SPS can be used as a Remote Desktop Gateway. The terminal service clients must be configured to use SPS as the Remote Desktop Gateway. SPS will connect the server (selected inband) after authentication.
- Remote Desktop Gateway will require a certificate. Decide whether you want to use a fix certificate, or an on-the-fly generated certificate before performing the steps below and prepare the certificate.
- You may also need to adjust the port settings of the connections. The default port for RDP connections is 3389, but the Remote Desktop Gateway Server Protocol uses port 443. However, the SPS web interface uses port 443 as well, and other connection policies might already use port 443. Therefore, if administrator or user login is enabled on the interface that receives the Remote Desktop Services connections, add a new alias IP address to the interface of SPS and use this alias in your connection policy and the client configurations. For details on creating IP aliases on SPS, see Managing logical interfaces on page 120.

To use SPS as a Remote Desktop Gateway

1. Navigate to **RDP Control > Connections** and create a new connection policy that will handle the incoming client connections that use the Remote Desktop Gateway Server Protocol.
2. Enable the **Act as a Remote Desktop Gateway** option.

Figure 182: RDP Control > Connections — Configuring SPS as a Remote Desktop Gateway (or RD Gateway)



3. Set the target of the connections.
   - To direct every incoming connection to a single target server, select **Use fixed address** and specify the address of the target server.
   - To extract the destination address from the Remote Desktop Gateway Server Protocol, select **Inband destination selection** and set the address of the servers the clients are allowed to access in the **Target > Domain** fields. For details on using inband destination selection, see Modifying the destination address on page 441.

   NOTE:

   In non-transparent mode, enter the IP address generated for the Remote Desktop Gateway service into the To field. Do not enter the IP address configured for administrator or user login.

4. To act as a Remote Desktop Gateway, SPS needs to display a certificate to the clients.
   - To display always the same certificate, select **Use the same certificate for every connection** and upload the X.509 certificate and the matching private key.

     One Identity recommends using 2048-bit RSA keys (or stronger).

CAUTION:

The Common Name (CN) of the certificate must be the FQDN of SPS, which is the address of the Remote Desktop Gateway specified in the client applications. Otherwise the clients will reject the connections.

- To automatically create new certificates on SPS for every client, select **Generate certificate on-the-fly**, then select the Certificate Authority (CA) to sign the generated certificates with from the **Signing CA** field. For details on creating a signing CA, see Signing certificates on-the-fly on page 478.

  By default, the Common Name (CN) of the generated certificate is `<SPS-hostname.domainname>`. You can set a custom Common Name in the **Custom Common Name** field.

  NOTE:

  Save the CA certificate used to sign the certificate that SPS shows into DER format and import it to the clients into the **Local Computer > Trusted Root Certificate** store of the clients so that the clients can verify the identity of SPS.

5. Under **Authentication backend**:
   - To use Active Directory for authentication, select **Active Directory**.
   - To use a Local User Database for authentication, select **Local User Database**, enter the **Domain**, and select the **Local User Database** from the list.
6. Configure other parameters of the connection policy as needed for your environment.
7. Click Commit.

Configuring Remote Desktop clients for gateway authentication

To configure the Remote Desktop applications of the clients to perform two separate authentications. One of these authentications is on the Remote Desktop Gateway, that is, on One Identity Safeguard for Privileged Sessions (SPS). The other authentication is on the target server. For details on configuring SPS to act as a Remote Desktop Gateway (or RD Gateway), see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on page 531.

Prerequisites

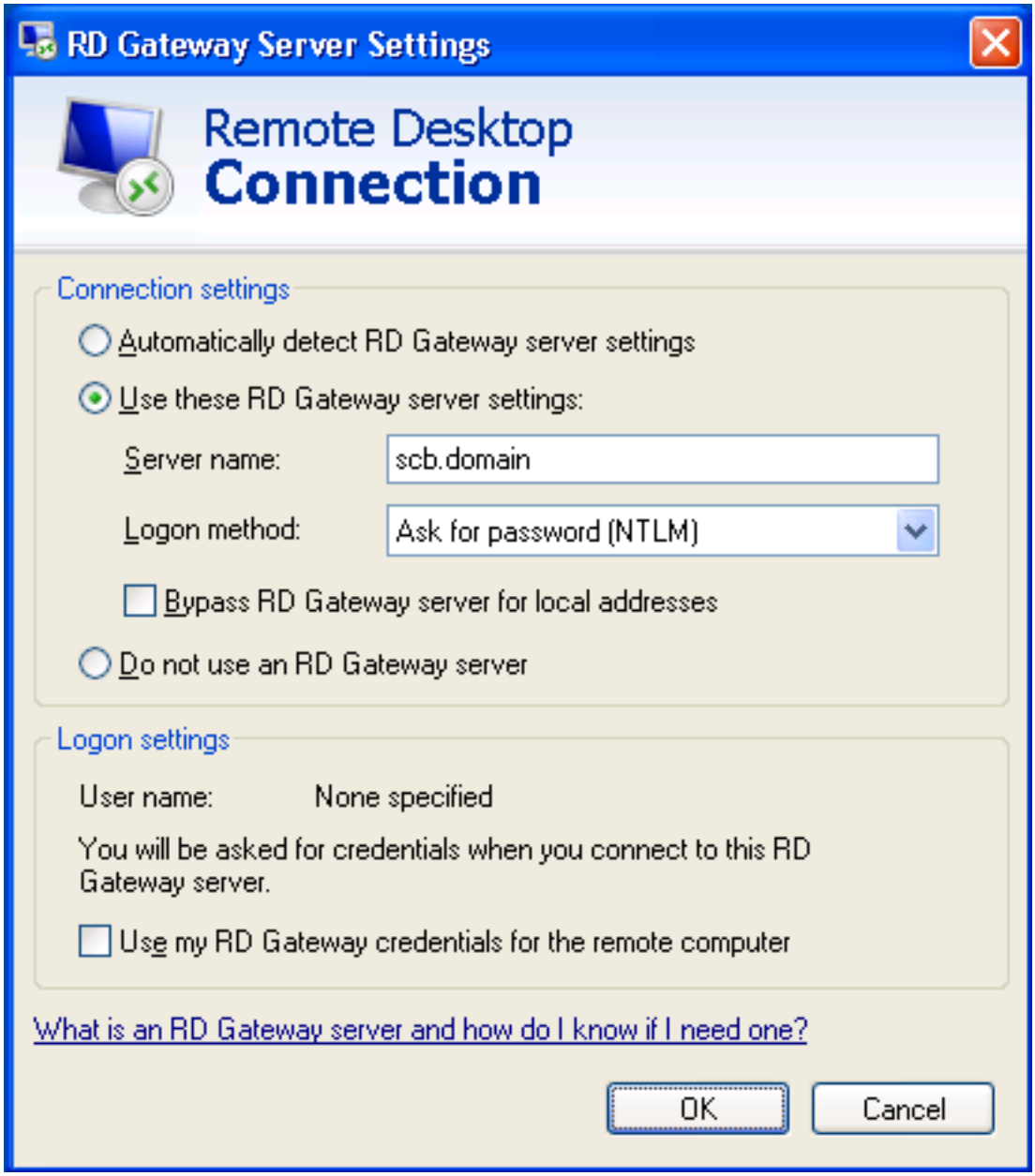
- SPS must be configured to act as a Remote Desktop Gateway. For details, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on page 531.

- The client must use version 6.1 or newer of the Remote Desktop application.
- The target server must be member of a domain.
- The logical interface of SPS must be accessible from the client. You might have to add the address of the logical interface to the `Windows/System32/Drivers/etc/hosts` file to accomplish this.

To configure Remote Desktop clients for gateway authentication

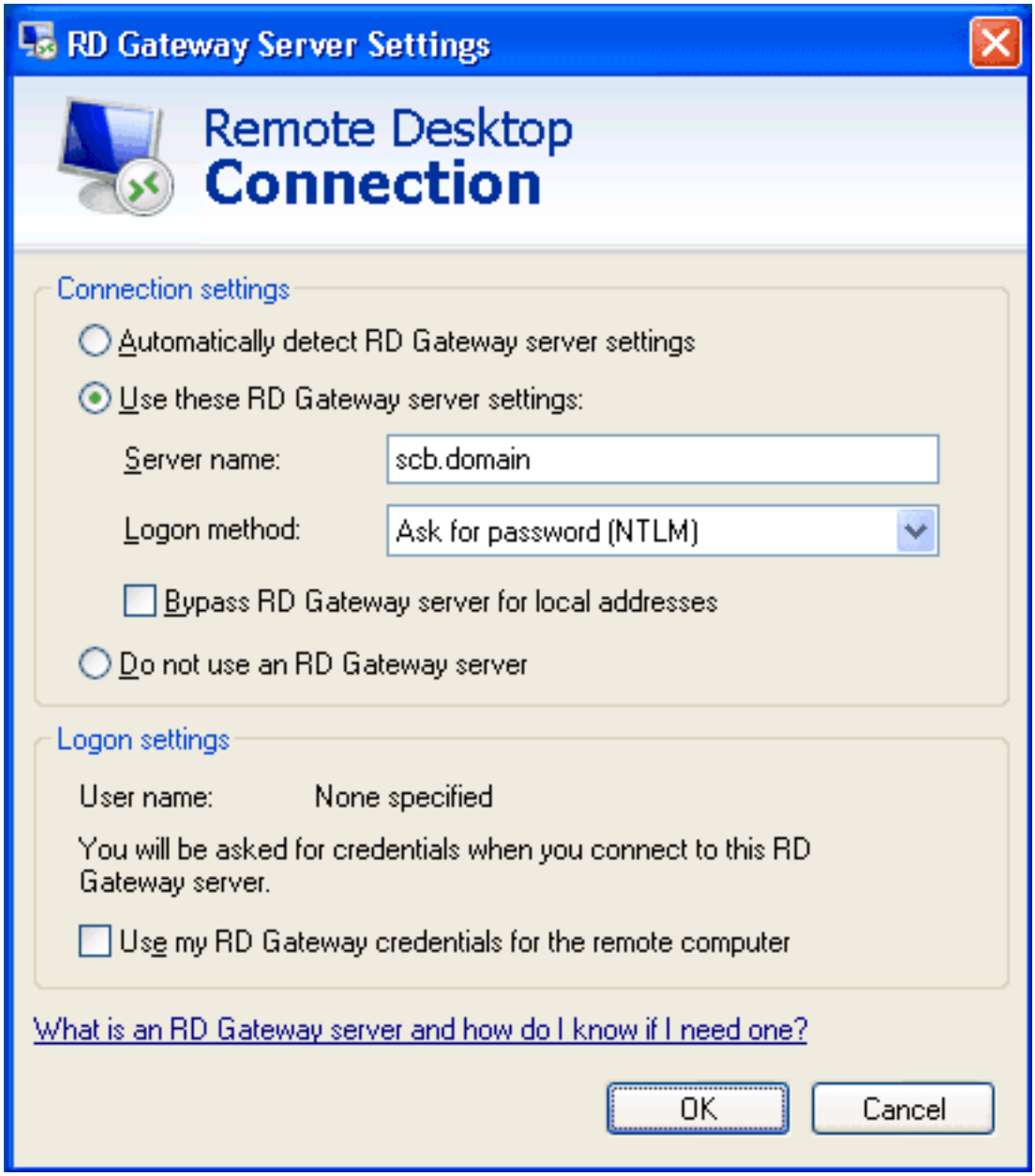
1. On your Windows client, start the **Remote Desktop Connection** application and select **Advanced > Settings**.

Figure 183: Configuring Remote Desktop clients to use SPS as a Remote Desktop Gateway (or RD Gateway)



2. Configure the client to use SPS as its Remote Desktop Gateway. Select **Connection settings > Use these RD Gateway settings**.

Figure 184: Configuring Remote Desktop clients to use SPS as a Remote Desktop Gateway (or RD Gateway)



3. Enter the address of SPS into the **Server name** field. Use the address of the SPS's logical interface that you have configured to accept RDP connections.
4. Select **Logon method > Ask for password (NTLM)**.
5. Uncheck the **Bypass RD Gateway server for local addresses** and **Use my RD Gateway credentials for the remote computer** options.

NOTE:

Technically, gateway authentication is performed even if the **Use my RD Gateway credentials for the remote computer** option is selected, but the same credentials are used on the gateway and on the remote server.

6. Click **OK**.
7. Into the **Username** enter the domain username (for example, `exampledomain\exampleusername`).
8. Click **Connect**.

NOTE:

Depending on your network environment, it might take up to a minute until the connection is established.

Inband destination selection in RDP connections

To use inband destination selection with RDP connections, it is recommended to use One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway (or RD Gateway). For details, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on page 531.

To use inband destination selection with RDP connections without using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway (or RD Gateway), you must use SSL-encrypted RDP connections (see Enabling TLS-encryption for RDP connections on page 529).

Configure your RDP clients so SPS can record the username of client uses in the connection. If you do not configure these settings on the clients, SPS will automatically display a login screen for the users to enter their usernames and passwords. Note that although SPS automatically displays a login screen if it cannot determine the username used in the connection, currently you cannot specify the destination address in this login screen, only in your RDP client application.

- *On Windows Vista SP1 and newer platforms (Remote Desktop Protocol 6.1 or newer)*:

  Navigate to **Local Group Policy Editor > Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Connection Client** and enable the **Prompt for credentials on the client computer** option in the clients. For details, see the Microsoft Documentation.

- *On Windows Vista and older platforms (Remote Desktop Protocol 6.0 or older)*:

  Configure your RDP clients to save the credentials, or make sure that the **Allow me to save credentials** option is selected in the RDP client.

Also, your users have the option to encode the address of the destination server in their username, in the username field of their client application. Note that SPS automatically displays a login screen if it cannot determine the username used in the connection, or you have not encoded a destination server in the username field. You can specify the destination address in the login screen when prompted.

When encoding the address of the destination server in the username, there are a few points to keep in mind. Since most RDP client applications limit which special characters can be used in usernames, this is not always intuitive. For the Microsoft Remote Desktop application (`mstsc`) and the login screen that SPS displays, note the following points:

- Use `%` character to separate the fields, for example: `username%my-targetserver`
- To specify the port number of the server (if it does not use the default port), use the caret `^` character, for example: `username%my-targetserver^6464`
- To specify an IPv6 address, replace the colons with carets, and enclose the address in parentheses. For example, to target the `::1` IP address, use `username%(^^1)`. To target port 6464 of the same server, use `username%(^^1)^6464`.

Usernames in RDP connections

When processing RDP connections, One Identity Safeguard for Privileged Sessions (SPS) attempts to extract the username from the connection. For example, you need the username to:

- Use gateway authentication for the connection. For details on gateway authentication, see "Configuring gateway authentication" in the Administration Guide.
- Use usermapping policies. In this case, SPS compares the username on the server with the username on the gateway. For details on usermapping policies and gateway authentication, see "Configuring usermapping policies" in the Administration Guide and "Configuring gateway authentication" in the Administration Guide, respectively.

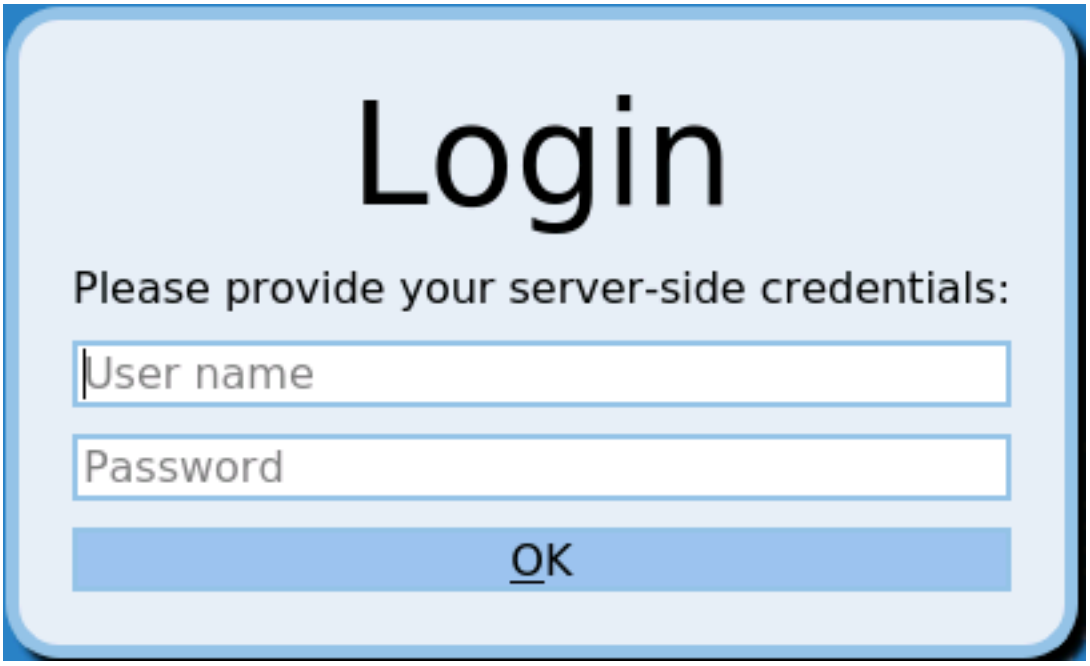
  NOTE:

  In certain cases, SPS receives an empty username from the server, and the connection will be denied by the usermapping policy unless a policy is set for the connection that allows every user for the given group. To add such a policy, specify * in the **Username on the server** field of the usermapping policy. For a list of cases when SPS receives empty username, see Windows settings that interfere with username extraction on page 540.
- Search or filter connections by the username on the SPS search interface, or create automatic statistics based on the username.
- Find the connection of the user on the **Four Eyes** and **Active Connections** pages.
- Usernames are also essential if you want to use One Identity Safeguard for Privileged Analytics. If you are interested in One Identity Safeguard for Privileged Analytics, contact our Sales Team, or your One Identity representative.

SPS can record the username automatically if the RDP connection is using Network Level Authentication (CredSSP), and usually in other scenarios as well. If SPS cannot automatically extract the username, it displays the following login screen to the user (note that SPS can display this login screen only in TLS-encrypted connections).

The known scenarios that interfere with RDP usernames are listed in Windows settings that interfere with username extraction.

Figure 185: Server-side login in RDP



To ensure that your users can access the target servers only when their username is recorded, you can configure SPS to terminate RDP connections if it cannot reliably extract the username. To terminate such connections, clear the **RDP Control > Settings > Permit unreliable usernames** option.

Windows settings that interfere with username extraction

The following settings on the Windows client or server can prevent SPS from correctly extracting the username from the RDP connection. As a result, the username is not visible on the **Search**, **Four Eyes** and **Active Connections** pages.

- The **DontDisplayLastUserName** option is enabled on the server. The **DontDisplayLastUserName** security setting of Windows servers specifies whether the username from the last successful login is displayed on the login screen as a default for the next login. To disable the **DontDisplayLastUserName** security setting, do one of the following.
  - Disable the `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\DontDisplayLastUsername` registry setting. For more details, see the

DontDisplayLastUserName TechNet article.

NOTE:

Registry settings can be overridden by Group Policy settings.

- Disable this option in the `Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options` policy. For details, see Do not display last user name in logon screen TechNet article.

- There is no server-side authentication. To avoid this problem, ensure that your server requires authentication from the users.
- If the server is Windows 2003 Server or Windows XP and the **Allow to save credentials** or **Remember my credentials** options are enabled in the Remote Desktop client application. In this case, disable these options on the client, and delete any credentials that have already been saved on the client.

Saving login credentials for RDP on Windows

You can use automatic RDP login on Windows, but the stored credentials are not trusted by default, and you have to enter the password for each connection. Create the following local policies on the client to allow delegating saved credentials:

1. Start the Group Policy Editor: run **gpedit.msc**.
2. Navigate to **Local Computer Policy > Computer Configuration > Administrative Templates > System > Credentials Delegation**.
3. Open the **Allow Delegating Saved Credentials with NTLM-only Server Authentication** policy.
4. Click **Show** and enter `TERMSRV/*`.
5. Click **Apply**.
6. Open the **Allow Delegating Saved Credentials** policy.
7. Click **Show** and enter `TERMSRV/*`.
8. Click **Apply**.
9. Open the **Allow Delegating Default Credentials with NTLM-only Server Authentication** policy.
10. Click **Show** and enter `TERMSRV/*`.
11. Click **Apply**.
12. Open the **Allow Delegating Default Credentials** policy.
13. Click **Show** and enter `TERMSRV/*`.
14. Click **Apply**.

15. Verify that the **Deny Delegating Saved Credentials** policy does not contain `TERMSRV/*` in the list.
16. Close the Group Policy Editor.
17. From the command line, issue the **gpupdate /force** command.

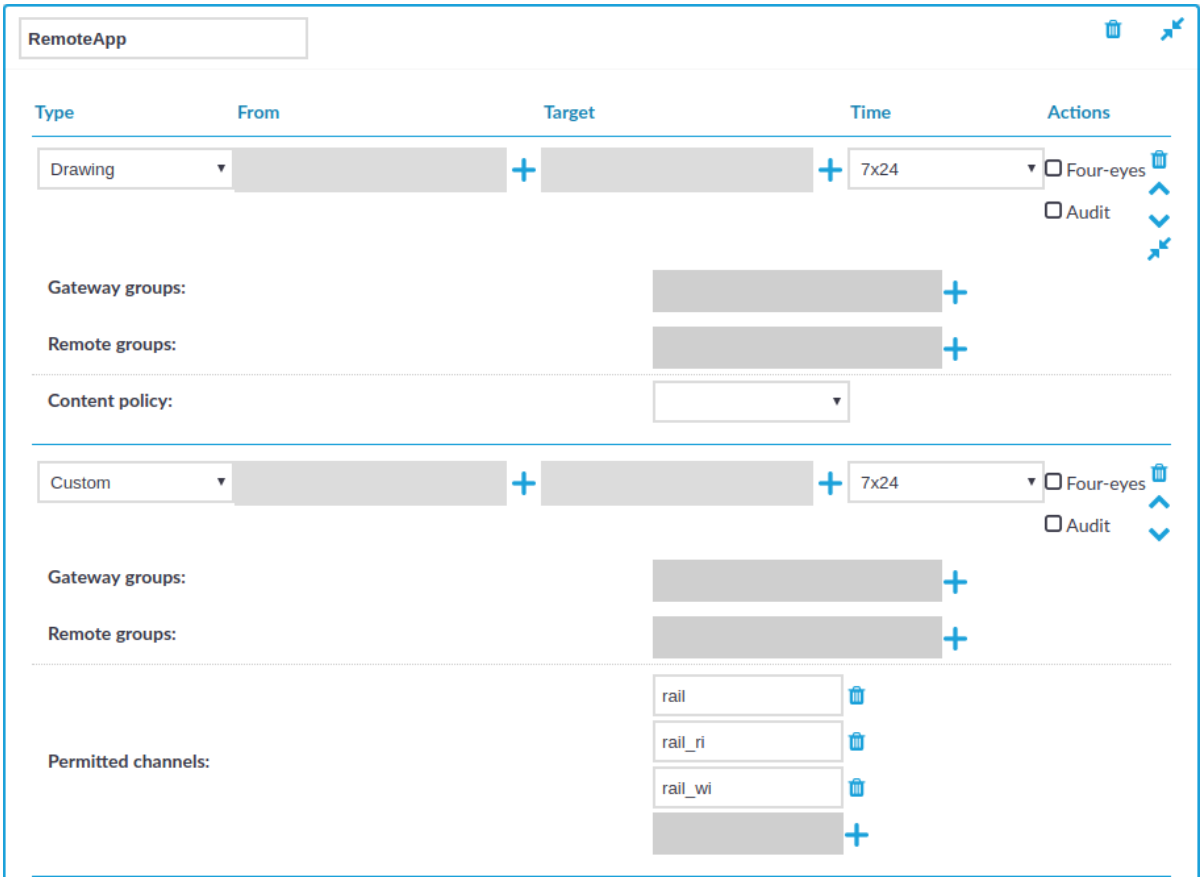
Configuring RemoteApps

Overview

RemoteApps use RDP channels that are denied by default. When configuring RDP connections for RemoteApps on One Identity Safeguard for Privileged Sessions (SPS), create a custom channel policy which enables the following channels:

- Drawing
- rail
- rail_ri
- rail_wi

Figure 186: RDP Control > Channel Policies — Configuring the required channels for RemoteApps



Prerequisites

- You must disable the **Use advanced RemoteFX graphics for RemoteApp** group policy on the RDP server.

  The policy is available at **Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment > Use advanced RemoteFX graphics for RemoteApp**.
- You must have the Remote Desktop (RD) Licensing role installed.

To configure RemoteApps

1. Navigate to **RDP Control > Channel Policies**.
2. Click + to create a new channel policy.
3. Enter the name for the channel policy.
4. Choose **Drawing** as the channel type.
5. Click + to add an additional channel type.

6. Choose **Custom** as the second channel type.
7. In **Permitted channels**, click + to add the following channels:
   - rail
   - rail_ri
   - rail_wi

   (You have to click + for each channel.)
8. Click 

to save the channel policy.
9. You have created a channel policy for RemoteApps.

   When you configure a connection that uses RemoteApps in **RDP Control > Connections**, select this channel policy as the **Channel policy** of the connection.

Configuring SPS to enable exporting files from audit trails after RDP file transfer via clipboard

In SPS versions 6.2 and later, you can export files from audit trails after RDP file transfer via clipboard. For more information on the process in the Safeguard Desktop Player application, see Exporting files from an audit trail.

To configure SPS to enable extracting files from audit trails after RDP file transfer via clipboard

1. Navigate to **RDP Control > Connections** and open an existing connection (or create and configure a new connection).
2. Expand the connection tab, scroll down to the **Channel policy** drop-down list, and select a channel policy of your choice from the drop-down list options.
3. Navigate to **RDP Control > Channel Policies** and open the channel policy that you selected from the **Channel policy** drop-down list under **RDP Control > Connections**.
4. Ensure that the **Clipboard** drop-down list option under **Type** and the **Record audit trail** checkbox are both selected.
5. (Optional) Click 

to save your configuration.

SSH-specific settings

The following sections describe configuration settings available only for the SSH protocol. Use the following policies to control who, when, and how can access the SSH connection.

- *Hostkeys and host certificates*: One Identity Safeguard for Privileged Sessions (SPS) allows you to set how the identity of the client hosts and servers is verified. For details, see Setting the SSH host keys of the connection on page 546.
- *Authentication Policy*: Authentication policies describe the authentication methods allowed in a connection. Different methods can be used for the client and server-side connections. For details, see Authentication Policies on page 552.
- *User List*: A user list is a list of usernames permitted to use — or forbidden from using — the connection. Essentially it is a blacklist or a whitelist. All users matching the other requirements of the connection are accepted by default. For details, see Creating and editing user lists on page 457.
- *Channel Policy*: The channel policy determines which SSH channels (for example terminal session, SCP, and so on) can be used in the connection, and whether they are audited or not. The different channels may be available only under certain restrictions, as set in the channel policy. For details, see Creating and editing channel policies on page 448.
- *SSH settings*: SSH settings determine the parameters of the connection on the protocol level, including timeout value and greeting message of the connection. The following parameters determine which algorithms are used in the connections, and can be set independently for the client and the server side: key exchange, host key, cipher, MAC, and compression algorithms. The default values include all possible algorithms. For details, see Creating and editing protocol-level SSH settings on page 564.
- *Content Policy*: Content policies allow you to inspect the content of the connections for various text patterns, and perform an action if the pattern is found. For example, SPS can send an e-mail alert if a specific command is used in an SSH terminal session. For details, see Creating a new content policy on page 452.
- *Authentication and Authorization plugin*:

  One Identity Safeguard for Privileged Sessions (SPS) provides a plugin framework to integrate SPS to external systems to authenticate or authorize the user before authenticating on the target server. Such plugins can also be used to request additional information from the users, for example, to perform multi-factor authentication.

For details, see Integrating external authentication and authorization systems on page 768.

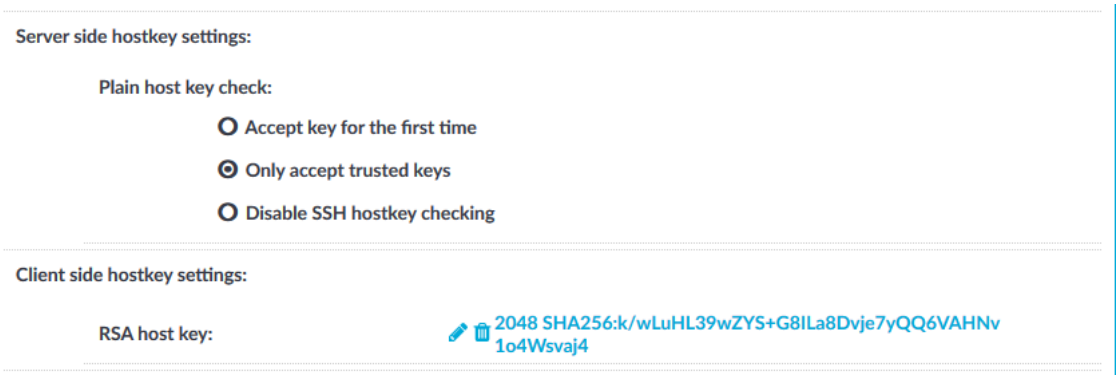
Setting the SSH host keys of the connection

By default, One Identity Safeguard for Privileged Sessions (SPS) accepts and stores the host key of the server when the connection is first established.

To manually set the SSH keys used and accepted in the connection

1. Navigate to **SSH Control > Connections** and click ⤢ to display the details of the connection.

   Figure 187: SSH Control > Connections — Configuring SSH host keys of the connection

2. Verify the identity of the servers based on their hostkeys.
   - Select **Accept key for the first time** to automatically record the key shown by the server on the first connection. SPS will accept only this key from the server in later connections. This is the default behavior of SPS.

     NOTE:

     When your deployment consists of two or more instances of SPS organized into a cluster, the SSH keys recorded on the Managed Host nodes before they were joined to the cluster are overwritten by the keys on the Central Management node. For details, see Configuration synchronization and SSH keys on page 356.
   - Select **Only accept trusted keys** if the key of the server is already available on SPS. SPS will accept only the stored key from the server. For further information on setting the host keys of the server, see Server host

keys on page 560.

> NOTE:
>
> When your deployment consists of two or more instances of SPS organized into a cluster, the SSH keys recorded on the Managed Host nodes before they were joined to the cluster are overwritten by the keys on the Central Management node. For details, see Configuration synchronization and SSH keys on page 356.

- Select **Disable SSH hostkey checking** to disable SSH host key verification.

  > **CAUTION:**
  >
  > **Disabling SSH host key verification makes it impossible for SPS to verify the identity of the server and prevent man-in-the-middle (MITM) attacks.**

3. You can choose to upload or paste an RSA host key, or generate a new one.

   > NOTE:
   >
   > One Identity recommends using 2048-bit RSA keys (or stronger).

   Click on the fingerprint to display the public part of the key.

4. Click Commit.

Supported SSH channel types

The available SSH channel types and their functionalities are described below. For details on configuring Channel Policies, see Creating and editing channel policies on page 448. For a list of supported client applications, see Supported protocols and client applications on page 38.

- **Agent**: Forwards the SSH authentication agent from the client to the server.

  > NOTE:
  >
  > To perform agent-based authentication on the target server, it is not required to enable the `Agent-forwarding` channel in the Channel Policy used by the connection. The `Agent-forwarding` channel is needed only to establish connections from the target server to other devices and authenticate using the agent running on the client.

- **X11 Forward**: Forwards the graphical X-server session from the server to the client. Enter the address of the client into the **Details > Target address** field to permit X11-forwarding only to the specified clients. Specify IP addresses or networks (in IP address/Prefix format).

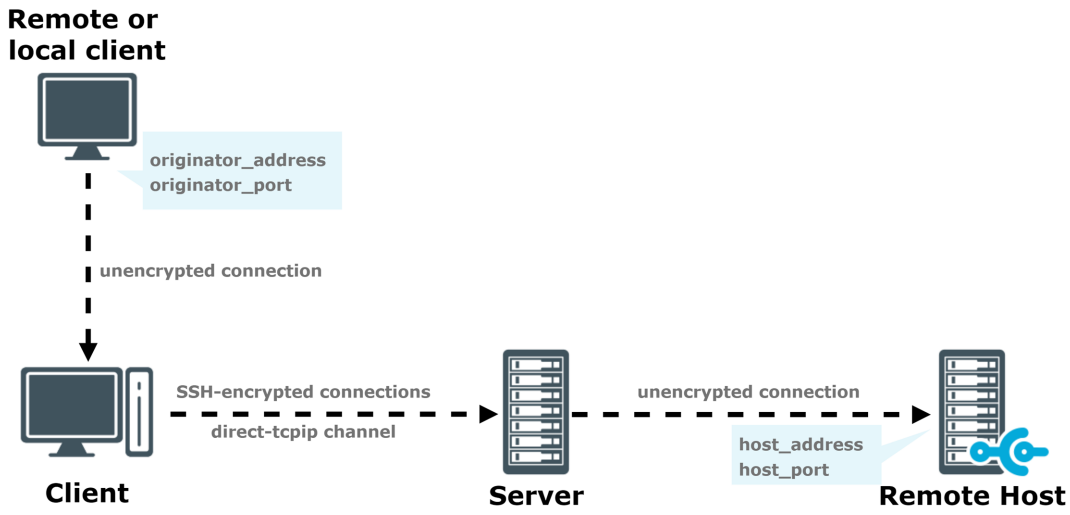
NOTE:

Certain client applications send the Target address as a hostname, while others as an IP address. If you are using a mix of different client applications, you might have to duplicate the channel rules and create IP-address and hostname versions of the same rule.

- **Local Forward**: Forwards traffic arriving to a local port of the client to a remote host. To enable forwarding only between selected hosts, enter their IP addresses into the **Details** field. If the **Details** field is empty, local forwarding is enabled without restriction, the client may forward any traffic to the remote host. Enter the source of the forwarded traffic into the **Originator**, the target of the traffic into the **Target** field. Specify IP addresses or networks (in IP address/Prefix format). These parameters are the end-points of the forwarded traffic (that is, the local host that sends data to the remote host), and not the SSH server or the client.

  For example, to enable forwarding from the `192.168.20.20` host to the remote host `192.168.50.50`, enter `192.168.20.20` into the **Originator**, and `192.168.50.50` into the **Target** field.

Figure 188: Local TCP forwarding



NOTE:

Certain client applications send the Originator and Target addresses as hostnames, while others as IP addresses. If you are using a mix of different client applications, you might have to duplicate the channel rules and create IP-address and hostname versions of the same rule.

CAUTION:

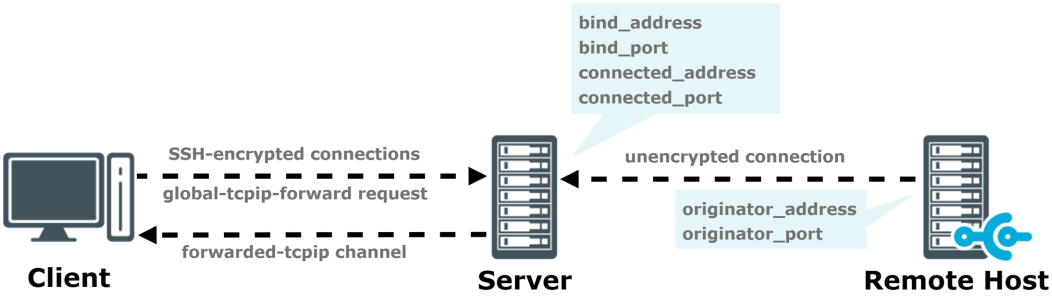
Port forwarding across One Identity Safeguard for Privileged Sessions (SPS) may fail for certain SSH client-server combinations. This happens if within the protocol, the address of the remote host is specified as a hostname during the port-forwarding request (`SSH_MSG_GLOBAL_REQUEST`), but the hostname is resolved to IP address in the channel opening request (`SSH_MSG_CHANNEL_OPEN`. By default, SPS rejects such connections.

To enable these connections, navigate to SSH Control > Settings, and disable the Strict mode option.

- **Remote Forward**: Forwards traffic arriving a remote port of the server to the client. To enable forwarding only between selected hosts, enter their IP addresses into the **Details** field. If the **Details** field is empty, remote forwarding is enabled without restriction, the SSH server may forward any traffic to the client. Enter the source of the forwarded traffic into the **Originator**, the target of the traffic into the **Target** field. Specify IP addresses or networks (in IP address/Prefix format). These parameters are the end-points of the forwarded traffic (that is, the remote host that sends data to the client), and not the SSH server.

  For example, to enable forwarding from the `192.168.20.20` remote host to the client `192.168.50.50`, enter `192.168.20.20` into the **Originator**, and `192.168.50.50` into the **Target** field.

Figure 189: Remote TCP forwarding



NOTE:

Certain client applications send the Originator and Target addresses as hostnames, while others as IP addresses. If you are using a mix of different client applications, you might have to duplicate the channel rules and create IP-address and hostname versions of the same rule.

> **CAUTION:**
>
> **Port forwarding across SPS may fail for certain SSH client-server combinations. This happens if within the protocol, the address of the remote host is specified as a hostname during the port-forwarding request (`SSH_MSG_GLOBAL_REQUEST`), but the hostname is resolved to IP address in the channel opening request (`SSH_MSG_CHANNEL_OPEN`. By default, SPS rejects such connections.**
>
> **To enable these connections, navigate to SSH Control > Settings, and disable the Strict mode option.**

- **Session Exec**: Execute a remote command (for example `rsync`) without opening a session shell. Enter the permitted command into the **Permitted commands** field. You can use regular expressions to specify the commands. This field can contain only letters (`a-z, A-Z`), numbers (`0-9`), and the following special characters (`{}()*?\\|[]`).

  > **CAUTION:**
  >
  > **Restricting the commands available in Session Exec channels does not guarantee that no other commands can be executed. Commands can be renamed, or executed from shell scripts to circumvent such restrictions.**

- **Session Exec SCP**: Transfers files using the Secure Copy (SCP) protocol.
  - To make the list of file operations available in the **File operations** column of the **Search** page, navigate to the Channel Policies page of the protocol, and enable the **Log file transfers to database** option. This option is disabled by default.
  - To send the file operations into the system log, enable the **Log file transfers to syslog** option. This option is disabled by default.

    > NOTE:
    >
    > Turning logging on might result in a slight performance penalty. If traffic load slows processes down, disable the option.

CAUTION:

The WinSCP application does not follow the RFC of the SCP protocol properly, but transfers files in a Session Shell channel instead of a Session Exec SCP channel. This has the following results:

- **If the Session Shell channel is enabled in a Channel Policy (this is needed for SSH terminal sessions as well), and your users use WinSCP using the File protocol > SCP option, they will be able to transfer files to and from the server. Also, these files will not be listed in the File operations field of the Search page.**
- **To avoid these problems, you have to enforce that your clients use WinSCP with the File protocol > SFTP option. WinSCP uses SFTP by default, but can be changed manually to use SCP, and also falls back to using SCP if a server rejects SFTP.**
- **To terminate the connection when a user transfers a file with WinSCP using the Session Shell channel, create a Content Policy that matches the `WinSCP: this is end-of-file` string in screen content, and use this policy in your Connection Policies. For details on Content Policies, see Real-time content monitoring with Content Policies on page 451. This solution has been tested with WinSCP version 5.1.5: if it does not work for your version, contact our Support Team.**

- **Session Subsystem**: Use a subsystem. Enter the name of the permitted subsystem into the **Details** field.
- **Session SFTP**: Transfers files using the Secure File Transfer Protocol (SFTP).
  - To make the list of file operations available in the **File operations** column of the **Search** page, navigate to the Channel Policies page of the protocol, and enable the **Log file transfers to database** option. This option is disabled by default.
  - To send the file operations into the system log, enable the **Log file transfers to syslog** option. This option is disabled by default.

    NOTE:

    Turning logging on might result in a slight performance penalty. If traffic load slows processes down, disable the option.
- **Session Shell**: The traditional remote terminal session.

CAUTION:

The WinSCP application does not follow the RFC of the SCP protocol properly, but transfers files in a Session Shell channel instead of a Session Exec SCP channel. This has the following results:

- **If the Session Shell channel is enabled in a Channel Policy (this is needed for SSH terminal sessions as well), and your users use WinSCP using the File protocol > SCP option, they will be able to transfer files to and from the server. Also, these files will not be listed in the File operations field of the Search page.**
- **To avoid these problems, you have to enforce that your clients use WinSCP with the File protocol > SFTP option. WinSCP uses SFTP by default, but can be changed manually to use SCP, and also falls back to using SCP if a server rejects SFTP.**
- **To terminate the connection when a user transfers a file with WinSCP using the Session Shell channel, create a Content Policy that matches the `WinSCP: this is end-of-file` string in screen content, and use this policy in your Connection Policies. For details on Content Policies, see Real-time content monitoring with Content Policies on page 451. This solution has been tested with WinSCP version 5.1.5: if it does not work for your version, contact our Support Team.**

Authentication Policies

An authentication policy is a list of authentication methods that can be used in a connection. Connection definitions refer to an authentication policy to determine how the client can authenticate to the target server. Separate authentication methods can be used on the client and the server-side of the connection.

Figure 190: Authentication policies



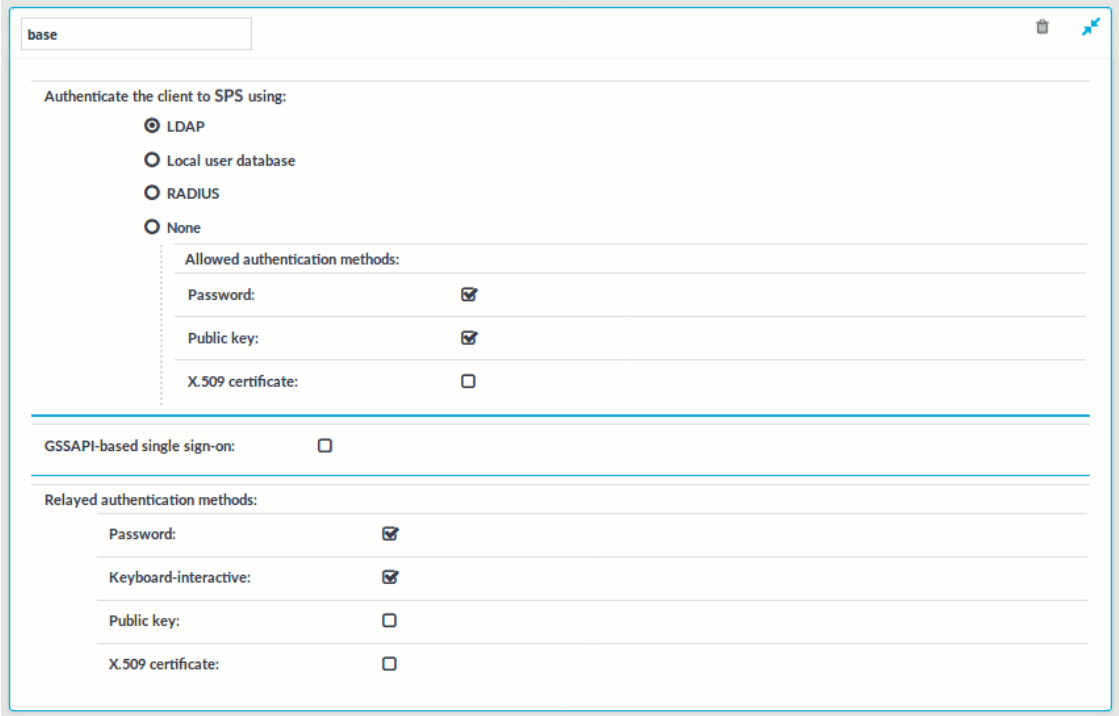
Creating a new authentication policy

The following describes how to create a new authentication policy.

To create a new authentication policy

1. Navigate to **SSH Control > Authentication Policies**, and click +.

   Figure 191: SSH Control > Authentication Policies — Configuring authentication policies

2. Enter a name for the policy into the **Name** field.
3. Select the authentication method used on the client-side in the **Authenticate the client to SPS using** field. For details on the client-side authentication settings, see Client-side authentication settings on page 554.
4. Select the authentication method used on the server-side in the **Relayed authentication methods** field. For details on the relayed authentication settings, see Relayed authentication methods on page 557.

   If you selected **Public key > Agent** as the relayed authentication method:

   If this option is used, SPS requests the client to use its SSH agent to authenticate on the target server. Therefore, you must configure your clients to enable agent forwarding, otherwise authentication will fail. For details on enabling agent forwarding in your SSH application, see the documentation of the application.

5. Click Commit.

NOTE:

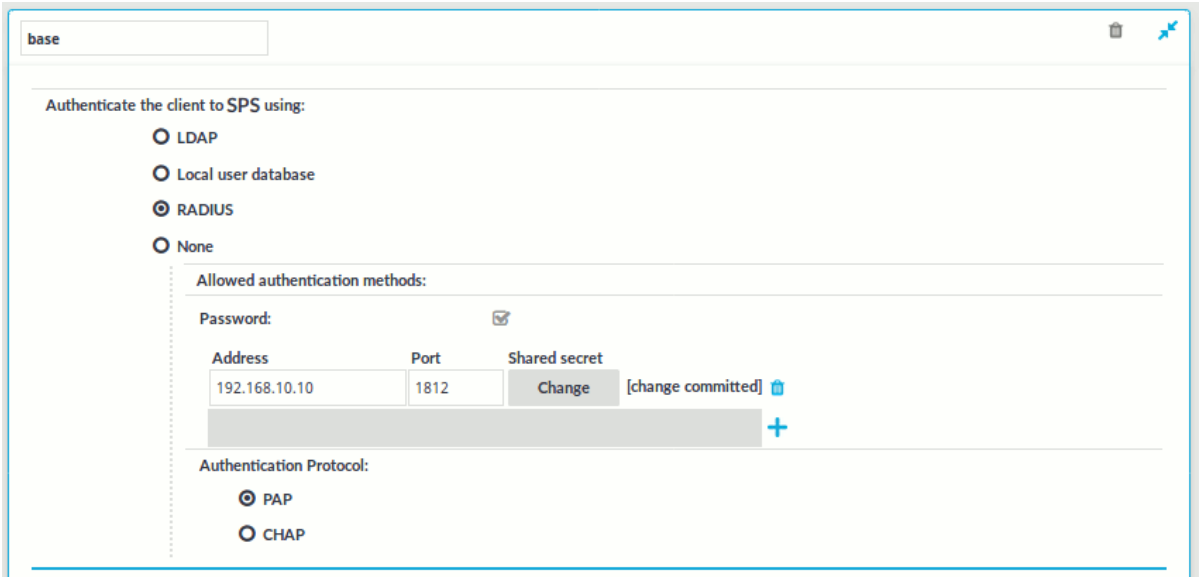
- The client-side authentication settings apply for authenticating the user inband (that is, within the SSH protocol) to the One Identity Safeguard for Privileged Sessions (SPS) gateway, and is independent from the gateway authentication performed on the SPS web interface. The web-based gateway authentication is an out-of-band gateway authentication method that can be required by the connection policy. For details on out-of-band gateway authentication, see Configuring out-of-band gateway authentication on page 744.

  Gateway authentication on the SPS web interface can be used together with authentication policies. In an extreme setting, this would mean that the user has to perform three authentications: a client-side gateway authentication within the SSH protocol to SPS, an out-of-band gateway authentication on the SPS web interface, and a final authentication on the target server.
- The Connection Policy will ignore the settings for server-side authentication (set under **Relayed authentication methods**) if a Credential Store is used in the Connection Policy.

Client-side authentication settings

For the client-side connection, One Identity Safeguard for Privileged Sessions (SPS) can authenticate the client inband (within the SSH protocol) using the following authentication methods:

Figure 192: SSH Control > Authentication Policies — Configuring client-side authentication methods



- **LDAP**: SPS will authenticate the client to the LDAP database set in the **LDAP Server** of the connection policy. To use LDAP authentication on the client side, select **Authenticate the client to SPS using > LDAP**, and select the permitted authentication methods (**Password**, **Public key**). More than one method can be permitted.

  NOTE:

  - SPS will authenticate the client-side connection to the LDAP server configured in the connection policy. This is not necessarily the same as the LDAP server used to authenticate the users accessing the SPS web interface.
  - The public keys of the users stored in the LDAP database must be in OpenSSH format.

- **Local user database**: Authenticate the client locally on the SPS gateway. For details, see Local client-side authentication on page 556.
- **RADIUS**: SPS will authenticate the client to the specified RADIUS server. Select **Authenticate the client to SPS using > RADIUS**, enter the IP address or hostname of the RADIUS server into the **Address** field, the port number of the RADIUS server into the **Port** field, and the shared secret of the RADIUS server into the **Shared secret** field. Only password-authentication is supported (including one-time passwords), challenge-response based authentication is not.

  To use the Password Authentication Protocol, select **PAP**. To use the Challenge-Handshake Authentication Protocol, select **CHAP**.

  Use an IPv4 address.

  To add more RADIUS servers, click + and fill in the respective fields.
- **None**: Do not perform client-side authentication, the client will authenticate only on the target server.

  CAUTION:

  Hazard of security breach. If the None authentication option is selected on the client side and SPS is configured to use public-key or certificate based authentication on the server, the user will not be authenticated at all unless gateway authentication is required for the connection.

To use certificates to authenticate the client, you can use the **LDAP** and the **Local user database** backends.

Figure 193: Client-side inband gateway authentication with different certificates

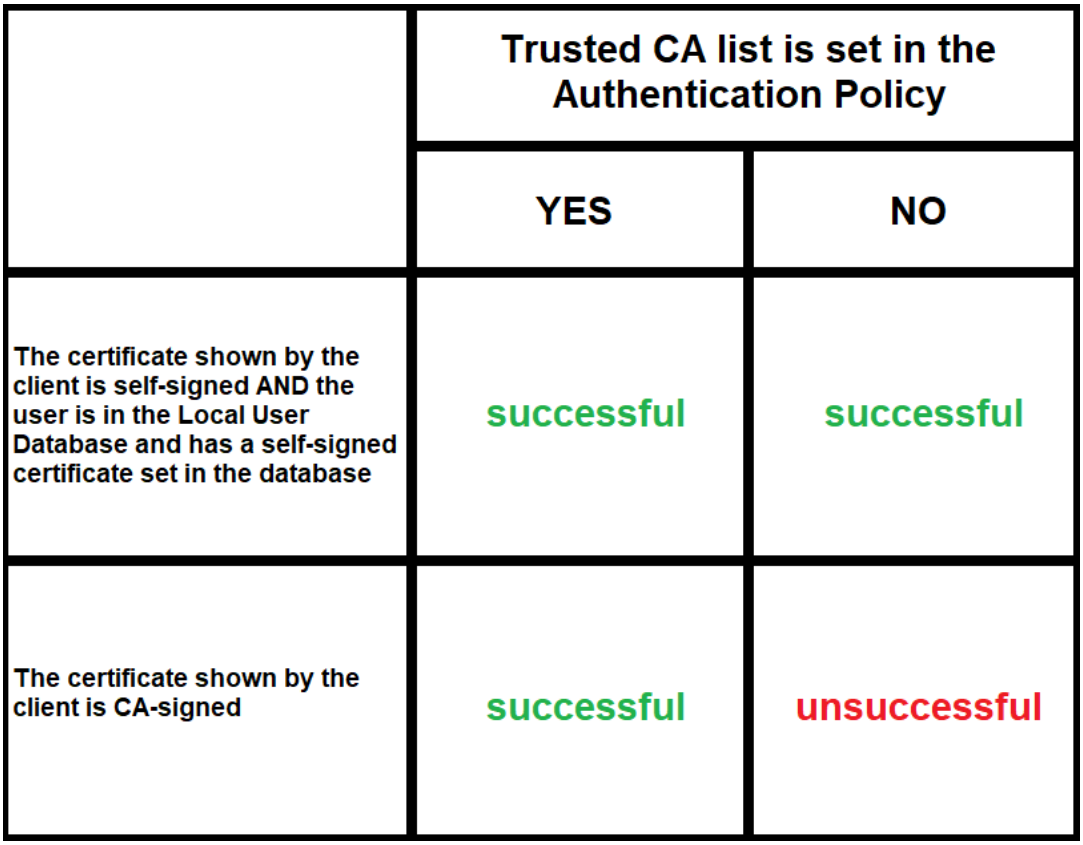


Table 8: Client-side inband gateway authentication with different certificates

	Trusted CA list is set in the Authentication Policy	
	YES	**NO**
The certificate shown by the client is self-signed AND the user is in the Local User Database and has a self-signed certificate set in the database	successful	successful
The certificate shown by the client is CA-signed	successful	unsuccessful

Local client-side authentication

The following describes how to perform authentication locally on One Identity Safeguard for Privileged Sessions (SPS) for client-side connections.

NOTE:

The users can be authenticated to their passwords or public-keys uploaded to SPS.

The accounts created to access the SPS web interface cannot be used to authenticate SSH connections.

Prerequisites

To perform authentication locally on SPS for client-side connections, an existing **Local User Database** is needed. To create a **Local User Database**, complete the following procedure: Creating a Local User Database on page 486.

To perform authentication locally on SPS for client-side connections

1. Navigate to **SSH Control > Authentication Policies**, and select the authentication policy to modify.
2. Select **Authenticate the client to SPS using > Local user database**, and select the permitted authentication methods (**Password**, **Public key**).
3. Select the **Local user database** from the list that defines the users who can access the server.
4. Click Commit.

Relayed authentication methods

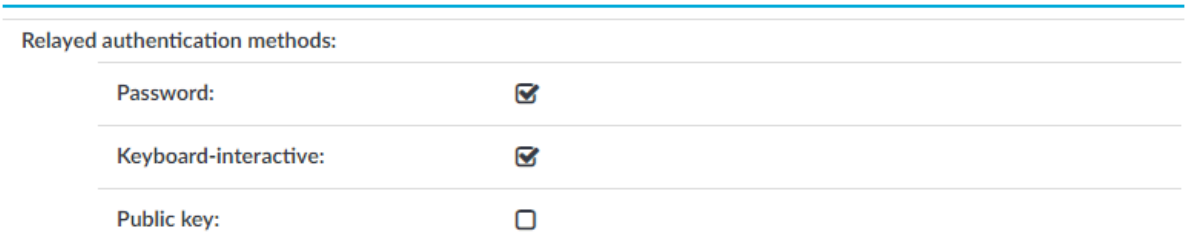
For the server-side connection (between One Identity Safeguard for Privileged Sessions (SPS) and the target server), the following authentication methods are available.

NOTE:

Even though these settings refer to the server-side connection, the client must support the selected authentication method and have it enabled. For example, to use publickey authentication on the server side, the client must support publickey authentication as well as provide a fake publickey, even if a different authentication method is used on the client side.

The Connection Policy will ignore the settings for server-side authentication (set under **Relayed authentication methods**) if a Credential Store is used in the Connection Policy.

Figure 194: SSH Control > Authentication Policies — Configuring relayed authentication methods



- **Password**: Authentication based on username and password. The server will request a password from the user, even if a password-based authentication was

already successful on the client-side.

- **Keyboard-Interactive**: Authentication based on exchanging messages between the user and the server. This method includes authentication schemes like S/Key or TIS authentication. Note that depending on the configuration of the SSH server, password-based authentication can also require using the keyboard-interactive authentication method.
- **Public Key**: Authentication based on public-private encryption keypairs. SPS supports the following public-key authentication scenarios:
  - **Publish to LDAP**: SPS generates a keypair, and uses this keypair in the server-side connection. The public key of this keypair is also uploaded to the LDAP database set in the LDAP Server of the connection policy. That way the server can authenticate the client to the generated public key stored under the user's username in the LDAP database.
  - **Fix**: Uses the specified private key in the server-side connection.
  - **Agent**: Allow the client to use agent-forwarding, and use its own keypair on the server-side.

    If this option is used, SPS requests the client to use its SSH agent to authenticate on the target server. Therefore, you must configure your clients to enable agent forwarding, otherwise authentication will fail. For details on enabling agent forwarding in your SSH application, see the documentation of the application.

  One Identity recommends using 2048-bit RSA keys (or stronger).

Configuring your Kerberos environment

To integrate One Identity Safeguard for Privileged Sessions (SPS) with your Kerberos environment, so that your clients can authenticate on the target servers using Kerberos tickets, you have to configure your environment appropriately.

To configure your Kerberos environment

1. Configure your DNS server.
   a. On your Domain Name Server (DNS), add SRV records that describe which Key Distribution Center (KDC) belongs to the domain. Add both TCP and UDP entries for each domain. For example, if your domain is `example.com` and the hostname of your KDC server is `kdc.example.com`, this entry looks like:

      ```
      _kerberos_tcp_example.com 0 0 88 kdc.example.com
      _kerberos_udp_example.com 0 0 88 kdc.example.com
      ```

   b. If your environment uses multiple realms, repeat the previous step for every realm.

c. Verify that the servers that your clients will connect to via SPS have proper reverse-dns entries. Otherwise, your clients cannot access the target servers if you use the **Inband destination selection** feature of SPS.

2. Create a keytab file for SPS.
   a. On your KDC server, create a principal for the SPS host, using the domain name of your SPS. For example:

      ```
      host/scb.example.com@EXAMPLE.COM
      ```

   b. If your environment uses multiple realms, repeat the previous step on the KDC of every realm.
   c. Export the key of the principal into a keytab file.
   d. If your environment uses multiple realms, merge the keytab files of the different realms into a single file, for example, using the **ktadd** or the **ktutil** utilities.
   e. If your environment uses multiple realms, repeat the previous step on the KDC of every realm.
3. Configure the SSH application of your client hosts to enable Kerberos (GSSAPI) ticket forwarding. (In most applications this is disabled by default.)

Expected result

You have configured your environment to use Kerberos authentication with SPS, and created a keytab file for your SPS host. For details on uploading the keytab file and configuring SPS see Kerberos authentication settings on page 559.

Kerberos authentication settings

The following describes how to perform authentication with Kerberos.

> NOTE:
>
> If Kerberos authentication has been configured for the connection, it is not possible to fall back to other authentication methods.

Prerequisites

Before configuring Kerberos authentication on One Identity Safeguard for Privileged Sessions (SPS), make sure you have configured your Kerberos environment correctly and have retrieved the keytab file. For details, see Configuring your Kerberos environment on page 558.

To perform authentication with Kerberos

1. Navigate to **SSH Control > Authentication Policies**.
2. Create a new Authentication Policy and enable **GSSAPI-based single sign-on**. This will disable all other authentication methods. Click Commit.
3. Navigate to **SSH Control > Global Options > GSSAPI**.
4. **Browse** for the **Kerberos keytab file**, and click **Upload**. The uploaded principals are displayed in **Currently uploaded principals**.

   If a Connection Policy uses an SSH Authentication Policy with **GSSAPI-based single sign-on** together with a Usermapping Policy, then SPS stores the user principal as the gateway user, and the target username as the server username in the session database. If you want to allow your users to use a username on the target server that is different from their principal, configure a Usermapping Policy for your SSH connections. For details, see "Configuring usermapping policies" in the Administration Guide.
5. (Optional) If more than one realm is deployed on your network, you have to specify the mapping from the server's DNS domain name to the name of its realm. To map hostnames onto Kerberos realms, click +.
6. Navigate to **SSH Control > Connections** and configure the SSH connection as follows. For details on configuring connections in general, see Configuring connections on page 434.
   a. Select **Use fixed address** or **Inband destination selection** as **Target**.
   b. Select the Kerberos **Authentication policy**.

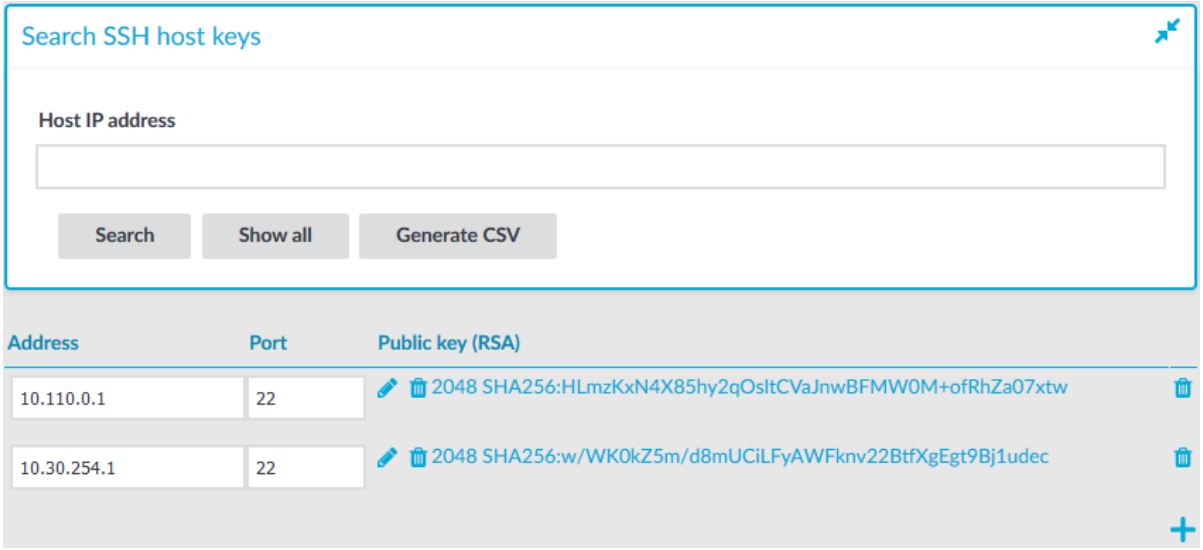
Server host keys

The host keys of the trusted servers can be managed on the **SSH Control > Server Host Keys** page. When a client tries to connect to a server, One Identity Safeguard for Privileged Sessions (SPS) verifies the host key of the server. SPS allows connections only to the servers listed on this page, unless the **Accept key for the first time** or the **Accept certificate for the first time** option is enabled in the connection policy.

- To display the stored host keys of a host, enter its IP address into **Host IP address** field and click **Search**. Note that the search requires the exact IP address, and does not support wildcard characters.

  You can use IPv4 and IPv6 addresses as well.
- To display the list of stored host keys, click **Show All**. Note that this function does not work if there are more than 250 hosts in the database. In this case, use **Generate CSV** instead.
- To export the entire list into as a comma-separated list, click **Generate CSV**.

Figure 195: SSH Control > Server Host Keys — Server host keys



Automatically adding the host keys of a server to One Identity Safeguard for Privileged Sessions (SPS)

The host keys of the servers can be added either automatically or manually.

To add the host key automatically

1. Navigate to the **SSH Control > Connections**.
2. Configure a connection: fill the **From**, **To**, and **Port** fields.

   You can use IPv4 and IPv6 addresses as well.

   - To configure a transparent connection, enter the IP address of the server into the **To** field.
   - To configure a non-transparent connection, enter the IP address of SPS into the **To** field, and the address of the target server into the **Target** field.
3. Click to display the advanced settings and verify that the **Server side hostkey settings > Plain host key check** option is set to **Accept key for the first time**.

   Click Commit.
4. Initiate an SSH connection from the client to the server. SPS will automatically record the host key of the server — the server's IP address and the host key will be listed on the **SSH Control > Server Host Keys** page.

Manually adding the host key of a server

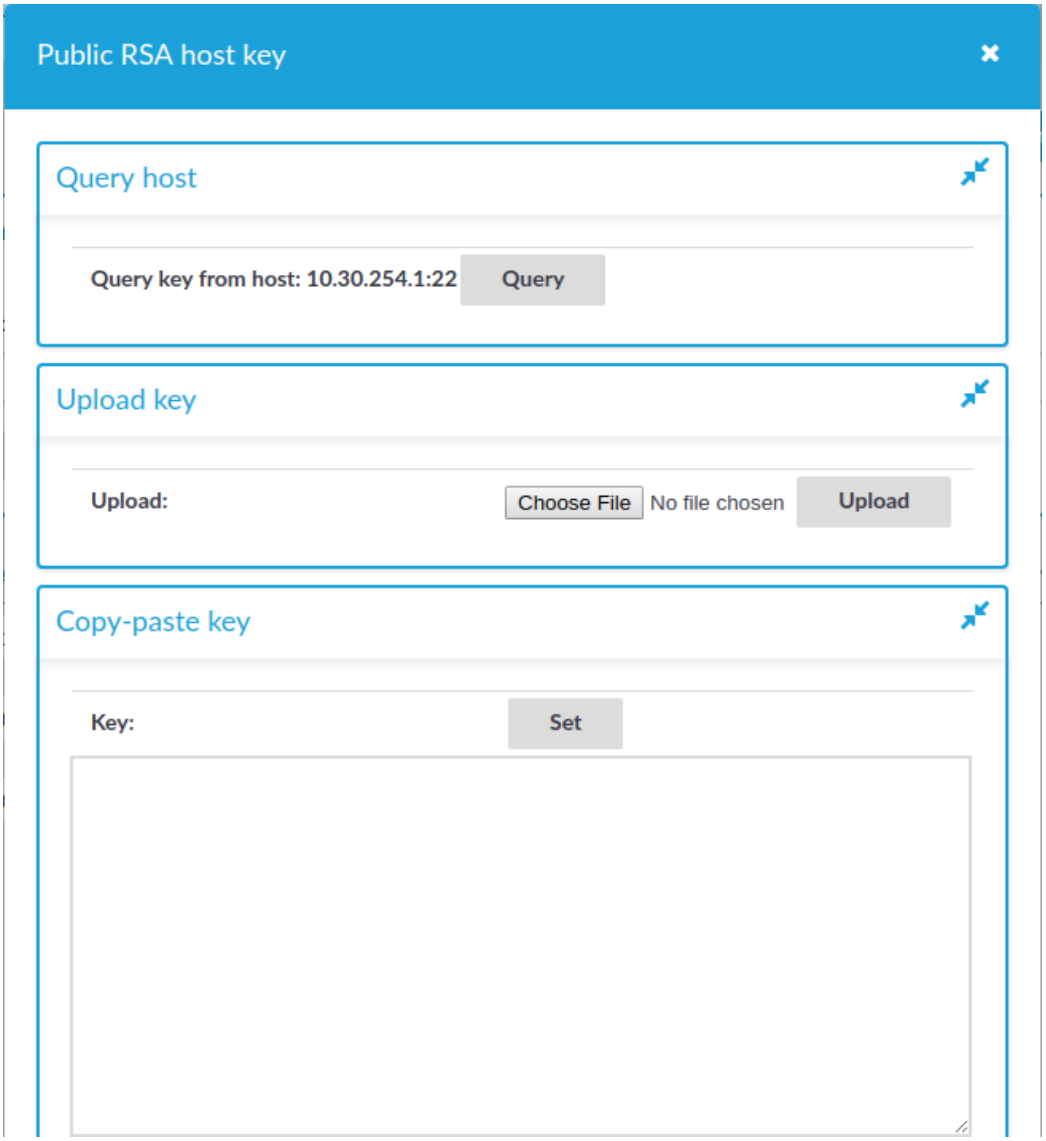
The following describes how to add the host key manually.

NOTE:
One Identity recommends using 2048-bit RSA keys (or stronger).

To add the host key manually

1. Navigate to the **SSH Control > Server Host Keys** and click +.
2. Enter the IP address and port of the server into the **Address** and **Port** fields.
   You can use IPv4 and IPv6 addresses as well.
3. To set the host key of the server, complete the following steps:
   a. To add the RSA fingerprint of the server, click in the **Public key (RSA)** field.
      A pop-up window is displayed.

b. **Figure 196: SSH Control > Server Host Keys — Uploading server host keys**



- Select **Query** to retrieve the host key from the server.
- To upload the host key manually, select **Browse**, select the file, and click **Upload**. Optionally, you can also paste the key into the **Copy-paste key** section and select **Upload**.

  Close the window.

4. Click Commit.

Creating and editing protocol-level SSH settings

SSH settings determine the parameters of the connection on the protocol level. For example, when the server-side connection is built, the timeout value, and greeting message of the connection. The following parameters determine which algorithms are used in the connections, and can be set independently for the client and the server side: key exchange, host key, cipher, MAC, and compression algorithms.

CAUTION:

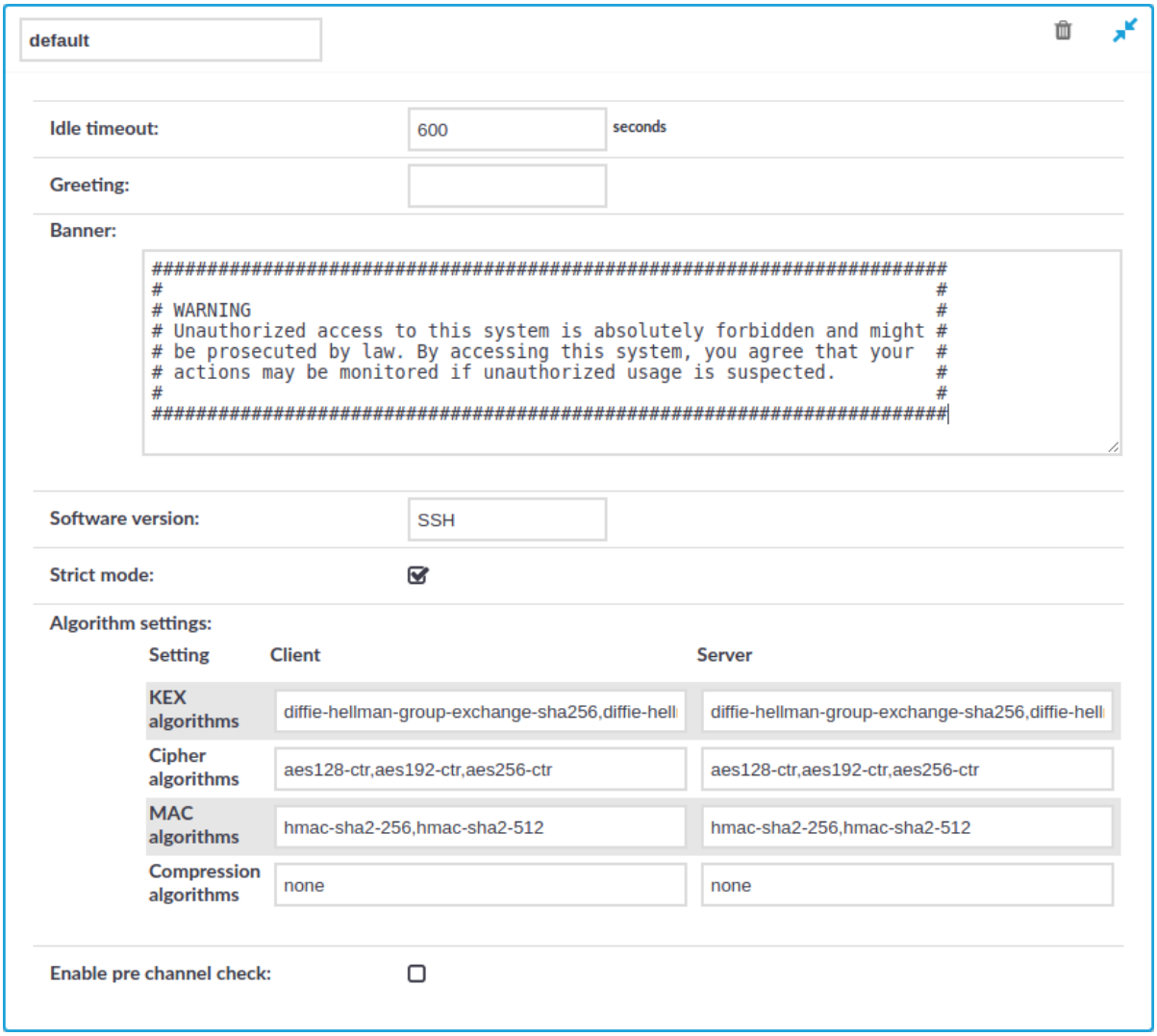
Before modifying any of the algorithm settings, check whether the default algorithms are supported by your SSH client and server.

If yes, then you can leave these settings untouched.

If not and you need to amend the default algorithm settings, ensure that the client and server sides are harmonized. You can either do that in One Identity Safeguard for Privileged Sessions (SPS) or on the client/server itself.

Note that modifying algorithm settings in SPS is recommended to advanced users only. If you are unsure about which settings to amend, then contact our Support Team for assistance.

Figure 197: SSH Control > Settings — SSH settings



To create a new SSH settings profile or edit an existing one

1. Navigate to the **SSH Control > Settings** and click + to create an SSH setting profile. Enter a name for the profile (for example `strongencryption`).
2. Click ⤢ to display the parameters of the SSH connection.
3. To set a connection timeout value, enter a value in the **Idle timeout** field in milliseconds. To avoid early timeout, set it to a larger value, for example a week (604800000 milliseconds).

CAUTION:

Determining if a connection is idle is based on the network traffic generated by the connection, not the activity of the user. For example, if an application or the taskbar of a graphical desktop displays the time which is updated every minute, it generates network traffic every minute, negating the effects of timeout values greater than one minute and preventing One Identity Safeguard for Privileged Sessions (SPS) from closing the connection.

4. To display a greeting message to the clients after connecting the server, enter the message into the **Greeting** field.
5. To display a banner message to the clients before authentication (as specified in RFC 4252 — The Secure Shell (SSH) Authentication Protocol), enter the message into the **Banner** field. For example, this banner can inform the users that the connection is audited.
6. Optional. You can specify additional text to append to the SSH protocol banner, for example to mask the OpenSSH version upon connection. Enter the text in the **Software version** field.
7. If needed, modify the encryption parameters. SPS enforces policies on the various elements of the encrypted SSH communication, such as the MAC, key-exchange, and cipher algorithms that are permitted to be used. The parameters can be set separately for the client and for the server side. The attributes are comma-separated strings listing the enabled methods/algorithms, in the order of preference.

   For a complete list of the available parameters, see Supported encryption algorithms on page 567.

   NOTE:

   Do not use the CBC block cipher mode, or the `diffie-hellman-group1-sha1` key exchange algorithm. For details, see "Supported encryption algorithms" in the Administration Guide.
8. To check the protocol-level parameters of the connections very strictly, select the **Strict mode** option. This option is enabled by default. When this option is enabled:

   SPS will reject connections that use unrealistic parameters, for example:

   - The number of columns and rows of the terminal is bigger or equal than 512.
   - The size of the screen is greater than 8192 pixels in either directions.

   SPS will reject port-forwarding connections where the address in the port-forwarding request and the channel-opening request does not match.

   NOTE:

   Strict mode can interfere with certain client or server applications.

NOTE:

Strict mode is not working with the Windows 10 internal Bash/WSL feature, because it uses a very large terminal window size. Using Windows 10 internal Bash/WSL is not supported.

9. Before establishing the server-side connection, SPS can evaluate the connection and channel policies to determine if the connection might be permitted at all, for example it is not denied by a Time Policy. To enable this function, select the **Enable pre channel check** option. That way SPS establishes the server-side connection only if the evaluated policies permit the client to access the server.
10. Click 

.
11. Select this settings profile in the **SSH settings** field of your connections.

Supported encryption algorithms

The following tables contain all the encryption algorithms you can configure One Identity Safeguard for Privileged Sessions (SPS) to recognize. If you use a configuration that is only partially supported, SPS might ignore the connection without warning.

NOTE:

Do not use the CBC block cipher mode, or the `diffie-hellman-group1-sha1` key exchange algorithm.

Key exchange algorithms

The default SPS configuration for both the client and the server is the following:

```
diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1
```

The following key exchange (KEX) algorithms are recognized:

Table 9: Key exchange (KEX) algorithms

Key exchange (KEX)	Default	Comment
diffie-hellman-group1-sha1	–	Not recommended
diffie-hellman-group14-sha1	✓	
diffie-hellman-group-exchange-sha1	✓	
diffie-hellman-group-exchange-sha256	✓	

Cipher algorithms

The default SPS configuration for both the client and the server is the following:

```
aes128-ctr,aes192-ctr,aes256-ctr
```

The following cipher algorithms are recognized:

Table 10: Cipher algorithms

Cipher algorithm	Default	Comment
3des-cbc	–	Not recommended
blowfish-cbc	–	Not recommended
twofish256-cbc	–	Not recommended
twofish-cbc	–	Not recommended
twofish192-cbc	–	Not recommended
twofish128-cbc	–	Not recommended
aes256-cbc	–	Not recommended
aes192-cbc	–	Not recommended
aes128-cbc	–	Not recommended
aes256-ctr	✓	
aes192-ctr	✓	
aes128-ctr	✓	
serpent256-cbc	–	Not recommended
serpent192-cbc	–	Not recommended
serpent128-cbc	–	Not recommended
arcfour	–	Not recommended
idea-cbc	–	Not recommended
cast128-cbc	–	Not recommended
none	–	Means no cipher algorithm; not recommended

Message authentication code (MAC) algorithms

The default SPS configuration for both the client and the server is the following:

```
hmac-sha2-256,hmac-sha2-512
```

The following MAC algorithms are recognized:

Table 11: Message Authentication Code (MAC) algorithms

MAC	Default
hmac-sha1	–
hmac-sha1-96	–
hmac-md5	–
hmac-md5-96	–
hmac-sha2-256	✓
hmac-sha2-512	✓

SSH compression algorithms

The default SPS configuration for both the client and the server is the following:

```
none
```

The following SSH compression algorithms are recognized:

Table 12: SSH compression algorithms

SSH compression algorithm	Default	Comment
zlib	–	
none	✓	Means no compression

Host key algorithms

The SPS configuration for both the client and the server is the following:

```
rsa-sha2-512,rsa-sha2-256,ssh-rsa
```

This list cannot be configured.

Telnet-specific settings

The following sections describe configuration settings available only for the Telnet protocol. Use the following policies to control who, when, and how can access the Telnet connection. For a list of supported client applications, see Supported protocols and client applications on page 38.

- *Channel Policy*: The Telnet protocol has only one channel type with no special configuration options. The available channel policy options are the following: **Type**, **From**, **Target**, **Time policy**, **4 eyes**, **Record audit trail**, **Gateway groups**, **Remote groups**, and **Content policy**. For details on configuring these options, see Creating and editing channel policies on page 448.
- *TLS support*: To enable TLS-encryption for your Telnet connections, see Enabling TLS-encryption for Telnet connections on page 571.
- *Authentication Policy*: Authentication policies describe the authentication methods allowed in a connection. Different methods can be used for the client and server-side connections. For details, see Creating a new Telnet authentication policy on page 574.
- *Telnet settings*: Telnet settings determine the parameters of the connection on the protocol level, including timeout value, and so on. For details, see Creating and editing protocol-level Telnet settings on page 578.
- *User lists in Channel Policies*: User lists affect Telnet connections only when they are used together with Gateway Authentication. For details, see Configuring gateway authentication on page 742.
- *Content Policy*: Content policies allow you to inspect the content of the connections for various text patterns, and perform an action if the pattern is found. For example, One Identity Safeguard for Privileged Sessions (SPS) can send an e-mail alert if a specific command is used in a Telnet terminal session. For details, see Creating a new content policy on page 452.
- *Authentication and Authorization plugin*:

  One Identity Safeguard for Privileged Sessions (SPS) provides a plugin framework to integrate SPS to external systems to authenticate or authorize the user before authenticating on the target server. Such plugins can also be used to request additional information from the users, for example, to perform multi-factor authentication.

  For details, see Integrating external authentication and authorization systems on page 768.

Enabling TLS-encryption for Telnet connections

The following steps describe how to enable TLS-encryption in a Telnet connection policy. Note that when using encryption, One Identity Safeguard for Privileged Sessions (SPS) automatically changes the port number of the connection policy to 992.

Prerequisites

Depending on your requirements, one or more of the following might be needed:

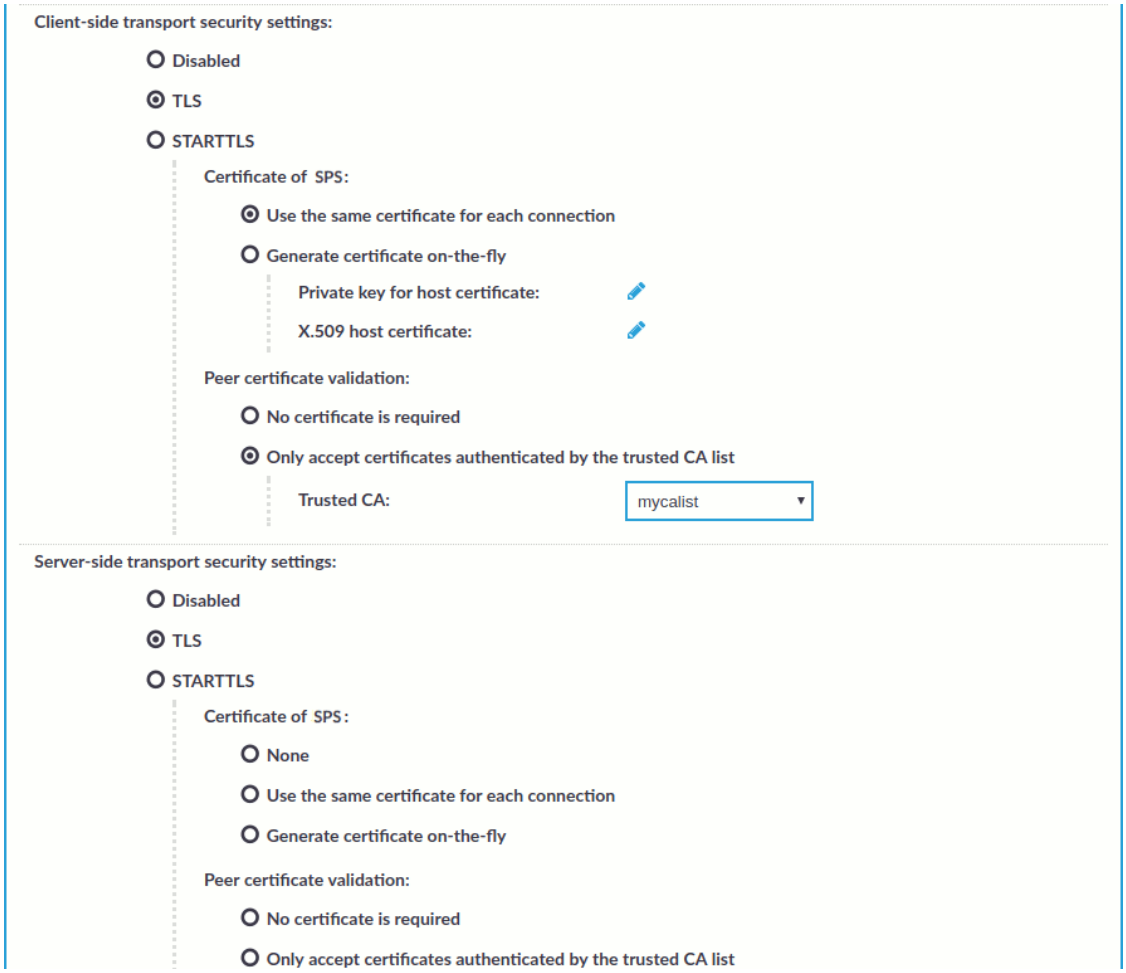
- An X.509 certificate and its private key. SPS can display the same certificate to the peers on both the client and the server side. You can also use different certificates for the client and server sides. Use your own PKI system to generate these certificates, as they cannot be created on SPS. Note that the Common Name of the certificate must contain the domain name or the IP address of SPS, otherwise the clients might reject the certificate.
- To generate certificates on-the-fly for a connection, a signing certificate authority is required. For details on creating a signing CA, see Signing certificates on-the-fly on page 478.
- To require the peers of SPS to have an X.509 certificate signed by a specific Certificate Authority, a list of the trusted certificate authorities is needed. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities on page 476.

One Identity recommends using 2048-bit RSA keys (or stronger).

To enable TLS-encryption in a Telnet connection policy

1. Navigate to **Telnet Control > Connections** and select the connection policy in which you want to enable TLS.

Figure 198: Telnet Control > Connections — Enabling TLS-encryption for Telnet connections



2. Set the encryption settings used between the client and SPS in the **Client-side transport security settings** section.

    - To require encryption, select **TLS**. When the connection is encrypted, SPS has to show a certificate to the peer.
    - To enable encrypted connections that use the STARTTLS method, select **STARTTLS**. Note that the peer must use the STARTTLS method. Unencrypted connections will be terminated after a brief period.

3. Select the certificate to show to the peers.

    - To use the same certificate for every peer, complete the following steps.

        1. Generate and sign a certificate for SPS in your PKI system, and export the certificate and its private key.
        2. Select **Use the same certificate for each connection**.

3. Select **Private key for host certificate**, click and upload the private key.
4. Select **X.509 host certificate**, click and upload the certificate.

- To use a separate certificate for every connection, complete the following steps.
  1. Create a certificate authority that will be used to sign the certificates that SPS shows to the peer. For details, see Signing certificates on-the-fly on page 478.
  2. Select **Generate certificate on-the-fly**.
  3. In the **Signing CA** field, select the certificate authority to use.

4. Select how SPS should authenticate the peers.
   - To permit connections from peers without requesting a certificate, select **No certificate is required**.
   - To permit connections only from peers with a valid certificate that was signed by a specific CA, complete the following steps.
     1. Create a list of trusted Certificate Authorities that will be used to validate the certificates of the peers. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities on page 476.
     2. Select **Only accept certificates authenticated by the trusted CA list**.
     3. Select the certificate authority list to use in the **Trusted CA** field.
5. Set the encryption settings used between SPS and the server in the **Server-side transport security settings** section.
   - To require encryption, select **TLS**. When the connection is encrypted, SPS has to show a certificate to the peer.
   - To enable encrypted connections that use the STARTTLS method, select **STARTTLS**. Note that the peer must use the STARTTLS method. Unencrypted connections will be terminated after a brief period.
6. Select the certificate to show to the server.
   - If the server does not require a certificate from SPS, select **None**.
   - To use the same certificate for every peer, complete the following steps.
     1. Generate and sign a certificate for SPS in your PKI system, and export the certificate and its private key.
     2. Select **Use the same certificate for each connection**.
     3. Select **Private key for host certificate**, click and upload the private key.
     4. Select **X.509 host certificate**, click and upload the certificate.
   - To use a separate certificate for every connection, complete the following steps.

1. Create a certificate authority that will be used to sign the certificates that SPS shows to the peer. For details, see Signing certificates on-the-fly on page 478.
2. Select **Generate certificate on-the-fly**.
3. Select the certificate authority to use in the **Signing CA** field.

Limitations

NOTE:

When using the **Use the same certificate for each connection** option and the connection policy that allows access to multiple servers using HTTPS, the client applications will display a warning because the certificate used in the connection will be invalid (namely, the Common Name of the certificate will not match the hostname or IP address of the server).

NOTE:

Import the certificate of the signing Certificate Authority to your clients. Otherwise, the client applications will display a warning due to the unknown Certificate Authority.

7. Select how SPS should authenticate the peers.
   - To permit connections from peers without requesting a certificate, select **No certificate is required**.
   - To permit connections only from peers with a valid certificate that was signed by a specific CA, complete the following steps.
     1. Create a list of trusted Certificate Authorities that will be used to validate the certificates of the peers. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities on page 476.
     2. Select **Only accept certificates authenticated by the trusted CA list**.
     3. Select the certificate authority list to use in the **Trusted CA** field.
8. Click Commit.

Expected result

The encryption settings are applied to the connection policy.

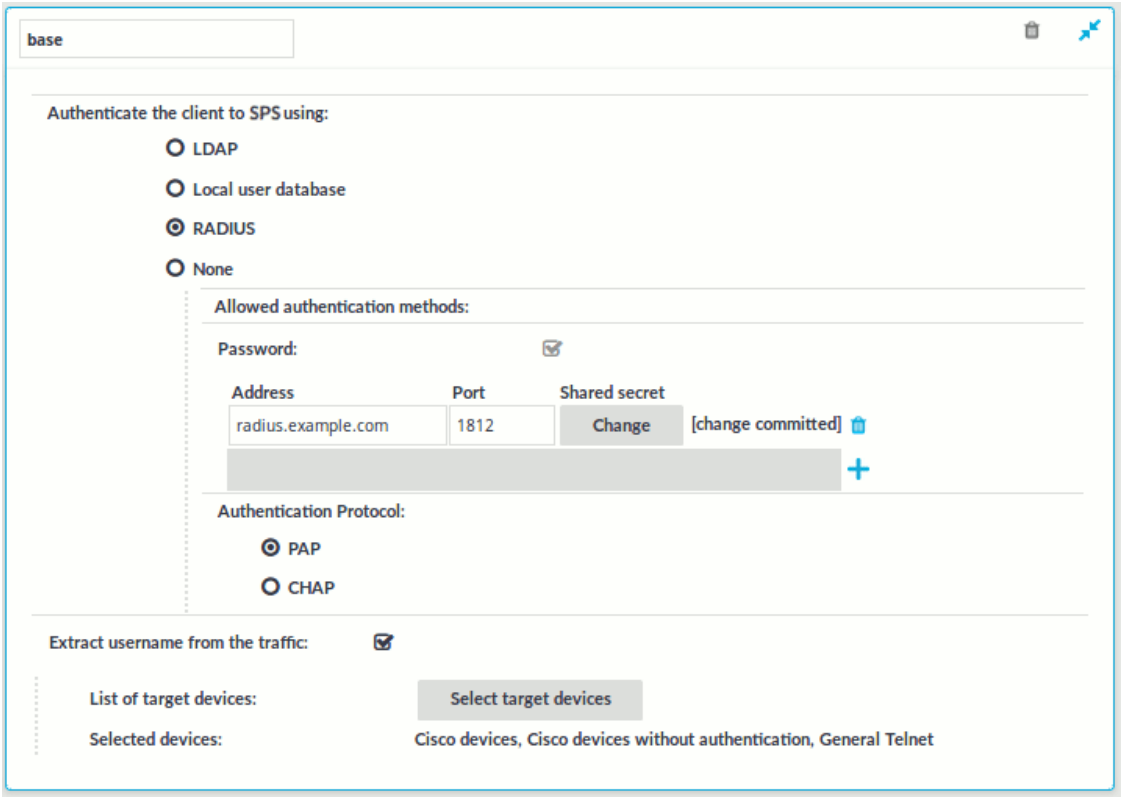
Creating a new Telnet authentication policy

An authentication policy is a list of authentication methods that can be used in a connection. Connection definitions refer to an authentication policy to determine how the client can authenticate to the target server. Separate authentication methods can be used on the client and the server-side of the connection.

To create a new authentication policy

1. Navigate to **Telnet Control > Authentication Policies**, and click +.

   Figure 199: Telnet Control > Authentication Policies — Configuring Telnet authentication policies

2. Enter a name for the policy into the **Name** field.
3. Select the authentication method used on the client-side in the **One Identity Safeguard for Privileged Sessions (SPS)Authenticate the client to SPS using** field. For the client-side connection, SPS can authenticate the client inband (within the Telnet protocol) using the following authentication methods:
   - **LDAP**: SPS will authenticate the client to the LDAP database set in the **LDAP Server** of the connection policy. To use LDAP authentication on the client side, select **Authenticate the client to SPS using > LDAP**.

NOTE:

SPS will authenticate the client-side connection to the LDAP server configured in the connection policy. This is not necessarily the same as the LDAP server used to authenticate the users accessing the SPS web interface.

- **Local user database**: Authenticate the client locally on the SPS gateway using a **Local user database**. Select the database to use in the **Local user database** field. For details on creating a **Local User Database**, see Creating a Local User Database on page 486.
- **RADIUS**: SPS will authenticate the client to the specified RADIUS server. Select **Authenticate the client to SPS using > RADIUS**, enter the IP address or hostname of the RADIUS server into the **Address** field, the port number of the RADIUS server into the **Port** field, and the shared secret of the RADIUS server into the **Shared secret** field. Only password-authentication is supported (including one-time passwords), challenge-response based authentication is not.

  Use an IPv4 address.

  To add more RADIUS servers, click + and fill in the respective fields.
- **None**: Do not perform client-side authentication, the client will authenticate only on the target server.

  CAUTION:

  Hazard of security breach. If the None authentication option is selected on the client side and SPS is configured to use public-key or certificate based authentication on the server, the user will not be authenticated at all unless gateway authentication is required for the connection.

4. Click Commit.

NOTE:

- The client-side authentication settings apply for authenticating the user inband (that is, within the SSH protocol) to the SPS gateway, and is independent from the gateway authentication performed on the SPS web interface. The web-based gateway authentication is an out-of-band gateway authentication method that can be required by the connection policy. For details on out-of-band gateway authentication, see Configuring out-of-band gateway authentication on page 744.

  Gateway authentication on the SPS web interface can be used together with authentication policies. In an extreme setting, this would mean that the user has to perform three authentications: a client-side gateway authentication within the SSH protocol to SPS, an out-of-band gateway authentication on the SPS web interface, and a final authentication on the target server.
- The Connection Policy will ignore the settings for server-side authentication (set under **Relayed authentication methods**) if a Credential Store is used in the Connection Policy.

Extracting username from Telnet connections

For specific devices, it is now possible to extract the username from Telnet connections with the help of patterns (including TN3270 and TN5250 systems).

To select patterns or request a custom pattern

1. Navigate to **Telnet Control > Authentication Policies** and enable **Extract username from the traffic**.
2. Click **Select target devices** to display the list of available target devices. Select the respective device(s) in the **Available devices** column and click **Add**.

   NOTE: You can only add one TN3270 specific device to the authentication policy.

   To remove a device from the **Target devices** column, select it and click **Remove**.
3. Click **OK**. The target devices are listed after **Selected devices**.
4. If you cannot find your device in the list of available target devices, request a custom Pattern Set. To do this, contact our Support Team.
5. To upload the custom pattern set you received, navigate to **Telnet Control > Pattern Sets**, browse for the file and click **Upload**.
6. To delete a custom Pattern Set from One Identity Safeguard for Privileged Sessions (SPS), click in the respective row. Generic Pattern Sets cannot be deleted.

Creating and editing protocol-level Telnet settings

Procedure

Telnet settings determine the parameters of the connection on the protocol level, including timeout value, and so on. Complete the following procedure to create a new Telnet settings profile or edit an existing one:

CAUTION:

Modifying the Telnet settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

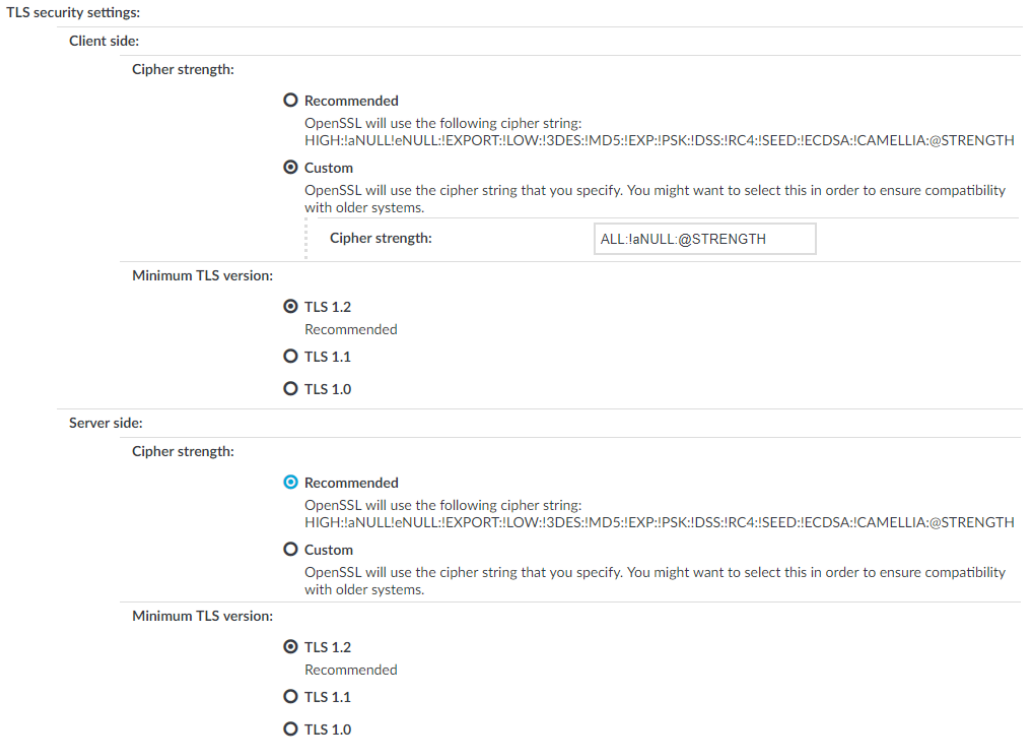
To create and edit protocol-level Telnet settings

1. Navigate to the **Settings** tab of the **Telnet Control** menu item and click + to create a Telnet setting profile. Enter a name for the profile (for example `telnet_special`).
2. Click ⤢ to display the parameters of the connection.
3. Modify the parameters as needed. The following parameters are available:
   - **Idle timeout**: Timeout value for the connection in milliseconds. To avoid early timeout, set it to a larger value, for example a week (604800000 milliseconds).

     CAUTION:

     Determining if a connection is idle is based on the network traffic generated by the connection, not the activity of the user. For example, if an application or the taskbar of a graphical desktop displays the time which is updated every minute, it generates network traffic every minute, negating the effects of timeout values greater than one minute and preventing One Identity Safeguard for Privileged Sessions (SPS) from closing the connection.
   - **Enable pre channel check**: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all, SPS does not establish the server-side connection.
   - To configure TLS security settings on both the **Client side** and the **Server side**, proceed to **TLS security settings**.

Figure 200: <Protocol> Control > Settings > TLS security settings - configuring TLS security settings



- **Cipher strength** specifies the cipher string OpenSSL will use. The following settings options are possible:
  - **Recommended**: this setting only uses ciphers with adequate security level.
  - **Custom**: this setting allows you to specify the list of ciphers you want to permit SPS to use in the connection. This setting is only recommended in order to ensure compatibility with older systems. For more details on customizing this list, check the 'openssl-ciphers' manual page on your SPS appliance.

    For example: `ALL:!aNULL:@STRENGTH`
- **Minimum TLS version** specifies the minimal TLS version SPS will offer during negotiation. The following settings options are possible:
  - **TLS 1.2**: this setting will only offer TLS version 1.2 during negotiation. This is the recommended setting.
  - **TLS 1.1**: this setting will offer TLS version 1.1 and later versions during negotiation.
  - **TLS 1.0**: this setting will offer TLS version 1.0 and later versions during negotiation.

NOTE:

Note that SPS only permits TLS-encrypted connections. SSLv3 is not supported.

4. Click Commit.
5. To display a banner message to the clients before authentication, enter the message into the **Banner** field. For example, this banner can inform the users that the connection is audited.
6. Select this settings profile in the **TELNET settings** field of your connections.

Inband destination selection in Telnet connections

When using inband destination selection in Telnet connections, the user can provide the server address and the username using the following methods:

- By setting the `TELNET ENVIRON` option using the `SERVER` environment variable in the `server:port` format.
- By setting the `TELNET ENVIRON` option using the `USER` environment variable in the `user@server:port` format.
- If neither the `SERVER` nor the `USER` environment variable, One Identity Safeguard for Privileged Sessions (SPS) displays a terminal prompt where the user can enter the username and the server address.

Limitations of using TN5250 protocol with IBM iSeries Access for Windows

Using the TN5250 protocol with IBM iSeries Access for Windows is not supported in non-transparent mode if the client tries to set up all its connections using One Identity Safeguard for Privileged Sessions's (SPS's) IP addresses (for example, when both the client and SPS are within the same zone and the firewall is behind SPS). This is problematic in the case of an IBM iSeries Access for Windows client, which initiates administrative communication with components other than the Telnet server itself. Bypassing non-audited traffic goes against the purpose of non-transparent mode.

Possible workarounds:

- Use a local service file instead of active client-server communication.
- Use transparent mode.

- Use single-interface transparent mode and policy-based routing.

VMware Horizon View connections

The following sections describe how to use One Identity Safeguard for Privileged Sessions (SPS) to control and audit VMware Horizon View (formerly known as VMware View) connections. When using SPS to control and audit VMware Horizon View connections, the following requirements and restrictions apply:

- Only connections using the Remote Desktop (RDP) display protocol are supported. Connections using the PCoIP or HP Remote Graphics Software display protocols are not supported.
- Both direct connections and tunnel connections are supported.
- The VMware Horizon View connections must pass SPS directly. It is best if SPS is deployed directly before the Virtual Desktops accessed with VMware Horizon View, and connections are configured in transparent mode.

  Deploying SPS that way has the advantage of auditing connections even if the clients access the Virtual Desktops directly, without using a View Connection Server.

  NOTE:

  Using non-transparent mode is also possible if the VMware Horizon View traffic is routed to SPS with an external device (for example, a firewall).

SPS treats VMware Horizon View connections that satisfy these criteria as common RDP connections. All the features of SPS that are available for RDP connections can be used with VMware Horizon View connections as well, for example, four-eyes authorization, auditing and replaying, indexing the recorded audit trails, and so on. For details on RPD-specific settings, see RDP-specific settings on page 515.

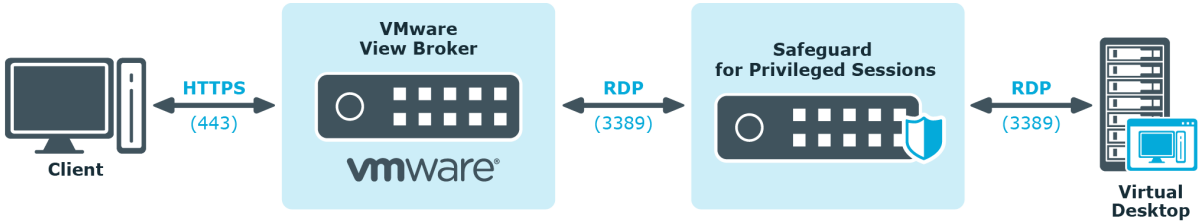
One Identity Safeguard for Privileged Sessions (SPS) deployment scenarios in a VMware environment

One Identity Safeguard for Privileged Sessions (SPS) supports a variety of deployment scenarios, which make it really flexible when it comes to deployment. The following network topologies illustrate typical SPS VMware Horizon View deployment scenarios.

Client - Broker - SPS - Server

SPS is deployed between the Broker and the virtual desktop, where the RDP traffic is embedded into a HTTPS tunnel between the Client and the Broker.

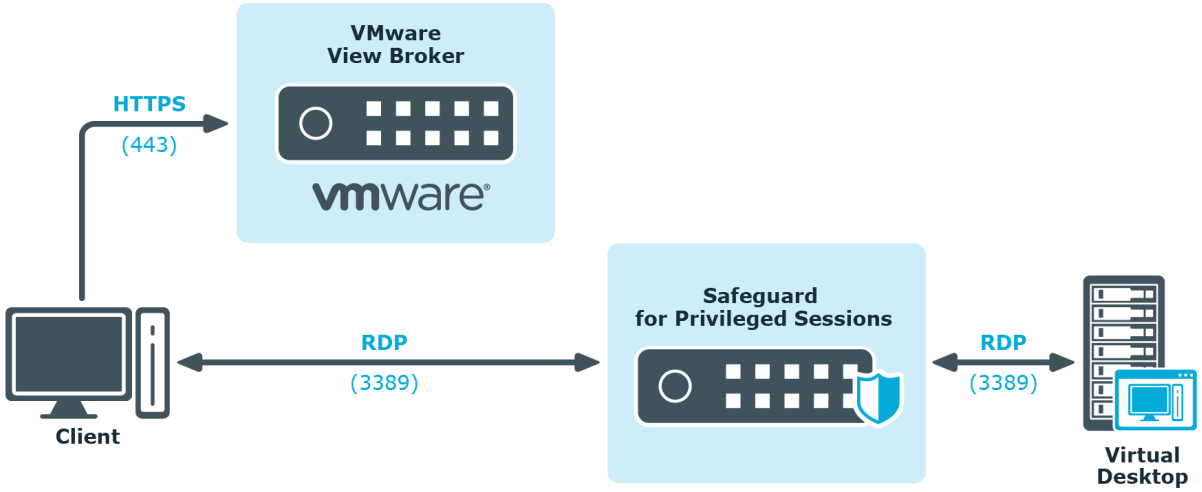
Figure 201: Client - Broker - SPS - Server



Client - SPS - Server

SPS is deployed between the Client and the virtual desktop, the client makes a direct RDP connection to the Server, without tunneling it through the Broker.

Figure 202: Client - SPS - Server



VNC-specific settings

The following sections describe configuration settings available only for the Virtual Networking (VNC) protocol. Use the following policies to control who, when, and how can access the VNC connections. For a list of supported client applications, see Supported protocols and client applications on page 38.

CAUTION:

To monitor VNC connections, enable user authentication on your VNC server. One Identity Safeguard for Privileged Sessions (SPS) automatically terminates unauthenticated connections.

- *Channel Policy*: The VNC protocol has only one channel type with no special configuration options. The available channel policy options are the following: **Type**, **From**, **Target**, **Time policy**, **Four-eyes**, **Record audit trail**, **Gateway groups**, **Remote groups**, and **Content policy**. For details on configuring these options, see Creating and editing channel policies on page 448.
- *TLS support*: To enable TLS-encryption for your VNC connections, see Enabling TLS-encryption for VNC connections on page 584.
- *VNC settings*: VNC settings determine the parameters of the connection on the protocol level, including timeout value, and so on. For details, see Creating and editing protocol-level VNC settings on page 589.
- *User lists in Channel Policies*: User lists affect VNC connections only when they are used together with Gateway Authentication. For details, see Configuring gateway authentication on page 742.
- *Content Policy*: Content policies allow you to inspect the content of the connections for various text patterns, and perform an action if the pattern is found. For example, SPS can send an e-mail alert if a specific window title appears in RDP and VNC connections. For details, see Creating a new content policy on page 452.

Enabling TLS-encryption for VNC connections

The following steps describe how to enable TLS-encryption in a VNC connection policy.

NOTE:

Some vendors may use custom protocol elements and TLS-encryption that do not have available documentation. As a result, these cannot be audited by One Identity Safeguard for Privileged Sessions (SPS). Regardless of vendors, only the custom features described in the RFC 6143 are supported. As for encryptions, only those completely TLS-encapsulated streams can be processed where the TLS encryption process was started before the VNC protocol handshake.

Prerequisites

Depending on your requirements, one or more of the following might be needed:

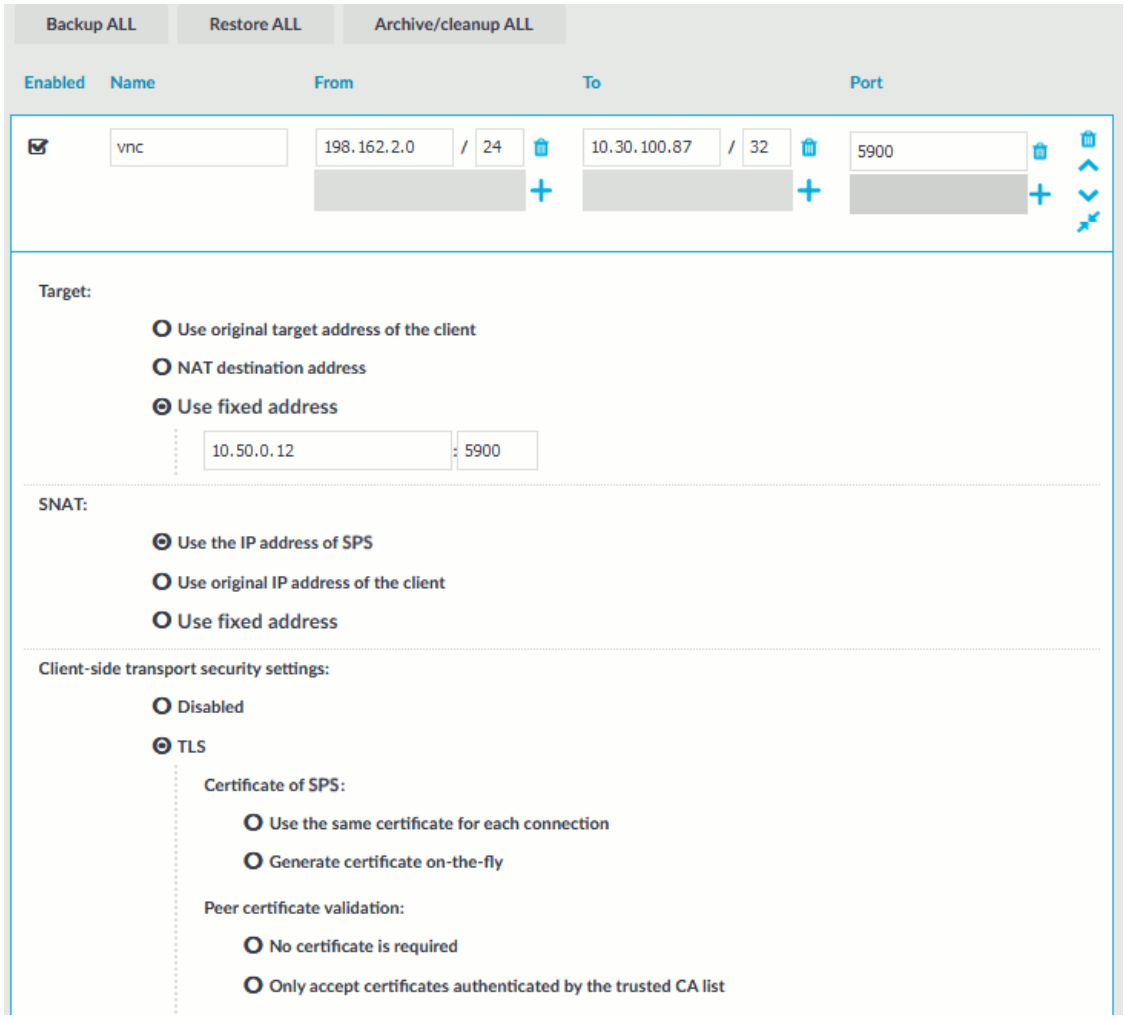
- An X.509 certificate and its private key. SPS can display the same certificate to the peers on both the client and the server side. You can also use different certificates for the client and server sides. Use your own PKI system to generate these certificates, as they cannot be created on SPS. Note that the Common Name of the certificate must contain the domain name or the IP address of SPS, otherwise the clients might reject the certificate.
- To generate certificates on-the-fly for a connection, a signing certificate authority is required. For details on creating a signing CA, see Signing certificates on-the-fly on page 478.
- To require the peers of SPS to have an X.509 certificate signed by a specific Certificate Authority, a list of the trusted certificate authorities is needed. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities on page 476.

One Identity recommends using 2048-bit RSA keys (or stronger).

To enable TLS-encryption in a VNC connection policy

1. Navigate to **VNC Control > Connections** and select the connection policy in which you want to enable TLS.

Figure 203: VNC Control > Connections — Enabling TLS-encryption for VNC connections



2. Set the encryption settings used between the client and SPS in the **Client-side transport security settings** section.

   To require encryption, select **TLS**. When the connection is encrypted, SPS has to show a certificate to the peer.

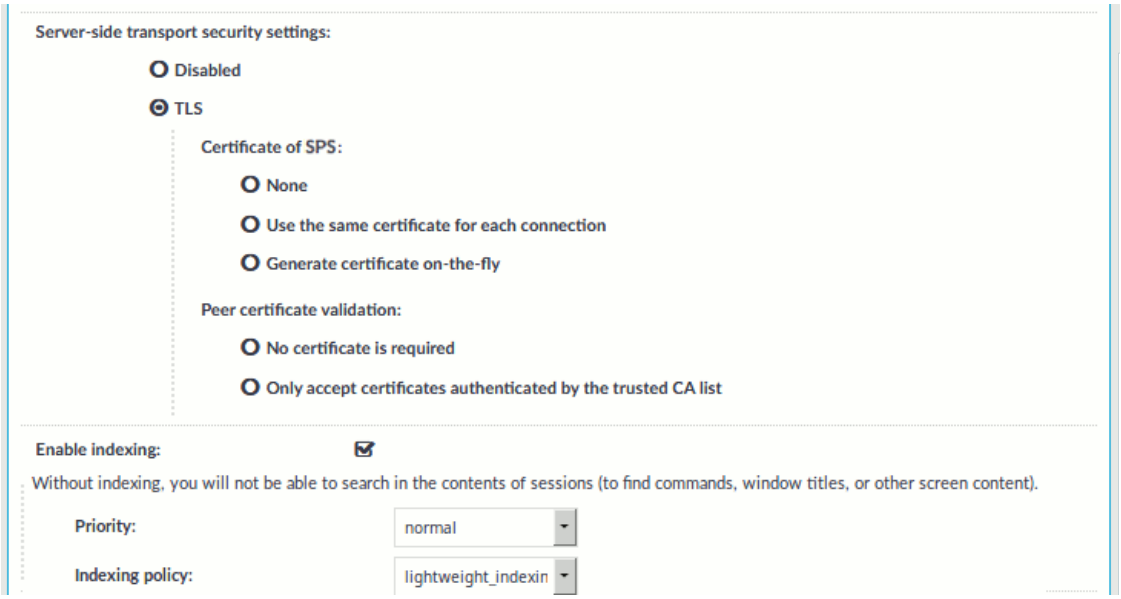
3.
   - To use the same certificate for every peer, complete the following steps.
     1. Generate and sign a certificate for One Identity Safeguard for Privileged Sessions (SPS) in your PKI system, then export the certificate and its private key.
     2. Select **Use the same certificate for each connection**.
     3. Select **Private key for host certificate**, click and upload the

private key.

4. Select **X.509 host certificate**, click and upload the certificate.

- To use a separate certificate for every connection, complete the following steps.
  1. Create a certificate authority that will be used to sign the certificates that SPS shows to the peer. For details, see Signing certificates on-the-fly on page 478.
  2. Select **Generate certificate on-the-fly**.
  3. Select the certificate authority to use in the **Signing CA** field.

4. Select how SPS should authenticate the peers.
   - To permit connections from peers without requesting a certificate, select **No certificate is required**.
   - To permit connections only from peers with a valid certificate that was signed by a specific CA, complete the following steps.
     1. Create a list of trusted Certificate Authorities that will be used to validate the certificates of the peers. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities on page 476.
     2. Select **Only accept certificates authenticated by the trusted CA list**.
     3. Select the certificate authority list to use in the **Trusted CA** field.
5. Set the encryption settings used between SPS and the server in the **Server-side transport security settings** section.



   To require encryption, select **TLS**. When the connection is encrypted, SPS has to show a certificate to the peer.
6. Select the certificate to show to the server.

- If the server does not require a certificate from SPS, select **None**.
- To use the same certificate for every peer, complete the following steps.
    1. Generate and sign a certificate for SPS in your PKI system, and export the certificate and its private key.
    2. Select **Use the same certificate for each connection**.
    3. Select **Private key for host certificate**, click and upload the private key.
    4. Select **X.509 host certificate**, click and upload the certificate.
- To use a separate certificate for every connection, complete the following steps.
    1. Create a certificate authority that will be used to sign the certificates that SPS shows to the peer. For details, see Signing certificates on-the-fly on page 478.
    2. Select **Generate certificate on-the-fly**.
    3. Select the certificate authority to use in the **Signing CA** field.

Limitations

NOTE:

When using the **Use the same certificate for each connection** option and the connection policy that allows access to multiple servers using HTTPS, the client applications will display a warning because the certificate used in the connection will be invalid (namely, the Common Name of the certificate will not match the hostname or IP address of the server).

NOTE:

Import the certificate of the signing Certificate Authority to your clients. Otherwise, the client applications will display a warning due to the unknown Certificate Authority.

7. Select how SPS should authenticate the peers.
    - To permit connections from peers without requesting a certificate, select **No certificate is required**.
    - To permit connections only from peers with a valid certificate that was signed by a specific CA, complete the following steps.
        1. Create a list of trusted Certificate Authorities that will be used to validate the certificates of the peers. For details on creating a trusted CA list, see Verifying certificates with Certificate Authorities on page 476.
        2. Select **Only accept certificates authenticated by the trusted CA list**.
        3. Select the certificate authority list to use in the **Trusted CA** field.

8. Click Commit.

Expected result

The encryption settings are applied to the connection policy.

Creating and editing protocol-level VNC settings

VNC settings determine the parameters of the connection on the protocol level, including timeout value, and so on.

CAUTION:

Modifying the VNC settings is recommended only to advanced users. Do not modify these settings unless you exactly know what you are doing.

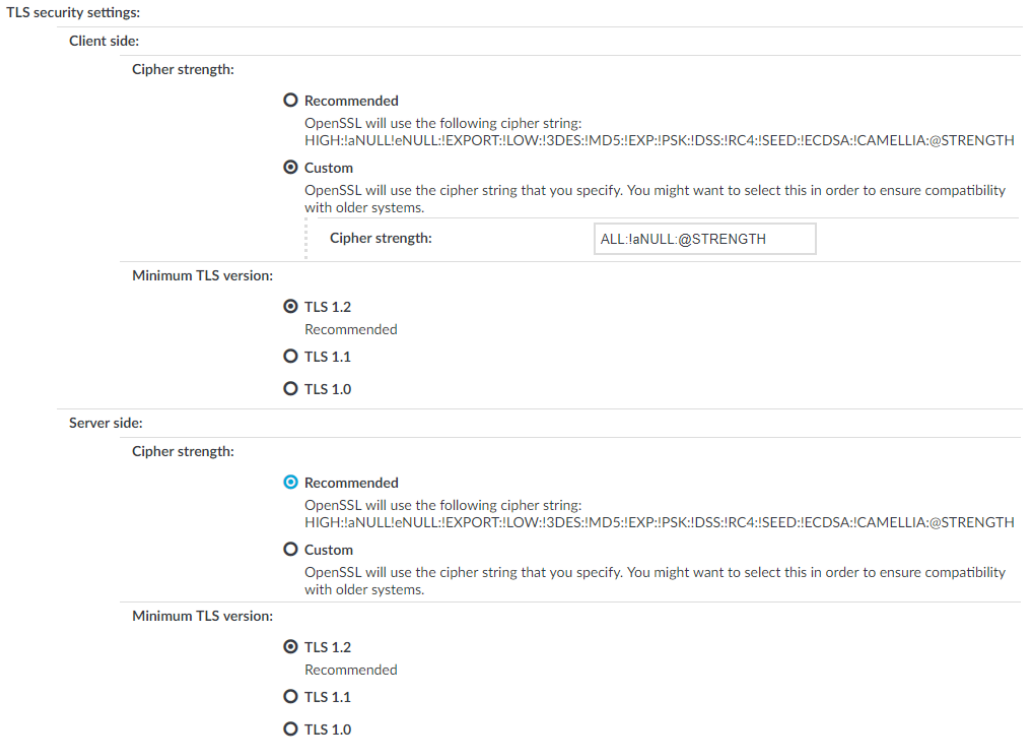
To create a new VNC settings profile or edit an existing one

1. Navigate to **VNC Control > Settings** and click + to create a VNC setting profile. Enter a name for the profile (for example `vnc_special`).
2. Click ⤢ to display the parameters of the connection.
3. Modify the parameters as needed. The following parameters are available:
   - **Idle timeout**: Timeout value for the connection in milliseconds. To avoid early timeout, set it to a larger value, for example a week (604800000 milliseconds).

     CAUTION:

     Determining if a connection is idle is based on the network traffic generated by the connection, not the activity of the user. For example, if an application or the taskbar of a graphical desktop displays the time which is updated every minute, it generates network traffic every minute, negating the effects of timeout values greater than one minute and preventing One Identity Safeguard for Privileged Sessions (SPS) from closing the connection.
   - **Enable pre channel check**: Select this option to evaluate the connection and channel policies before establishing the server-side connection. That way if the connection is not permitted at all, SPS does not establish the server-side connection.
   - To configure TLS security settings on both the **Client side** and the **Server side**, proceed to **TLS security settings**.

Figure 204: <Protocol> Control > Settings > TLS security settings - configuring TLS security settings



- **Cipher strength** specifies the cipher string OpenSSL will use. The following settings options are possible:
  - **Recommended**: this setting only uses ciphers with adequate security level.
  - **Custom**: this setting allows you to specify the list of ciphers you want to permit SPS to use in the connection. This setting is only recommended in order to ensure compatibility with older systems. For more details on customizing this list, check the 'openssl-ciphers' manual page on your SPS appliance.

    For example: `ALL:!aNULL:@STRENGTH`
- **Minimum TLS version** specifies the minimal TLS version SPS will offer during negotiation. The following settings options are possible:
  - **TLS 1.2**: this setting will only offer TLS version 1.2 during negotiation. This is the recommended setting.
  - **TLS 1.1**: this setting will offer TLS version 1.1 and later versions during negotiation.
  - **TLS 1.0**: this setting will offer TLS version 1.0 and later versions during negotiation.

> **NOTE:**
>
> Note that SPS only permits TLS-encrypted connections. SSLv3 is not supported.

4. Click 

.
5. Select this settings profile in the **VNC settings** field of your connections.

Indexing audit trails

One Identity Safeguard for Privileged Sessions (SPS) can index the contents of audit trails using its own indexer service or external indexers. Indexing extracts the text from the audit trails and segments it to tokens. A token is a segment of the text that does not contain whitespace: for example words, dates (`2009-03-14`), MAC or IP addresses, and so on. The indexer returns the extracted tokens to SPS, which builds a comprehensive index from the tokens of the processed audit trails.

Once indexed, the contents of the audit trails can be searched from the web interface. SPS can extract the commands typed and the texts seen by the user in terminal sessions, and text from graphical protocols like RDP, Citrix ICA, and VNC. Window titles are also detected.

SPS has an internal indexer, which runs on the SPS appliance. In addition to the internal indexer, external indexers can run on Linux hosts.

Processing and indexing audit trails requires significant computing resources. If you have to audit lots of connections, or have a large number of custom reports configured, consider using an external indexer to decrease the load on SPS. For sizing recommendations, ask your One Identity partner or contact our Support Team.

- *The internal indexer service* runs on the SPS appliance. It supports languages based on the Latin-, Greek- and Cyrillic alphabets, as well as Chinese, Japanese and Korean languages, allowing it to recognize texts from graphical audit trails in 100+ languages. It can also generate screenshots for content search results.

  Recognizing and OCR-ing CJK (Chinese, Japanese and Korean) languages must be licensed separately.
- *The external indexer* runs on Linux hosts and instances. It uses the same engine as the indexer service of SPS, and has the same capabilities and limitations.

  SPS can work with multiple external indexers to process audit trails.

NOTE:

If a text is displayed for less than 1 second, it is not indexed.

If you have indexed trails, the index itself is also archived:

When using the **Indexer service**: Every 30 days, unless the **Backup & Archive/Cleanup > Archive/Cleanup policies > Delete data from SPS after** is configured to occur less frequently (more than 30 days). For example, if the **Delete data from SPS after** is 60 days, the index will be archived every 60 days. The content of the archived index will be the content that was available X days before the archival date, where X is the number in the **Delete data from SPS after** field.

CAUTION:

Hazard of data loss Make sure you also backup your data besides archiving (for details, see Data and configuration backups on page 146). If a system crash occurs, you can lose up to 30 days of index, since the index is only archived in every 30 days.

- To configure SPS to index the entire content of the audited connections, complete Configuring the internal indexer on page 593.

  Indexing also needs to be enabled in the connection policy of the monitored connections.
- To configure external indexers, complete Configuring external indexers on page 599.
- To monitor the status of the servers indexing the audit trails, see Monitoring the status of the indexer services on page 617.
- To create custom reports from the contents of the audit trails, complete Creating reports from audit trail content on page 782.

Reindex audit trails

In certain cases, reindexing already indexed audit trails might be necessary, for example, if the audit trails were indexed without full screen content but you still need to search in screen content. In this case, the audit trails can be reindexed with a different indexer configuration to perform screen content extraction. For more information, contact our Support Team.

Configuring the internal indexer

The following describes how to configure One Identity Safeguard for Privileged Sessions (SPS) to index the audit trails.

Indexing is a resource intensive (CPU and hard disk) operation, and depending on the number of processed audit trails and parallel connections passing SPS, may affect the performance of SPS. Test it thoroughly before enabling it in a production environment that is under heavy load. If your SPS appliance cannot handle the connections and the indexing, consider using external indexers (see "Configuring external indexers" in the Administration Guide) to decrease the load on SPS. For sizing recommendations, ask your One Identity partner or contact our Support Team.

Note that the minimum value of **Backup & Archive/Cleanup > Archive/Cleanup policies > Delete data from SPS after** is `30` days when using the indexer service. If you previously had a setting lower than this, it will still archive the index after `30` days when the indexer service is used.

NOTE:

Only those audit trails will be processed that were created after full-text indexing had been configured for the connection policy. It is not possible to process already existing audit trails.

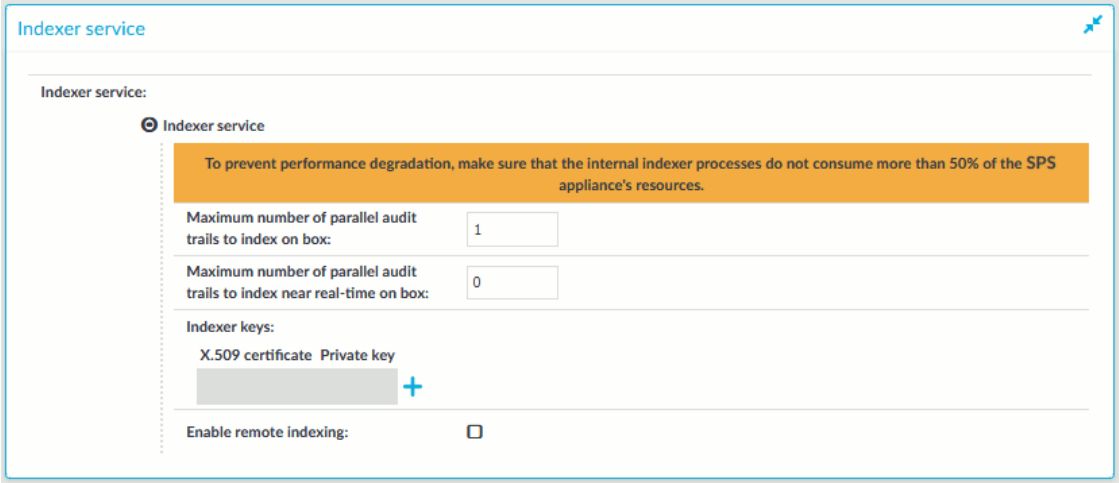
NOTE:

Using content policies significantly slows down connections (approximately 5 times slower), and can also cause performance problems when using the indexer service.

To configure SPS to index the audit trails

1. Navigate to **Basic Settings > Local Services > Indexer service**.

   Figure 205: Basic Settings > Local Services > Indexer service > Configure the Indexer service of SPS

2. Define the **Maximum number of parallel audit trails to index on box**.

   This option determines the maximum number of parallel indexing tasks that the SPS appliance performs. The default value is set to the number of detected CPU cores. Note that indexing audit trails requires about 50-100 Mbytes of memory for terminal sessions (SSH, Telnet, TN3270), and 150-300 Mbytes for graphical sessions (RDP, ICA, VNC, X11). Consider the memory usage of your SPS host before modifying this value.

3. Define the **Maximum number of parallel audit trails to index near real-time on box**.

   This option determines the maximum number of parallel indexing tasks that the SPS appliance performs near real-time, meaning that indexing starts when sessions are still ongoing. The default value is set to 0.

   NOTE:

   A connection policy configured with near real-time priority (**Connection policy > Enable indexing > Priority**) requires that you set **Maximum number of parallel audit trails to index near real-time on box** to a value other than 0.

4. (Optional) If you have encrypted audit trails and you want to index them, upload the necessary RSA keys (in PEM-encoded X.509 certificates).

NOTE:

Certificates are used as a container and delivery mechanism. For encryption and decryption, only the keys are used.

a. Click +, and then click the first icon to upload the new certificate. A pop-up window is displayed.

   Select **Browse**, select the file containing the certificate, and click **Upload**. Alternatively, you can also copy-paste the certificate into the **Certificate** field and click **Set**.

b. To upload the private key corresponding to the certificate, click the second icon. A pop-up window is displayed.

   Select **Browse**, select the file containing the private key, provide the **Password** if the key is password-protected, and click **Upload**. Alternatively, you can also copy-paste the private key into the **Key** field, provide the **Password** there, and click **Set**.

c. To add more certificate-key pairs, click + and repeat steps 3a and 3b.

TIP:

If you want to search in the trail content on the web interface: to view screenshots generated from encrypted audit trails, you also have to upload the necessary certificates to your private keystore. For more information, see Replaying encrypted audit trails in your browser on page 724.

5. Click Commit.
6. Navigate to **Policies > Indexer Policies**.
7. Two Indexer Policies are available by default, both with automatic language detection:
   - `full_indexing`: Slower, indexes the complete content of the screen, including all events.
   - `lightweight_indexing`: Significantly faster, but it extracts only the executed commands (Command event) and the window titles (Window title event) that appear on the screen. It does not index any other screen content (for example, text that is displayed in a terminal or that appears in an RDP window).

     For example, in the case of an SSH protocol, `lightweight_indexing` will index a command with parameters, such as **cat --help**, but will not index terminal printouts such as the help content itself.

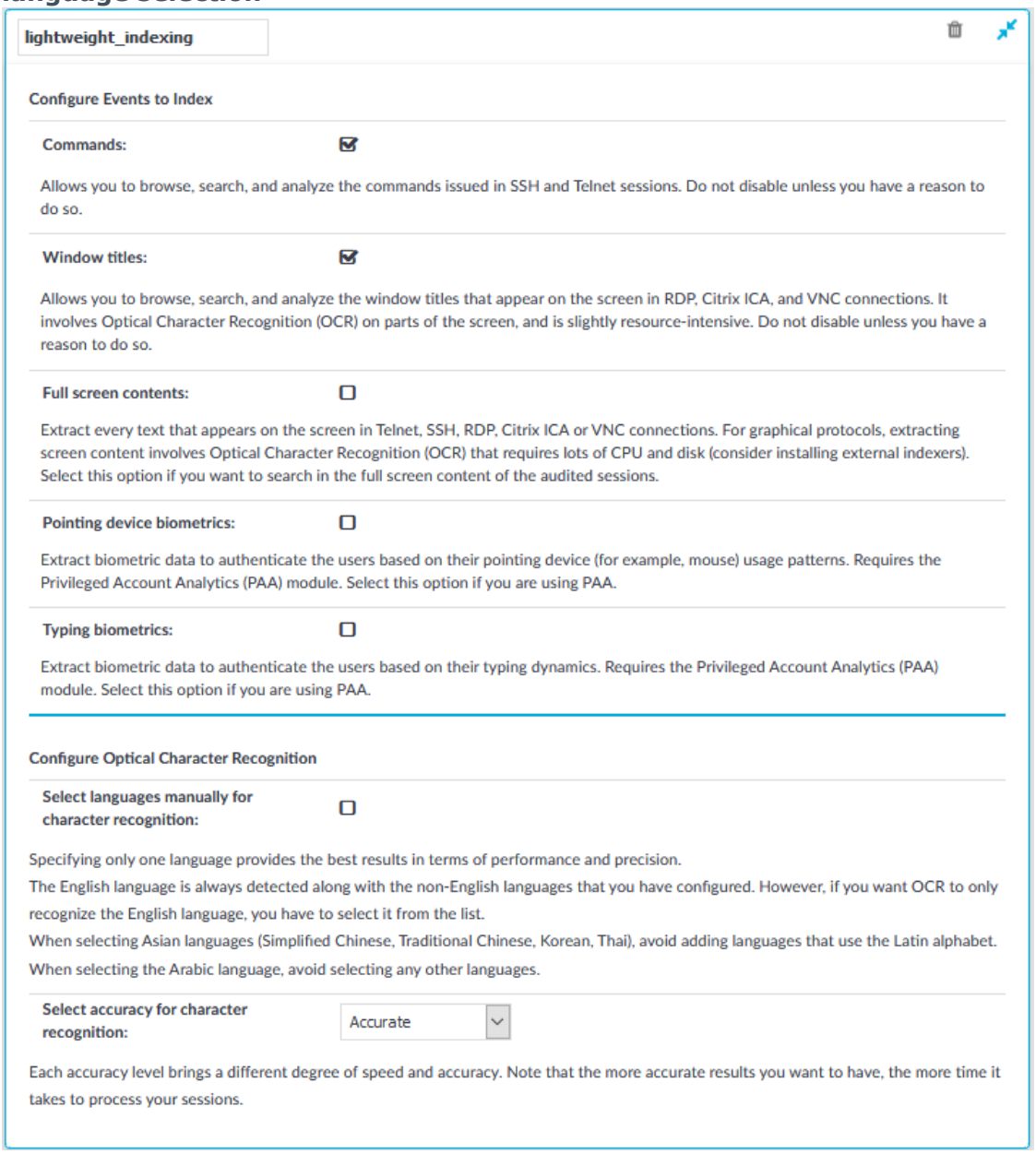
     When you add a new Connection Policy, the `lightweight_indexing` Indexer Policy is assigned to it by default.

NOTE:

In the case of graphical protocols, the default Optical Character Recognition (OCR) configuration is automatic language detection. This means that the OCR engine will attempt to detect the languages of the indexed audit trails automatically. However, if you know in advance what language(s) will be used, create a new Indexer Policy.

To create a new Indexer Policy, click +.

Figure 206: Policies > Indexer Policies > Indexing options and manual language selection



8. Select from the indexing options as follows:
    - **Commands:** Allows you to browse, search, and analyze the commands issued in SSH and Telnet sessions.

      CAUTION:

      Do not disable unless you have a reason to do so.
    - **Window titles:** Allows you to browse, search, and analyze the window titles that appear on the screen in RDP, Citrix ICA, and VNC connections. It involves Optical Character Recognition (OCR) on parts of the screen, and is slightly resource-intensive.

      CAUTION:

      Do not disable unless you have a reason to do so.
    - **Full screen contents:** Select this option if you want to search in the full screen content of the audited sessions.

      Extract every text that appears on the screen in Telnet, SSH, RDP, Citrix ICA or VNC connections. For graphical protocols, extracting screen content involves Optical Character Recognition (OCR) that requires lots of CPU and disk (consider installing external indexers).
    - **Pointing device biometrics:** Select this option only if you are using One Identity Safeguard for Privileged Analytics (SPA)).

      Extract biometric data to authenticate the users based on their pointing device (for example, mouse) usage patterns. It requires the SPA) module.
    - **Typing biometrics:** Select this option only if you are using One Identity Safeguard for Privileged Analytics (SPA)).

      Extract biometric data to authenticate the users based on their typing dynamics. It requires the SPA) module.
9. To configure what languages to detect, select **Select languages manually for character recognition**. Select the language(s) to detect. Note the following:
    - Specifying only one language provides the best results in terms of performance and precision.
    - The English language is always detected along with the non-English languages that you have configured. However, if you want the OCR to only recognize the English language, you have to select it from the list of languages.
    - There are certain limitations in the OCR engine when recognizing languages with very different character sets. For this reason, consider the following:
        - When selecting Asian languages (Simplified Chinese, Traditional Chinese, Korean), avoid adding languages that use the Latin alphabet.
        - When selecting the Arabic language, avoid selecting any other languages.

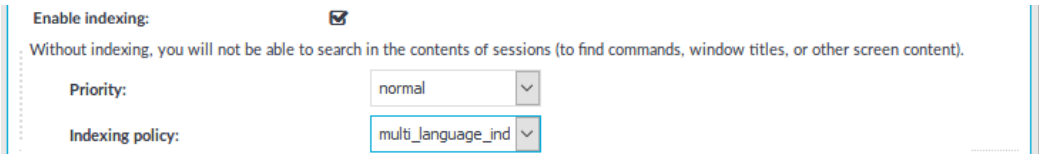
- The Thai language is currently not supported. If you are interested in using SPS to index Thai texts, contact our Sales Team.

10. Specify an accuracy level for Optical Character Recognition (OCR). Each accuracy level brings a different degree of speed and accuracy:
    - **Fast**: The fastest option with potentially less accurate results. Select this option if speed is more important to you than getting the most accurate results possible.
    - **Balanced** (default setting): Fairly accurate option with less than optimum speed. Select this option if you want results to be fairly accurate but you have more than a few sessions to process and processing time is less of a concern.
    - **Accurate**: The most accurate option with less optimal speed. Select this option if you must have the most accurate results possible and speed is less important or you only have a few sessions to process.
11. Configure the Indexing policy for the Connection policy that you want to index:

    By default, the `lightweight_indexing` Indexing policy is enabled for every Connection policy with `normal` priority. If this is ideal for you, skip this step and continue with the next step. If you want to use a different policy, for example because you want to OCR the complete screen content, or because you have created a language-specific indexer policy, complete the following substeps.

    a. Navigate to the **Control > Connections** page of the traffic type (for example **SSH Control**), and select the connection policy to index.

    b. **Figure 207: <Protocol name> Control > Connections > Enable indexing — Select Indexing Policy**

    Select the **Indexing Policy** to be used. Both built-in Indexer Policies feature automatic language detection. To specify a particular language detection configuration, select the Indexing Policy you have created before (in Step 6).

    c. To determine the priority level of indexing this connection, select the appropriate **Priority** level. Selecting a high priority level means that the trails of this connection will be indexed first. Selecting a low priority level means that the trails of this connection will be indexed also, but there might be a delay in indexing if there are a lot of high-priority connections waiting to be indexed. Selecting **near real-time** means that the indexing of sessions starts when sessions are still ongoing.

    d. Click Commit.

12. Check which channel policy is used in the connection, and navigate to the **<Protocol name> Control > Connections** page. Select the channel policy used in the

connection to index.

13. On the **<Protocol name> Control > Channel Policies** page, verify that the **Record audit trail** option is selected for the channels you want to index (for example, the `Session shell` channel in SSH, or the `Drawing` channel in RDP).
14. Click 

.

> TIP:
>
> To verify that indexing works as configured, start a session that uses this connection policy (connect from a client to a server).
>
> When the session is finished, navigate to the **Indexer > Indexer status** page to verify that the indexer service is processing the audit trail.
>
> If the audit trails are encrypted, ensure that the required decryption keys have been uploaded to **Basic Settings > Local Services > Indexer service > Indexer keys**.

Configuring external indexers

If One Identity Safeguard for Privileged Sessions (SPS) audits lots of connections, processing and indexing the created audit trails requires significant computing resources, which may not be available in the SPS appliance. To decrease the load on the SPS appliance, you can install the indexer service on external Linux hosts. These external indexer hosts run the same indexer service as the SPS appliance, and can index audit trails, or generate screenshots and replayable video files from the audit trails as needed. The external indexers register on SPS, wait for SPS to send an audit trail to process, process the audit trail, then return the processed data to SPS. The external indexer hosts do not store any data, thus any sensitive data is available on the host while it is being processed.

To use external indexers to process your audit trails, you have to complete the following steps.

- Read the conditions and limitations related to external indexers in Prerequisites and limitations on page 600.
- Install and configure the hosts (physical or virtual) that will run the external indexer service. For details on the hardware requirements, see Hardware requirements for the external indexer host on page 600.
- Configure SPS to use external indexers. For details, see Configuring One Identity Safeguard for Privileged Sessions (SPS) to use external indexers on page 601.
- Install and configure the indexer application on the external hosts. For details, see Installing the external indexerConfiguring the external indexer on page 603.
- If you enabled audit trail encrypting on SPS, you will also need to upload the necessary certificates to the external indexer to allow indexing the encrypted trails. For details, Uploading decryption keys to the external indexer on page 605.

Prerequisites and limitations

Before starting to use One Identity Safeguard for Privileged Sessions (SPS) with external indexers, consider the following:

- If there is a firewall between the host of the external indexer and SPS, enable two-way communication between them.

  The default port is TCP/12345. To change the port number, you have to modify the indexer settings on SPS, and upload the new configuration to the external indexer(s).
- To protect the sensitive data in the audit trails, ensure that the audit trails are encrypted. For details on encrypting audit trails, see Encrypting audit trails on page 465.
- Make sure to permit indexer access only to the hosts that really run external indexers on the **Basic Settings > Local Services > Indexer service** page of the SPS web interface.
- To process graphical audit trails that contain Asian characters, make sure that you have uploaded a license to SPS that enables indexing Asian characters.

  NOTE:

  The current OCR engine cannot guarantee accurate character recognition for Asian characters smaller than 30 x 30 pixels. If you encounter problems with character recognition for Asian characters, increase resolution settings in your connection.
- The external indexer can be installed on the following 64-bit operating systems: Red Hat Enterprise Linux Server 6.7, Red Hat Enterprise Linux Server 7, CentOS 6.7, and CentOS 7. The installer is a self-contained package that includes every required dependency of the indexer.

If your security policy does not permit the above limitations, or your environment does not make it possible to fulfill them, do not use external indexers with SPS.

Hardware requirements for the external indexer host

- CPU: You can configure the number of audit trails that an indexer host processes at the same time. For optimal performance, each indexer process should have a dedicated CPU core.
- Memory requirements: In addition to the memory requirements of the operating system of the host, the indexer requires about 300 MB memory for each worker process, depending on the protocol of the indexed audit trails. The audit trails of terminal connections require less memory.

- Disk: The indexer requests the data from One Identity Safeguard for Privileged Sessions (SPS) in small chunks, it does not store the entire audit trail nor any temporary files. You will need only disk space for the operating system, and a few GB to store logs.

For example, if you want to have a host that can process 6 audit trails at the same time, you need 6 CPU cores and 1.8 GB of memory for the indexer service. If you install only a minimal operating system and the external indexer on the host, 6 GB disk space should be enough.

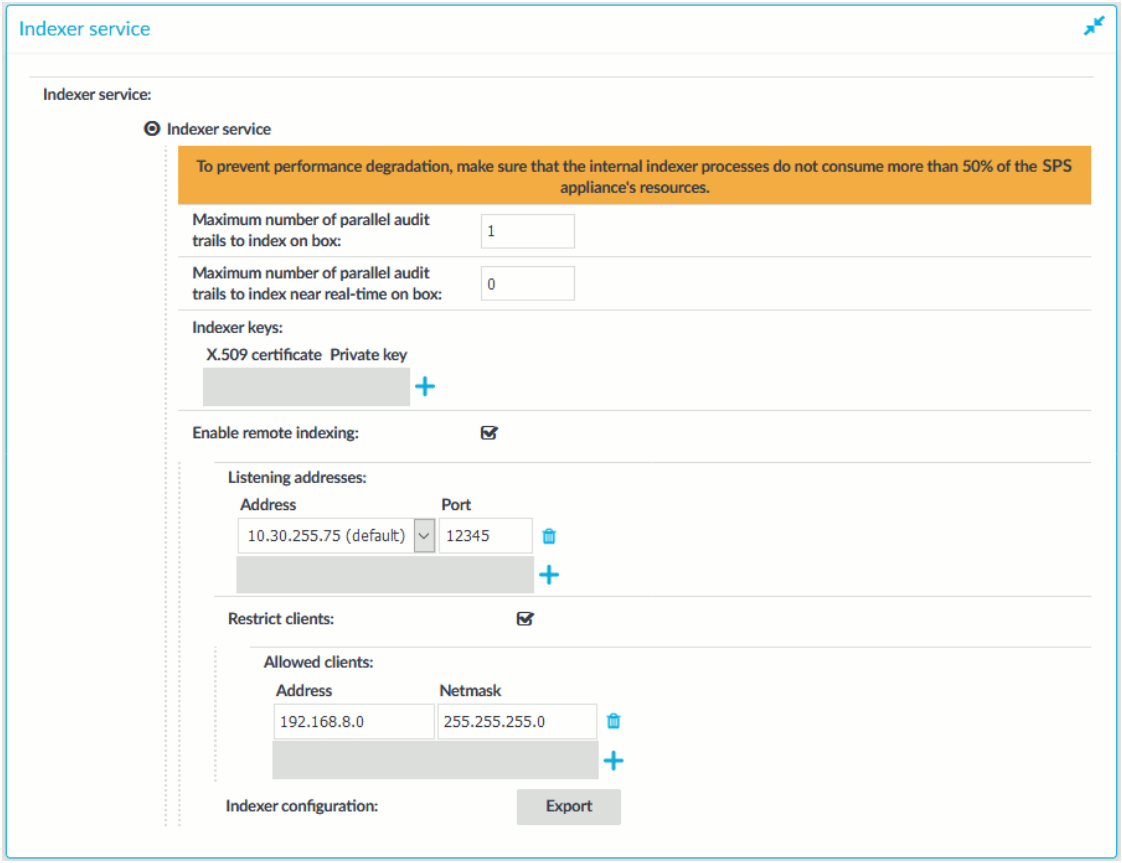
Configuring One Identity Safeguard for Privileged Sessions (SPS) to use external indexers

The following describes how to configure One Identity Safeguard for Privileged Sessions (SPS) to accept connections from external indexer services.

To configure SPS to accept connections from external indexer services

1. Log in to the SPS web interface, and navigate to **Basic Settings > Local Services > Indexer service**.
2. Select **Indexer service**.
3. Select **Enable remote indexing**.

Figure 208: Basic Settings > Local Services > Indexer service > Enable remote indexing — Configure external indexers



4. In the **Listening addresses > Address** field, select the network interface where SPS should accept external indexer connections. Repeat this step to add other interfaces if needed.

   The available addresses correspond to the interface addresses configured in **Basic Settings > Network > Interfaces**. Only IPv4 addresses can be selected.
5. Select **Restrict clients**, and list the IP address and netmask of your external indexer hosts.

   Use an IPv4 address.
6. Click **Commit**.

Installing the external indexer

Prerequisites

The external indexer can be installed on the following 64-bit operating systems: Red Hat Enterprise Linux Server 6.7, Red Hat Enterprise Linux Server 7, CentOS 6.7, and CentOS 7.

The installer is a self-contained package that includes every required dependency of the indexer.

To install the external indexer

1. Log in as root to the host that you want to use to index your audit trails.
2. Copy the installer package to the host.

   NOTE: Due to legal reasons, installation packages of the external indexer application will be available only from the SPS web interface. After SPS versions 6.4 and 6.0.3 are released, the installation packages will be removed from our website.

3. Install the package using the package manager of the operating system. For example:

   ```
   yum install external-indexer-standalone-<version-number>.x86_64.rpm
   ```

   The installer performs the following steps automatically.

   - Unpacks the indexer files into the `/opt/external-indexer/` directory.
   - Installs the related init scripts (the `/etc/init.d/external-indexer` init script, and adds the init script configuration to `/etc/sysconfig/external-indexer`).
   - Creates the `indexer` user and usergroup. This is an unprivileged user that is used to run the indexer application.
   - Registers the service to start automatically on system boot. Note that the indexer init script uses bind mount points.

4. Edit the firewall rules of the host to accept connections from One Identity Safeguard for Privileged Sessions (SPS) on the TCP/12345 port. For example:

   ```
   iptables -A INPUT -p tcp --destination-port 12345 --jump ACCEPT
   ```

   NOTE:

   If there is a firewall between the indexer host and SPS, enable two-way communication between them on the TCP/12345 port.

5. Configure the indexer. For details, see Configuring the external indexer on page 603.

Configuring the external indexer

In order to connect to One Identity Safeguard for Privileged Sessions (SPS) and index the audit trails, you must configure the external indexer.

CAUTION:

Unless you know exactly what you are doing, modify only the parameters you are instructed to.

To configure the external indexer

1. Log in to the SPS web interface, and navigate to **Basic Settings > Local Services > Indexer service**.
2. Export the configuration file for the external indexer: click **Export**. (Note that the **Export** button is displayed only after the configuration to enable SPS to use remote indexers has been committed.)

   The configuration file contains the listening address (IP and port) of SPS, the OCR license, and the necessary keys for SSL authentication.

   Upload the file to the host of the external indexer.
3. On the external indexer host, import the configuration file with the following command:

   ```
   indexer-box-config <configuration-file>.config
   ```
4. Configure the external indexer service: open the `/etc/indexer/indexerworker.cfg` configuration file for editing.
5. To edit the number of worker groups assigned to a certain worker process type, find the `worker_groups` line.

   A worker group has the following parameters:

   - `name`: the name of the worker group
   - `count`: the number of workers to use for processing
   - `capabilities`: the type of job(s) this process will perform (`index`, `screenshot`, `video`, `near-realtime`)

     NOTE:

     Setting the `near-realtime` capability exclusively determines whether active or closed sessions will be indexed. Take the following examples:

     - When setting `[near-realtime, index]` capabilities for a worker, that worker will only index active, ongoing sessions.
     - When setting `[index, screenshot, video]` capabilities for a worker, that worker will only index closed sessions.

     NOTE:

     A connection policy configured with near real-time priority (**Connection policy > Enable indexing > Priority**) requires that you also configure indexer workers that are capable of near real-time indexing. To configure such indexer workers, set the `near-realtime` capability for the relevant workers.

> **NOTE:**
>
> Indexer workers with the `near-realtime` capability require fewer CPU cores but more memory than indexer workers that do not have this capability.

Make sure that the sum value of the workers are equal to the number of CPU cores in the host (or the number of CPU cores minus one if you want to save resources for other tasks).

One Identity recommends using dedicated hosts for external indexing. If the host is not dedicated exclusively to the external indexer, decrease the number of workers accordingly.

6. (Optional) To fine-tune performance, you can configure the number of OCR threads each worker can initiate using the `ocr_thread_count` key.

   The default setting is `3`. When configuring this setting, pay attention to the available CPU cores, as raising the number of possible threads too high can impact performance negatively.

7. (Optional) If instructed by One Identity Support, configure the OCR engine.

   Find the `engine` key, and change its value to one of the following options:

   - `omnipage-external` is the default setting. It provides the best performance and stability by allowing workers to initiate multiple OCR threads.

     This setting also allows you to search for images where OCR could not be performed. On the search UI of SPS enter the `OOCCRRCCRRAASSHH` search string to list all such images. If possible, contact our Support Team, so we can continue improving the engine.

     Note that multiple OCR threads can only speed up processing graphical protocols (RDP, VNC and ICA trails), and do not affect the processing speed for terminal-based protocols (telnet and SSH).

   - `omnipage` only supports one OCR thread per worker.

     If you have to use this option, make sure to also set the `ocr_thread_count` to `0`.

8. Save your changes. To continue with uploading decryption keys (for indexing encrypted audit trails), see Uploading decryption keys to the external indexer on page 605.

   To start the indexer service, see Starting the external indexer on page 614.

Uploading decryption keys to the external indexer

If the audit trails you want to index are encrypted, complete the following steps to make the decryption keys available for the indexer.

To make the decryption keys available for the external indexer

1. Obtain the RSA private keys and the matching x.509 certificates in PKCS-1 PEM format, and copy them to the external indexer's host. Other certificate formats are not supported.
2. Use the **indexer-keys-json** utility to transform the certificate and the private key to the required JSON format. When executed, the script asks for the path to the certificate and the private key, and the password of the private key. After the conversion, the password is removed.

   The utility automatically adds the certificate and the private key to the `/etc/indexer/indexer-certs.cfg` keystore file. If you want to use a different keystore file, use the `--keystore` argument to specify another file. If the keystore already includes the certificate and the private key you want to add, they will be ignored.

   a. In the `/opt/external-indexer/usr/bin/` folder, issue the following command: **indexer-keys-json**

   b. Enter the absolute path to the X.509 certificate. Alternatively, you can include this information as a parameter: **indexer-keys-json --cert <path-to-certificate>**

   c. Provide the absolute path to the corresponding private key. Alternatively, you can include this information as a parameter: **indexer-keys-json --private-key <path-to-private-key>**

   d. If the key is password protected, enter the password to the private key.

   e. To add additional certificates, re-run the **indexer-keys-json** command.
3. You can now start the indexer service. For more information, see Starting the external indexer on page 614.

Configuring a hardware security module (HSM) or smart card to integrate with external indexer

It is possible to use a hardware security module (HSM) or a smart card to store the decryption keys required for decrypting audit trails. An HSM or a smart card is a tamper-resistant physical, software, or cloud solution that can securely store digital keys used for authentication.

The main steps of configuring a hardware security module (HSM) or smart card to integrate with an external indexer are as follows:

1. Set up and test the environment.
2. Encrypt the PKCS#11 PIN.

To see examples of how to configure various HSM or smart card solutions that you wish to integrate with your external indexer(s), consult the following sections:

- Configuring SoftHSM on page 609
- Configuring AWS CloudHSM on page 611
- Configuring a smart card on page 612

Setting up and testing the environment

To access an HSM or smart card with the external indexer, a PKCS#11 shared library plugin must be used. In most cases, these libraries also need a background daemon or environment variables set. The PKCS#11 library must be accessible to the external indexer with a proper environment.

To set up the environment and test it, complete the following steps.

1. Load the environment for the indexer commands:

   ```
   source /etc/indexer/external-indexer.env
   ```

2. Test your environment.
   - *Option #1: Use the pkcs11-tool to test your environment:*
     1. List the available slots.

        ```
        pkcs11-tool --modul <path-to-pkcs11-library> -L
        ```

     2. List the objects in a slot.

        ```
        pkcs11-tool --modul <path-to-pkcs11-library> -l --slot <id> -O
        ```

   - *Option #2: Use the indexerworker with the log level set to dump to see the available keys:*

     ```
     indexerworker -l -v 7 --pkcs11-lib <path-to-pkcs11-library> --pkcs11-slot-id <id> --pkcs11-pin <pin>
     ```

3. Assuming that the environment is ready, the external indexer must be configured to use the PKCS#11 library. To do so, edit `/etc/indexer/indexerworker.cfg` as follows:

   ```
   ...
   "settings": {
     "pkcs11": {
           "custom_password": false
           "slots": [
             {
               "library": "<path-to-pkcs11-library>",
               "slot_id": <slot-number>,
               "pin": "<your-encrypted-PIN>"
   ```

```
          }
        ]
   }
}
...
```

Encrypting a PKCS#11 PIN

The PKCS#11 PIN(s) must be protected by additional encryption. The indexerconfigcrypter tool must be used to encrypt the PIN(s).

To encrypt the PIN(s)

1. Encrypt the PIN.

   The PINs can be encrypted with a custom passphrase or a default one is used if no custom passphrase is provided. A custom passphrase is more secure, but interaction is needed to start or restart the external-indexer service. Using a custom passphrase is supported on hosts running CentOS 7 or later.

   Issue either of the following commands:

   - Using a default password (CentOS 6 or 7): **indexerconfigcrypter --input <your-PIN>**
   - Using a custom password (CentOS 7 or later): **indexerconfigcrypter --input <your-PIN> --password**

   It is possible to configure multiple slots. In that case, the PINs must be encrypted using the same passphrase.

2. Update the `"pkcs11"` object in the `indexerworker.cfg` file.

   The encrypted PINs must be stored in the `"pin"` field of the configuration file (in the example, a SoftHSM is used):

```
...
     "pkcs11": {
           "custom_password": true
           "slots": [
              {
                 "library": "/usr/lib/softhsm/libsofthsm.so",
                 "slot_id": 0,
                 "pin": "<your-encrypted-PIN>"
              }
           ]
     }
 ...
```

Starting and restarting the external-indexer service when using a custom password for PKCS#11 PIN encryption

When you choose to encrypt the PKCS#11 PIN(s) using a custom password, on starting or restarting the external-indexer service, you are asked to enter your password using a special tool.

To provide your password using the required tool

1. Start the external-indexer service:

   ```
   systemctl start external-indexer
   ```

2. The external-indexer service prompts you to provide a password using the systemd-ask-password tool. Issue:

   ```
   systemd-tty-ask-password-agent
   ```

3. Provide the password at the prompt. You can use multiple agents to enter the password.
4. Once the external indexer(s) have been started or restarted, make sure that all the indexers have started up successfully.

   For example, on CentOS 7, you can use:

   ```
   systemctl status external-indexer
   ```

Configuring SoftHSM

SoftHSM is the software implementation of an HSM. It can be installed from the EPEL repository. The configuration of SoftHSM can be found at `/etc/softhsm2.conf` (CentOS 7), or `/etc/softhsm.conf` (CentOS 6).

The following describes how to configure SoftHSM.

NOTE:

Depending on the exact SoftHSM solution that you are using, the steps described here may slightly differ.

NOTE:

The following steps assume that:

- You are on the host operating system.
- The external indexer has been installed.

Prerequisites

The indexer user/group has the rights to read the data directory of SoftHSM and its contents, which defaults to `/var/lib/softhsm`.

To configure SoftHSM

1. Initialize directories for SoftHSM.

   ```
   mkdir -p /var/lib/softhsm
   chgrp -R indexer /var/lib/softhsm
   ```

2. Configure slots for softhsm1 (CentOS 6). For softhsm2 (CentOS 7), you can skip this step.

   ```
   cat /etc/softhsm.conf
   0:/var/lib/softhsm/slot0.db
   1:/var/lib/softhsm/slot1.db
   ```

3. Initialize slot 0 (softhsm1).

   ```
   softhsm --init-token --slot 0 --label "<your-slot-label>" --<so-pin>
   topsecret --pin <your-SoftHSM-PIN>
   ```

4. Initialize a new slot (softhsm2) and get the slot ID:

   ```
   softhsm2-util --init-token --free --label "<your-slot-label>" --<so-pin>
   topsecret --pin <your-SoftHSM-PIN>
   SLOT_ID=$(softhsm2-util --show-slots | grep -B 15 "<your-slot-label>" |
   grep "Slot [0-9]" | head -n 1 | cut -d ' ' -f 2)
   ```

5. Import your keys. Your keys must be in the `.der` format.

   For softhsm1, use:

   ```
   pkcs11-tool --module /usr/lib/softhsm/libsofthsm.so -l -y privkey --slot 0
   -w key.der -d 001 -a <your-key-label> --pin <your-SoftHSM-PIN>
   ```

   For softhsm2, use:

   ```
   pkcs11-tool --module /usr/lib/softhsm/libsofthsm2.so -l -y privkey --slot
   0 -w key.der -d 001 -a <your-key-label> --pin <your-SoftHSM-PIN>
   ```

6. Make sure that the indexer user/group has execute right to the token directory and read right to the token files below the `/var/lib/softhsm/tokens/` directory.
7. Test your SoftHSM configuration with the indexer.

```
source /etc/indexer/external-indexer.env
indexerworker -l -v 7 --pkcs11-lib "<your-SoftHSM-library>" --pkcs11-slot-
id 0 --pkcs11-pin "<your-SoftHSM-PIN>"
```

8. Encrypt the PKCS#11 PIN(s). For detailed instructions, see Encrypting a PKCS#11 PIN on page 608.
9. Update the `"pkcs11"` object in the `/etc/indexer/indexerworker.cfg` file.

```
...
     "pkcs11": {
           "slots": [
              {
                 "library": "/usr/lib/softhsm/libsofthsm.so",
                 "slot_id": 0,
                 "pin": "<your-encrypted-PIN>"
              }
           ]
     }
...
```

Configuring AWS CloudHSM

Amazon Web Services (AWS) CloudHSM provides hardware security modules in the AWS Cloud.

The following describes how to configure CloudHSM.

NOTE:

The following steps assume that:

- You have set up your AWS CloudHSM, that is, you have created a user for the indexer, imported/generated keys, and so on.

  For detailed information on AWS CloudHSM, see the AWS CloudHSM User Guide.
- The CloudHSM PKCS#11 library is installed.
- The external indexer has been installed.

To configure CloudHSM

1. Test your environment as described in Setting up and testing the environment on page 607.

   Note that you will need to provide your CloudHSM PIN in the following format:

```
"<your-CloudHSM-username:your-CloudHSM-PIN>"
```

2. Encrypt the PKCS#11 PIN(s). For detailed instructions, see Encrypting a PKCS#11 PIN on page 608.
3. Update the "pkcs11" object in the /etc/indexer/indexerworker.cfg file.

```
...
     "pkcs11": {
           "custom_password": true
           "slots": [
              {
                 "library": "/opt/cloudhsm/lib/libcloudhsm_pkcs11.so",
                 "slot_id": 1,
                 "pin": "<your-encrypted-PIN>"
              }
           ]
     }
...
```

Configuring a smart card

NOTE:

Using the external indexer with a smart card is currently an experimental feature only.

To configure a smart card

1. Install OpenSC, for example, from the EPEL repository of CentOS.
2. Ensure that the PC/SC Smart Card Daemon (pcscd) service is running:
   - On CentOS 6:

     ```
     service pcscd start
     ```

   - On CentOS 7:

     ```
     systemctl enable pcscd
     systemctl start pcscd
     ```

     Alternatively, you can use:

     ```
     systemctl enable pcscd.socket
     systemctl start pcscd.socket
     ```

     This ensures that the pcscd service will not start at system startup, it will only start when there is an attempt (for example, by the indexerworker) to connect to it.

3. Test your environment as described in Setting up and testing the environment on page 607.
4. Encrypt the PKCS#11 PIN(s). For detailed instructions, see the Encrypting a PKCS#11 PIN on page 608.
5. Update the "pkcs11" object in the /etc/indexer/indexerworker.cfg file, for example:

```
...
     "pkcs11": {
           "slots": [
              {
                 "library": "/usr/lib64/pkcs11/opensc-pkcs11.so",
                 "slot_id": 1,
                 "pin": "encrypted_pin"
              }
           ]
     }
...
```

Customizing the indexing of HTTP traffic

Use this section to customize how One Identity Safeguard for Privileged Sessions (SPS) indexes HTTP traffic.

Prerequisites

You can customize only the configuration of external indexers. The indexer running on the SPS host always uses the default HTTP configuration, which is the following:

```
{
    "General": {
       "Whitelist": ["text/.*", ".*json.*", "application/x-www-form-
urlencoded", "multipart/.*"],
       "Blacklist": ["text/css", "application/javascript", "text/xslt",
".*xml.*"]
    },
    "Form": {
        "Blacklist": ["password", "pass"]
    },
    "Html": {
       "Attributes": ["href", "name", "value", "title", "id", "src"],
       "StrippedTags": ["script", "object", "style", "noscript", "embed",
"video", "audio", "canvas", "svg"]
    }
}
```

To customize how SPS indexes HTTP traffic

1. Create a configuration file for the HTTP indexer using a text editor. The configuration file uses the JSON format. For details on the configuration format, see HTTP indexer configuration format on page 618.

   NOTE:

   If you want to index HTTP POST messages, include the "`application/x-www-form-urlencoded`" Content-Type in the `General > WhiteList` list. The indexer will decode URL encoding (percentage encoding), and create `key=value` pairs from the form fields and their values. Note that in the values, the indexer will replace whitespace with the underscore (_) character. To avoid indexing sensitive information (for example, passwords from login forms), use the `Form > Blacklist` option.

2. Copy the configuration file to the external hosts, to the `/opt/external-indexer/usr/share/adp/httpconfig.json` file.
3. Reload the indexer service: **systemctl restart external-indexer.service**
4. Repeat the above steps for your other external indexer hosts. Otherwise, it is possible that certain audit trails will be indexed using different indexer configuration.
5. Disable the indexer that is running on the SPS host. Otherwise, it is possible that certain audit trails will be indexed using different indexer configuration.

   Navigate to **Basic Settings > Local Services > Indexer service**, and set the **Maximum parallel audit trails to index on box** option to `0`.

Starting the external indexer

When you have configured the external indexer, and added all decryption keys, you can start running the service.

To start the external indexer

1. Start the indexer service using the following command.
   - On Red Hat or CentOS 6.5:

     ```
     service external-indexer start
     ```

   - On Red Hat or CentOS 7:

     ```
     systemctl start external-indexer.service
     ```

2. Verify that the indexer service is running. Execute the **ps aux** command. In the output, you should see a `workercontroller` and one or more `indexerworker` processes. The number of the `indexerworker` processes should be the same number you set for the `number_of_workers` key of the `/etc/indexer/indexerworker.cfg` file.

3. Verify the `indexer-certs.cfg` configuration file.

   Check the system logs of the host of the external indexer. The "Error loading key store" log indicates that there was a problem with the `indexer-certs.cfg` configuration file.
4. Verify that the indexer host is displayed in the list of indexers on the **Indexer > Indexer status** page of the One Identity Safeguard for Privileged Sessions (SPS) web interface.

Disabling indexing on One Identity Safeguard for Privileged Sessions (SPS)

To reduce load on One Identity Safeguard for Privileged Sessions (SPS), you can disable indexing audit trails on the box. Note that this introduces delays when generating on-demand screenshots for audit trail searches.

Prerequisites

Disabling indexing on the SPS box works only if an external indexer is available. If SPS cannot detect the presence of an external indexer (for example, because of a network outage), indexing is re-enabled on SPS automatically with one indexing process.

To disable indexing on SPS

1. On the SPS web interface, navigate to **Basic Settings > Local Services > Indexer service**.
2. Set the **Maximum parallel audit trails to index on box** to `0`.
3. Click 

.

Managing the indexers

The indexers that run on an external host send log messages into the standard syslog of the external host. These log messages are not visible on One Identity Safeguard for Privileged Sessions (SPS).

The indexers use the standard `init.d` framework of the host. You can restart the indexer processes using the **/etc/init.d/indexerworker restart** command, and the entire indexer service using the **/etc/init.d/external-indexer restart** command. Note that restarting the indexer service automatically restarts the worker processes as well.

The hosts that are running indexers should be visible in the list of indexers on the **Indexer > Indexer status** page of the SPS web interface.

Upgrading the external indexer

The following describes how to upgrade the indexer application on your external indexer hosts.

CAUTION:

One Identity Safeguard for Privileged Sessions (SPS) 5 F4 and later versions use a new encryption algorithm to encrypt the recorded audit trails (AES128-GCM). This change has the following effects:

- **If you are using external indexers to index your audit trails, you must upgrade them to the latest version. Earlier versions will not be able to index encrypted audit trails recorded with SPS 5 F4 and later.**
- **To replay an encrypted audit trail recorded with SPS 5 F4 or later, you can use the latest version of the Safeguard Desktop Player application, or the browser-based player of SPS. You cannot replay such audit trails using earlier versions of Safeguard Desktop Player, nor any version of the Audit Player application.**

Prerequisites

Before you start, create a backup copy of the `/etc/indexer/indexerworker.cfg` and `/etc/indexer/indexer-certs.cfg` indexer configuration files.

To upgrade the indexer application on your external indexer hosts

1. Download the latest indexer `.rpm` package from the **Basic Settings > Local Services > Indexer service** page of the SPS web interface.

   NOTE: Due to legal reasons, installation packages of the external indexer application will be available only from the SPS web interface. After SPS versions 6.4 and 6.0.3 are released, the installation packages will be removed from our website.

2. Copy the downloaded `.rpm` package to your external indexer hosts.
3. Stop the indexer by using the following command.
   - On Red Hat or CentOS 6.5:

     ```
     service external-indexer stop
     ```

   - On Red Hat or CentOS 7:

     ```
     systemctl stop external-indexer.service
     ```

4. Execute the following command: **yum upgrade -y indexer.rpm**
5. Resolve any warnings displayed during the upgrade process.
6. Restart the indexer by using the following command.

- On Red Hat or CentOS 6.5:

  ```
  service external-indexer start
  ```

- On Red Hat or CentOS 7:

  ```
  systemctl start external-indexer.service
  ```

7. Repeat this procedure on every indexer host.

Troubleshooting external indexers

The indexers that run on an external host send log messages into the standard syslog of the external host. These log messages are not visible on One Identity Safeguard for Privileged Sessions (SPS). If a problem occurs, check the logs of SPS and the external indexer to find out which component on which host causes the problem. If the problem is on the external indexer host, verify that the required decryption keys are available on the host, then restart the indexer service using the following command.

- On Red Hat or CentOS 6.5:

  ```
  service external-indexer restart
  ```

- On Red Hat or CentOS 7:

  ```
  systemctl restart external-indexer.service
  ```

If the problem persists, contact our Support Team. You can increase the log level of the indexer processes from the configuration file.

Monitoring the status of the indexer services

The status of audit-trail processing is displayed on the **Indexer** page of the **Main Menu**.

Elements of the Indexer page

The following list describes the elements of the **Indexer** page and their functions.

- **Worker status**: displays various data about the worker groups.
  - **Indexer IP address**: displays the IP address of the indexer (either the indexer running on One Identity Safeguard for Privileged Sessions (SPS) or an external indexer). It may display the following:

- **127.0.0.1**: indicated the indexer running on SPS.
- An IP address other than**127.0.0.1**: indicates an external indexer.

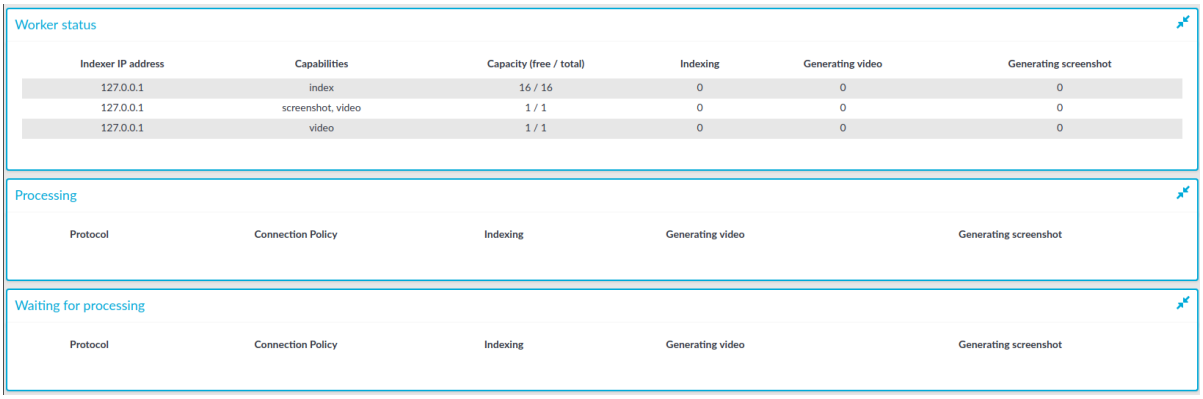
- **Capabilities**: the type of job(s) this worker will perform.
- **Capacity**: the available and total **Capacity** (Maximum parallel audit trails to process) of the indexer, and also the number of active processes that are **Indexing** an audit trail, or **Generating video** or **Generating screenshot**.

- **Processing**: audit trails that are currently being processed per connection policy.
- **Waiting for processing**: audit trails waiting to be processed.

  When you see audit trails in the **Indexing** column, that could mean any of the following:

  - The maximal queue size is 1000. If there are many trails waiting to be indexed, SPS will keep lots of trails in the queue.
  - The worker with the appropriate key for decryption is not available at the moment, and other workers do not have the required key.
  - There are no workers available that have the required capability.

Figure 209: Indexer > Indexer status — Monitoring the status of the indexers



To automatically refresh the **Indexer Status** page every 5 seconds, select **Auto refresh**. To refresh the page immediately, click **Refresh now**.

HTTP indexer configuration format

This section describes the configuration format and options of the HTTP indexer (that is, how and which fields of the HTTP audit trails are indexed). For details on how to customize HTTP indexing, see Customizing the indexing of HTTP traffic on page 613.

NOTE:

If you want to index HTTP POST messages, include the "application/x-www-form-urlencoded" Content-Type in the General > WhiteList list. The indexer will decode URL encoding (percentage encoding), and create key=value pairs from the form fields and their values. Note that in the values, the indexer will replace whitespace with the underscore (_) character. To avoid indexing sensitive information (for example, passwords from login forms), use the Form > Blacklist option.

HTTP indexer configuration options

General

Type: Top level item

Description: Determines which HTTP Content-Types are indexed. An HTTP message is indexed only if its Content-Type is listed in Whitelist and is not listed in Blacklist.

For example:

```
"General": {
                  "Whitelist": ["text/.*", ".*json.*", "multipart/.*",
"application/x-www-form-urlencoded"],
                  "Blacklist": ["text/css", "application/javascript",
"text/xslt", ".*xml.*"]
},
```

General (Whitelist)

Type: list

Description: The list of HTTP Content-Types to index. Every entry of the list is treated as a regular expression.

For example:

```
"Whitelist": ["text/.*", ".*json.*", "multipart/.*", "application/x-www-form-urlencoded"],
```

General (Blacklist)

Type: list

Description: The list of HTTP Content-Types that are not indexed. Every entry of the list is treated as a regular expression.

For example:

```
"Blacklist": ["text/css", "application/javascript", "text/xslt", ".*xml.*"]
```

Form

Type: Top level item

Description: Determines which fields are indexed in HTTP POST messages.

For example:

```
"Form": {
            "Blacklist": ["password", "pass"]
},
```

NOTE:

If you want to index HTTP POST messages, include the `"application/x-www-form-urlencoded"` Content-Type in the `General > WhiteList` list. The indexer will decode URL encoding (percentage encoding), and create `key=value` pairs from the form fields and their values. Note that in the values, the indexer will replace whitespace with the underscore (_) character. To avoid indexing sensitive information (for example, passwords from login forms), use the `Form > Blacklist` option.

Form (Blacklist)

Type: list

Description: The list of fields that are not indexed in HTTP POST messages (for example, when submitting forms, such as login forms). Every entry of the list is treated as a regular expression.

For example:

```
"Blacklist": ["password", "pass"]
```

Html

Type: Top level item

Description: Include this section in the configuration to process `text/html` messages. HTML tags are stripped from the text, and only their content is indexed (for example, `<html><title>Title</title></html>` becomes `Title`).

For example:

```
"Html": {
            "Attributes": ["href", "name", "value", "title", "id", "src"],
            "StrippedTags": ["script", "object", "style", "noscript",
"embed", "video", "audio", "canvas", "svg"]
}
```

Html (Attributes)

Type: list

Description: The list of HTML attributes that extracted as `key=value` pairs and indexed. Note that in the values, the indexer will replace whitespace with the underscore (_) character, and decode URL encoding. For example:

```
"Attributes": ["href", "name", "value", "title", "id", "src"],
```

Note that for the `content` attribute of the `meta name="description"`, `meta name="keywords"`, `meta name="author"` and `meta name="application-name"` is always indexed.

For example, if an audit trail contains the following HTML:

```
<head>
      <meta name="description" content="Web page description">
      <meta name="keywords" content="HTML,CSS,XML,JavaScript">
      <meta name="author" content="OI SA">
      <meta charset="UTF-8">
</head>
```

Then the index will contain the following text:

```
description=Web_page_description keywords=HTML,CSS,XML,JavaScript author=OI_SA
```

Html (StrippedTags)

Type: list

Description: The list of HTML tags that are not indexed.

For example:

```
"StrippedTags": ["script", "object", "style", "noscript", "embed", "video",
"audio", "canvas", "svg"]
```

Using the Search interface

This section provides an overview on how to use the Search interface. It describes how you can access the Search interface, lists the steps to take to search effectively, view the details of a connection, replay the audit trails, or export the search results as a comma-separated text file.

Prerequisites

Users need the **Search** privilege to access the Search interface.

NOTE:

Assigning the **Search** privilege to a user on the **AAA > Access Control** page, automatically enables the **Search in all connections** privilege, and grants the user access to every audit trail, even if the user is not a member of the groups listed in the **Access Control** option of the particular connection policy.

If you want users to access audit trails only for connections for which they are granted permission, see Assigning search privileges.

For information on configuring:

- Authorizers for a connection, see Configuring four-eyes authorization on page 752.
- User rights, see Managing user rights and usergroups on page 320.

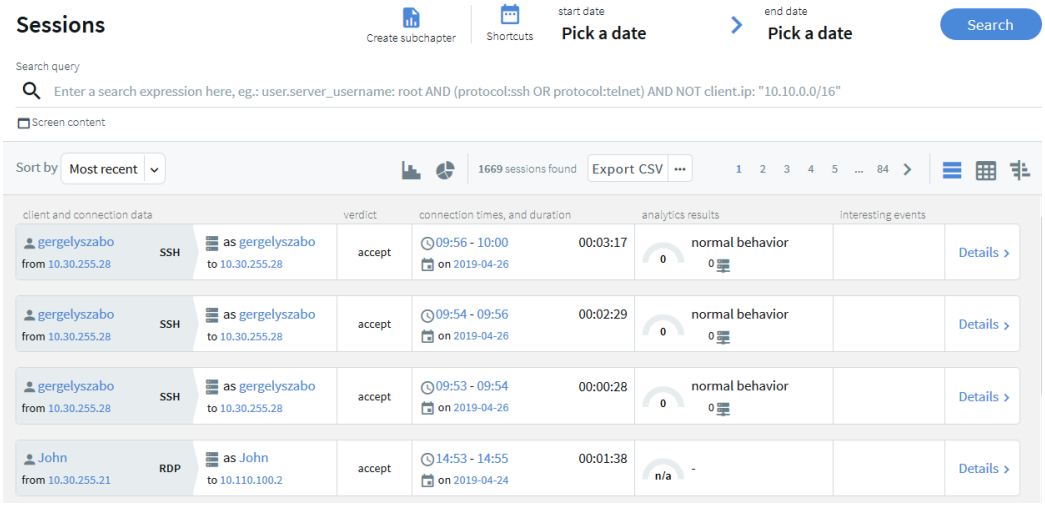
1. To access the Search interface, navigate to **Search**.

   You can view sessions in a card, table or flow view.

   Click:

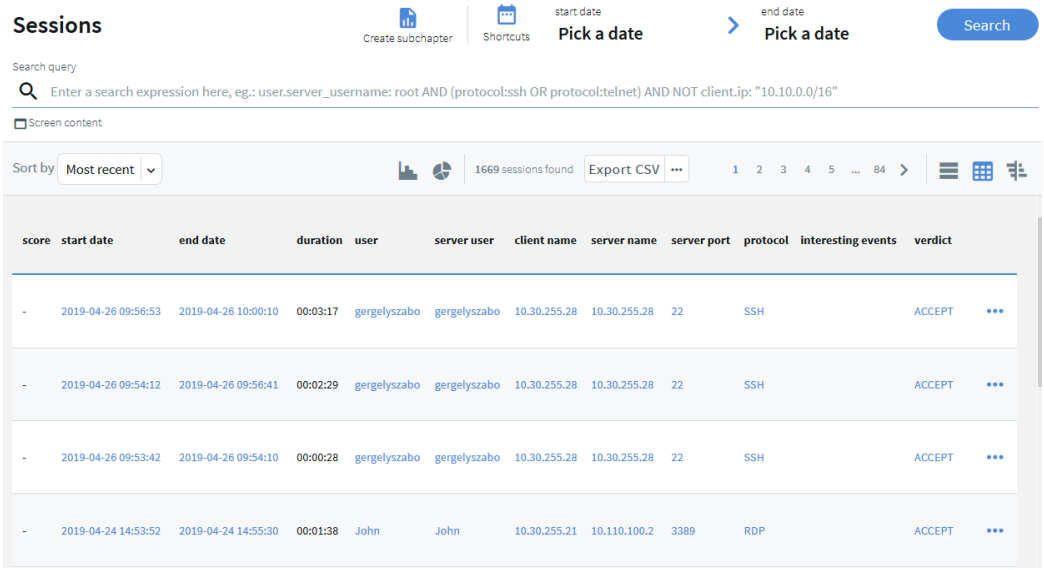
-  for the card view.

Figure 210: Search — Card view



-  for the table view.

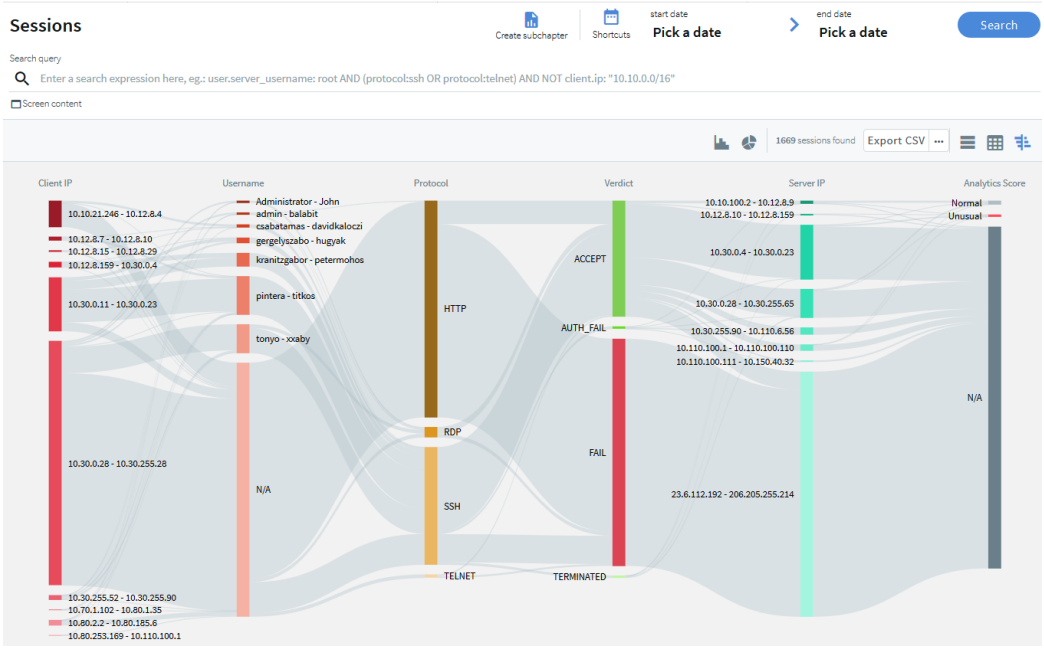
Figure 211: Search — Table view



Sessions are displayed sorted by date. For ongoing sessions, the Search interface is updated in real-time to always show the most up-to-date information.

-  for the flow view.

Figure 212: Search — Flow view



The flow view allows you to:

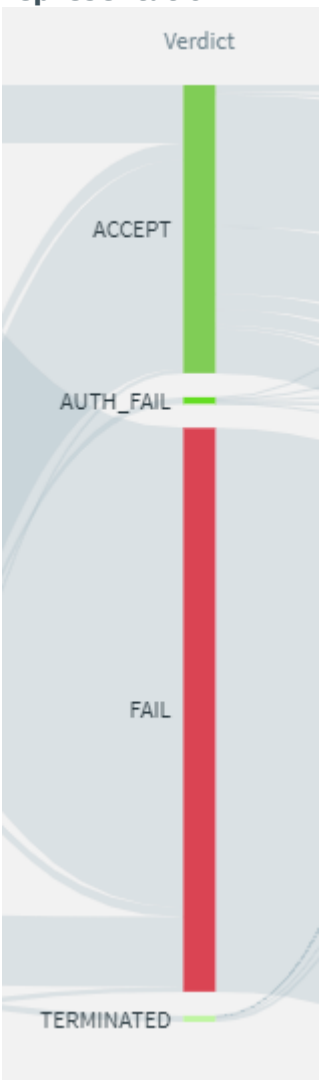
- Quickly visualize the distribution of the sessions based on their various metadata, such as, client address, username, protocol, verdict, server address, and One Identity Safeguard for Privileged Analytics (SPA) score.

  The metadata of the sessions are presented as vertical bars and each bar represents the proportional value of the data.

  Example: Proportional data representation

  The **Verdict** column shows that most of the sessions failed, a large number were accepted, and the rest of the sessions fall into the category of **AUTH_FAIL**, and **TERMINATED**.

Figure 213: Search > Flow view — proportional data representation



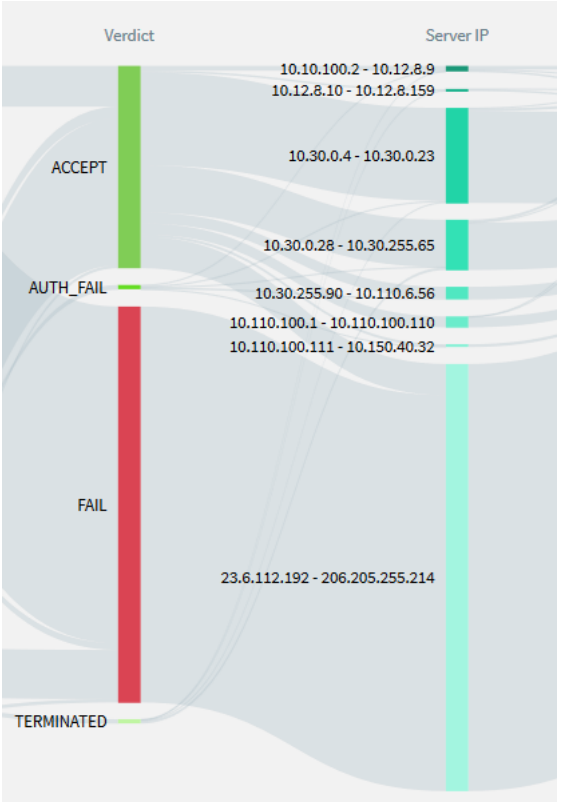
- See at a glance the relationship between various metadata and identify patterns in user behavior.

Example: Relationship between metadata

You want to have an overview of activities where access was denied.

A quick look at the **Verdict** column shows that there were several accesses where the authentication failed (**AUTH_FAIL**) and the lines from the **AUTH_FAIL** field point to several server addresses.

Figure 214: Search > Flow view — relationship between metadata



- Use it interactively to drill down further on information.

  To drill down on information, click on an item, then click **Search**.

  TIP:

  To exclude an item, press `Ctrl` while clicking the item.

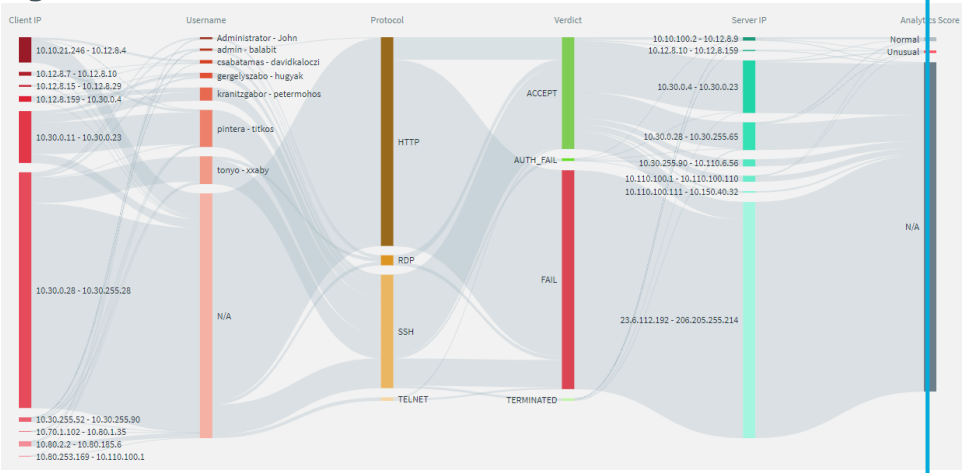
Example: Interactive drill down

You want to investigate if there were any unusual activities. To take a closer look, in the **Analytics Score** column, click **Unusual**,

then click **Search**.

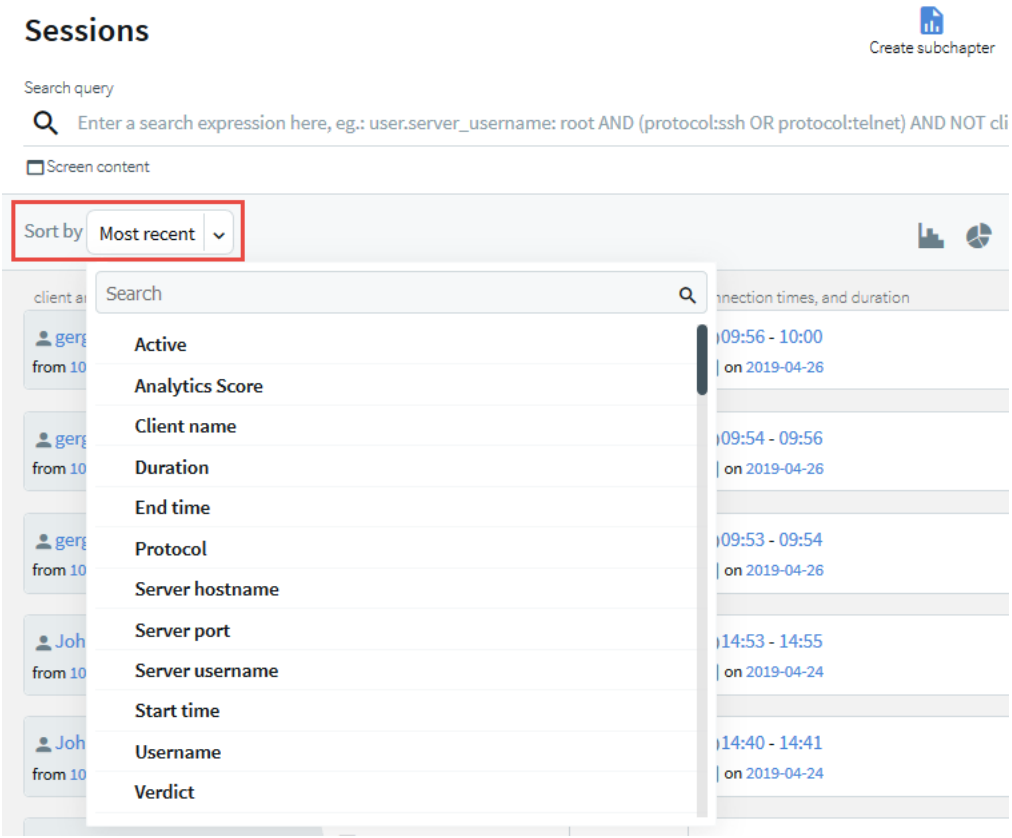
The flow view now only displays the unusual session activities. You can further narrow your search as required.

Figure 215: Search > Flow view — interactive drill down



2. To sort columns, from the card or table view, click the **Sort by** drop-down menu and select from the list.

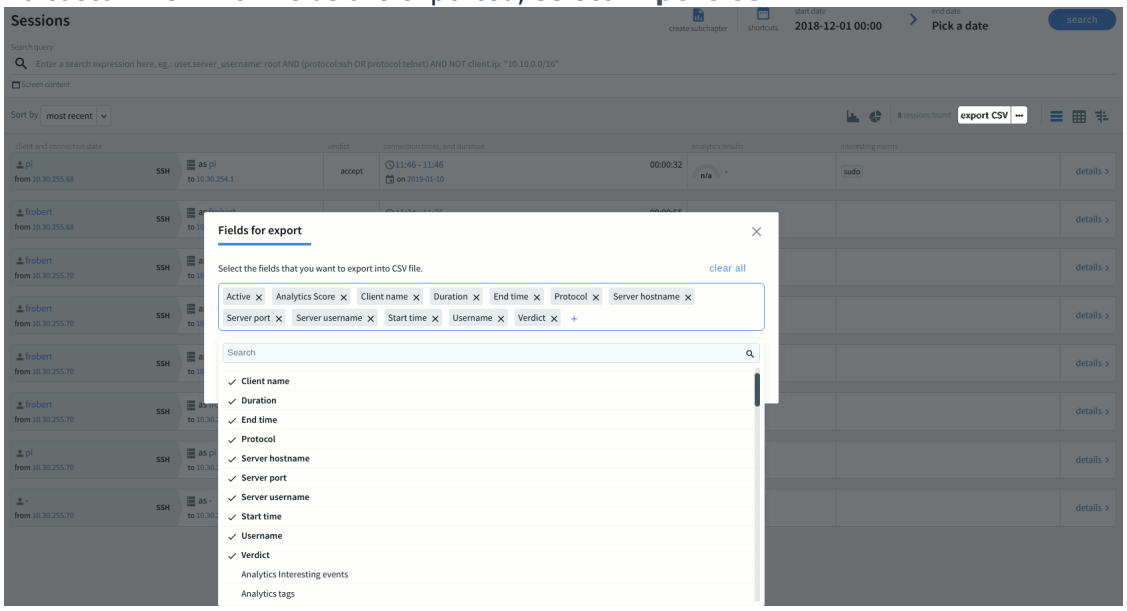
Figure 216: Search — Sort columns



For example, to see the shortest session, select **Duration** from the list. Sessions are now sorted based on duration with the shortest session first. To see the longest session, click (**ascending**).

3. Specify a date and time range to restrict your search criteria as described in Specifying time ranges on page 632.
4. Filter connections as described in Using search filters on page 635.
5. Search the contents of audit trails as described in Searching in the contents of audit trails on page 694.
6. View connection details as described in Viewing session details on page 716.
7. Download and replay audit trails as described in Replaying audit trails in your browser on page 721.
8. To export the search results as a comma-separated text file, select **Export CSV**. Note that if your search returns more than 10.000 results, only the first 10.000 rows are exported. If you want to see all results, refine your search.

To customize which fields are exported, select **Export CSV ...**.



Assigning search privileges

The following describes how to assign users to access sessions only for connections for which they are granted permission.

Users need the **Search** privilege to access the Search interface.

Assigning the **Search** privilege to a user on the **AAA > Access Control** page, automatically enables the **Search in all connections** privilege, and grants the user access to every session, even if the user is not a member of the groups listed in the **Access Control** option of the particular connection policy.

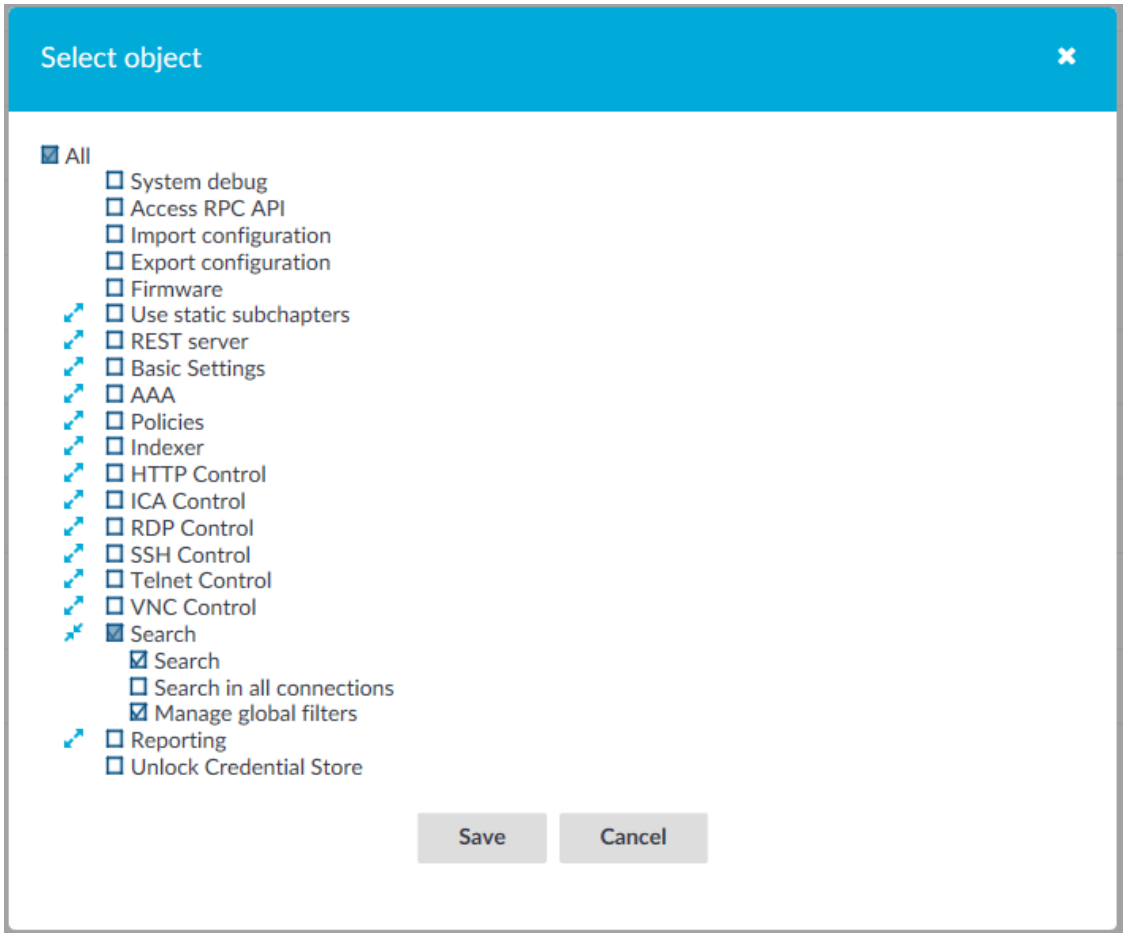
Prerequisites

- You have created a user for which you want to assign the search privilege. For more information, see Creating local users in One Identity Safeguard for Privileged Sessions (SPS).
- You have created a usergroup. For more information, see Managing local usergroups.

To assign users to access sessions only for connections for which they are granted permission

1. Navigate to **AAA > Access Control**.

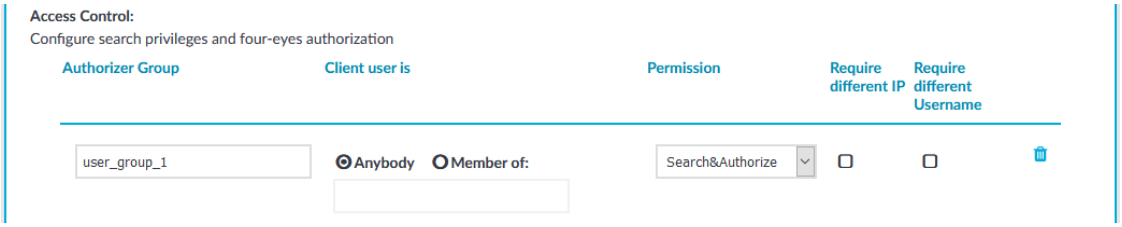
2. **Figure 217: AAA > Access Control — Configuring search privileges**



   Assign the **Search** privilege to your usergroup as described in Assigning privileges to usergroups for the One Identity Safeguard for Privileged Sessions (SPS) web interface.

3. Deselect the **Search in all connections** privilege so that users can access sessions only for connections for which they are granted permission.
4. To grant permission to a specific connection, navigate to the **Connections** page of the traffic (for example to **SSH Control > Connections**), and select the connection policy to modify.

5. **Figure 218: <Protocol name> Control > Connections > Access Control — Configuring search privileges**



   Navigate to **Access Control** and click +.

6. Enter the name of the usergroup whose members are permitted to access the Search interface into the **Authorizer Group** field. This group must exist on the **AAA > Group Management** page.

   CAUTION:

   Usernames, the names of user lists, and the names of usergroups are case sensitive.

7. Set the permissions of the usergroup.
   - If the usergroup can authorize (that is, enable) and audit (that is, monitor in real-time and download the audit trails) the sessions, select **Permission > Search&Authorize**.
   - If the usergroup can only audit the sessions but cannot authorize, select **Permission > Search**.

   NOTE:

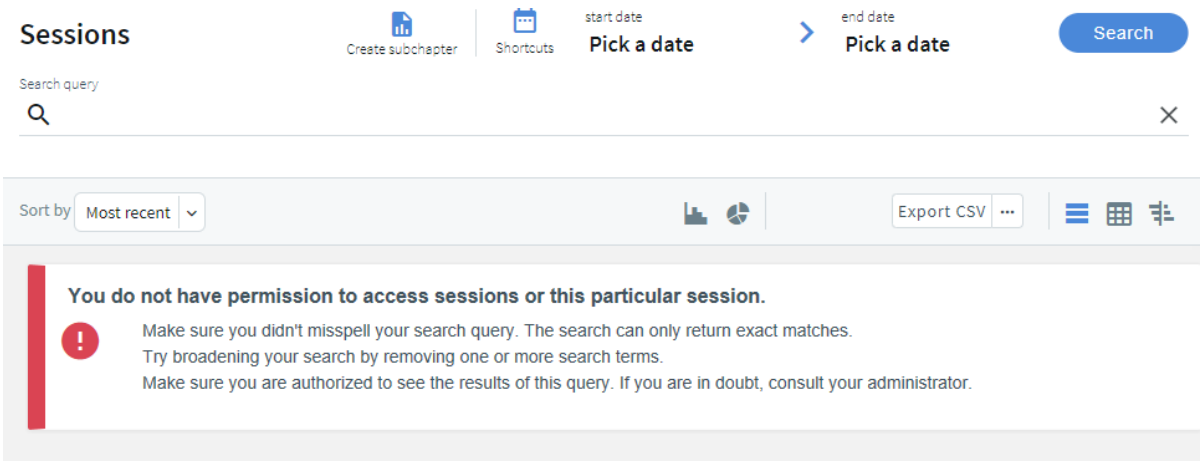
   If the **Client user is > Member of** field is set, the auditor can only monitor the sessions of the specified usergroup. However, if **Client user is > Member of** field is set, the Auditor cannot access the **Search** page. To avoid this problem, add another Access Control rule for the **Authorizer Group** without setting the **Client user is**field.

   The `admin` user of One Identity Safeguard for Privileged Sessions (SPS) can audit and authorize every connection.

Result

Users with the relevant privileges can now access the sessions for which they are granted permission. If users do not have the required permission to access sessions, a warning message is displayed and no session is visible as shown below:

Figure 219: Search — Permission denied



Specifying time ranges

Specify a time range to restrict, or filter your search criteria by setting boundaries on your searches. You can restrict the search to one of the preset time ranges, or use a custom time range for a more specific search.

When you specify a time range, the search result includes:

- Connections started and finished anywhere between the start time and end time you specified.
- Connections started anywhere between the start time and end time you specified.
- Connections ended anywhere between the start time and end time you specified.
- Active connections if they were started anywhere between the start time and the end time you specified.

For example, at `17:00` PM you specify a start date of `10:00` AM and end date of `15:00` PM for your search. The search result includes:

- Connections started at `8:00` AM and ended at `14:00` PM.
- Connections started at `11:00` AM and ended at `14:00` PM.
- Connections started at `11:00` AM and ended at `16:00` PM.
- Active connections started at `11:00` AM.
- Active connections started at `10:00` AM.

To specify time ranges

1. To select the start date of your search, click **Pick a date**.

   Alternatively, use the **shortcuts** button to restrict the search to one of the preset time ranges. For example, to investigate an incident that occurred sometime in the last hour, you can select **Today**, but a better option is **Last 60 minutes**.

   Figure 220: Search — Pick a date

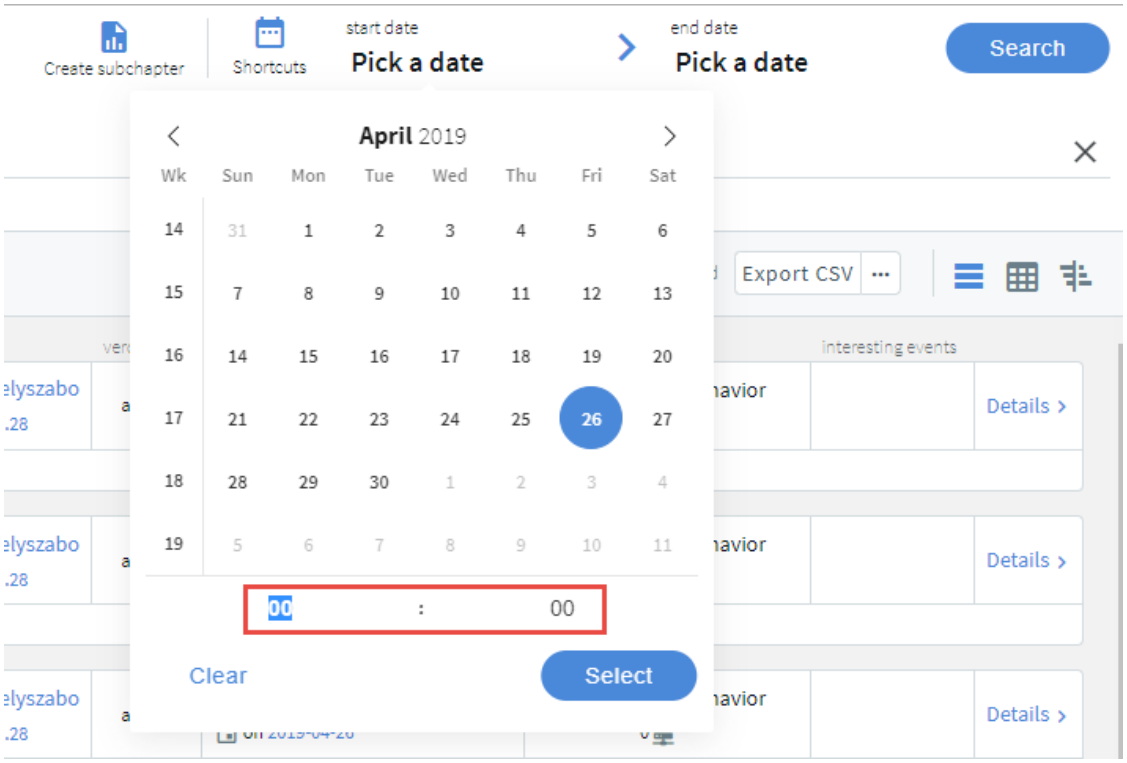
2. From the calendar, select the start date as required.

   NOTE:

   The date refers to the timezone configured on SPS.

3. For exact time ranges, specify to search by the hour and minute.

   Figure 221: Search — Specify hour and minute

4. To select the end date of your search, click **Pick a date** and select a date as required.

   If you specify only the start date, the end date is set to the current time.

5. *Optional:* To clear the start and end date, click **shortcuts** > **All time**.

6. *Optional:* You can use the timeline for a quick time range selection and visual representation of sessions in the selected interval.

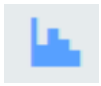
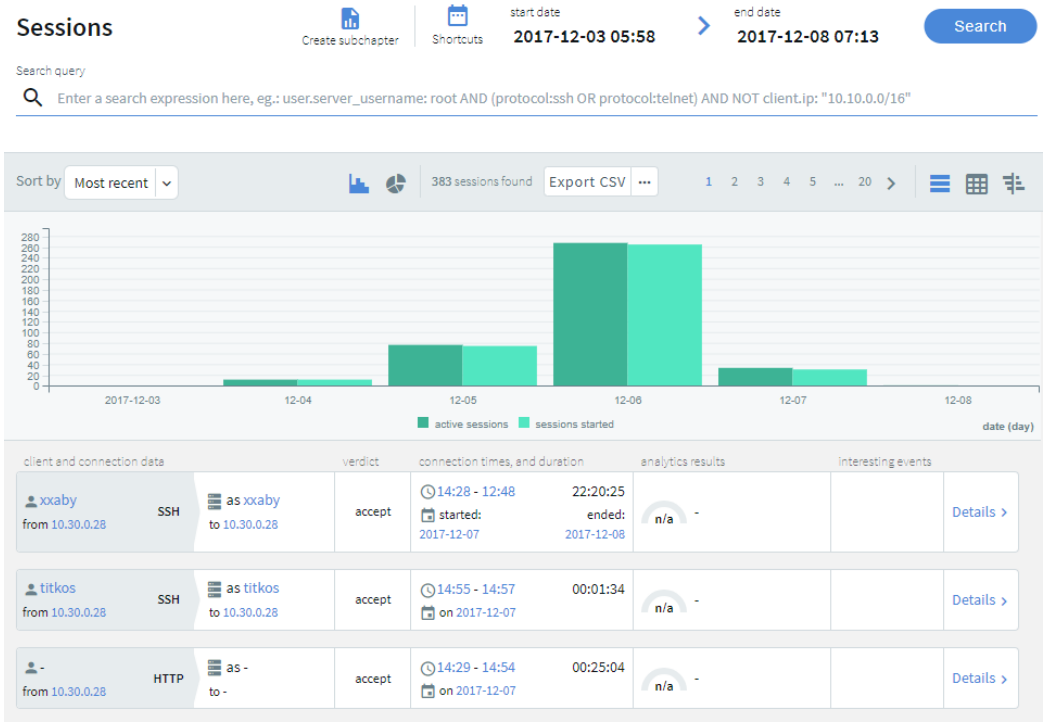
   Click the  icon.

   Figure 222: Search — Using the timeline

   a.

   The bars display the number of results in the selected interval.

   The **active sessions** columns indicate all the sessions, which were active in the selected interval. The **sessions started** columns indicate all the sessions started during the selected interval. For example, if the selected interval is today between `8:00` AM and `9:00` AM, then a session started at `7:00` AM but lasting after `8:00` AM is displayed in the **active sessions** column. A session started at `8:30` AM is displayed in the **sessions started** column. Since the session was active during the selected time interval, the session started at `8:30` AM is also displayed in the **active sessions** column.

   To disable the active sessions and view only the started sessions in the timeline, click active sessions. To disable the started sessions and view only the active sessions in the timeline, click sessions started.

   Hovering the mouse above a bar displays the number of entries and the start and end date of the period that the bar represents.

   b. To select a range, drag the mouse pointer across the timeline or use Shift+Click and select multiple bars.

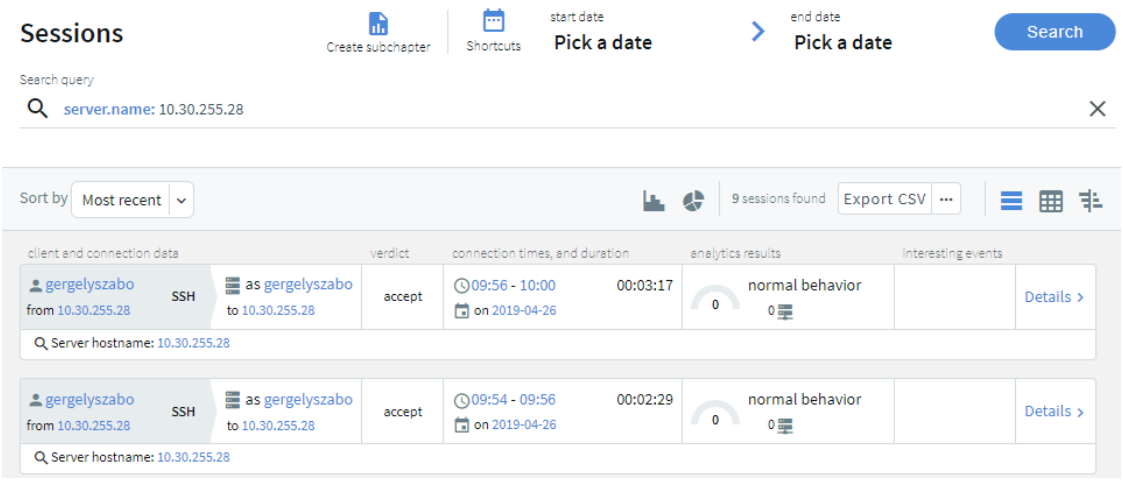
Using search filters

The following describes how you can use search filters to perform a more specific search.

To search using search filters

1. Enter a search filter in the **Search query** field, or click on an entry in the table.

   Figure 223: Search — Search filters

   There are search filters that are not displayed but you can still use them to filter the sessions. For example, you can search for active connections using the `active` search filter, and search results are listed accordingly, but there is no **active** field displayed in the search table or in the **Overview**, **Details**, **Events**, **Alerts**, or **Contents** tabs.

   For the list of search filters that you can use, see List of available search filters on page 636.

   > TIP:
   >
   > Search is case insensitive. To make the search case sensitive, enclose the search keywords in double quotes.
   >
   > The search queries can include only alphanumerical characters. You can use complex expressions and boolean operators. For more information, see Using the content search on page 950.

2. After specifying the relevant filters, click **Search**.

   > TIP:
   >
   > To save the filters for future use, simply save the URL or bookmark it in your browser.

Expected result

Connection metadata is displayed in columns that you can filter for any parameter, or a combination of parameters. You can view the connection metadata in the search columns and also displayed as fields in the **Overview**, **Details**, **Events**, **Alerts**, or **Contents** tabs.

List of available search filters

This section lists the search filters that you can use to perform a more specific search. For information about how to use the search filters listed below, see Using search filters.

The following table provides an explanation to the search filter tables listed in this section.

Name:	Specifies the meaningful and easily readable name of the search filter.
Search filter:	Specifies the filter expression that you can use to filter the audit trails. For example, to narrow your search to a specific server-side IP address, you can enter the `server.address: 10.30.255.70` search filter in the **Search query** field. All search results that contain that specific server IP address are listed.
Displayed:	Specifies if the search filter result is displayed as a field in the search columns or in the **Overview**, **Details**, **Events**, **Alerts**, or **Contents** tabs. There are search filters that are not displayed but you can still use them to filter the audit trails. For example, you can search for active connections using the `active` search filter, and search results are listed accordingly, but there is no **active** field displayed in the search table or in the **Overview**, **Details**, **Events**, **Alerts**, or **Contents** tabs.

The following search filters are available:

alert

Alert type

Name:	Alert type
Search filter:	alert_type
Type:	enum
Displayed:	True

The type of the alert.

Possible values:

- *adp.event.command*: A command entered in SSH or Telnet.
- *adp.event.screen.content*: Alert triggered by the screen content.
- *adp.event.screen.creditcard*: Credit card numbers detected. Displayed only as an alert, not visible in the events.
- *adp.event.screen.windowtitle*: The title of the window in graphic protocols.

Channel ID

Name:	Channel ID
Search filter:	channel_id
Type:	string
Displayed:	True

The id of the channel the alert belongs to.

Matched regexp on action

Name:	Matched regexp on action
Search filter:	matched_action
Type:	string
Displayed:	True

The regular expression that matched the command line without prompt

Matched content

Name:	Matched content
Search filter:	matched_content
Type:	string
Displayed:	True

The content the alert matched.

Note that this value contains the context of the match as well. For example, if a Content Policy triggers an alert if a user types the `sudo` command, then the `psm.alerts.matched_content` value contains the entire command line, including the command prompt, for example, `myuser@examplehost:~$ man sudo`.

Matched regexp

Name:	Matched regexp
Search filter:	matched_regexp
Type:	string
Displayed:	True

The regular expression that matched the content.

For details, see Real-time content monitoring with Content Policies on page 451.

Alert ID

Name:	Alert ID
Search filter:	record_id
Type:	long
Displayed:	True

The identifier of the alert within the audit trail (.zat file).

Rule name

Name:	Rule name
Search filter:	rule_name
Type:	string
Displayed:	True

The name of the content policy rule.

Note that this is not the name of the Content Policy.

Alert time

Name:	Alert time
Search filter:	time
Type:	date
Displayed:	False

The timestamp of the alert.

channel

Channel is active

Name:	Channel is active
Search filter:	active
Type:	boolean
Displayed:	True

True if the session has not ended yet.

Application

Name:	Application
Search filter:	application
Type:	string
Displayed:	True

The name of the application accessed in a seamless Citrix ICA connection.

Audit stream ID

Name:	Audit stream ID
Search filter:	audit_stream_id
Type:	string
Displayed:	True

The identifier of the channel's audit stream. If the session does not have an audit trail, this element is not used.

Channel ID

Name:	Channel ID
Search filter:	channel_id
Type:	long
Displayed:	True

The unique ID of the channel.

Client X.509 Subject

Name:	Client X.509 Subject
Search filter:	client_x509_subject
Type:	string
Displayed:	True

The client's certificate in TELNET or VNC sessions. Available only if the 'Client-side transport security settings > Peer certificate validation' option is enabled in One Identity Safeguard for Privileged Sessions.

Executed commands

Name:	Executed commands
Search filter:	command
Type:	string
Displayed:	True

Lists the commands executed in an SSH session.

Port-forward target IP

Name:	Port-forward target IP
Search filter:	connected.ip
Type:	ip
Displayed:	True

The traffic was forwarded to this IP address in Remote Forward and Local Forward channels.

Port-forward target name

Name:	Port-forward target name
Search filter:	connected.name

Type:	text
Displayed:	True

The traffic was forwarded to this host in Remote Forward and Local Forward channels. If the hostname is not available, this field contains the IP address of the host

Port-forward target port

Name:	Port-forward target port
Search filter:	connected.port
Type:	port
Displayed:	True

The traffic was forwarded to this port in Remote Forward and Local Forward channels.

Device name

Name:	Device name
Search filter:	device_name
Type:	string
Displayed:	True

The name or ID of the shared device (redirect) used in the RDP connection.

Description: Used with the `serial redirect`, `parallel redirect`, `printer redirect`, `disk redirect`, and `scard redirect` RDP channel types.

The name of the device.

Channel duration

Name:	Channel duration
Search filter:	duration
Type:	long
Displayed:	True

The length of the channel (how long the channel lasted).

Dynamic channel

Name:	Dynamic channel
Search filter:	dynamic_channel
Type:	string
Displayed:	True

The name or ID of the dynamic channel opened in the RDP session.

Channel end time

Name:	Channel end time
Search filter:	end_time
Type:	date
Displayed:	True

Date when the channel was closed.

Environment

Name:	Environment
Search filter:	environment
Type:	string
Displayed:	True

Date when the channel was closed.

Four-eyes authorizer

Name:	Four-eyes authorizer
Search filter:	four_eyes_authorizer
Type:	string
Displayed:	True

The username of the user who authorized the session. Available only if four-eyes authorization is required for the channel.

Four-eyes description

Name:	Four-eyes description
Search filter:	four_eyes_description
Type:	string
Displayed:	True

The description submitted by the authorizer of the session.

Channel originator IP address

Name:	Channel originator IP address
Search filter:	originator.ip
Type:	ip
Displayed:	True

The IP address of the host initiating the channel in Remote Forward and Local Forward channels. Note that this host is not necessarily the client or the server of the SSH connection.

Channel originator name

Name:	Channel originator name
Search filter:	originator.name
Type:	text
Displayed:	True

The hostname of the host initiating the channel in Remote Forward and Local Forward channels. Note that this host is not necessarily the client or the server of the SSH connection. If the hostname is not available, this field contains the IP address of the host.

Originator port

Name:	Originator port
Search filter:	originator.port
Type:	port
Displayed:	True

The number of the forwarded port in Remote Forward and Local Forward SSH channels.

Rule number

Name:	Rule number
Search filter:	rule_num
Type:	string
Displayed:	True

The number of the line in the Channel policy applied to the channel.

SCP path

Name:	SCP path
Search filter:	scp_path
Type:	string
Displayed:	True

Name and path of the file copied via SCP. Available only for SCP sessions (Session exec SCP SSH channels) if the Log file transfers to database option isenabled in the Channel Policy of the connection.

Channel start time

Name:	Channel start time
Search filter:	start_time
Type:	date
Displayed:	True

Date when the channel was started.

Subsystem name

Name:	Subsystem name
Search filter:	subsystem_name
Type:	string
Displayed:	True

Name of the SSH subsystem used in the channel.

Channel type

Name:	Channel type
Search filter:	type
Type:	enum
Displayed:	True

Type of the channel.

Possible values:

- *#drawing*: Drawing
- *CTXCAM*: Audio
- *CTXCDM*: Drive
- *CTXCLIP*: Clipboard
- *CTXCOM1*: Printer (COM1)
- *CTXCOM2*: Printer (COM2)
- *CTXCPM*: Printer Spooler
- *CTXFLSH*: HDX Mediastream
- *CTXLPT1*: Printer (LPT1)
- *CTXLPT2*: Printer (LPT2)
- *CTXSCRD*: Smartcard
- *CTXTW*: Drawing (Thinwire)
- *CTXTWI*: Seamless
- *CTXUSB*: USB
- *SPDBRS*: Speedbrowse
- *auth-agent*: Agent
- *cliprdr*: Clipboard
- *custom*: Custom
- *direct-tcpip*: Local forward
- *drawing*: Drawing
- *drdynvc*: Dynamic virtual channel
- *forwarded-tcpip*: Remote forward
- *http*: HTTP
- *rdpdr*: Redirects

- *rdpdr-disk*: Disk redirect
- *rdpdr-parallel*: Parallel redirect
- *rdpdr-printer*: Printer redirect
- *rdpdr-scard*: SCard redirect
- *rdpdr-serial*: Serial redirect
- *rdpsnd*: Sound
- *seamrdp*: Seamless
- *session-exec*: Session exec
- *session-exec-scp*: Session exec SCP
- *session-shell*: Session shell
- *session-subsystem*: Session subsystem
- *session-subsystem-sftp*: Session SFTP
- *telnet*: Telnet
- *vnc*: VNC
- *x11*: X11 forward

Channel verdict

Name:	Channel verdict
Search filter:	verdict
Type:	enum
Displayed:	True

Indicates what One Identity Safeguard for Privileged Sessions decided about the channel. Possible values:

- *ACCEPT*: Accepted
- *DENY*: Denied
- *FOUR_EYES_DEFERRED*: Waiting for remote username
- *FOUR_EYES_ERROR*: Internal error during four-eyes authorization
- *FOUR_EYES_REJECT*: Four-eyes authorization rejected
- *FOUR_EYES_TIMEOUT*: Four-eyes authorization timed out

content

Window title

Name:	Window title
Search filter:	title
Type:	string
Displayed:	True

The content of the title bar in the active window. The window title typically contains the name of the application, or the name of the dialogue box. Only available in graphical sessions (for example, RDP), if indexing is enabled.

Command

Name:	Command
Search filter:	command
Type:	string
Displayed:	True

The commands that the user executed in the session. Only available in terminal sessions (for example, SSH), if indexing is enabled.

event

Event Action

Name:	Event Action
Search filter:	action
Type:	string
Displayed:	True

The command line without prompt in commands

Channel ID

Name:	Channel ID
Search filter:	channel_id
Type:	string
Displayed:	True

The id of the channel the event belongs to.

Event content

Name:	Event content
Search filter:	content
Type:	string
Displayed:	True

The command executed, or the window title detected in the channel (for example, ls, exit, or Firefox).

Protocol details

Name:	Protocol details
Search filter:	details
Type:	string
Displayed:	True

The details of the protocol used for the operation.

Operation

Name:	Operation
Search filter:	operation
Type:	string
Displayed:	True

The type of the operation that occurred, for example, Create file (in the case of FTP) or GET (in the case of HTTP).

Path

Name:	Path
Search filter:	path
Type:	string
Displayed:	True

The path (if any) used by the operation that occurred.

Event ID

Name:	Event ID
Search filter:	record_id
Type:	long
Displayed:	True

The identifier of the event within the audit trail (.zat file).

Response code

Name:	Response code
Search filter:	response_code
Type:	long
Displayed:	True

The status code of the protocol response (if any) returned.

Event date

Name:	Event date
Search filter:	time
Type:	date
Displayed:	False

The date when the event happened.

Event type

Name:	Event type
Search filter:	type
Type:	string
Displayed:	True

The type of the event, for example, command, screen_content, window_title.

indexer_info

Commands indexed

Name:	Commands indexed
Search filter:	config.command.enabled
Type:	boolean
Displayed:	True

True if commands were extracted while indexing the session.

Keyboard buffering interval

Name:	Keyboard buffering interval
Search filter:	config.keyboard.buffer_interval
Type:	double
Displayed:	True

The buffering interval in milliseconds used when extracting keyboard events while indexing the session.

Keyboard extracted

Name:	Keyboard extracted
Search filter:	config.keyboard.enabled
Type:	boolean
Displayed:	True

True if keyboard events were extracted while indexing the session.

Mouse buffering interval

Name:	Mouse buffering interval
Search filter:	config.mouse.buffer_interval
Type:	double
Displayed:	True

The buffering interval in milliseconds used when extracting mouse events while indexing the session.

Mouse extracted

Name:	Mouse extracted
Search filter:	config.mouse.enabled
Type:	boolean
Displayed:	True

True if mouse events were extracted while indexing the session.

Near real-time indexing

Name:	Near real-time indexing
Search filter:	config.near_realtime
Type:	boolean
Displayed:	True

True if indexing this session was done near real-time (when the session was still active).

OCR languages

Name:	OCR languages
Search filter:	config.ocr_languages
Type:	string
Displayed:	True

The language configuration for optical character recognition used when indexing the session.

Screen content indexed

Name:	Screen content indexed
Search filter:	config.screen.enabled
Type:	boolean
Displayed:	True

True if screen content was extracted while indexing the session.

OCR tradeoff

Name:	OCR tradeoff
Search filter:	config.screen.omnipage_trade_off
Type:	string
Displayed:	True

The tradeoff used for optical character recognition when extracting screen content while indexing the session.

Titles indexed

Name:	Titles indexed
Search filter:	config.title.enabled
Type:	boolean
Displayed:	True

True if window titles were extracted while indexing the session.

Indexing error

Name:	Indexing error
Search filter:	error.message
Type:	string
Displayed:	True

The reason why indexing failed

Indexing cpu time

Name:	Indexing cpu time
Search filter:	statistics.cpu_time
Type:	long
Displayed:	True

The CPU time that indexing this session took in milliseconds.

Indexing duration

Name:	Indexing duration
Search filter:	statistics.duration
Type:	long
Displayed:	True

The duration of time that indexing this session took in milliseconds.

Indexing start time

Name:	Indexing start time
Search filter:	statistics.start_time
Type:	date
Displayed:	True

The time and date when indexing this session started.

Indexing status

Name:	Indexing status
Search filter:	status
Type:	string
Displayed:	True

Shows if the channel has been indexed successfully or not.

Indexer ADP version

Name:	Indexer ADP version
Search filter:	version.adp
Type:	string
Displayed:	True

The version of the audit data processor used for indexing the session

Indexer version

Name:	Indexer version
Search filter:	version.worker
Type:	string
Displayed:	False

The version of the indexer worker used for indexing the session

ZAC created

Name:	ZAC created
Search filter:	config.zac.enabled
Type:	boolean
Displayed:	False

True if an Audit Content file was created while indexing the session.

screen

Screen content

Name:	Screen content
Search filter:	content
Type:	string
Displayed:	False

Text that appeared on the screen in the session.

Channel id in trail

Name:	Channel id in trail
Search filter:	channel_id_in_trail
Type:	long
Displayed:	False

The ID of the channel where this content appeared. To check the channel ID (channel_id), select a session and click details. Navigate to details > Channels and click the channel type.

Screen content creation time

Name:	Screen content creation time
Search filter:	time
Type:	screen
Displayed:	False

The creation time of the indexed screen content.

Screen content ID

Name:	Screen content ID
Search filter:	id
Type:	string
Displayed:	False

The ID of a screen content event.

session

Active

Name:	Active
Search filter:	active
Type:	boolean
Displayed:	True

The session is still open.

Analytics Interesting events

Name:	Analytics Interesting events
Search filter:	analytics.interesting_events

Type:	string
Displayed:	True

Collection of interesting command(s) and window title(s) from the session.

Analytics Score

Name:	Analytics Score
Search filter:	analytics.score.aggregated
Type:	long
Displayed:	True

The risk score that the Analytics Module assigned to the session.Ranges from 0 to 100, 100 is the highest risk score.

Score time

Name:	Score time
Search filter:	analytics.score.time
Type:	date
Displayed:	False

The scoring time of the given analytics. The different analytics are scored at different times based on the type of the analytics and certain configuration settings.

Command score

Name:	Command score
Search filter:	analytics.score.details.command.score
Type:	long
Displayed:	True

Score given by the Command algorithm.

FIS score

Name:	FIS score
Search filter:	analytics.score.details.fis.score
Type:	long
Displayed:	True

Score given by the Frequent Item Set (FIS) algorithm

Host login score

Name:	Host login score
Search filter:	analytics.score.details.hostlogin.score
Type:	long
Displayed:	True

Score given by the Host login algorithm.

Login time score

Name:	Login time score
Search filter:	analytics.score.details.logintime.score
Type:	long
Displayed:	True

Score given by the Login time algorithm.

Keystroke score

Name:	Keystroke score
Search filter:	analytics.score.details.keystroke.score
Type:	long
Displayed:	True

Score given by the Keystroke algorithm.

Windowtitle score

Name:	Windowtitle score
Search filter:	analytics.score.details.windowtitle.score
Type:	long
Displayed:	True

Score given by the Window title algorithm.

Scripted

Name:	Scripted
Search filter:	analytics.scripted
Type:	boolean
Displayed:	True

True if the One Identity Safeguard for Privileged Analytics module marked the session as scripted because of non-human activity

Similar Sessions

Name:	Similar Sessions
Search filter:	analytics.similar_sessions
Type:	string
Displayed:	True

Collection of similar sessions from different sources.

Analytics tags

Name:	Analytics tags
Search filter:	analytics.tags
Type:	string
Displayed:	True

The Analytics tags section in Search > details.

Client IP

Name:	Client IP
Search filter:	client.ip
Type:	ip
Displayed:	True

The IP address of the client that initiated the session.

Client name

Name:	Client name
Search filter:	client.name
Type:	string
Displayed:	True

The name of the client that initiated the session.

Client port

Name:	Client port
Search filter:	client.port
Type:	port
Displayed:	True

The port number of the client that initiated the session.

Creation time

Name:	Creation time
Search filter:	creation_time
Type:	date
Displayed:	True

The first time the pipeline created the session. It is different from start_time and can be later than start_time.

Duration

Name:	Duration
Search filter:	duration
Type:	long
Displayed:	True

The length of the session (how long the session lasted).

End time

Name:	End time
Search filter:	end_time
Type:	date
Displayed:	True

Date when the session was closed.

For ongoing connections, the value is `null`.

Starting with SPS 5 LTS, the timestamp is in ISO 8601 format, for example, `2018-10-11T09:23:38.000+02:00`. In earlier versions, it was in UNIX timestamp format.

Log adapter

Name:	Log adapter
Search filter:	log.adapter_name
Type:	string
Displayed:	True

The name of the Log Adapter Plugin. This plugin can be uploaded at Basic Settings > Plugins.

Log auth method

Name:	Log auth method
Search filter:	log.auth_method
Type:	string
Displayed:	True

SSH relayed authentication method. It is configured at SSH Control > Authentication Policies > Relayed authentication methods.

Log syslog time

Name:	Log syslog time
Search filter:	log.syslog_time
Type:	date
Displayed:	True

Date of the message in the ISO 8601 compatible standard timestamp format.

Node ID

Name:	Node ID
Search filter:	node_id
Type:	string
Displayed:	True

The node ID of the Safeguard for Privileged Sessions machine

Origin

Name:	Origin
Search filter:	origin
Type:	string
Displayed:	True

How One Identity Safeguard for Privileged Analytics received this session. Can be One Identity Safeguard for Privileged Sessions for sessions based on an audit trail recorded by One Identity Safeguard for Privileged Sessions, or LOG for sessions built from log data.

Protocol

Name:	Protocol
Search filter:	protocol

Type:	enum
Displayed:	True

The protocol used in the session: Citrix ICA, HTTP, RDP, SSH, Telnet (including TN3270 and TN5250), or VNC.

Possible values:

- *HTTP*: HTTP
- *ICA*: ICA
- *RDP*: RDP
- *SSH*: SSH
- *TELNET*: TELNET
- *VNC*: VNC

Additional metadata

Name:	Additional metadata
Search filter:	recording.additional_metadata
Type:	string
Displayed:	False

Data about the session recorded by the different plugins of One Identity Safeguard for Privileged Sessions, for example, when using an Authentication and Authorization plugin.

Recording Archive date

Name:	Recording Archive date
Search filter:	recording.archive.date
Type:	date
Displayed:	True

The date when the connection was archived or cleaned up.

Recording Archive path

Name:	Recording Archive path

Search filter:	recording.archive.path
Type:	string
Displayed:	True

The path where the audit trail was archived on the remote server.

Recording Archive policy

Name:	Recording Archive policy
Search filter:	recording.archive.policy
Type:	string
Displayed:	True

The archive policy used to archive the audit trail.

Recording Archive server

Name:	Recording Archive server
Search filter:	recording.archive.server
Type:	ip
Displayed:	True

The hostname or IP address of the remote server where the audit trail was archived.

Recording Archived

Name:	Recording Archived
Search filter:	recording.archived
Type:	boolean
Displayed:	True

Shows if the data (metadata, audit trail) about the session was archived to a remote server.

Audit trail path

Name:	Audit trail path
Search filter:	recording.audit_trail
Type:	string
Displayed:	False

The path to the audit trail file on One Identity Safeguard for Privileged Sessions. If One Identity Safeguard for Privileged Sessions has already archived the audit trail, see the Archive path field instead.

. If the session does not have an audit trail, this element is not used. To download the audit trail, see Replaying audit trails in your browser on page 721.

Audit trail download link

Name:	Audit trail download link
Search filter:	trail_download_link
Type:	string
Displayed:	True

The download link to the audit trail file on One Identity Safeguard for Privileged Sessions.

Recording Authentication method

Name:	Recording Authentication method
Search filter:	recording.auth_method
Type:	string
Displayed:	True

The authentication method used in the session.

Recording Channel policy

Name:	Recording Channel policy
Search filter:	recording.channel_policy
Type:	string
Displayed:	True

The Channel policy applied to the session. Channel policy determines the channels permitted in the connection, and if the channel is audited or not. The Channel policy can restrict access based on IP address, user list, user group, or time policy.

You can find the list of channel policies for each protocol at the **<Protocol> Control > Channel Policies** page.

Commands available

Name:	Commands available
Search filter:	recording.command_extracted
Type:	boolean
Displayed:	True

True if commands have been extracted from the session. The extracted commands are in the Events field.

Recording Connection policy

Name:	Recording Connection policy
Search filter:	recording.connection_policy
Type:	string
Displayed:	True

The name of the Connection policy that handled the client's connection request.

This is the name displayed on the **<Protocol> Control > Connections** page of the SPS web interface, and in the `name` field of the Connection Policy object. You can find the list of connection policies for each protocol at the **<Protocol> Control > Connections** page.

Recording Connection policy ID

Name:	Recording Connection policy ID
Search filter:	recording.connection_policy_id
Type:	string
Displayed:	True

The ID of the Connection policy that handled the client's connection request.

You can find the list of connection policies for each protocol at the **<Protocol> Control > Connections** page.

Recording Content reference ID

Name:	Recording Content reference ID
Search filter:	recording.content_reference_id
Type:	long
Displayed:	True

The unique identifier for the session content search.

Recording Indexing status

Name:	Recording Indexing status
Search filter:	recording.index_status
Type:	enum
Displayed:	True

Shows if the channel has been indexed.

Possible values:

- *CHANNEL_OPEN*: Session is active
- *INDEXED*: Session indexed
- *INDEXING_FAILED*: Session indexing failed
- *INDEXING_IN_PROGRESS*: Session indexing in progress
- *INDEXING_NOT_REQUIRED*: Session indexing not required
- *NOT_INDEXED*: Session is not indexed
- *NO_TRAIL*: Auditing not enabled

Has ZAC

Name:	Has ZAC
Search filter:	recording.has_zac
Type:	boolean
Displayed:	False

Audit Content file is available for the session. This file allows the user to search the content of graphical sessions using the Safeguard Desktop Player.

Recording Network namespace

Name:	Recording Network namespace
Search filter:	recording.network_id
Type:	string
Displayed:	True

The ID of the Linux network namespace where the session originated from.

Server local IP address

Name:	Server local IP address
Search filter:	recording.server_local.ip
Type:	ip
Displayed:	True

The IP address of One Identity Safeguard for Privileged Sessions used in the server-side connection.

Server local name

Name:	Server local name
Search filter:	recording.server_local.name
Type:	text
Displayed:	True

The hostname of One Identity Safeguard for Privileged Sessions used in the server-side connection. If the hostname is not available, this field contains the IP address of One Identity Safeguard for Privileged Sessions.

Recording Server local port

Name:	Recording Server local port
Search filter:	recording.server_local.port
Type:	port
Displayed:	True

The port number of One Identity Safeguard for Privileged Sessions used in the server-side connection.

Recording Session ID

Name:	Recording Session ID
Search filter:	recording.session_id
Type:	string
Displayed:	True

A globally unique string that identifies the session. Log messages related to the session contain this ID.

Target IP address

Name:	Target IP address
Search filter:	recording.target.ip
Type:	ip
Displayed:	True

The client originally tried to access this IP address. This can differ from the destination address, for example, when One Identity Safeguard for Privileged Sessions is configured to redirect the connection. The address that the client actually connected to is in the Server address field.

Target name

Name:	Target name
Search filter:	recording.target.name
Type:	text
Displayed:	True

The client originally tried to access this host. This can differ from the destination address, for example, when One Identity Safeguard for Privileged Sessions is configured to redirect the connection. The address that the client actually connected to is in the Server address field. If the hostname is not available, this field contains the IP address of the host.

Recording Target port

Name:	Recording Target port
Search filter:	recording.target.port
Type:	port
Displayed:	True

The client originally tried to access this port. This can differ from the port of the destination server, for example, when One Identity Safeguard for Privileged Sessions is configured to redirect the connection. The port that the client actually connected to is in the Server port field.

Recording Verdict

Name:	Recording Verdict
Search filter:	recording.verdict
Type:	enum
Displayed:	True

Indicates what One Identity Safeguard for Privileged Sessions decided about the session.

Possible values:

- *ACCEPT*: Accepted
- *ACCEPT_TERMINATED*: Terminated by a content policy
- *AUTH_FAIL*: Authentication failed
- *DENY*: Connection rejected
- *FAIL*: Connection timed out on the server
- *GW_AUTH_FAIL*: Gateway authentication failed
- *KEY_ERROR*: Hostkey mismatch
- *USER_MAPPING_FAIL*: Usermapping failed

Recording Window titles available

Name:	Recording Window titles available
Search filter:	recording.window_title_extracted
Type:	boolean
Displayed:	True

True if window titles have been extracted from the session. The extracted window titles are in the Window title field.

Server IP

Name:	Server IP
Search filter:	server.ip
Type:	ip
Displayed:	True

The IP address of the server that One Identity Safeguard for Privileged Sessions connected to. This address was the remote end of the server-side connection.

Server ID

Name:	Server ID
Search filter:	server.id
Type:	string
Displayed:	True

The id of the server that One Identity Safeguard for Privileged Sessions connected to.

Server hostname

Name:	Server hostname
Search filter:	server.name
Type:	string
Displayed:	True

The hostname of the server that One Identity Safeguard for Privileged Sessions connected to.

Server port

Name:	Server port
Search filter:	server.port

Type:	port
Displayed:	True

The port number of the server that One Identity Safeguard for Privileged Sessions connected to.

Start time

Name:	Start time
Search filter:	start_time
Type:	date
Displayed:	True

Date when the session was started.

Starting with SPS 5 LTS, the timestamp is in ISO 8601 format, for example, `2018-10-11T09:23:38.000+02:00`. In earlier versions, it was in UNIX timestamp format.

Gateway username

Name:	Gateway username
Search filter:	user.gateway_username
Type:	string
Displayed:	True

The username used to authenticate on the One Identity Safeguard for Privileged Sessions gateway (that is, in the client-side connection). Sometimes it is also called client-side username.

Gateway username domain

Name:	Gateway username domain
Search filter:	user.gateway_username_domain
Type:	string
Displayed:	True

The domain of the username used to authenticate on the One Identity Safeguard for Privileged Sessions gateway (that is, in the client-side connection).

User ID

Name:	User ID
Search filter:	user.id
Type:	string
Displayed:	True

The ID of the user.

Username

Name:	Username
Search filter:	user.name
Type:	string
Displayed:	True

This field contains the username which was used by the user to authenticate to the remote server. Its value is the same as the gateway username when it is available. Otherwise, it will be filled with the server username.

Name domain

Name:	Name domain
Search filter:	user.name_domain
Type:	string
Displayed:	True

This field contains the domain of the username which was used by the user to authenticate to the remote server. Its value is the same as the gateway domain when it is available. Otherwise, it will be filled with the server domain.

Server username

Name:	Server username
Search filter:	user.server_username
Type:	string
Displayed:	True

The username used to log in to the remote server. This username can differ from the client-side username if usermapping is used in the connection.

Server username domain

Name:	Server username domain
Search filter:	user.server_username_domain
Type:	string
Displayed:	True

The domain of the username used to log in to the remote server.

Verdict

Name:	Verdict
Search filter:	verdict
Type:	enum
Displayed:	True

Indicates what One Identity Safeguard for Privileged Sessions decided about the session. A session verdict that originates from log events or other external events.

Possible values:

- *ACCEPT*: Accepted
- *AUTH_FAIL*: Authentication failed
- *DENY*: Connection rejected
- *FAIL*: Connection timed out on the server
- *PENDING*: Connection is pending
- *TERMINATED*: Connection terminated

Channel is active

Name:	Channel is active
Search filter:	channel.active
Type:	boolean
Displayed:	False

True if the session has not ended yet.

Application

Name:	Application
Search filter:	channel.application
Type:	string
Displayed:	False

The name of the application accessed in a seamless Citrix ICA connection.

Audit stream ID

Name:	Audit stream ID
Search filter:	channel.audit_stream_id
Type:	string
Displayed:	False

The identifier of the channel's audit stream. If the session does not have an audit trail, this element is not used.

Channel ID

Name:	Channel ID
Search filter:	channel.channel_id
Type:	long
Displayed:	False

The unique ID of the channel.

Client X.509 Subject

Name:	Client X.509 Subject
Search filter:	channel.client_x509_subject
Type:	string
Displayed:	False

The client's certificate in TELNET or VNC sessions. Available only if the 'Client-side transport security settings > Peer certificate validation' option is enabled in One Identity Safeguard for Privileged Sessions.

Executed commands

Name:	Executed commands
Search filter:	channel.command
Type:	string
Displayed:	False

Lists the commands executed in an SSH session.

Port-forward target IP

Name:	Port-forward target IP
Search filter:	channel.connected.ip
Type:	ip
Displayed:	False

The traffic was forwarded to this IP address in Remote Forward and Local Forward channels.

Port-forward target name

Name:	Port-forward target name
Search filter:	channel.connected.name
Type:	text
Displayed:	False

The traffic was forwarded to this host in Remote Forward and Local Forward channels. If the hostname is not available, this field contains the IP address of the host

Port-forward target port

Name:	Port-forward target port

Search filter:	channel.connected.port
Type:	port
Displayed:	False

The traffic was forwarded to this port in Remote Forward and Local Forward channels.

Device name

Name:	Device name
Search filter:	channel.device_name
Type:	string
Displayed:	False

The name or ID of the shared device (redirect) used in the RDP connection.

Channel duration

Name:	Channel duration
Search filter:	channel.duration
Type:	long
Displayed:	False

The length of the channel (how long the channel lasted).

Dynamic channel

Name:	Dynamic channel
Search filter:	channel.dynamic_channel
Type:	string
Displayed:	False

The name or ID of the dynamic channel opened in the RDP session.
Used with the `dynamic virtual` RDP channel type.

Channel end time

Name:	Channel end time
Search filter:	channel.end_time
Type:	date
Displayed:	False

Date when the channel was closed.

Environment

Name:	Environment
Search filter:	channel.environment
Type:	string
Displayed:	False

Date when the channel was closed.

Four-eyes authorizer

Name:	Four-eyes authorizer
Search filter:	channel.four_eyes_authorizer
Type:	string
Displayed:	False

The username of the user who authorized the session. Available only if four-eyes authorization is required for the channel.

Four-eyes description

Name:	Four-eyes description
Search filter:	channel.four_eyes_description
Type:	string
Displayed:	False

The description submitted by the authorizer of the session.

Channel originator IP address

Name:	Channel originator IP address
Search filter:	channel.originator.ip
Type:	ip
Displayed:	False

The IP address of the host initiating the channel in Remote Forward and Local Forward channels. Note that this host is not necessarily the client or the server of the SSH connection.

Channel originator name

Name:	Channel originator name
Search filter:	channel.originator.name
Type:	text
Displayed:	False

The hostname of the host initiating the channel in Remote Forward and Local Forward channels. Note that this host is not necessarily the client or the server of the SSH connection. If the hostname is not available, this field contains the IP address of the host.

Originator port

Name:	Originator port
Search filter:	channel.originator.port
Type:	port
Displayed:	False

The number of the forwarded port in Remote Forward and Local Forward SSH channels.

Rule number

Name:	Rule number
Search filter:	channel.rule_num
Type:	string
Displayed:	False

The number of the line in the Channel policy applied to the channel.

SCP path

Name:	SCP path
Search filter:	channel.scp_path
Type:	string
Displayed:	False

Name and path of the file copied via SCP. Available only for SCP sessions (Session exec SCP SSH channels) if the Log file transfers to database option isenabled in the Channel Policy of the connection.

Channel start time

Name:	Channel start time
Search filter:	channel.start_time
Type:	date
Displayed:	False

Date when the channel was started.

Subsystem name

Name:	Subsystem name
Search filter:	channel.subsystem_name
Type:	string
Displayed:	False

Name of the SSH subsystem used in the channel.

Channel type

Name:	Channel type
Search filter:	channel.type
Type:	enum
Displayed:	False

Type of the channel.

Possible values:

- *#drawing*: Drawing
- *CTXCAM*: Audio
- *CTXCDM*: Drive
- *CTXCLIP*: Clipboard
- *CTXCOM1*: Printer (COM1)
- *CTXCOM2*: Printer (COM2)
- *CTXCPM*: Printer Spooler
- *CTXFLSH*: HDX Mediastream
- *CTXLPT1*: Printer (LPT1)
- *CTXLPT2*: Printer (LPT2)
- *CTXSCRD*: Smartcard
- *CTXTW*: Drawing (Thinwire)
- *CTXTWI*: Seamless
- *CTXUSB*: USB
- *SPDBRS*: Speedbrowse
- *auth-agent*: Agent
- *cliprdr*: Clipboard
- *custom*: Custom
- *direct-tcpip*: Local forward
- *drawing*: Drawing
- *drdynvc*: Dynamic virtual channel
- *forwarded-tcpip*: Remote forward
- *http*: HTTP
- *rdpdr*: Redirects
- *rdpdr-disk*: Disk redirect
- *rdpdr-parallel*: Parallel redirect
- *rdpdr-printer*: Printer redirect
- *rdpdr-scard*: SCard redirect
- *rdpdr-serial*: Serial redirect
- *rdpsnd*: Sound
- *seamrdp*: Seamless
- *session-exec*: Session exec
- *session-exec-scp*: Session exec SCP

- *session-shell*: Session shell
- *session-subsystem*: Session subsystem
- *session-subsystem-sftp*: Session SFTP
- *telnet*: Telnet
- *vnc*: VNC
- *x11*: X11 forward

Channel verdict

Name:	Channel verdict
Search filter:	channel.verdict
Type:	enum
Displayed:	False

Indicates what One Identity Safeguard for Privileged Sessions decided about the channel. Possible values:

- *ACCEPT*: Accepted
- *DENY*: Denied
- *FOUR_EYES_DEFERRED*: Waiting for remote username
- *FOUR_EYES_ERROR*: Internal error during four-eyes authorization
- *FOUR_EYES_REJECT*: Four-eyes authorization rejected
- *FOUR_EYES_TIMEOUT*: Four-eyes authorization timed out

Event Action

Name:	Event Action
Search filter:	event.action
Type:	string
Displayed:	False

The command line without prompt in commands

Channel ID

Name:	Channel ID

Search filter:	event.channel_id
Type:	string
Displayed:	False

The id of the channel the event belongs to.

Event content

Name:	Event content
Search filter:	event.content
Type:	string
Displayed:	False

The command executed, or the window title detected in the channel (for example, ls, exit, or Firefox).

Protocol details

Name:	Protocol details
Search filter:	event.details
Type:	string
Displayed:	False

The details of the protocol used for the operation.

Operation

Name:	Operation
Search filter:	event.operation
Type:	string
Displayed:	False

The type of the operation that occurred, for example, Create file (in the case of FTP) or GET (in the case of HTTP).

Path

Name:	Path
Search filter:	event.path
Type:	string
Displayed:	False

The path (if any) used by the operation that occurred.

Event ID

Name:	Event ID
Search filter:	event.record_id
Type:	long
Displayed:	False

The identifier of the event within the audit trail (.zat file).

Response code

Name:	Response code
Search filter:	event.response_code
Type:	long
Displayed:	False

The status code of the protocol response (if any) returned.

Event date

Name:	Event date
Search filter:	event.time
Type:	date
Displayed:	False

The date when the event happened.

Event type

Name:	Event type
Search filter:	event.type
Type:	string
Displayed:	False

The type of the event, for example, command, screen_content, window_title.

Alert type

Name:	Alert type
Search filter:	alert.alert_type
Type:	enum
Displayed:	False

The type of the alert.

Possible values:

- *adp.event.command*: A command entered in SSH or Telnet.
- *adp.event.screen.content*: Alert triggered by the screen content.
- *adp.event.screen.creditcard*: Credit card numbers detected. Displayed only as an alert, not visible in the events.
- *adp.event.screen.windowtitle*: The title of the window in graphic protocols.

Channel ID

Name:	Channel ID
Search filter:	alert.channel_id
Type:	string
Displayed:	False

The id of the channel the alert belongs to.

Matched regexp on action

Name:	Matched regexp on action

Search filter:	alert.matched_action
Type:	string
Displayed:	False

The regular expression that matched the command line without prompt

Matched content

Name:	Matched content
Search filter:	alert.matched_content
Type:	string
Displayed:	False

The content the alert matched.

Matched regexp

Name:	Matched regexp
Search filter:	alert.matched_regexp
Type:	string
Displayed:	False

The regular expression that matched the content.

Alert ID

Name:	Alert ID
Search filter:	alert.record_id
Type:	long
Displayed:	False

The identifier of the alert within the audit trail (.zat file).

Rule name

Name:	Rule name
Search filter:	alert.rule_name
Type:	string
Displayed:	False

The name of the content policy rule.

Alert time

Name:	Alert time
Search filter:	alert.time
Type:	date
Displayed:	False

The timestamp of the alert.

From API

Name:	From API
Search filter:	trail_download.from_api
Type:	boolean
Displayed:	False

The audit trail downloaded via API or not.

Trail download ID

Name:	Trail download ID
Search filter:	trail_download.id
Type:	string
Displayed:	False

The ID of an audit trail download event.

Download ip

Name:	Download ip
Search filter:	trail_download.ip_address
Type:	ip
Displayed:	False

The ip address from where the download is requested.

Download time

Name:	Download time
Search filter:	trail_download.time
Type:	date
Displayed:	False

The exact time when the user downloaded the audit trail file.

Downloader username

Name:	Downloader username
Search filter:	trail_download.username
Type:	string
Displayed:	False

The name of user who downloaded the audit trail of the session.

Commands indexed

Name:	Commands indexed
Search filter:	indexer_info.config.command.enabled
Type:	boolean
Displayed:	False

True if commands were extracted while indexing the session.

Keyboard buffering interval

Name:	Keyboard buffering interval
Search filter:	indexer_info.config.keyboard.buffer_interval
Type:	double
Displayed:	False

The buffering interval in milliseconds used when extracting keyboard events while indexing the session.

Keyboard extracted

Name:	Keyboard extracted
Search filter:	indexer_info.config.keyboard.enabled
Type:	boolean
Displayed:	False

True if keyboard events were extracted while indexing the session.

Mouse buffering interval

Name:	Mouse buffering interval
Search filter:	indexer_info.config.mouse.buffer_interval
Type:	double
Displayed:	False

The buffering interval in milliseconds used when extracting mouse events while indexing the session.

Mouse extracted

Name:	Mouse extracted
Search filter:	indexer_info.config.mouse.enabled
Type:	boolean
Displayed:	False

True if mouse events were extracted while indexing the session.

Near real-time indexing

Name:	Near real-time indexing
Search filter:	indexer_info.config.near_realtime
Type:	boolean
Displayed:	False

True if indexing this session was done near real-time (when the session was still active).

OCR languages

Name:	OCR languages
Search filter:	indexer_info.config.ocr_languages
Type:	string
Displayed:	False

The language configuration for optical character recognition used when indexing the session.

Screen content indexed

Name:	Screen content indexed
Search filter:	indexer_info.config.screen.enabled
Type:	boolean
Displayed:	False

True if screen content was extracted while indexing the session.

OCR tradeoff

Name:	OCR tradeoff
Search filter:	indexer_info.config.screen.omnipage_trade_off
Type:	string
Displayed:	False

The tradeoff used for optical character recognition when extracting screen content while indexing the session.

Titles indexed

Name:	Titles indexed
Search filter:	indexer_info.config.title.enabled
Type:	boolean
Displayed:	False

True if window titles were extracted while indexing the session.

Indexing error

Name:	Indexing error
Search filter:	indexer_info.error.message
Type:	string
Displayed:	False

The reason why indexing failed

Indexing cpu time

Name:	Indexing cpu time
Search filter:	indexer_info.statistics.cpu_time
Type:	long
Displayed:	False

The CPU time that indexing this session took in milliseconds.

Indexing duration

Name:	Indexing duration
Search filter:	indexer_info.statistics.duration
Type:	long
Displayed:	False

The duration of time that indexing this session took in milliseconds.

Indexing start time

Name:	Indexing start time
Search filter:	indexer_info.statistics.start_time
Type:	date
Displayed:	False

The time and date when indexing this session started.

Indexing status

Name:	Indexing status
Search filter:	indexer_info.status
Type:	string
Displayed:	False

Shows if the channel has been indexed successfully or not.

Indexer ADP version

Name:	Indexer ADP version
Search filter:	indexer_info.version.adp
Type:	string
Displayed:	False

The version of the audit data processor used for indexing the session

Indexer version

Name:	Indexer version
Search filter:	indexer_info.version.worker
Type:	string
Displayed:	False

The version of the indexer worker used for indexing the session

ZAC created

Name:	ZAC created
Search filter:	indexer_info.config.zac.enabled
Type:	boolean
Displayed:	False

True if an Audit Content file was created while indexing the session.

Screen content

Name:	Screen content
Search filter:	screen.content
Type:	string
Displayed:	False

Text that appeared on the screen in the session.

Channel id in trail

Name:	Channel id in trail
Search filter:	screen.channel_id_in_trail
Type:	long
Displayed:	False

The ID of the channel where this content appeared. To check the channel ID (channel_id), select a session and click details. Navigate to details > Channels and click the channel type.

Screen content creation time

Name:	Screen content creation time
Search filter:	screen.time
Type:	screen
Displayed:	False

The creation time of the indexed screen content.

Screen content ID

Name:	Screen content ID
Search filter:	screen.id
Type:	string
Displayed:	False

The ID of a screen content event.

trail_download

From API

Name:	From API
Search filter:	from_api
Type:	boolean
Displayed:	True

The audit trail downloaded via API or not.

Trail download ID

Name:	Trail download ID
Search filter:	id
Type:	string
Displayed:	True

The ID of an audit trail download event.

Download ip

Name:	Download ip
Search filter:	ip_address
Type:	ip
Displayed:	True

The ip address from where the download is requested.

Download time

Name:	Download time
Search filter:	time
Type:	date
Displayed:	False

The exact time when the user downloaded the audit trail file.

Downloader username

Name:	Downloader username
Search filter:	username
Type:	string
Displayed:	True

The name of user who downloaded the audit trail of the session.

Searching in the contents of audit trails

NOTE:

This feature is available only if auditing and content indexing was requested for the connection.

For more information, see Configuring the internal indexer on page 593.

You can search in the contents of the audit trails as follows:

- **From your browser**: Use this method to find all the sessions containing your search query.

  Enter the **screen.content: expression** search filter in the **Search query** field. For example: **screen.content="exit"**. The search returns all the sessions where **exit** was on the screen.

- **From the Safeguard Desktop Player application**: Use this method to find the exact location of the search query within a specific audit trail.

  Download the relevant audit trail, open it in the Safeguard Desktop Player application, and use the Search feature. You can also search in the contents of the audit trails for trails of graphical sessions created and indexed with One Identity Safeguard for Privileged Sessions (SPS) 6.0.

There are various ways you can refine your content query, you can:

- use wildcards
- use boolean expressions
- search in the commands of terminal connections (for example, `command:"sudo su"`)
- search in the window titles of graphical connections (for example, `title:settings`)

Search query examples

The following sections provide examples for different search queries.

- For examples of exact matches, see Searching for exact matches on page 695.
- For examples of using boolean operators to combine search keywords, see Combining search keywords on page 696.
- For examples of wildcard searches, see Using wildcard searches on page 697.
- For examples of searching with special characters, see Searching for special characters on page 699.
- For examples of fuzzy search that finds words with similar spelling, see Searching for fuzzy matches on page 701.
- For examples of proximity search to find words that appear within a special distance, see Proximity search on page 701.
- For examples of adjusting the relevance of a search term, see Adjusting the relevance of search terms on page 701.

For details on how to use more complex keyphrases that are not covered in this guide, see the Apache Lucene documentation.

Searching for exact matches

By default, One Identity Safeguard for Privileged Sessions (SPS) searches for keywords as whole words and returns only exact matches. Note that if your search keywords include special characters, you must escape them with a backslash (\) character. For details on special characters, see Searching for special characters on page 699. The following characters are special characters: `+ - & | ! ( ) { } [ ] ^ " ~ * ? : \ /`

Example: Searching for exact matches

Search expression	`example`
Matches	example
Does not match	examples example.com

query-by-example

exam

To search for an exact phrase, enclose the search keywords in double quotes.

Search expression	`"example command"`
Matches	example command
Does not match	example command example: command

To search for a string that includes a backslash characters, for example, a Windows path, use two backslashes (`\\`).

Search expression	`C\:\\Windows`
Matches	C:\Windows

Combining search keywords

You can use boolean operators – `AND`, `OR`, `NOT`, and `+` (required), `–` to combine search keywords. More complex search expressions can also be constructed with parentheses. If you enter multiple keywords,

Example: Combining keywords in search

Search expression	`keyword1 AND keyword2`
Matches	(returns hits that contain both keywords)

Search expression	`keyword1 OR keyword2`
Matches	(returns hits that contain at least one of the keywords)

Search expression	`"keyword1 keyword2" NOT "keyword2 keyword3"`
Matches	(returns hits that contain the first phrase, but not the second)

Search expression	`+keyword1 keyword2`
Matches	(returns hits that contain keyword1, and may contain keyword2)

To search for expressions that can be interpreted as boolean operators (for example: `AND`), use the following format: `"AND"`.

Example: Using parentheses in search

Use parentheses to create more complex search expressions:

Search expression	`(keyword1 OR keyword2) AND keyword3`
Matches	(returns hits that contain either keyword1 and keyword3, or keyword2 and keyword3)

Using wildcard searches

You can use the ? and * wildcards in your search expressions.

Example: Using wildcard ? in search

The ? (question mark) wildcard means exactly one arbitrary character. Note that it does not work for finding non-UTF-8 or multibyte characters. If you want to search for these characters, the expression ?? might work, or you can use the * wildcard instead.

You cannot use a * or ? symbol as the first character of a search.

Search expression	`example?`
Matches	example1 examples example?
Does not match	example.com

	example12 query-by-example
Search expression	`example??`
Matches	example12
Does not match	example.com example1 query-by-example

Example: Using wildcard * in search

The * wildcard means 0 or more arbitrary characters. It finds non-UTF-8 and multibyte characters as well.

Search expression	`example*`
Matches	example examples example.com
Does not match	query-by-example example*

Example: Using combined wildcards in search

Wildcard characters can be combined.

Search expression	`ex?mple*`
Matches	example1 examples example.com exemple.com

	example12
Does not match	exmples query-by-example

Searching for special characters

To search for the special characters, for example, question mark (`?`), asterisk (`*`), backslash (`\`) or whitespace ( ) characters, you must prefix these characters with a backslash (`\`). Any character after a backslash is handled as character to be searched for. The following characters are special characters: `+ - & | ! ( ) { } [ ] ^ " ~ * ? : \ /`

Example: Searching for special characters

To search for a special character, use a backslash (`\`).

Search expression	`example\?`
Matches	example?
Does not match	examples example1

To search for a string that includes a backslash characters, for example, a Windows path, use two backslashes (`\\`).

Search expression	`C\:\\Windows`
Matches	C:\Windows

To search for a string that includes a slash character, for example, a UNIX path, you must escape the every slash with a backslash (`\/`).

Search expression	`\/var\/log\/messages`
Matches	/var/log/messages

Search expression	`\(1\+1\)\:2`
Matches	(1+1):2

Searching in commands and window titles

For terminal connections, use the `command:` prefix to search only in the commands (excluding screen content). For graphical connections, use the `title:` prefix to search only in the window titles (excluding screen content). To exclude search results that are commands or window titles, use the following format: `keyword AND NOT title:[* TO *]`.

You can also combine these search filters with other expressions and wildcards, for example, `title:properties AND gateway`.

Example: Searching in commands and window titles

Search expression	`command:"sudo su"`
Matches	`sudo su` as a terminal command
Does not match	`sudo su` in general screen content

Search expression	`title:settings`
Matches	`settings` appearing in the title of an active window
Does not match	`settings` in general screen content

To find an expression in the screen content and exclude search results from the commands or window titles, see the following example.

Search expression	`properties AND NOT title:[* TO *]`
Matches	`properties` appearing in the screen content, but not as a window title.
Does not match	`properties` in window titles.

You can also combine these search filters with other expressions and wildcards.

Search expression	`title:properties AND gateway`
Matches	A screen where `properties` appears in the window title, and `gateway` in the screen content (or as part of the window title).
Does not match	Screens where both `properties` and `gateway` appear, but `properties` is not in the window title.

Searching for fuzzy matches

Fuzzy search uses the tilde ~ symbol at the end of a single keyword to find hits that contain words with similar spelling to the keyword.

Example: Searching for fuzzy matches

Search expression	`roam~`
Matches	roams foam

Proximity search

Proximity search uses the tilde ~ symbol at the end of a phrase to find keywords from the phrase that are within the specified distance from each other.

Example: Proximity search

Search expression	`"keyword1 keyword2"~10`
Matches	(returns hits that contain keyword1 and keyword2 within 10 words from each other)

Adjusting the relevance of search terms

By default, every keyword or phrase of a search expression is treated as equal. Use the caret ^ symbol to make a keyword or expression more important than the others.

Example: Adjusting the relevance of search terms

Search expression	`keyword1^4 keyword2`
Matches	(returns hits that contain keyword1 and keyword2, but keyword1 is 4-times more relevant)

Search expression	`"keyword1 keyword2"^5 "keyword3 keyword4"`
Matches	(returns hits that contain keyword1 keyword2 and keyword3 keyword4, but keyword1 keyword2 is 5-times more relevant)

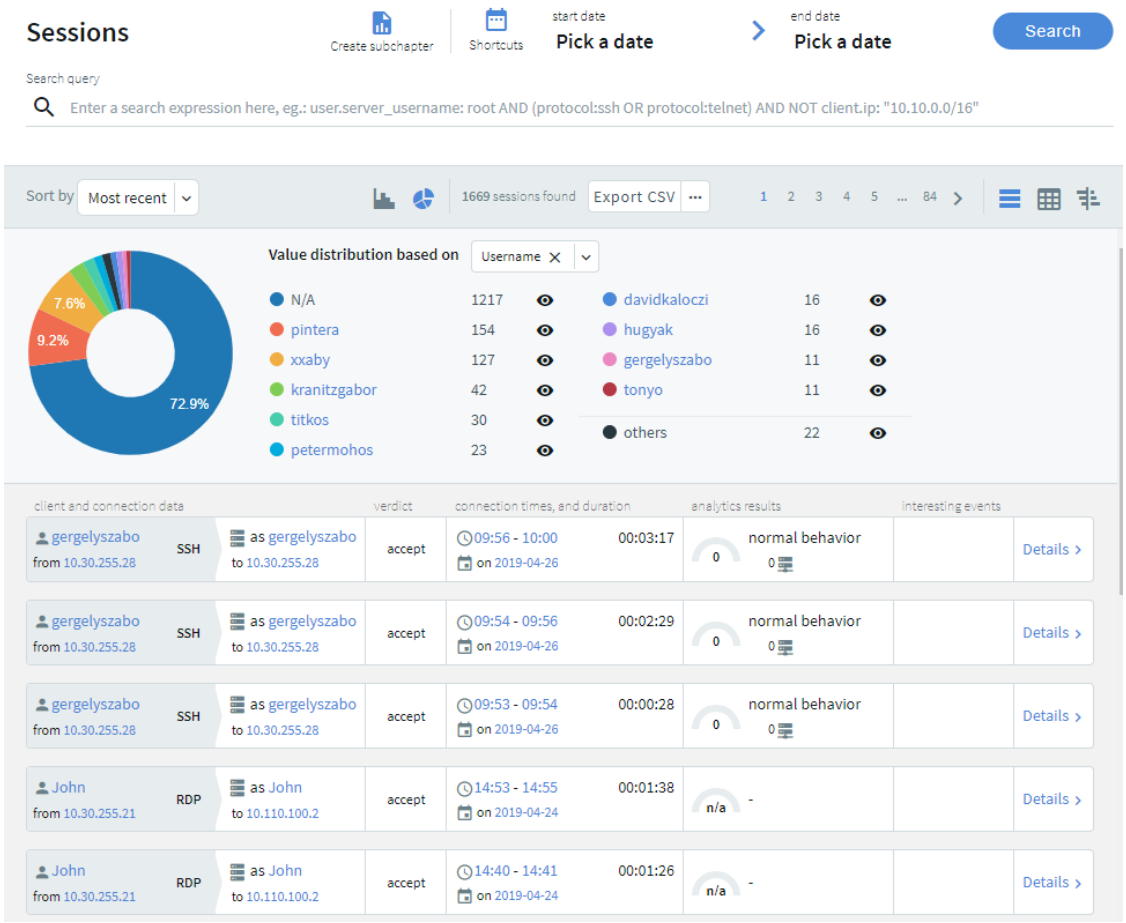
Displaying statistics on search results

You can quickly sort and visualize the distribution of the sessions based on their various metadata, for example, username, server address, and so on.

To display statistics on search results

1. Click the  icon.
2. Select the type of metadata you want to create statistics on from the **Value distribution based on** field, for example, select **Username** to display sessions based on username.

Figure 224: Search — Displaying statistics



3. To exclude items from the pie chart, click the  icon next to the metadata you want to exclude.

   For example, if you want to exclude results by a user called **testbot**, select the  icon next to the item.

Figure 225: Search — Excluding items from the pie chart



The pie chart now does not display results for the excluded item. The percentages always add up to 100%.

You can continue to restrict or refine your search results and view statistics as required.

Analyzing data using One Identity Safeguard for Privileged Analytics

One Identity Safeguard for Privileged Sessions (SPS) integrates data from SPS to use as the basis of user behavior analysis. SPA uses machine learning algorithms to scrutinize behavioral characteristics (using data from SPS), and generates user behavior profiles for each individual privileged user. SPA compares actual user activity to user profiles in real time, with profiles being continually adjusted using machine learning. When SPA detects unusual activity, this is indicated on the user interface of SPS in the form of high scores and visualized insight.

Prerequisites

Make sure that you have session data from network traffic that:

- contains real, unique usernames linked to users other than root/administrator or a shared account

  To check this, navigate to **Search**, and check whether the **Username** column contains data. This is important, because session data will be linked to users.

  If you do not have unique usernames in your session data, review your authentication settings and consult with the One Identity Professional Services team to learn about your options to tie accounts to users.

- has commands extracted (using lightweight or full indexing, or in real-time through content policies)

  For instructions on how to configure indexing and include commands in the scope of indexing, see "Indexing audit trails" in the Administration Guide.

  For details on how to configure real-time command extraction using a content policy, see "Creating a new content policy" in the Administration Guide.

- has keystrokes extracted (using lightweight or full indexing, or in real-time through content policies)

  The minimum required amount of data for reliable insight is 5 sessions with approximately 200 keystrokes each.

  For instructions on how to configure indexing and include typing biometrics in the scope of indexing, see "Indexing audit trails" in the Administration Guide.

  For details on how to configure real-time extraction of keystroke-related data using a content policy, see "Creating a new content policy" in the Administration Guide.

- has window titles extracted (using lightweight or full indexing, or in real-time through content policies)

  For instructions on how to configure indexing and include window titles in the scope of indexing, see "Indexing audit trails" in the Administration Guide.

  For details on how to configure real-time window title extraction using a content policy, see "Creating a new content policy" in the Administration Guide.

The following describes how to analyze data using One Identity Safeguard for Privileged Analytics.

Limitations

SPS used in combination with SPA currently has the following limitations:

- SPA requires at least 12GB RAM to operate. If you are interested in upgrading your appliance, contact our Support Team.
- SPA requires a lot of computation, which can put pressure on SPS:
  - The keystroke algorithm is much more resource-hungry than the other algorithms, therefore our recommendation is to start analyzing data using the algorithms that require less resources.
  - Before you start using SPA, make sure that at least half the capacity of SPS is available.
- SPA only analyzes audit trails and SPS metadata, it does not analyze log data.

To start using SPA

1. Start getting scores.

   Scoring happens in real-time, meaning that as soon as new data (even data from an ongoing session) is available, SPA immediately scores it.

   TIP:

   When data is not immediately available to you and you are unable to wait until sufficient amount of data comes in from production traffic, you can resort to the following:

   - Manually reindex historical sessions. For details, see "Reindex historical sessions" in the Safeguard for Privileged Analytics Configuration Guide.
   - Specifically for window title data, run the **pam-process-historical-window-titles** command to invoke window title processing for sessions that have been both closed and indexed.

     This can be useful, for example, when you have upgraded from a SPS version earlier than 5 F6 or you simply have never used the window title algorithm, and therefore SPS has not done any window title processing before.

Scores represent an aggregated amount. Session data is scored by multiple algorithms independent from each other. Scores given by individual algorithms are aggregated to create a single score.

For detailed instructions on how to configure SPA, see Safeguard for Privileged Analytics Configuration Guide.

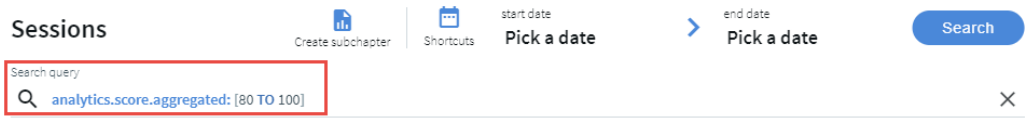
2. Search for sessions with high scores.
    a. Go to **Search**.

        Sessions are displayed sorted by date. For ongoing sessions, the Search interface is updated in real-time to always show the most up-to-date information.

    b. In the **Search query** field, type `analytics.score.aggregated: [80 TO 100]`, and click **Search**.

        A score between 80 and 100 indicates unusual user behavior.

Figure 226: Searching for sessions with unusual user behavior using a search query



Results that show sessions with high scores are displayed.

Figure 227: Sessions with high scores — table view

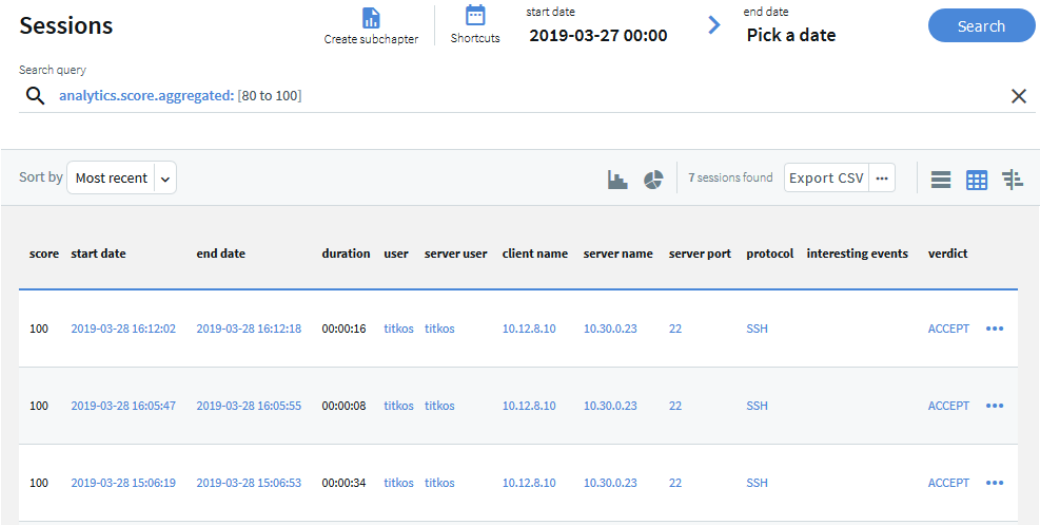
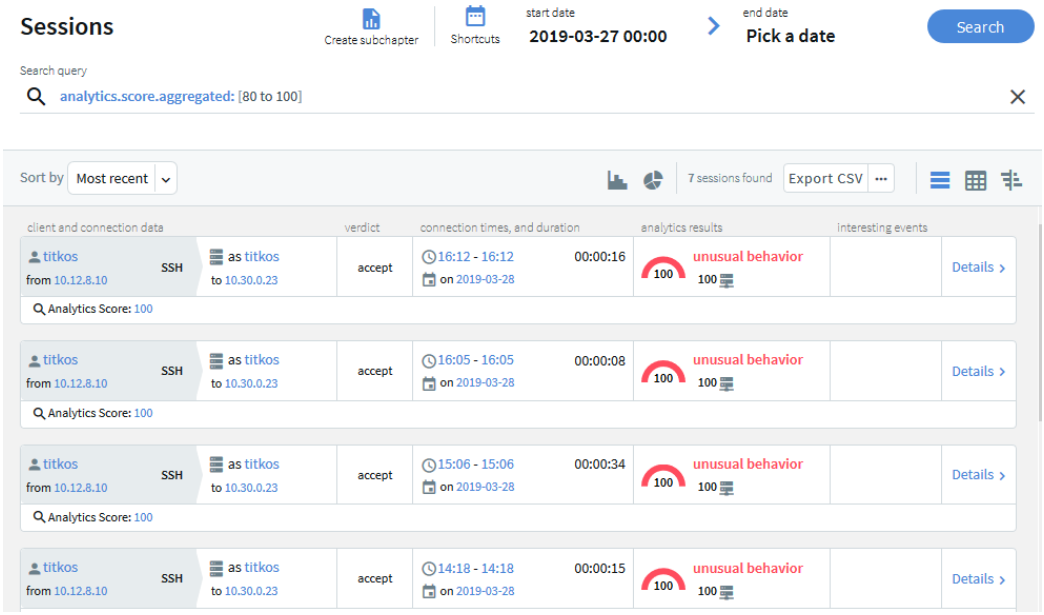


Figure 228: Sessions with high scores — card view



3. Alternatively, search for scripted sessions.

   In the **Search query** field, type `analytics.scripted:true`, and click **Search**.

4. To view details of a session, click Details › when you are in card view.

   Alternatively, click ••• when you are in table view.

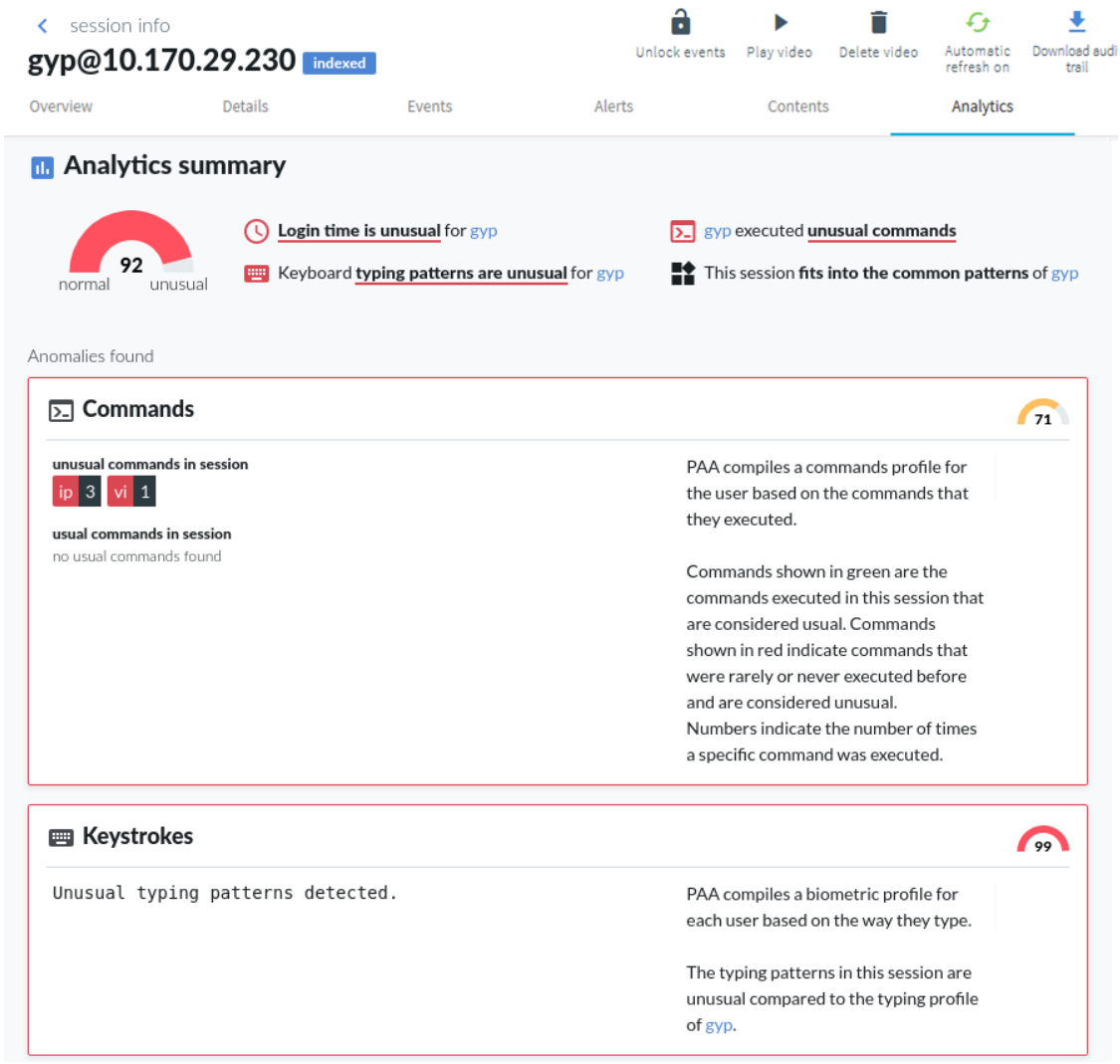
5. Click the **Analytics** tab.

   The top of the page displays a summary of key insights about the session, such as:

   - The aggregated score (indicated by a gauge). The following color codes are used:
     - Scores between 80-100 indicate unusual behavior, their color code is red.
     - Scores between 70-79 indicate behavior that might require further analysis and attention, their color code is amber.
     - Scores between 0-69 indicate normal behavior, their color code is gray.
   - A one-sentence summary of each algorithm's verdict about the session and user behavior.

   The **Anomalies found** and **Normal behavior** sections of the page display detailed analyses provided by each of the configured algorithms. This includes short information on how a particular algorithm works and how to read the visualized insight, as well as scores given by the individual algorithms.

Figure 229: Search — Viewing details on the Analytics tab: Anomalies found



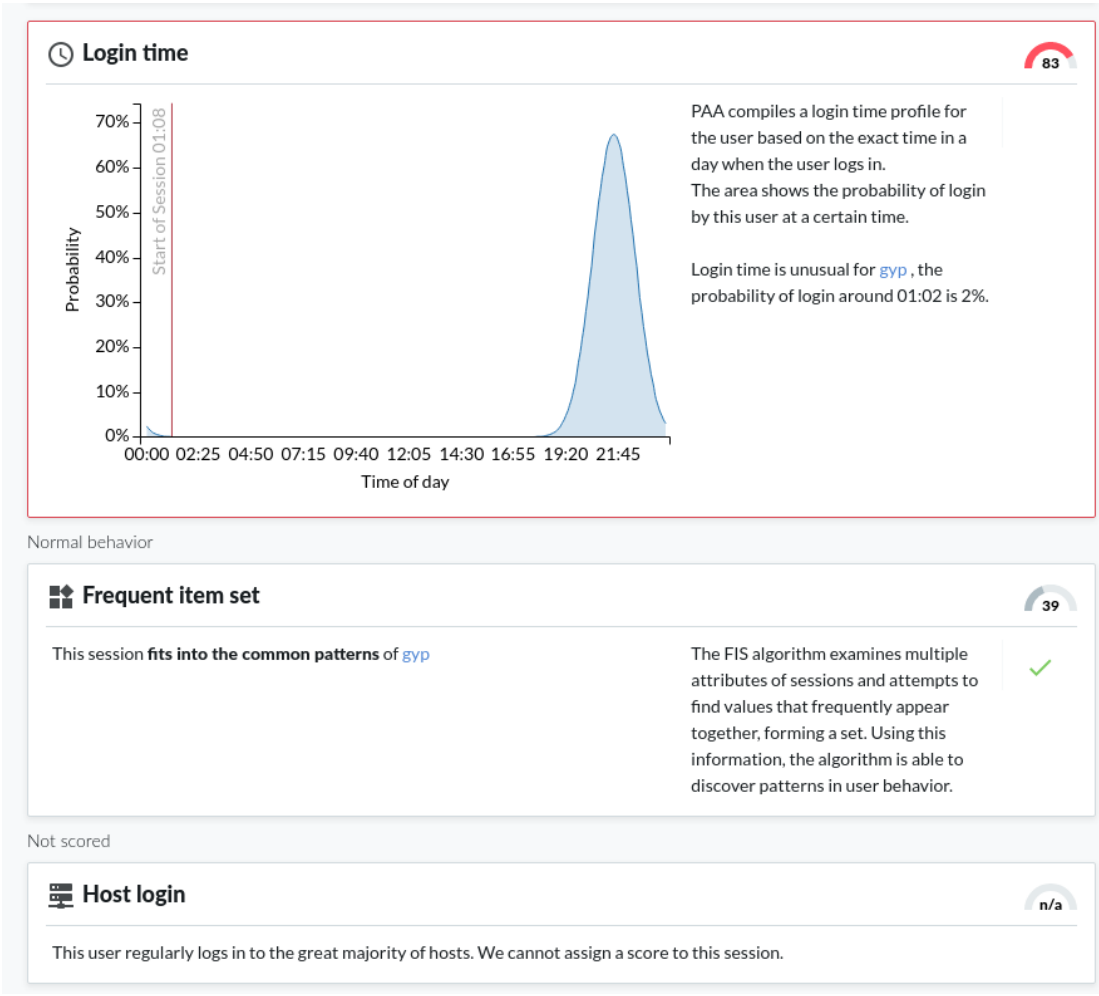
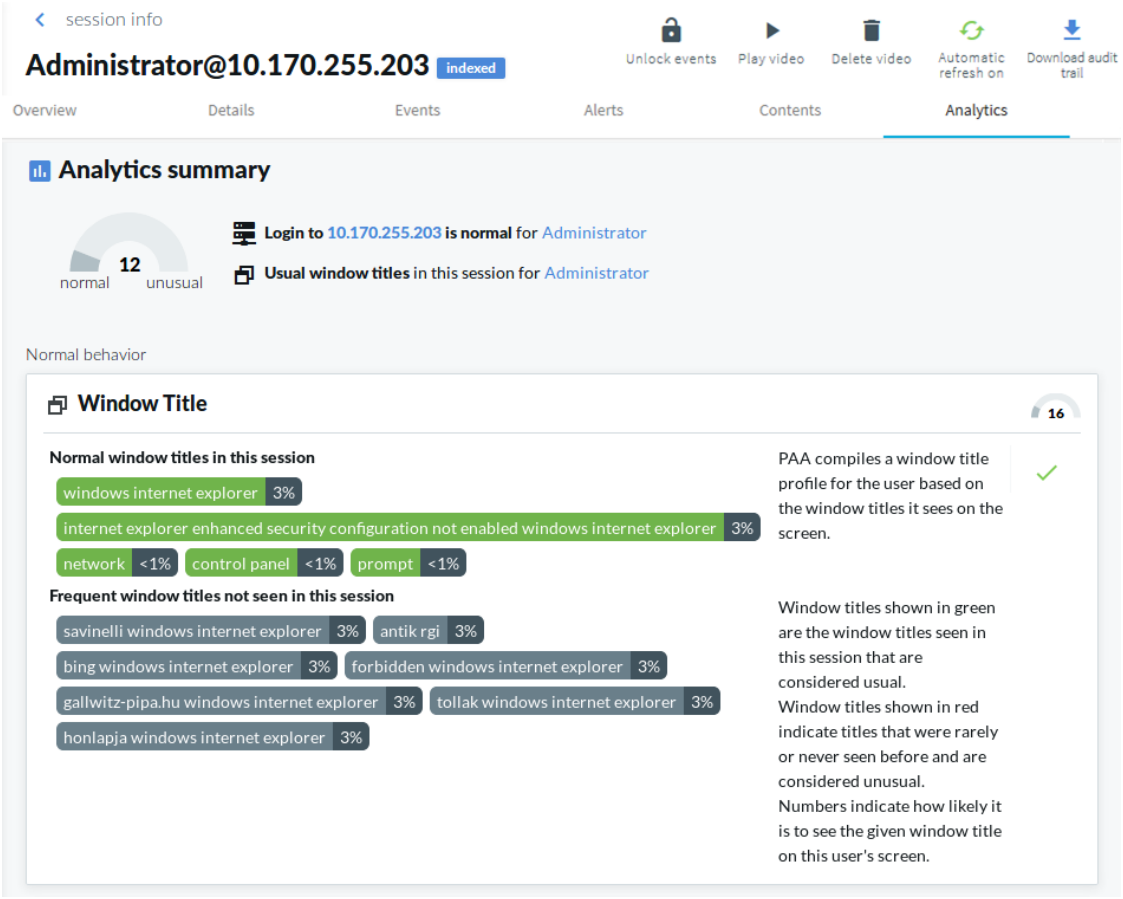
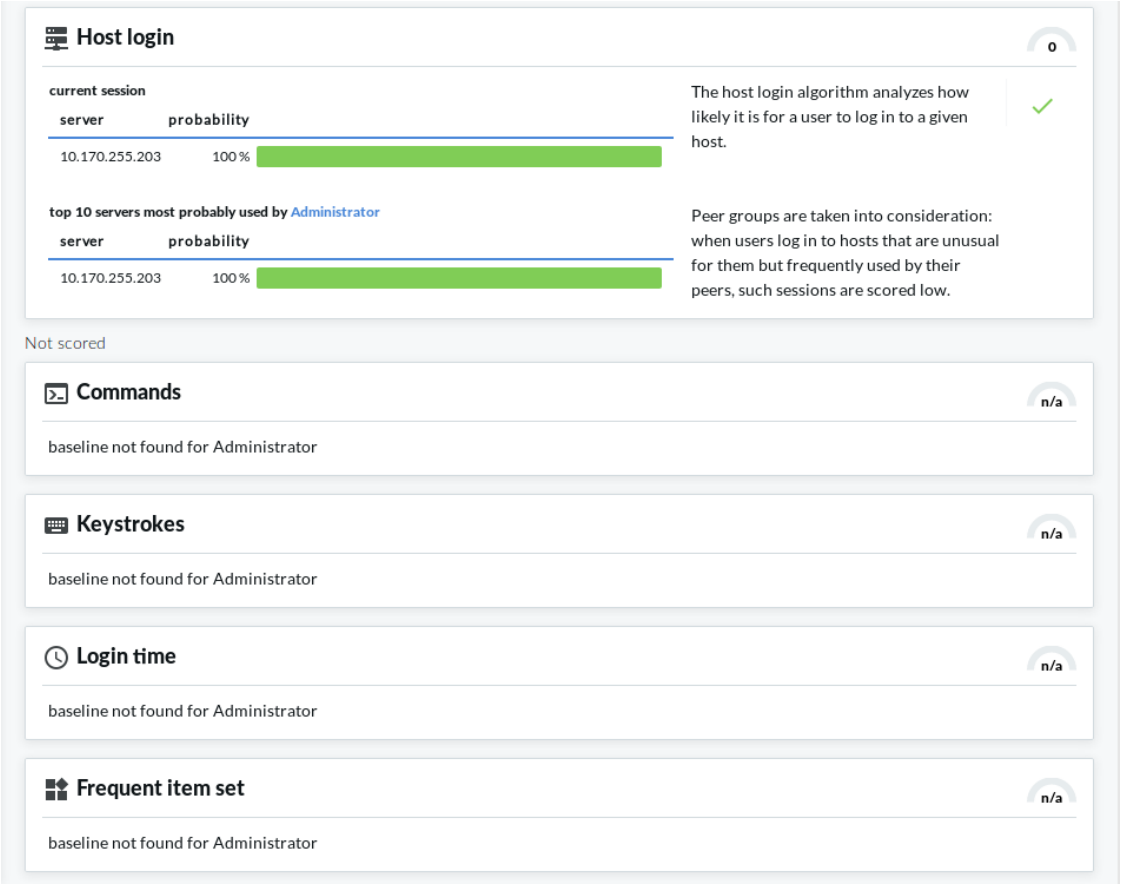

Login time
83
70%
60%
50%
40%
30%
20%
10%
0%
Probability
Start of Session 01:08
00:00 02:25 04:50 07:15 09:40 12:05 14:30 16:55 19:20 21:45
Time of day
PAA compiles a login time profile for the user based on the exact time in a day when the user logs in.
The area shows the probability of login by this user at a certain time.
Login time is unusual for gyp , the probability of login around 01:02 is 2%.
Normal behavior
Frequent item set
39
This session fits into the common patterns of gyp
The FIS algorithm examines multiple attributes of sessions and attempts to find values that frequently appear together, forming a set. Using this information, the algorithm is able to discover patterns in user behavior.
Not scored
Host login
n/a
This user regularly logs in to the great majority of hosts. We cannot assign a score to this session.

Figure 230: Search — Viewing details on the Analytics tab: Normal behavior

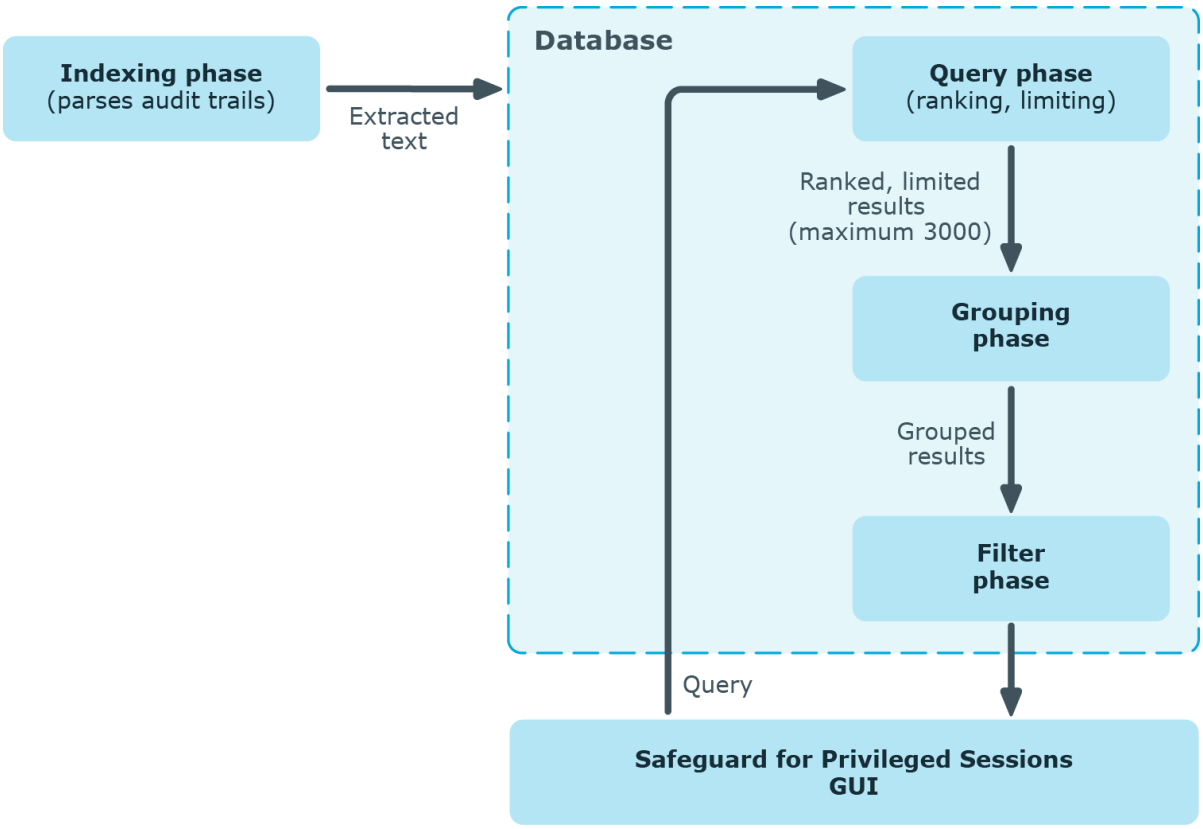




The search and filter process

The screen content is first indexed, then processed with the search backend, and finally, the filter expressions are applied. This process is described in detail in the following sections.

Figure 231: The search and filter process



Prerequisites - Indexing phase

First, as a prerequisite of the search process, screen content is indexed. The indexing phase generates a database that the search and filter processes will run on.

The indexer parses the audit trail files, and builds an "inventory" of the privileged user's activity data based on what appeared on their screen.

1. In the case of a terminal session, screen content corresponds to the activity data that is captured in a terminal window. In the case of graphical protocols, screen content is whatever is visible in the graphical user interface of the applications the user is interacting with. In the latter case, the indexer's Optical Character Recognition (OCR) engine extracts text that appeared on the screen (for example, window titles).

   NOTE:

   If a piece of text is displayed for less than 1 second, it is not extracted.

2. The indexer returns the information extracted from the parsed audit trail files to One Identity Safeguard for Privileged Sessions (SPS). In the case of a terminal session, the captured text is put in the backend database as one document per one second of screen content. Because of this, the content that you have searched for might only partially appear in the screenshot. In the case of graphical protocols, the captured

text is put in the backend database as one document per screenshot.

3. The queries will be run on this database during the search process.

For details on indexing, see Indexing audit trails on page 592.

Search and filter process phases

The search and filter process consists of three major phases:

- Query phase
- Grouping phase
- Filter phase

Query phase

In the query phase, the backend ranks and then limits the number of results.

1. The result of one query is the top 3000 documents, ordered by the default ranking system of the backend.

   This means that if there are more than 3000 results, those of the lowest rank will not be passed to the next phase at all.

   The ranking system cannot be modified, so there is no way to "upvote" those results of lower ranks.

   If you want to ensure that all important results are passed to the grouping phase, use a smaller time range that you run the query on. If there are fewer than 3000 results, it is certain that the events you are interested in will be included in the grouping phase.

2. The grouping phase receives the results.

Grouping phase

The grouping phase groups the results that were passed on from the query phase.

1. First, the results with the same trail IDs are grouped together. A trail ID group contains all search hits that are in that trail.
2. The trail ID groups are then further grouped by seach expression and time range. This group is essentially the time range during which the expression is displayed on the screen (for example, if the text `root` is displayed from 00:00:12 to 00:01:45, this will be one group).
3. This grouped result is displayed in the search screen as one row.

Filter phase

The filter phase applies filter expressions to these grouped results.

> NOTE:
>
> If there were screen content search results that were excluded during the query phase, the filter expressions will not be applied to them.

Example: Filtering for search results that were excluded in the query phase

For example, if you want to filter for Telnet connections where the text `root` was displayed, the following can happen:

You search for the **Screen content**: `root`. There are 3100 search results that consist of 3050 SSH connections and 50 Telnet connections. In this example, Telnet connections received the lowest ranks for some reason. 100 results that have received the lowest rank are excluded, and in this example it means all Telnet connections.

If you filter for protocol `Telnet` now, you will not see any results.

To remedy this situation, try searching in a smaller time range to make sure that there are less than 3000 search results. If you are unsure about the time range, you might want to attempt fine-tuning the backend search manually. For details, see: Fine-tuning the backend search manually on page 714.

Fine-tuning the backend search manually

You can fine-tune your search manually with the command line utility **lucenectl**. To do this, log on to the core shell. For details, see Accessing the One Identity Safeguard for Privileged Sessions (SPS) console on page 389.

- Specify more exact time ranges (use Unix timestamps).

  For example, to limit the time range to Thursday, June 30, 2016 11:39:51 AM - Thursday, November 3, 2016 2:44:46 PM, enter the following command:

  ```
  lucenectl search --from-to 1467286791 1478184286 --text remote --limit 3000 --aggregate-by-trail --normalize-rank
  ```

  NOTE:

  For converting timestamps to Unix timestamp, use https://www.epochconverter.com/.

- Increase the query limit of 3000 to a limit of your choice.

  For example, to increase the query limit of 3000 to 4500, enter the following command:

  ```
  lucenectl search --from-to 1467286791 1478184286 --text <your-screen-content-search-expression> --limit 4500 --aggregate-by-trail --normalize-rank
  ```

  ```
  lucenectl search --from-to 1467286791 1478184286 --text remote --limit 4500 --aggregate-by-trail --normalize-rank
  ```

NOTE:

If you do not receive more results with a larger query limit, it means that you have found all results with your search expression.

However, the downside of using **lucenectl** to fine-tune your search is that after the cli search, you have to manually extract the trails that you find interesting with the help of the metadb.

The following example shows the output of a lucenectl search:

```
{
  "hits": [
     {
        "hits_count": 1,
        "channel_id": 1,
        "trail_id": "58",
        "rank": 0.4068610216585047
     },
     {
        "hits_count": 7,
        "channel_id": 761,
        "trail_id": "12",
        "rank": 1.0
     },
     {
        "hits_count": 2,
        "channel_id": 1,
        "trail_id": "139",
        "rank": 0.5923645275802537
     }
  ]
}
```

- rank: the larger the number, the higher the rank
- hits_count: the number of times the screen content search expression is displayed in the audit trail
- trail_id: the ID of the trail
- channel_id: the ID of the channel

The most relevant audit trail will probably be the one with the highest rank.

If you have determined which audit trail you are interested in, enter the following command. The value of **_connection_channel_id** will be the value of the **trail_id** from the lucenectl output that you have determined as most relevant.

```
psql -U scb scb -c "select audit from channels where _connection_channel_
id = 12;"
```

The output of this command will be:

```
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:2
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:1
```

From this output, the audit trail file name path is as follows: `/<audittrailpath>/audit-scb_rdp-1467274538-0.zat`

NOTE:

If you cannot find the file at the path, check whether it has been archived and search for the file in the archive path. Use the following command:

```
psql -U scb scb -c "select audit, _archive_path  from channels where _
connection_channel_id = 12;"
```

The output of this command will be:

```
                    audit                                    | _archive_path
-------------------------------------------------------------+---------------
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:2 | /<myarchive/path>/
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:1 | /<myarchive/path>/
                                                             |
```

If you still cannot find the audit trail, contact our Support Team.

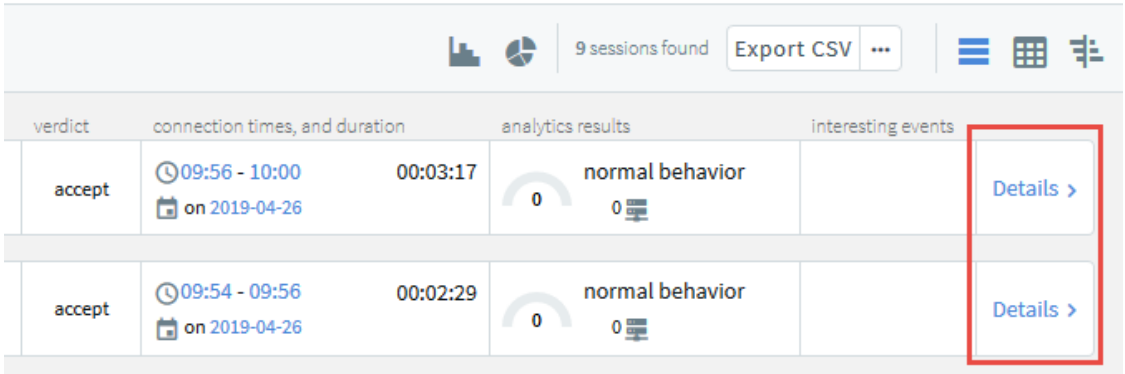
Viewing session details

The **session info** window provides in-depth information on each of the indexed session stored in the connection database. You can use it to gain contextual insight about the indexed session and its events.

Access the **session info** window in the following ways:

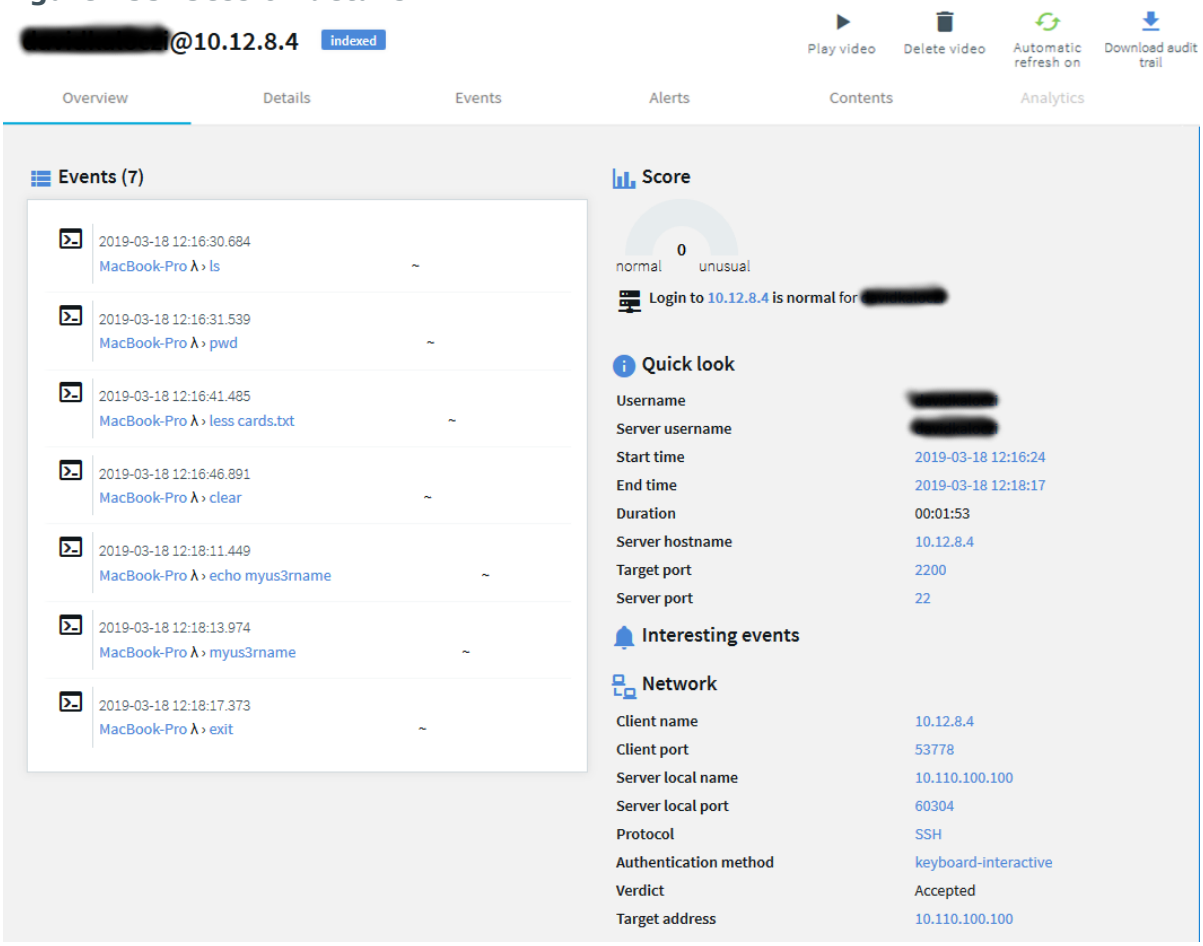
- In card view, click the Details › button in the last column of the relevant session.
- In table view, click the ••• button.

Figure 232: Search — Accessing session details



The **session info** window is displayed:

Figure 233: Session details



The **session info** window provides details about the sessions on tabs.

Overview tab

The **Overview** tab is divided into the following main areas:

- The *Events* area displays session events in chronological order.

  View the date and time of the event, the event type and event details. To filter events, use the **Events** tab.

- The *Score* area shows the risk score that the Analytics Module assigned to the session. Ranges from 0 to 100, 100 is the highest risk score.

- The *Quick look* area contains user information, for example, gateway and server username, start and end time of the session, and so on.

  The gateway username corresponds to the `Username` field of the connection metadata database, so note the following:

    - If the user performed inband gateway authentication in the connection, the field contains the username from the gateway authentication (gateway username).
    - Otherwise, the field contains the username used on the remote server.
- The *Interesting events* area displays events selected as interesting, for example, a list of commands and window titles from the session that could be interesting from a security point of view.

  The list of interesting events is currently hard-coded and cannot be modified. For terminal sessions it includes commands such as chmod, ssh, shutdown, sudo, su, mount, adduser, addgroup. For graphical sessions, it contains window titles such as Management Console, Control Panel, Server Manager, PowerShell, Security Settings, Windows Security Center.
- The *Network* area displays session information, for example, verdict, protocol, connection policy, client and server address.

Details tab

In addition to the *Quick look*, *Interesting events*, and *Network* screen areas, the **Details** tab provides monitoring information and channels information, too.

Events tab

The **Events** tab displays the:

- Session events in chronological order.

  You can narrow the event list by entering the event name in the **Filter events** field.
- Date and time of the event.
- Event type (command, screen content, window title).
- Event details.

Alerts tab

The **Alerts** tab displays the:

- Content policy alerts triggered in the session, in chronological order.

  You can narrow the alert list by entering the alert name in the **Filter alerts** field.
- Date and time of the alert.
- Alert details.
- Screenshots. If screenshots are available for the session, you can click each alert to view the corresponding screenshot.

Screenshots are generated for search results and alerts when the session is opened, and for subsequent searches.

Screenshots are not available for:

- Ongoing sessions.
- Unindexed sessions.
- Trails of HTTP sessions.
- Encrypted trails (without the necessary certificate).

Contents tab

You can search in the contents of the audit trails as follows:

- **From your browser**: Use this method to find all the sessions containing your search query.

  Enter the **screen.content: expression** search filter in the **Search query** field. For example: **screen.content="exit"**. The search returns all the sessions where **exit** was on the screen.
- **From the Safeguard Desktop Player application**: Use this method to find the exact location of the search query within a specific audit trail.

  Download the relevant audit trail, open it in the Safeguard Desktop Player application, and use the Search feature. You can also search in the contents of the audit trails for trails of graphical sessions created and indexed with One Identity Safeguard for Privileged Sessions (SPS) 6.0.

For more information, see Searching in the contents of audit trails.

Analytics tab

If you use the One Identity Safeguard for Privileged Analytics, you can view detailed analyses provided by the configured algorithms. For more information, see Analyzing data using One Identity Safeguard for Privileged Analytics on page 704.

Viewing active connections

If a connection is not closed and is still active, the 

label is displayed in the **session info** window.

To close an active connection, click the 

label.

You can also view the live connection as follows:

1. Click the 

label.
2. Download the audit trail.

   Trail data is exported in `.srs` format, which you can open with the Safeguard Desktop Player application.

For more information on the Safeguard Desktop Player, see Safeguard Desktop Player User Guide.

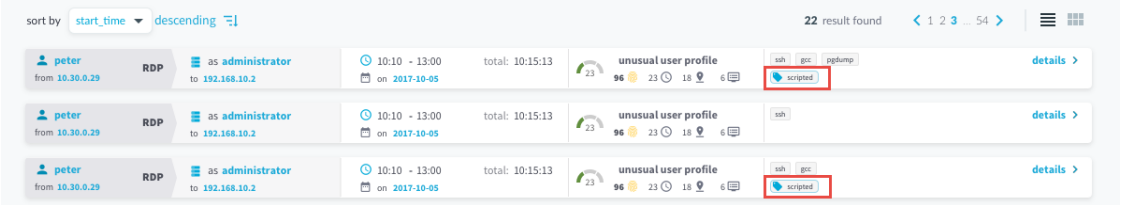
Session tags

Session tags allow you to get basic information about the session and its contents at a glance.

Scripted session tag: One Identity Safeguard for Privileged Sessions (SPS) currently supports the scripted session tag. SPS uses One Identity Safeguard for Privileged Analytics to detect if sessions are generated using human interaction or automation. If sessions are generated using automation, SPS displays the scripted tag in the search interface as shown below:

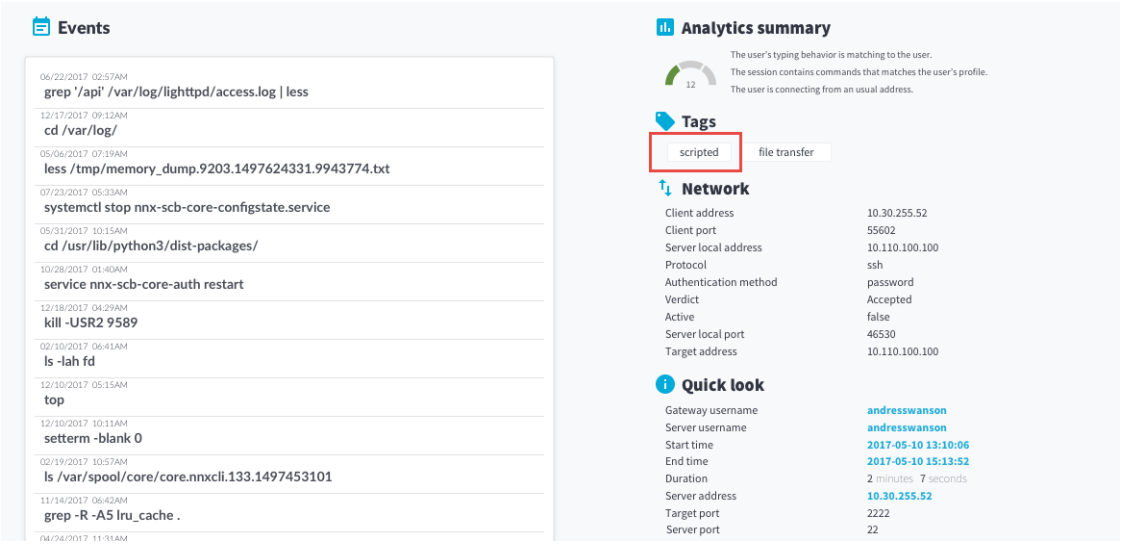
- Scripted sessions are shown on the main search screen.

Figure 234: Scripted sessions — cards view



- Scripted sessions are shown on the **Overview** tab.

Figure 235: Scripted sessions — Overview tab



Replaying audit trails in your browser

The following describes how to create and replay an audit trail in your browser.

CAUTION:

You can replay audit trails in your browser, or using the Safeguard Desktop Player application. Note that there are differences between these solutions.

	Browser	Safeguard Desktop Player
Works without installation	✓	-
Works on any operating system	✓	Windows, Linux, Mac
Can replay audit trails recorded with SPS 5 F4 and newer	✓	✓
Can replay TN5250 sessions	✓	✓
Can extract files from SCP, SFTP, HTTP and RDP sessions	-	✓
Can replay HTTP sessions	-	Only exports raw files from the command line
Can replay X11 sessions	✓	✓
Can start replay while rendering is in progress	-	✓
Can follow 4-eyes connections	-	✓
Can replay live streams in follow mode	-	✓
Can export to PCAP	-	✓
Can display user input	✓	✓
Can display subtitles for video	-	✓
Export audit trail as video	-	✓
Export screen content text	-	✓
Can search in the contents of the audit trails	-	✓

For details on the Safeguard Desktop Player application, see Safeguard Desktop Player User Guide.

CAUTION:

Even though the One Identity Safeguard for Privileged Sessions (SPS) web interface supports Internet Explorer and Microsoft Edge in general, to replay audit trails you need to use Internet Explorer 11, and install the Google WebM Video for Microsoft Internet Explorer plugin. If you cannot install Internet Explorer 11 or another supported browser on your computer, use the the Safeguard Desktop Player application. For details, see "Replaying audit trails in your browser" in the Administration Guide and Safeguard Desktop Player User Guide.

To replay an audit trail in your browser

1. On the **Search** page, find the audit trail you want to replay.
2. Click Details › to display the details of the connection.

   Alternatively, in the table view, click •••.

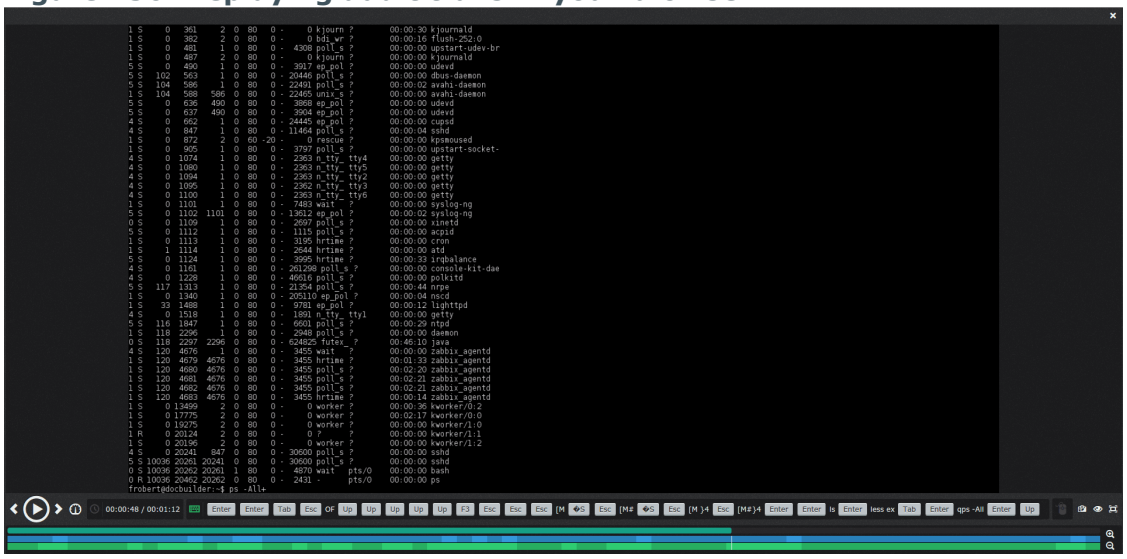
   

3. Click Start rendering to generate a video file from the audit trail you want to replay. Depending on the load of the indexer and the length and type of the audit trail, this can take several minutes.

   When the video is available, Start rendering changes to Play video and Delete video. You can use the Delete video button if you want to remove the generated video. After you remove the video file, the Start rendering button is available and you can use it to recreate the video file.
4. (Optional) If you have encrypted audit trails but the necessary certificates and private keys are not uploaded into your private keystore, you have to upload the keys first. After uploading them, click Unlock events. The Unlock events feature decrypts the encrypted upstream traffic elements. As a result, they will be displayed distributed in the generated video (see *List of keyboard events*, *Show / hide events*, and both versions of the *Progress bar* in the The Player window has the following controls: below).
5. To replay the video, click Play video.

   The Player window opens.

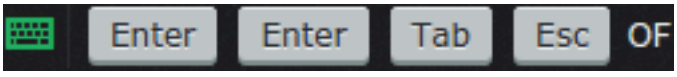
Figure 236: Replaying audit trails in your browser



TIP:

You can quickly zoom in or out by clicking anywhere in the Player window.

The Player window has the following controls:

-  : *Play*, *Pause*
-  : *Jump to previous event*, *Jump to next event*
-  : *Adjust replay speed*
-  : *Time since the audit trail started / Length of the audit trail*. Click on the time to show the date (timestamp) of the audit trail.
-  : *List of keyboard events*. Special characters like ENTER, F1, and so on are displayed as buttons. If the upstream traffic is encrypted, upload your permanent or temporary keys to the **User menu > Private keystore** to display the keyboard events. This will not be displayed if your upstream traffic is encrypted but not unlocked.
-  : *Active mouse button*
-  : *Create a screenshot*

-  : *Show / hide events*. Select the types of events to display. Depending on the protocol used and how the audit trail was processed, One Identity Safeguard for Privileged Sessions (SPS) can display keyboard events, commands, mouse events, and window titles. Commands and window titles are displayed as subtitles at the top of the screen. This will not be displayed if your upstream traffic is encrypted but not unlocked.
-  : *Fullscreen mode*
-  : Progress bar
-  : Shows the distribution of events. Blue - commands, green - keyboard events, yellow - mouse events, orange - window title. This will not be displayed if your upstream traffic is encrypted but not unlocked.
-  : *Close the player*, and return to the Connection details page.

Replaying encrypted audit trails in your browser

To view screenshots generated for encrypted audit trails and replay encrypted audit trails in your browser, you have to upload the necessary certificates and corresponding private keys to your private keystore. Depending on the encryption, decrypting the upstream part of an audit trail may require an additional set of certificates and keys.

Only RSA keys (in PEM-encoded X.509 certificates) can be uploaded to the private keystore.

> NOTE:
>
> Certificates are used as a container and delivery mechanism. For encryption and decryption, only the keys are used.

One Identity recommends using 2048-bit RSA keys (or stronger).

Use every keypair or certificate only for one purpose. Do not reuse cryptographic keys or certificates (for example, do not use the certificate of the One Identity Safeguard for Privileged Sessions (SPS) webserver to encrypt audit trails, or the same keypair for signing and encrypting data).

For more information on audit trail encryption, see Encrypting audit trails on page 465 and for more information about replaying audit trails in your browser, see Replaying audit trails in your browser.

You can upload certificates permanently or temporarily. The temporary certificates are deleted when you log out of SPS.

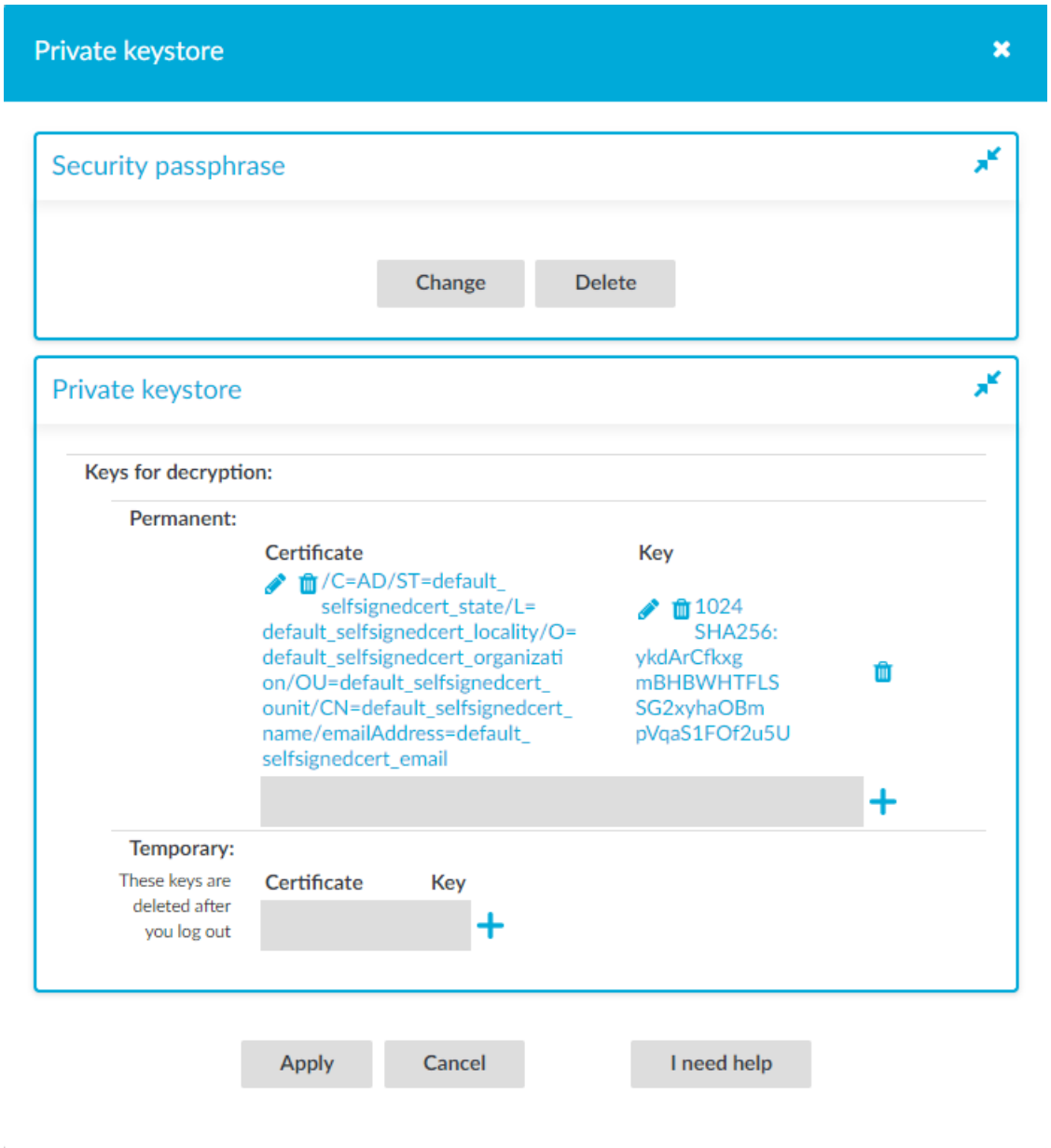
The certificates and private keys in your keystore can be protected with a passphrase. To use the certificates and private keys in a passphrase-protected keystore for decrypting

audit trails, you have to unlock the keystore first by providing the security passphrase. The keystore then remains unlocked for the duration of your session.

To replay encrypted audit trails in your browser

1. Click on **User Menu > Private keystore**.

Figure 237: User Menu > Private keystore — The private keystore



2. (Optional) Create a security passphrase, if you have not configured one yet.

a. In **Security passphrase**, click **Change**.

b. In the **New:** field, enter your new security passphrase. Repeat the same passphrase in the **Confirm:** field.

> NOTE:
>
> SPS accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

c. Click **Apply**.

If you forgot your security passphrase, contact our Support Team.

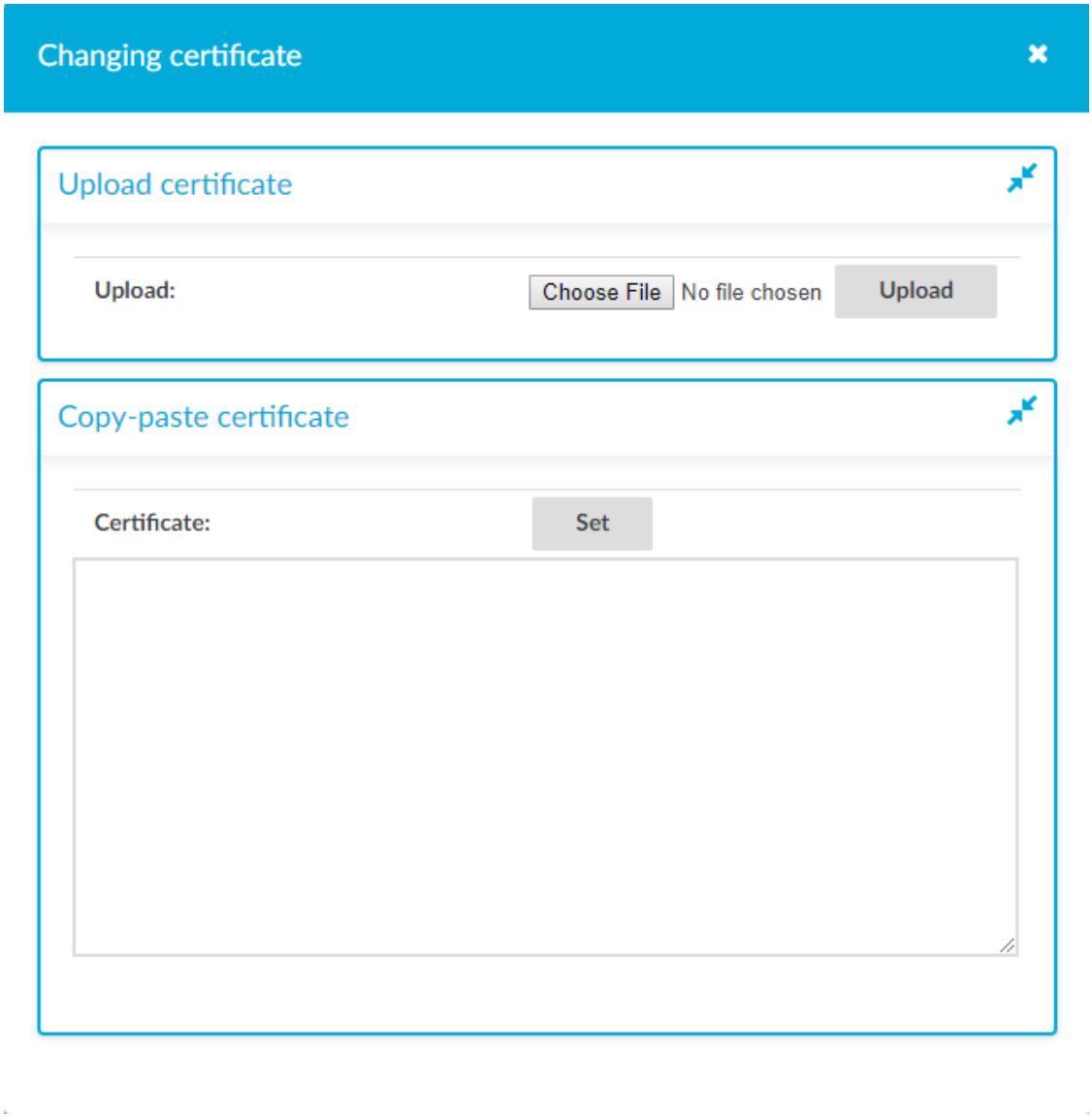
3. Click + to add a new certificate. A new empty row is added.

Figure 238: Adding certificates



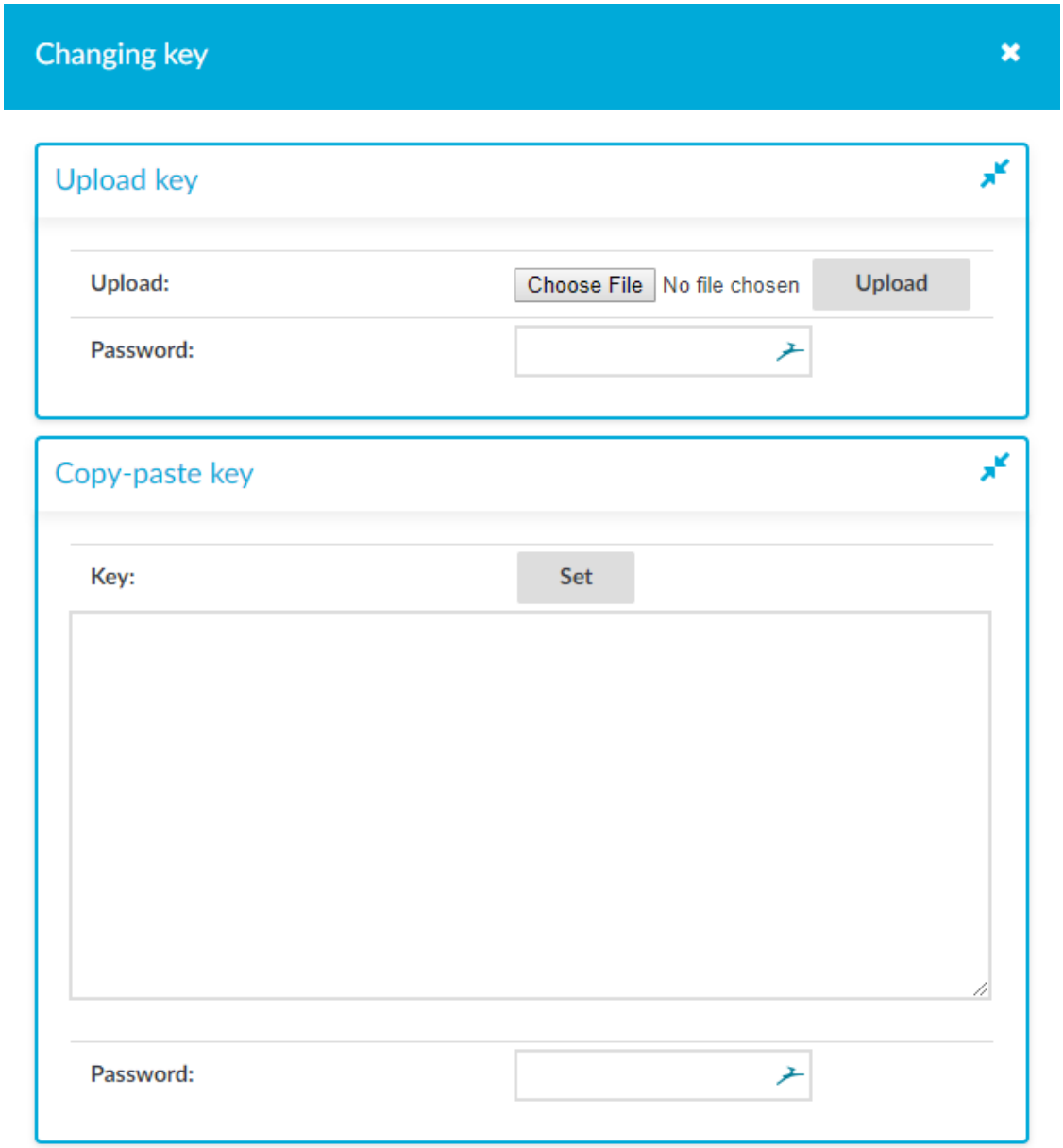
4. Click the first (under **Certificate**) to upload the new certificate. A pop-up window with the header **Changing certificate** is displayed.

Figure 239: Uploading certificates



5. Click **Choose File**, select the file containing the certificate, and click **Upload**. Alternatively, you can also copy-paste the certificate into the **Certificate** field and click **Set**.
6. To upload the private key corresponding to the certificate, click the second icon (under **Key**). A pop-up window with the header **Changing key** is displayed.

Figure 240: Uploading the private key



7. Click **Choose File**, select the file containing the private key, provide the **Password** if the key is password-protected, and click **Upload**. Alternatively, you can also copy-paste the private key into the **Key** field, provide the **Password** there, and click **Set**.
8. To add more certificate-key pairs, click + and repeat the steps above.
9. To finish uploading certificates and keys to your private keystore, click **Apply**.

Creating report subchapters from search queries

NOTE:

Creating report subchapters from search queries is currently an experimental feature of One Identity Safeguard for Privileged Sessions (SPS), therefore One Identity recommends that only administrators use this feature and only at their own risk.

You can turn any search query or statistics into a subchapter to add to your reports. This is an easy and flexible way of creating reports to monitor traffic, track certain parameters, or get alerted about particular events. The Search interface allows you to:

- Create search-based report subchapters from search results.
- Create search-based report subchapters from scratch.

Creating search-based report subchapters from search results

NOTE:

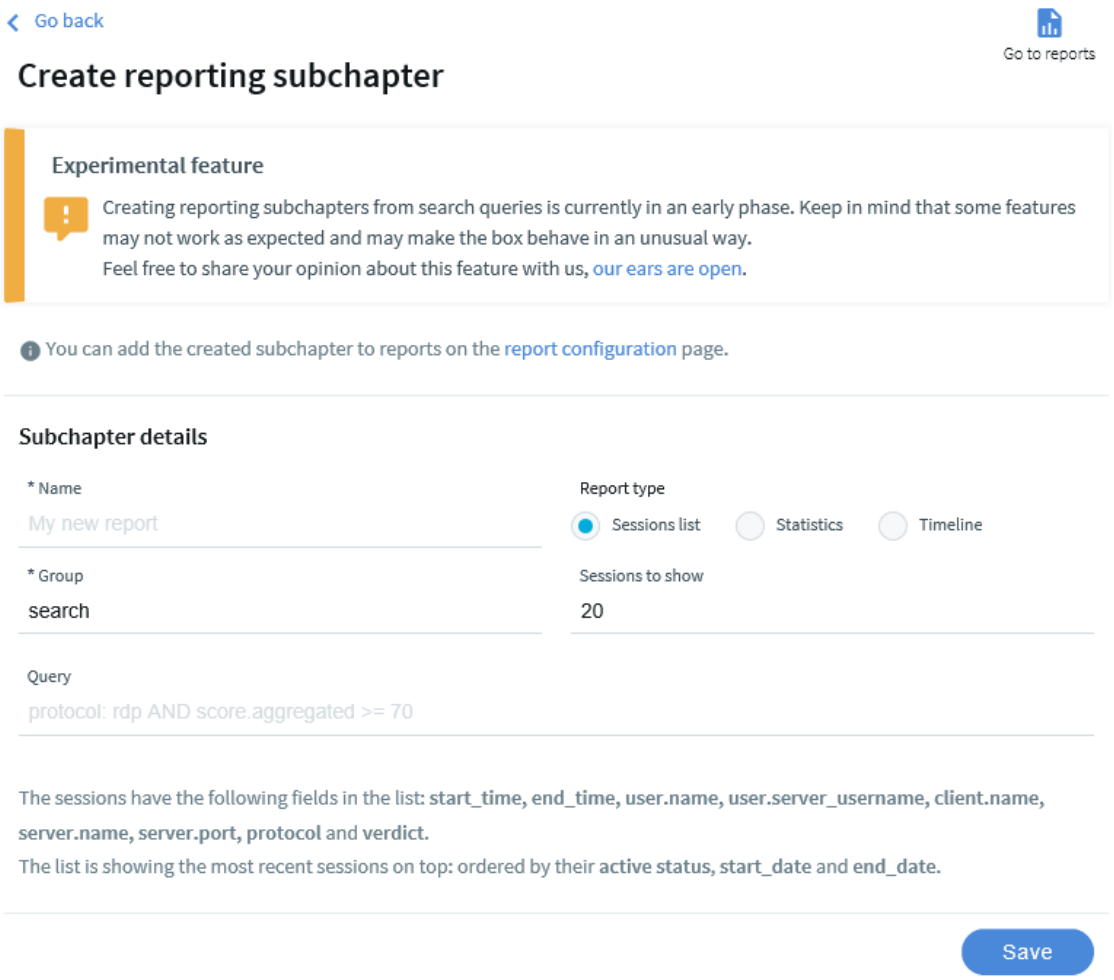
Creating report subchapters from search queries is currently an experimental feature of One Identity Safeguard for Privileged Sessions (SPS), therefore One Identity recommends that only administrators use this feature and only at their own risk.

The following describes how to create a search-based report subchapter from search results.

To create a search-based report subchapter from search results

1. Navigate to **Search**, and perform a query of your choice.
2. Click **Search**. Search results are displayed.
3. Click 

. The **Create reporting subchapter** page is displayed, with the **query** field populated with your query.



4. In the **name** field, add a name to your report.
5. In **Report type**, select the type that fits your query. You can choose from the following types:
   - **Sessions list**: Displays a list of sessions.

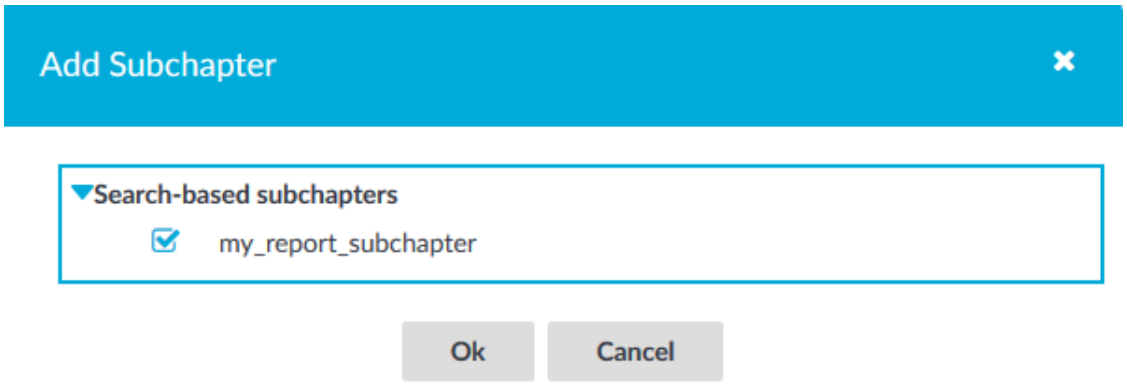
     Set the number of **sessions to show** in the report as required.
   - **Statistics**: Visualizes the distribution of sessions based on the selected metadata.

     Select a **Statistic presentation** for your report, such as **Pie chart**, **List**, **Bar chart**. Select the field (the metadata) to create your statistics on.
   - **Timeline**: Visualizes the distribution of sessions within a day/week/month, depending on the time range chosen for the report under **Reporting > Configuration > Generate this report every > Day/Week/Month**.
6. Click **Save**.
7. Click 

. Alternatively, navigate to **Reporting > Configuration**.

8. Configure a custom report from scratch, or add the subchapter to an existing report. For details, see Configuring custom reports on page 778.

   When adding the subchapter you created, look for it under **Search-based subchapters**.



Creating search-based report subchapters from scratch

> NOTE:
>
> Creating report subchapters from search queries is currently an experimental feature of One Identity Safeguard for Privileged Sessions (SPS), therefore One Identity recommends that only administrators use this feature and only at their own risk.

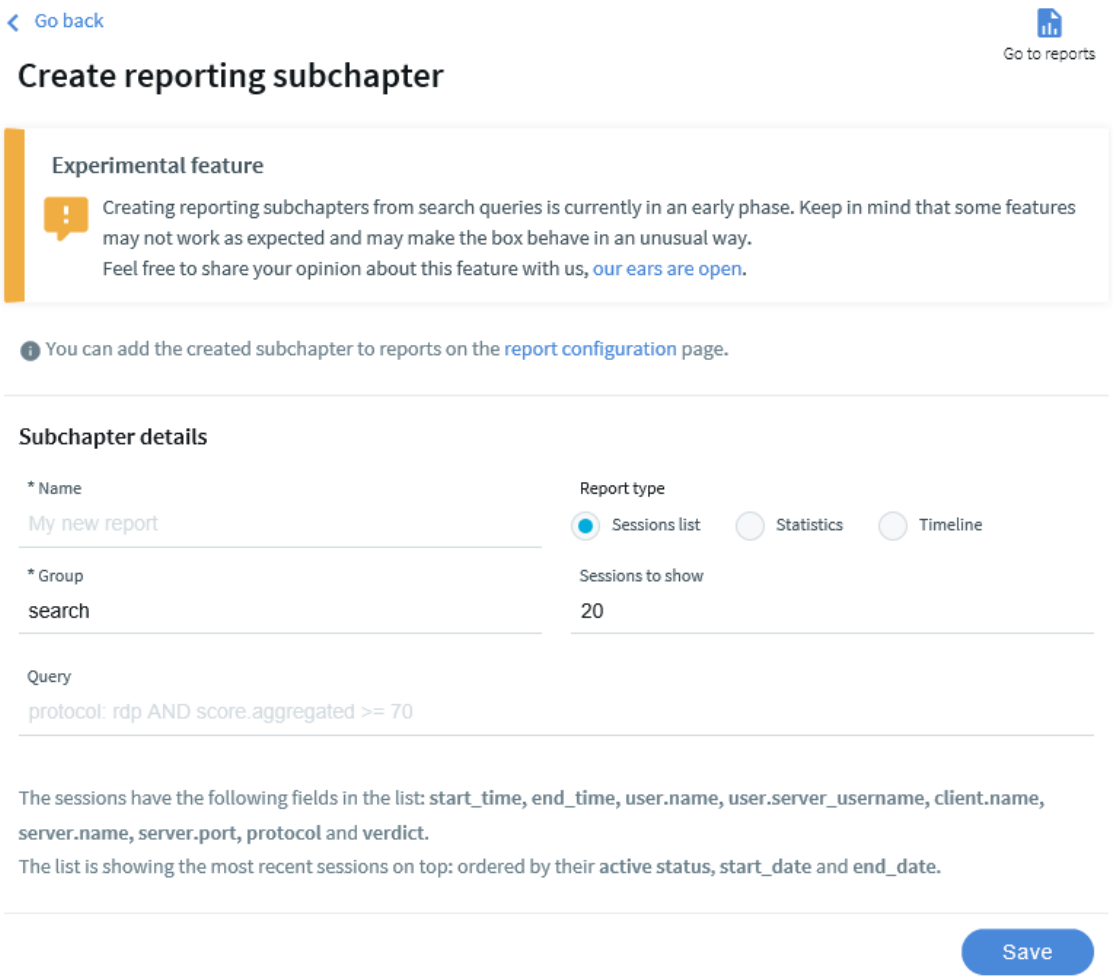
The following describes how to create a search-based report subchapter from scratch.

To create a search-based report subchapter from scratch

1. If you have multiple SPS appliances and they are organized into a cluster where one of the nodes is the Search Master (or Central Search) node, log in to that node.
2. Navigate to **Reporting > Search subchapters**.

3. Click + Add new. The **Create reporting subchapter** page is displayed.

Figure 241: Reporting > Search subchapters — Create reporting subchapter



4. In the **name** field, add a name to your report.
5. In the **query** field, type the query that you want to create a report from.
6. In **Report type**, select the type that fits your query. You can choose from the following types:
   - **Sessions list**: Displays a list of sessions.

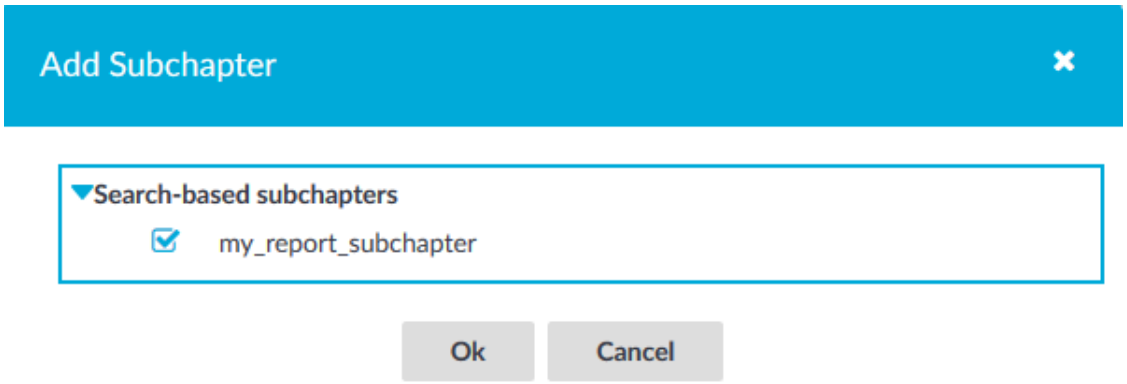
     Set the number of **sessions to show** in the report as required.
   - **Statistics**: Visualizes the distribution of sessions based on the selected metadata.

     Select a **Statistic presentation** for your report, such as **Pie chart**, **List**, **Bar chart**. Select the field (the metadata) to create your statistics on.

- **Timeline**: Visualizes the distribution of sessions within a day/week/month, depending on the time range chosen for the report under **Reporting > Configuration > Generate this report every > Day/Week/Month**.

7. Click **Save**.
8. Click Go to reports. Alternatively, navigate to **Reporting > Configuration**.
9. Configure a custom report from scratch, or add the subchapter to an existing report. For details, see Configuring custom reports on page 778.

   When adding the subchapter you created, look for it under **Search-based subchapters**.



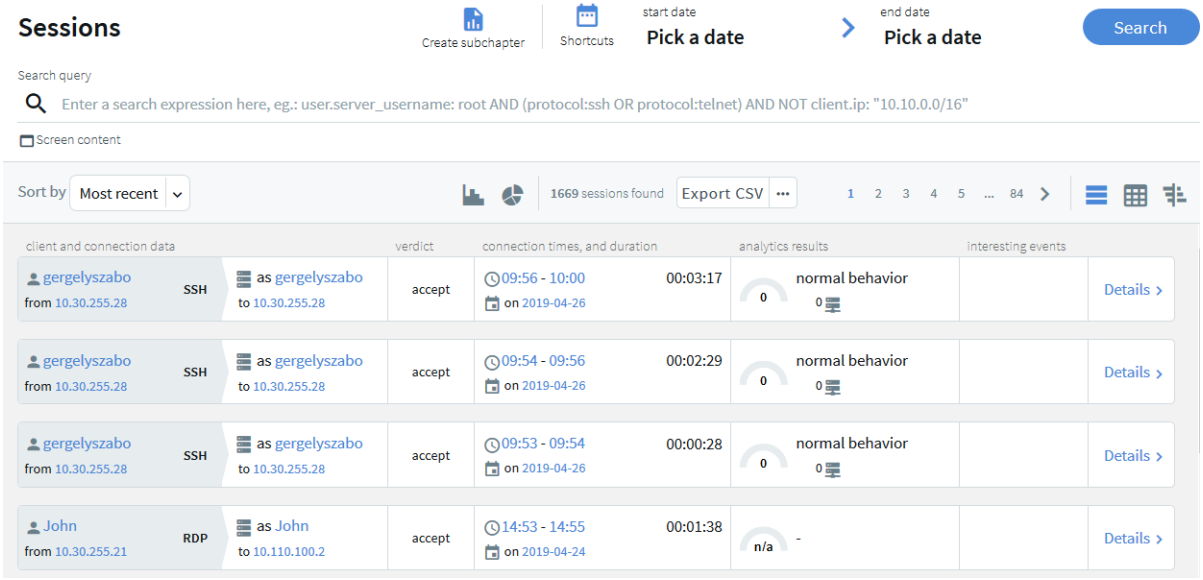
Search interface changes between version 5.0 and 6.0

Between versions 5 LTS (5.0) and 6.0 of One Identity Safeguard for Privileged Sessions (SPS), we have completely redesigned the **Search** page, and improved it with several new features. This section highlights the most important changes, and helps you find how to do the common search tasks on the new page. For the detailed documentation of the new **Search** page, see Using the Search interface on page 622.

Table view, card view

In addition to listing sessions and search results as a table, the new card view highlights the most important details of a session at a glance.

Figure 242: Search interface improvements

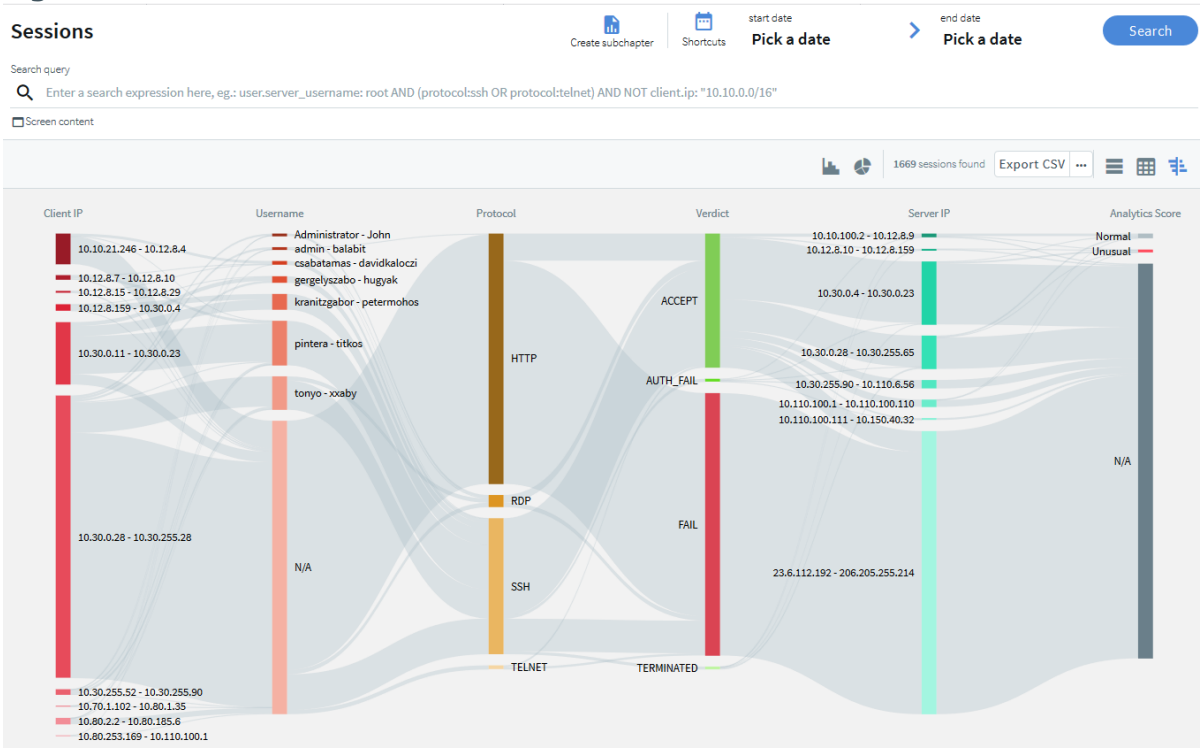


Note that in table view now the list of displayed columns is fixed and cannot be modified. However, if you search for specific values of fields that are not displayed, the values of these fields will be visible in card view.

Quick session analytics with the flow view

Display an interactive, visual overview of your search results to quickly visualize their distribution along multiple attributes, such as client and target IP addresses, protocol, or usernames. Helps to identify patterns in user behavior and to drill down fast to the most relevant sessions. For details, see Using the Search interface on page 622.

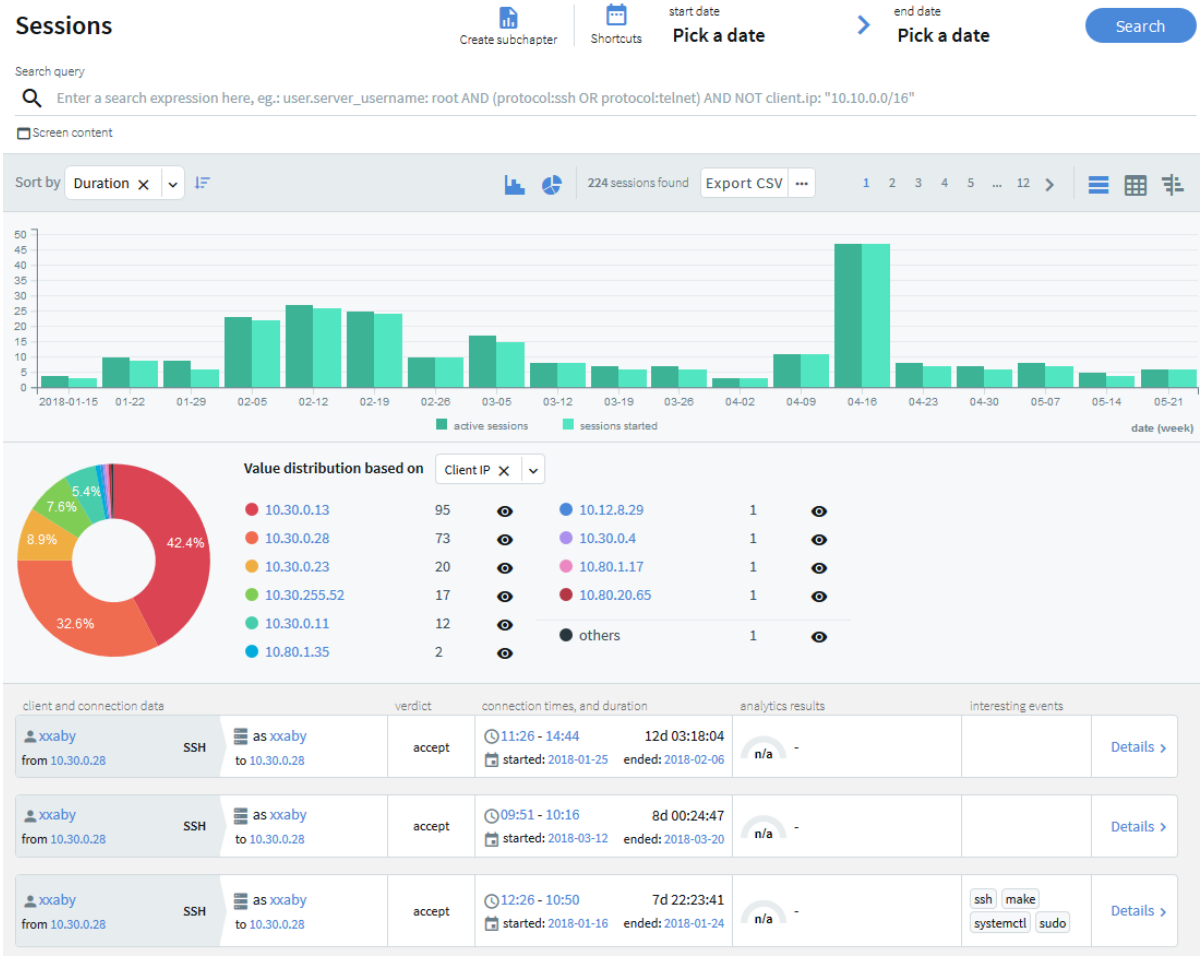
Figure 243: Search — Flow view



Timeline

The Search interface can now display a timeline showing the search results. Also, you can quickly sort and visualize the distribution of the sessions based on their various metadata, for example, username, server address, and so on.

Figure 244: Search — Displaying statistics and timeline



Set a custom or preset date range

Specify a time range to restrict or filter your search criteria by setting boundaries on your searches. Use one of the preset time ranges, or use a custom time range for a more specific search.

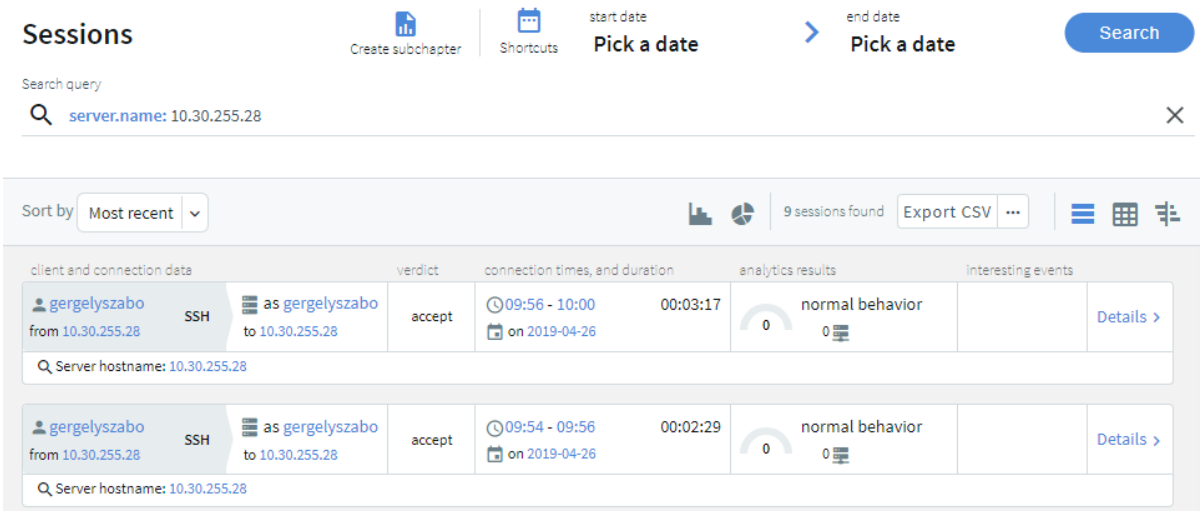
Figure 245: Search — Pick a date



Unified search field

Find everything from a single search box, filter search to a specific field, and combine searches in multiple fields using logical operators. You can also combine content search queries arbitrarily with other search queries. Flow view and quick statistics charts can handle content searches as well. For details, see Using search filters on page 635.

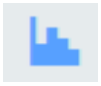
Figure 246: Search — Search filters



Further functions of the old search page (and where they are located on the new search page)

Some functions of the old search page are located somewhere else on the new page. Here is a list of the important ones.

Download the audit trail of a session: Click **details** or **...** to open the details of the session, then click **download trail**. For details, see Viewing session details on page 716.

Display the timeline: Click the  icon. To limit the date range of the search, click **Pick a date** or **shortcuts**. For details, see Specifying time ranges on page 632.

Change the time interval of the search: To limit the date range of the search, click **Pick a date** or **shortcuts**. For details, see Specifying time ranges on page 632. Alternatively, you can select a period on the timeline: click at the beginning of the interval, keep the button pressed, then move the pointer to the end of the interval. The timeline and the search results will be updated automatically.

Search in the screen content: To search in the content of the audited sessions, use the `screen.content` field in your search query. For example: `screen.content="exit"`. To search in the contents of a specific session, download the audit trail, open it in the Safeguard Desktop Player application, and use the Search feature of the Safeguard Desktop Player. For details, see Searching in the contents of audit trails on page 694.

Search or filter in a specific field: To search in a specific data field, start typing the name of the field into the search field. The possible field names and their description is automatically displayed. For example, to search for a specific username, use the `user.name:"my-username"` search query. For details, see Viewing session details on page 716.

Save a filter or a search query: SPS does not store filters anymore, but you can bookmark the page.

Searching session data on a central node in a cluster

The central search functionality is available when your deployment consists of two or more instances of One Identity Safeguard for Privileged Sessions (SPS) organized into a cluster. When you have a cluster of nodes set up, you have the possibility to search all session data recorded by all nodes in the cluster on a single node. This is achieved by assigning roles to the individual nodes in your cluster: you can set up one of your Safeguard for Privileged Sessions nodes to be the Search Master and the rest of the nodes to be Search Minions. Search Minions send session data that they record to the Search Master, and the Search Master acts as a central search node.

To set up your environment for central searching, complete the following steps:

1. Enable cluster management on the nodes that you want to be part of your cluster.
2. Build a cluster.
3. Assign roles to nodes in your cluster.

   Familiarize yourself with:

   - The available search roles before assigning them to nodes. For more information, see Cluster roles on page 346.
   - Managing a central search configuration. For more information, see Managing a cluster with central search configuration and configuration synchronization.

Once you have your cluster set up and the appropriate roles assigned, you can start searching session data using the Search interface.

NOTE:

Central search is not available on the **Search (classic)** interface.

Limitations of the central search functionality

Currently, the central search functionality comes with the following limitations:

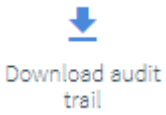
- Session data recorded by a node before it was joined to the cluster will not be searchable centrally. Only session data recorded after the node has been joined to

the cluster is available for central search.

- You cannot to run the indexer process on unindexed sessions after assigning the Search Master role to a node. Make sure all important sessions are indexed before assigning the Search Master role to the node.
- The Search Master node cannot run internal indexer processes, nor does it receive connections from external indexers. Indexers work only with Search Minion nodes.
- User behavior analysis provided by One Identity Safeguard for Privileged Analytics is not available.
- It is not possible to replay audit trail files in your browser from the Search Master node.
- When near real-time indexing is configured on a Search Minion node, while session data from active connections is visible on the Search interface of the Search Master node, it is not possible to:
    - export the audit trail of an active connection,
    - follow an active connection, and
    - terminate an active connection.

  Note, however, that you can terminate the active, ongoing connection on the Search Minion node that is recording the connection in question.
- A reliable, high-bandwidth connection is required between the nodes. Small loss of connection is handled well but if the connection between the Search Minions and the Search Master is lost for a longer period of time, the Search Minions will stop accepting new connections until the connection is repaired. Data is automatically pushed to the Search Master after the connection is restored.

NOTE:

Search Minion nodes do not send the files storing the audit trails to the Search Master node. When a user clicks 

, the Search Master node streams the trail files to the user from the original Search Minion node that recorded the sessions. If a Search Minion node does not have a backup policy set up and an error occurs that causes data loss, then session data recorded by that node will not be available.

Advanced authentication and authorization techniques

This section describes the advanced authentication and authorization techniques available in One Identity Safeguard for Privileged Sessions.

- For details on creating usermapping policies, see Configuring usermapping policies on page 740.
- For details on configuring gateway authentication, see Configuring gateway authentication on page 742.
- For details on configuring four-eyes authorization and real-time monitoring, see Configuring four-eyes authorization on page 752.
- For details on configuring Credential Stores, see Using credential stores for server-side authentication on page 757.

Configuring usermapping policies

For SSH, RDP, Telnet, and Citrix ICA connections, usermapping policies can be defined. A usermapping policy describes who can use a specific username to access the remote server: only members of the specified local or LDAP usergroups (for example, `administrators`) can use the specified username (for example, `root`) on the server.

CAUTION:

In SSH connections, the users must use the following as their username: `gu=username@remoteusername`, where `username` is the username used in the LDAP directory, One Identity Safeguard for Privileged Sessions will use this username to determine their group memberships, and `remoteusername` is the username they will use on the remote server. For example, to access the example.com server as root, use:

```
gu=yourldapusername@root@example.com
```

For the username of SSH users, only valid UTF-8 strings are allowed.

CAUTION:

In Telnet connections, usermapping policy works only if Extract username from the traffic is enabled. For details, see Extracting username from Telnet connections on page 577.

When configuring ICA connections, also consider the following:

CAUTION:

If the clients are accessing a remote application or desktop that is shared for Anonymous users (that is, the Users properties of the application is set to `Allow anonymous users` in the Citrix Delivery Services Console), the actual remote session will be running under an Anonymous account name (for example, `Anon001`, `Anon002`, and so on), not under the username used to access the remote server. Therefore, you need to enable usermapping to the `Anon*` usernames.

To accomplish this, create a usermapping policy and set the Username on the server option to `Anon*`, and the Groups option to `*`, then use this usermapping policy in your ICA connections. For details on using usermapping policies, see Configuring usermapping policies on page 740.

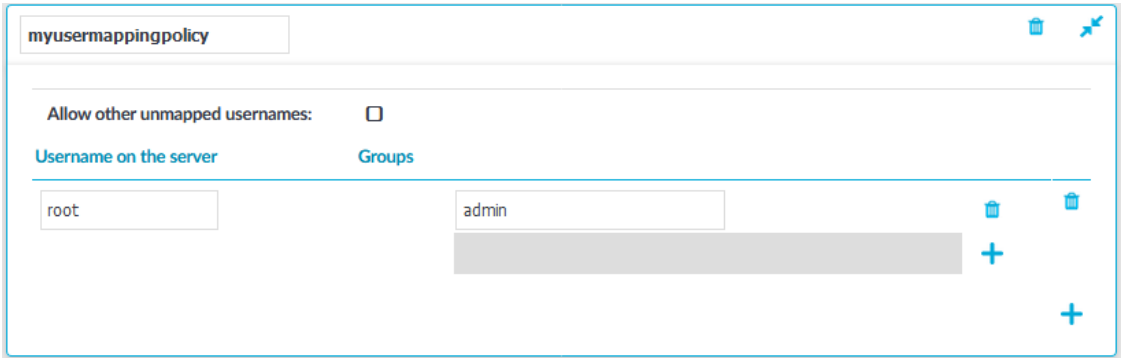
NOTE:

Starting from SPS version 3.2, usermapping is possible only when gateway authentication is used as well.

To configure usermapping

1. Navigate to **Policies > Usermapping Policies**.

   Figure 247: Policies > Usermapping Policies — Configuring usermapping policies

2. Click + to create a new policy, and enter a name for the policy.
3. Click + and enter the username that can be used to access the remote server (for example `root`) into the **Username on the server** field. SPS will use this username in the server-side connection. To permit any username on the server side, enter an asterisk (*).

4. Select **Groups**, click + and specify who is permitted to use the remote username set in the **Username on the server** field.
   - If you have an LDAP Server set in the connection policy where you will use usermapping, enter the name of the local or LDAP usergroup (for example `admins`) whose members will be permitted to use the remote username. For details on LDAP authentication, see Authenticating users to an LDAP server on page 459.

     NOTE:

     The LDAP server configured in the connection policy is not necessarily the same as the LDAP server used to authenticate the users accessing the SPS web interface.

   - If you do not authenticate the connections to an LDAP server, enter the name of the userlist whose members will be permitted to use the remote username. For details on using userlists, see Creating and editing user lists on page 457.

   Repeat this step to add further groups if needed.
5. Repeat steps 3-4 to add further usernames if needed.
6. To permit other users, who are not explicitly listed in the Usermapping Policy access the remote servers, select the **Allow other unmapped usernames** option. Note that these users must use the same username on the SPS gateway and the remote server.
7. Click Commit.
8. Navigate to the **Connections** page of the traffic (for example to **SSH Control > Connections**), and select the connection policy to modify.
9. Select the usermapping policy created in Step 2 from the **Usermapping policy** field.
10. Click Commit.

    NOTE:

    For RDP connections, usermapping is possible only when gateway authentication is used as well. When configuring usermapping for RDP connections, proceed to Configuring out-of-band gateway authentication on page 744 and configure gateway authentication.

Configuring gateway authentication

When gateway authentication is required for a connection, the user must authenticate on One Identity Safeguard for Privileged Sessions (SPS) as well. This additional authentication can be performed:

- Out-of-band, on the SPS web interface, for every protocol.
- Inband, using the incoming connection, for the SSH, Telnet, and RDP protocols.

For details about the concepts of gateway authentication, see The gateway authentication process. You can use gateway authentication to authenticate the real person when the user is using a shared account to access the target server.

NOTE:

For SSH, Telnet, and RDP connections, gateway authentication can be performed also inband, without having to access the SPS web interface.

- For SSH and Telnet connections, inband gateway authentication must be performed when client-side authentication is configured. For details on configuring client-side authentication, see Client-side authentication settings on page 554.
- For RDP connections, inband gateway authentication must be performed when SPS is acting as a Remote Desktop Gateway (or RD Gateway). In this case, the client authenticates to the Domain Controller or a local user database. For details, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway on page 531.

  In the case of RDP connections, inband gateway authentication can also be performed if an AA plugin is configured.

NOTE:

Gateway authentication can be used together with other advanced authentication and authorization techniques like four-eyes authorization, client- and server-side authentication, and so on.

CAUTION:

If the username used within the protocol to access the remote server is different from the username used to perform gateway authentication (for example, because the user uses a shared account in the remote server, but a personal account for gateway authentication), usermapping must be configured for the connection. For details on usermapping, see Configuring usermapping policies on page 740.

NOTE:

To configure a credential store for gateway authentication, see Using credential stores for server-side authentication on page 757.

Configuring out-of-band gateway authentication

CAUTION:

- **The `admin` user is a special One Identity Safeguard for Privileged Sessions (SPS) user and not a member of any user groups, nor can it belong to any group. Since usermapping policies are based on user groups, performing gateway authentication with the `admin` user is likely to result in usermapping errors.**
- **When using SSL-encrypted RDP connections, or connections that use the Credential Security Service Provider (CredSSP) authentication method, some Microsoft RDP clients restart the connection during the authentication process. This would require the user to perform gateway authentication on the SPS web interface twice. To avoid this situation, SPS temporarily caches the successful gateway authentication results if the client terminates the connection at a certain step while establishing the connection. The cache is used to automatically authenticate the restarted connection without user interaction.**

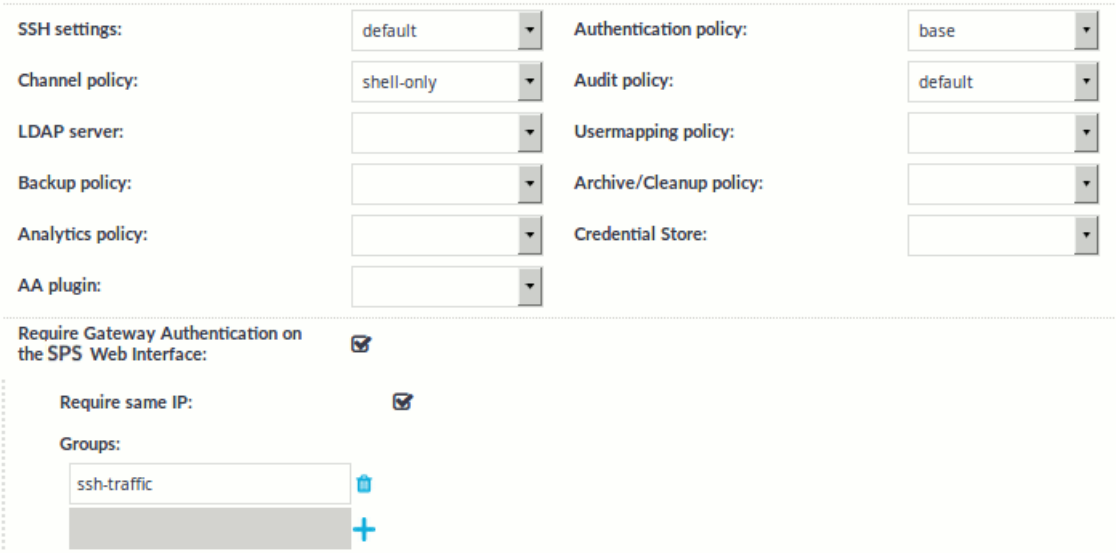
  In this case, the restarted connection coming from the same source IP and targeting the same destination IP:port pair will be authenticated from the cache. The cache is deleted after three minutes, or when a connection is authenticated from the cache.

  However, caching the authentication results has the following side-effect: if a different connection targets the same destination IP:port pair from seemingly the same source IP address within the brief period when SPS expects the original connection to be re-established, the new connection can access the target server without having to authenticate on the SPS gateway. Normally, this can occur only if the clients are behind a NAT.
- **If the clients are behind a device that performs network address translation (NAT), it will seem to SPS as if every connection was initiated from the same IP address. Therefore, in such cases using out-of-band gateway authentication is not recommended for security reasons, especially for RDP connections. If possible, use inband gateway authentication instead.**

To configure gateway authentication

1. Navigate to the **Connections** page of the traffic (for example to **SSH Control > Connections**), and select the connection policy to modify.
2. Select the **Require Gateway Authentication on the SPS Web Interface** option. This is the option to configure gateway authentication via the web interface of SPS.

Figure 248: <Protocol name> Connections > Require Gateway Authentication on the SPS Web Interface — Configuring gateway authentication



3. To accept the gateway authentication only from the host that initiated the connection, select **Require same IP**.

   NOTE:

   This option has no effect if the clients are behind a device that performs network address translation (NAT). In such cases, use inband gateway authentication instead.

4. By default, any user can perform gateway authentication for the connections. To allow only members of a specific group authenticate the connections of this connection policy, select **Groups**, click + and enter the name of the group whose members can authenticate the connections. This group must exist on the **AAA > Group Management** page. For details on creating and managing usergroups, see Managing user rights and usergroups on page 320. Repeat this step to add further groups if needed.
5. For SSH, RDP, Telnet and Citrix ICA connections, you may want to set a usermapping policy in the **Usermapping policy** field. For details on usermapping policies, see Configuring usermapping policies on page 740.
6. Click Commit. After that, users accessing these connections must perform gateway authentication as described in Performing out-of-band gateway authentication on One Identity Safeguard for Privileged Sessions (SPS) on page 748.
7. (Optional) To restrict the availability of selected channels of the connection based on the username used for gateway authentication, edit the channel policy used in the connection.

a. Navigate to the channel policy used in the connection (for example, **SSH Control > Channel Policies**).

b. Select **Gateway Group**, click + and enter the name of the user group allowed to use this type of the channel. The user group must correspond to the username used for the gateway authentication. Repeat this step until all permitted groups are listed.

   You may list local user lists as defined in Creating and editing user lists on page 457, or LDAP groups (for details on accessing LDAP servers from SPS, see Authenticating users to an LDAP server on page 459). Note the following behavior of SPS:

   - If you list multiple groups, members of any of the groups can access the channel.

     NOTE:

     When listing both a whitelist and blacklist in the **Gateway Group** section and a username appears on both lists, the user will be able to access the channel.

   - If a local user list and an LDAP group has the same name and the LDAP server is configured in the connection that uses this channel policy, both the members of the LDAP group and the members of the local user list can access the channel.

c. Click Commit.

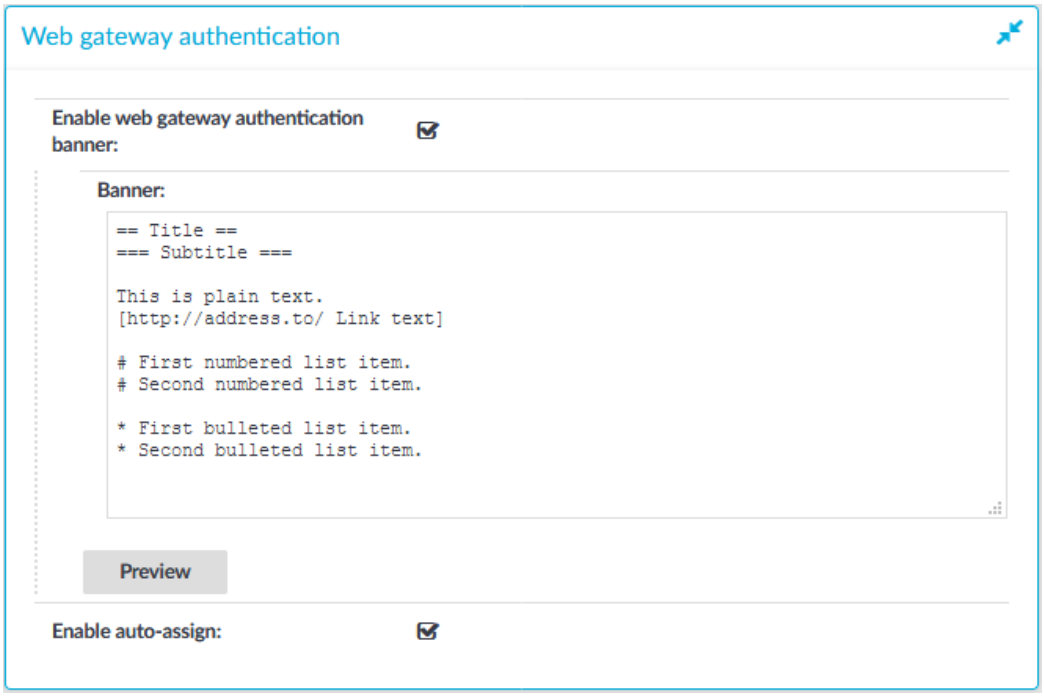
8. (Optional) If you want to provide a limited SPS web interface to your users that can be used only for gateway authentication and 4-eyes authorization, set up a dedicated user-only web login address. For details, see Configuring user and administrator login addresses on page 118.

9. (Optional) You can configure a message for users accessing SPS for out-of-band authentication. The message is displayed when they log in to SPS.

   a. Navigate to **Basic Settings > Management > Web gateway authentication**.

   b. Select **Enable web gateway authentication banner**.

Figure 249: Basic Settings > Management > Web gateway authentication — Configuring a banner



c. Enter the message in the **Banner** field. You can use the following text formatting options:

```
== Title ==
=== Subtitle ===

This is plain text.
[http://address.to/ Link text]

# First numbered list item.
# Second numbered list item.

* First bulleted list item.
* Second bulleted list item.
```

d. Click Commit.

10. (Optional) If your users have sessions to several remote server, or access a server several times a day, performing the gateway authentication for every session can be a nuisance. To permit your users to authenticate on the SPS web interface once, and open sessions without repeating the gateway authentication, select **Enable auto-assign** and click Commit. Note that the user must leave the browser window

(or tab) of SPS open.

Performing out-of-band gateway authentication on One Identity Safeguard for Privileged Sessions (SPS)

The following describes how to perform out-of-band gateway authentication on One Identity Safeguard for Privileged Sessions (SPS).

To perform out-of-band gateway authentication on SPS

1. Initiate a connection from a client. If gateway authentication is required for the connection, SPS will pause the connection.

   NOTE:

   For SSH and Telnet connections, when initiating the connection, you can use the following as your username: `gu=gatewayusername@remoteusername`, where `gatewayusername` is the username you will use to login to the SPS web interface (also called gateway user), and `remoteusername` is the username you will use on the remote server.

2. Open a browser, preferably on the same host you initiated the connection from, and navigate to the login page of SPS.

   CAUTION:

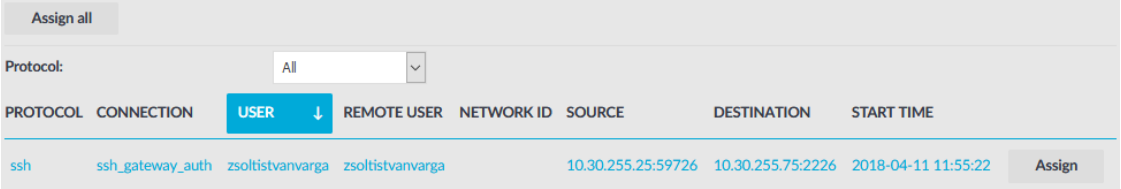
   If the username used within the protocol is different from the username used to access the SPS web interface to perform gateway authentication, usermapping must be configured for the connection. For details on usermapping, see Configuring usermapping policies on page 740.

3. Log in to SPS, and select **Gateway Authentication** from the main menu. The list of connections waiting for gateway authentication will be displayed.

   NOTE:

   - If users accessing the SPS web interface are authenticated to and LDAP server, the users must successfully authenticate to the LDAP server set on the **AAA > Settings** page.
   - No other SPS privilege is required to access this page.

Figure 250: Gateway Authentication — Performing gateway authentication



4. Select the connection that you started, and click **Assign**.
5. Continue to authenticate on the server.
6. To authenticate another session, you must either:
   - repeat this procedure, or
   - if your SPS administrator has enabled the auto-assign feature, you do not have to repeat this procedure as long as the browser tab you authenticated on SPS is open.

Performing inband gateway authentication in SSH and Telnet connections

The following describes how to perform inband gateway authentication in SSH and Telnet connections.

To perform inband gateway authentication in SSH and Telnet connections

1. Initiate a connection from a client. If gateway authentication is required for the connection, One Identity Safeguard for Privileged Sessions (SPS) will pause the connection.
2. SPS requests the username used for gateway authentication. Enter your gateway username into the **Gateway username** prompt. If password authentication is used, provide the password for the gateway user as well.
3. The login prompt for the remote server is displayed. Enter your username used on the remote server into the **Username** prompt. If password authentication is used, provide the password for the username as well.

   CAUTION:

   If the username used within the protocol to access the remote server is different from the username used to perform gateway authentication, usermapping must be configured for the connection. For details on usermapping, see Configuring usermapping policies on page 740.

NOTE:

When initiating the connection, you can use the following as your username: `gu=gatewayusername@remoteusername`, where `gatewayusername` is the username you will use to authenticate on SPS and `remoteusername` is the username you will use on the remote server. That way you do not have to provide the usernames in the prompt, only the passwords if password authentication is used.

If SPS is configured to require client-side authentication, the `gatewayusername` user must authenticate on the client side.

Performing inband gateway authentication in RDP connections

The following describes how to perform inband gateway authentication in RDP connections.

To perform inband gateway authentication in RDP connections

1. Initiate a connection from a client.
2. The graphical login window is displayed.
   - If the **Advanced > Remote Desktop Gateway > Logon Settings > Use my Remote Desktop Gateway credentials for the remote computer** option of your Remote Desktop application is enabled, login to the remote server using your usual credentials. One Identity Safeguard for Privileged Sessions (SPS) will use these credentials for the gateway authentication on the Domain Controller as well.
   - If the **Advanced > Remote Desktop Gateway > Logon Settings > Use my Remote Desktop Gateway credentials for the remote computer** option of your Remote Desktop application is disabled, first you have to authenticate on the SPS gateway. Enter your username and password for the Domain Controller.

     If the first authentication is successful, a second login window is displayed. Enter your username and password for the remote server you are trying to access.
   - If SPS is configured to use a Credential Store to login to the target server, enter the following:
     - In the **Username** field, enter the domain name, the `-AUTO` suffix, and your username. For example, `EXAMPLEDOMAIN-AUTO\Administrator`.

NOTE:

The `-AUTO` suffix is the default value of the **RDP Control > Settings > Autologon domain suffix** option of One Identity Safeguard for Privileged Sessions (SPS). If your SPS administrator has changed this option, use the appropriate suffix instead of -AUTO.

- Enter your username (only the username, without the domain, for example, `Administrator`) into the **Password** field.

3. If the authentication is successful, the desktop of the remote server is displayed.

Troubleshooting gateway authentication

If a user initiates a connection and then logs in to the One Identity Safeguard for Privileged Sessions (SPS) web interface, it might happen that his connection is not shown on the **Gateway Authentication** page. SPS checks the following points to determine if a pending connection is listed for a user:

CAUTION:

The `admin` user is a special One Identity Safeguard for Privileged Sessions (SPS) user and not a member of any user groups, nor can it belong to any group. Since usermapping policies are based on user groups, performing gateway authentication with the `admin` user is likely to result in usermapping errors.

- The username used to access the SPS web interface is a member of a group listed in the **Gateway authentication > Groups** field of the connection policy.
- If SPS knows from the protocol the username that will be used to access the SPS web interface to perform the gateway authentication, the connection is displayed only to this user.

  For SSH connections, SPS can determine the username if:

  - The client specifies the username for the web interface within the protocol using the `gu=webusername@server-side-username@server` format.
  - The client specifies the username within the protocol using an interactive prompt.
  - If the client does not use any of the above options, SPS uses the remote username. In this case, the username for the web interface must be the same as the remote username, otherwise the connection is not displayed.
- If the **Gateway authentication > Require same IP** option is enabled, the pending connection is displayed only if the user accesses the SPS web interface from the same IP address as the client in the pending connection.

NOTE:

The `admin` user sees every pending connection.

Configuring four-eyes authorization

When four-eyes authorization is required for a connection, a user (called authorizer) must authorize the connection on One Identity Safeguard for Privileged Sessions (SPS) as well. This authorization is in addition to any authentication or group membership requirements needed for the user to access the remote server. For details about the concepts of four-eyes authorization, see Four-eyes authorization on page 65.

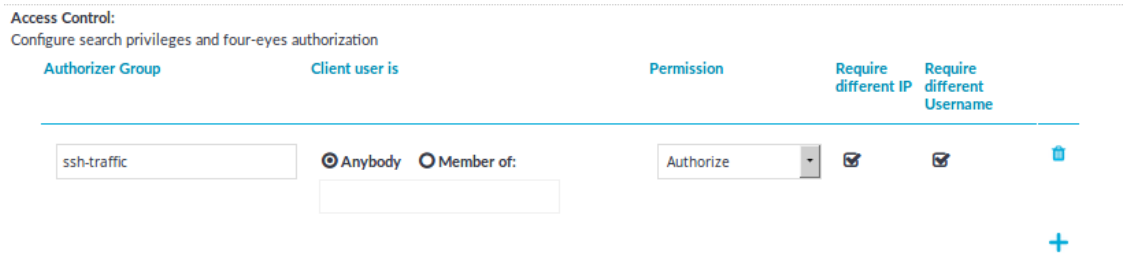
Configuring four-eyes authorization

The following describes how to configure four-eyes authorization.

To require four-eyes authorization for a connection

1. Navigate to the **Connections** page of the traffic (for example to **SSH Control > Connections**), and select the connection policy to modify.

2. **Figure 251: <Protocol name> Control > Connections > Access Control — Configuring four-eyes authorization**



   Navigate to **Access Control** and click +.

3. Enter the name of the usergroup whose members are permitted to authorize the sessions of the connection policy into the **Authorizer Group** field. This group must exist on the **AAA > Group Management** page. For details on creating and managing usergroups, see Managing user rights and usergroups on page 320.

   CAUTION:

   Usernames, the names of user lists, and the names of usergroups are case sensitive.

4. By default, the authorizer can authorize any session of the connection policy.

   If the authorizer is permitted to authorize only the sessions of a certain usergroup, select **Client user is > Member of**, and enter the name of the userlist whose sessions the authorizer can authorize. If you use four-eyes authorization without gateway authentication, you can specify an LDAP group instead of a userlist.

CAUTION:

Usernames, the names of user lists, and the names of usergroups are case sensitive.

CAUTION:

When using both gateway authentication and four-eyes authorization in a Connection Policy, specify the usergroup of the gateway username. The specified group must be a local or LDAP group.

5. Set the permissions of the usergroup set in the **Authorizer Group** field.
   - If the **Authorizer** group can authorize (that is, enable) and audit (that is, monitor in real-time and download the audit trails) the sessions, select **Permission > Search&Authorize**.
   - If the **Authorizer** group can only authorize (that is, enable) the sessions, select **Permission > Authorize**.

     NOTE:

     This option is not valid for HTTP connections.
   - If the **Authorizer** group can only audit (that is, monitor in real-time and download the audit trails) the sessions, select **Permission > Search**.

   NOTE:

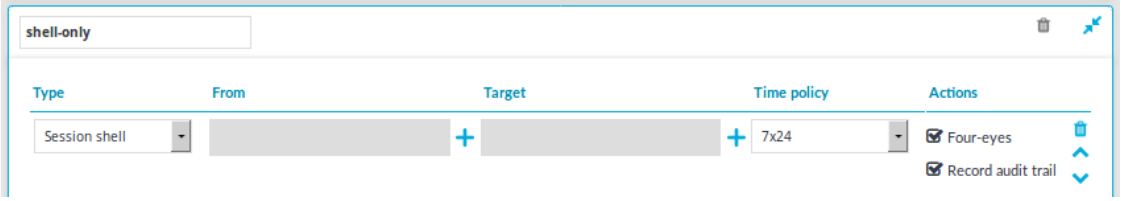
   If the **Client user is > Member of** field is set, the auditor can only monitor the sessions of the specified usergroup. However, if **Client user is > Member of** field is set, the Auditor cannot access the **Search** page. To avoid this problem, add another Access Control rule for the **Authorizer Group** without setting the **Client user is**field.

   The `admin` user of One Identity Safeguard for Privileged Sessions (SPS) can audit and authorize every connection.
6. To ensure that the client and the authorizer use different IP addresses and thus prevent self-authorization, enable **Require different IP**. If this is enabled, and the client and the authorizer do not have different IP addresses, it disables all actions for the connection and the four-eyes authorization, until they have different IP addresses.
7. To ensure that the client and the authorizer use different usernames and thus prevent self-authorization, enable **Require different Username**. If this is enabled, and the client and the authorizer do not have different usernames, it disables all actions for the connection and the four-eyes authorization, until they have different usernames.
8. Repeat steps 2-6 to add other authorizers or usergroups if needed.
9. Click 

.

10. Navigate to the **Channel Policies** page of the traffic (for example, to **SSH Control > Channel Policies**), and select the channel policy used in the connection.

Figure 252: <Protocol name> Control > Channel Policies — Configuring four-eyes authorization in the channel policy



11. Enable the **Four-eyes** option for the channels which should be accessed only using four-eyes authorization.

> NOTE:
>
> If a connection uses secondary channels that require four-eyes authorization — for example, a Remote Desktop connection allows a Drawing channel but requires four-eyes authorization for a Disk redirection channel — the connection is locked until the authorizer accepts the channel on the **Four-Eyes** page of SPS, or the four-eyes request times out. During this time, the client application can become nonresponsive, for example, display the graphical desktop but not react to mouse clicks.

> NOTE:
>
> In Citrix ICA connections, four-eyes authorization is required before the user logs in to the destination server. To request four-eyes authorization only after the log in, when the server-side username is already known, select the **Perform 4 eyes after user login** option.

12. Click Commit. After that, users accessing connections using the modified channel policy must be authorized as described in Performing four-eyes authorization on One Identity Safeguard for Privileged Sessions (SPS) on page 755.

13. (Optional) If you want to provide a limited SPS web interface to your users that can be used only for gateway authentication and 4-eyes authorization, set up a dedicated user-only web login address. For details, see Configuring user and administrator login addresses on page 118.

Performing four-eyes authorization on One Identity Safeguard for Privileged Sessions (SPS)

The following describes how to perform four-eyes authorization on One Identity Safeguard for Privileged Sessions (SPS).

To perform four-eyes authorization on SPS

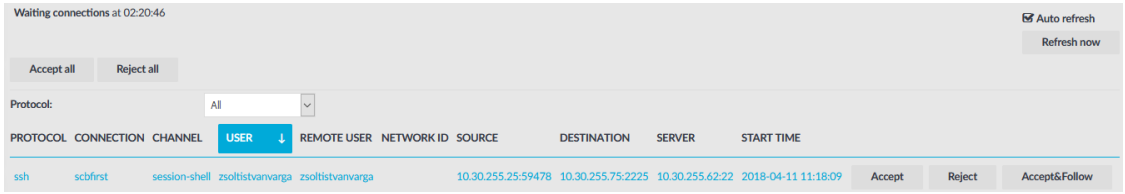
1. When a user initiates a connection from a client and four-eyes authorization is required for the connection, SPS will pause the connection.

   NOTE:

   Four-eyes authorization can be set separately for every channel. However, if a client of an existing connection opens a new channel that requires four-eyes authorization, every channel is paused until the authorization is completed.

2. Login to SPS, and select **Four-Eyes** from the main menu. The list of connections waiting for authorization will be displayed.

   Figure 253: Four-Eyes — Performing four-eyes authorization

   NOTE:

   Only those connections will be listed, where your usergroup has the `Authorize` or the `Search&Authorize` permissions. No other SPS privilege is required to access this page.

3. Select the connection and click **Accept** to enable the connection, **Reject** to deny the connection, or **Accept&Follow** to enable it and monitor in real-time.

NOTE:

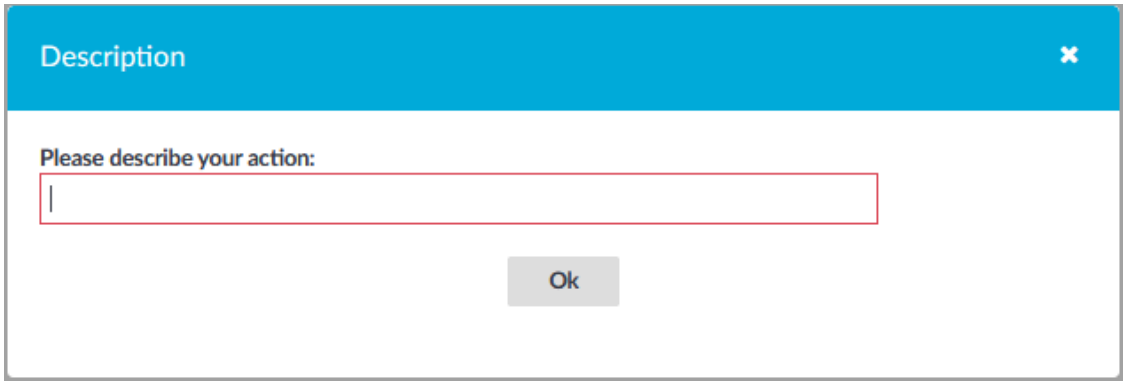
Following a session requires the following:

- The **Record audit trail** option must be enabled for the specific channel in the Channel policy of the connection.
- The Audit Player application must be installed on the computer of the auditor.
- If the Audit policy of the connection uses encryption, the appropriate decryption keys must be available on the computer of the auditor.

The Safeguard Desktop Player application replays the live streams in `live` mode. For details on how to monitor a connection in real-time using the Safeguard Desktop Player, see "Replay audit files in follow mode" in the Safeguard Desktop Player User Guide.

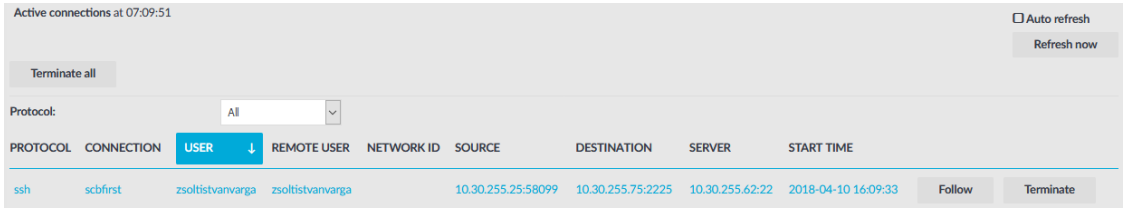
4. Enter a note why the connection was accepted/rejected into the appearing dialog box. This description will be stored in the connection database together with other metadata about the connection.

Figure 254: Describing why a connection was accepted/rejected



5. If you have to terminate an ongoing connection for some reason, select **Active Connections** from the main menu. The list of ongoing connections will be displayed.

Figure 255: Active Connections — Displaying active connections

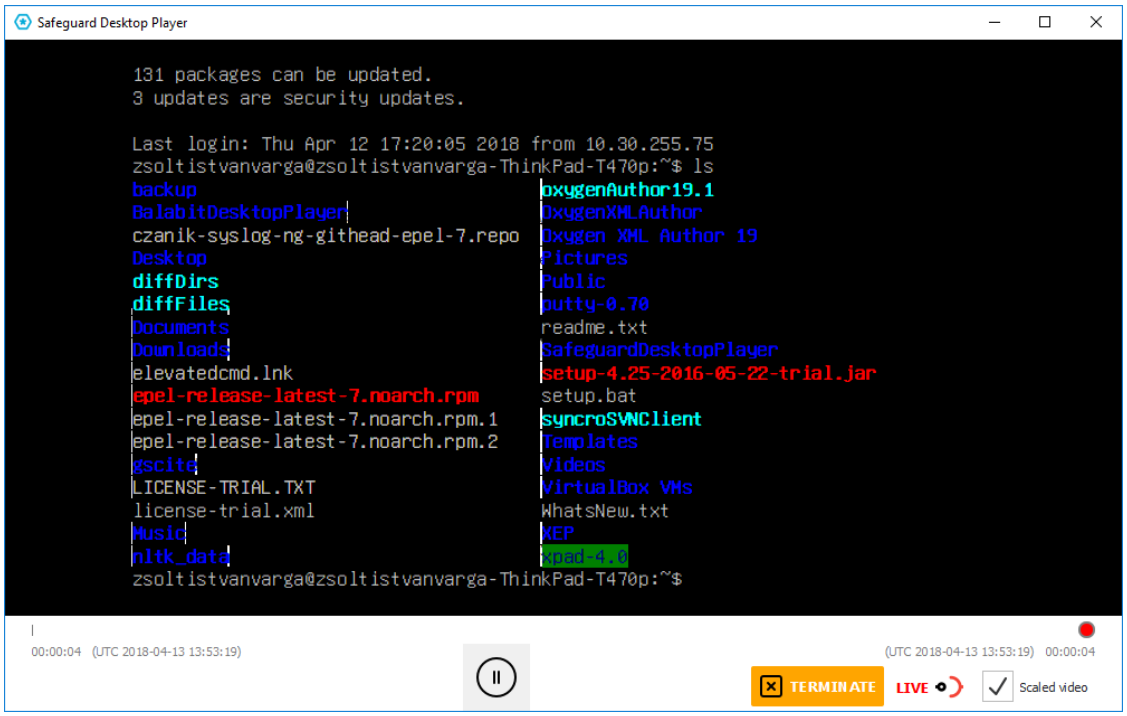


6. Select the connection to stop, and click **Terminate**.

NOTE:

When following a connection in the Safeguard Desktop Player application, the auditor can also terminate the connection from the Audit Player by clicking **Terminate**.

Figure 256: Terminating a connection in Safeguard Desktop Player



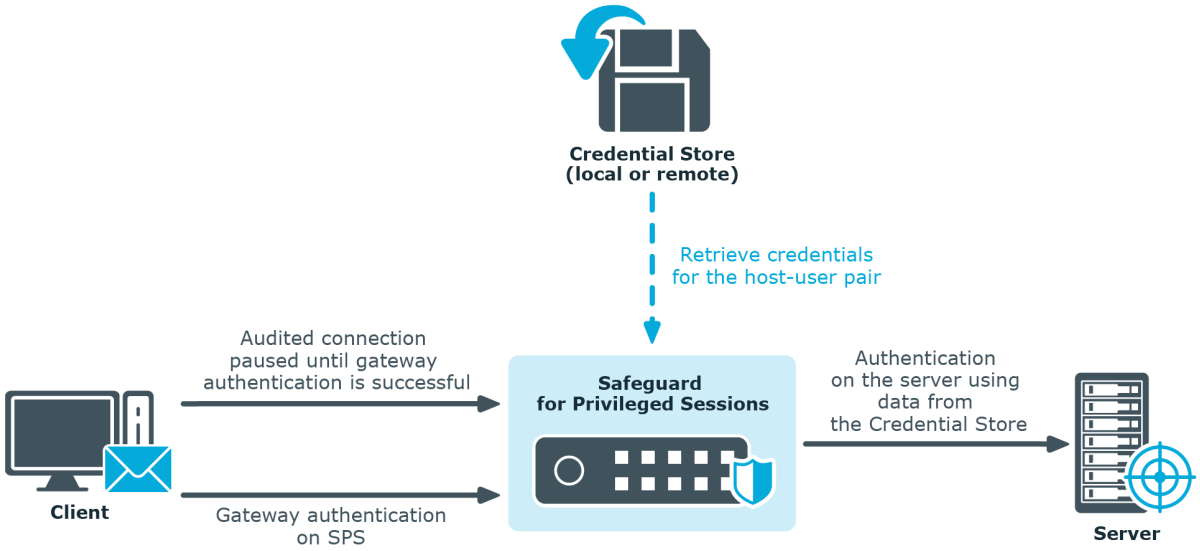
Using credential stores for server-side authentication

Credential Stores offer a way to store user credentials (for example, passwords, private keys, certificates) and use them to log in to the target server, without the user having access to the credentials. That way, the users only have to perform gateway authentication on One Identity Safeguard for Privileged Sessions (SPS) with their usual password (or to an LDAP database), and if the user is allowed to access the target server, SPS automatically logs in using the Credential Store. For details on gateway authentication, see Configuring gateway authentication on page 742.

NOTE:

Keyboard-interactive authentication is not supported when using credential stores.

Figure 257: Authenticating using Credential Stores



Credential Stores can be stored locally on SPS, or on a remote device. For remote Credential Stores, SPS integrates with external authentication and authorization systems using plugins.

- To configure a local Credential Store, see Configuring local Credential Stores on page 758.
- To configure a local, password-protected Credential Store, see Configuring password-protected Credential Stores on page 762.
- To unlock a local, password-protected Credential Store, see Unlocking Credential Stores on page 766.
- To configure a custom Credential Store plugin, see Using a custom Credential Store plugin to authenticate on the target hosts on page 767.

NOTE:

After performing a successful gateway authentication, if the credential store does not contain a password for the user, the user is prompted for the server-side password as a fallback.

In case of authenticating to RDP servers using Network Level Authentication (NLA), the server-side password is prompted at the start of the connection. If there is no password in the credential store for the user and the server-side password is incorrect, the connection is terminated.

Configuring local Credential Stores

The following describes how to configure a local Credential Store that stores the credentials used to login to the target host.

Prerequisites

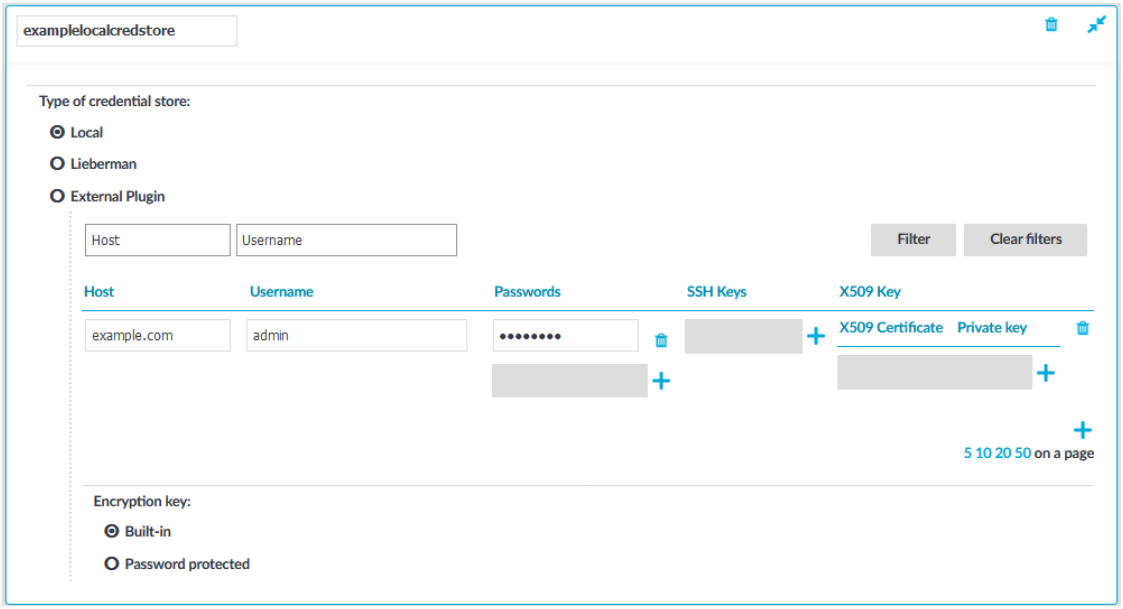
NOTE:

Users accessing connections that use Credential Stores to authenticate on the target server must authenticate on One Identity Safeguard for Privileged Sessions (SPS) using gateway authentication or an AA plugin. Therefore gateway authentication or an AA plugin must be configured for these connections. For details, see "Configuring gateway authentication" in the Administration Guide and "Integrating external authentication and authorization systems" in the Administration Guide.

To configure a local Credential Store that stores the credentials used to login to the target host

1. Navigate to **Policies > Credential Stores**.
2. Click + and enter a name for the Credential Store.
3. Select **Local**.
4. Select **Encryption key > Built-in**. That way the credentials will be encrypted with a built-in password, and the Credential Store is automatically accessible when SPS boots up. To use custom passwords to encrypt the Credential Store, see Configuring password-protected Credential Stores on page 762.

Figure 258: Policies > Credential Stores > Local — Configuring local Credential Stores



5. Add credentials to the Credential Store.
   a. Click + and enter the destination host and the username. For the destination host, you can use hostname, IP address, or subnet as well. To use the same

credentials for every destination host, enter the `0.0.0.0/0` subnet. To use the credentials only on the hosts of a specific domain, enter `*.domain`. Note that:

- Usernames are case sensitive.
- To authenticate users of a Windows domain, enter the name of the domain into the **Host** field.

Use an IPv4 address.

b. Set the credentials. SPS will use these credentials to login to the destination host if the credential store is selected in a Connection policy. If more than one credential is specified to a host-username pair, SPS will attempt to use the credentials as the destination host requests it.

- To add a password, click **Passwords** > +, then enter the password corresponding to the username.
- To upload a private key, click **SSH Keys** > + > ✎, then paste or upload a private key.

  NOTE:

  If the private key is protected by a passphrase, enter the passphrase. The passphrase is needed only once during the upload, it is not required for the later operation of the Credential Store.

- To generate a keypair on SPS click **SSH Keys** > + > ✎, set the length and type of the key, then click **Generate**. After that, click the fingerprint of the key to download the public part of the keypair. There is no way to download the private key from the SPS web interface.

  NOTE:

  One Identity recommends using 2048-bit RSA keys (or stronger).

- To upload a certificate and the corresponding private key, click **X509 Keys** > + > ✎, then paste or upload a certificate and the private key.

  NOTE:

  If the private key is protected by a passphrase, enter the passphrase. The passphrase is needed only once during the upload, it is not required for the later operation of the Credential Store.

NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

c. Repeat the previous step to add further credentials to the username as necessary.

6. Repeat the previous step to add further hosts or usernames as necessary.

NOTE:

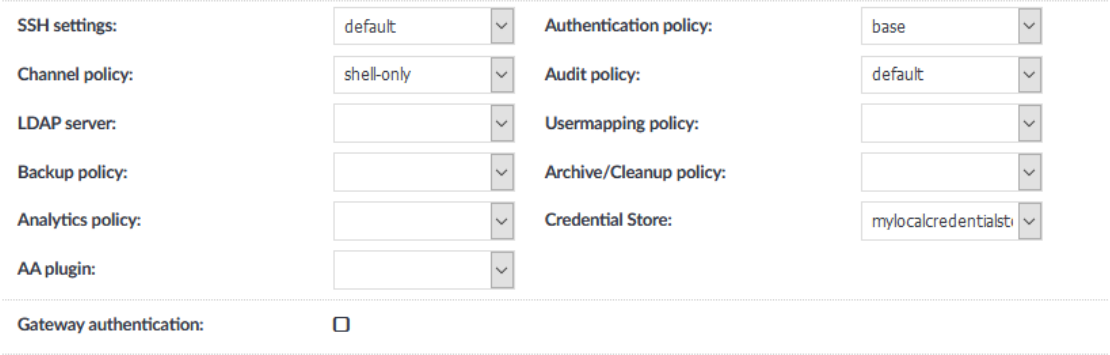
Credential Stores can be used together with usermapping policies to simplify the administration of users on the target hosts. For details, see Configuring usermapping policies on page 740.

7. Click Commit.
8. Navigate to the Connection policy where you want to use the Credential Store (for example, to **SSH Control > Connections**), select the Credential Store to use in the **Credential Store** field, then click Commit.

NOTE:

The Connection Policy will ignore the settings for server-side authentication (set under **Relayed authentication methods**) if a Credential Store is used in the Connection Policy.

Figure 259: <Protocol name> Control > Connections — Select a Credential Store to use



Performing gateway authentication to RDP servers using local Credential Store and NLA

The following describes how to perform a gateway authentication to RDP servers using local Credential Store and Network Level Authentication (NLA).

To perform a gateway authentication to RDP servers using local Credential Store and NLA

1. Initiate the RDP connection.
2. Enter your gateway credentials during the gateway authentication. This can be web gateway authentication, or inband gateway authentication using RD Gateway.
3. Enter the following:

- In the **Username** field, enter the domain name, the `-AUTO` suffix, and your username. For example, `EXAMPLEDOMAIN-AUTO\Administrator`.

  NOTE:

  The `-AUTO` suffix is the default value of the **RDP Control > Settings > Autologon domain suffix** option of One Identity Safeguard for Privileged Sessions (SPS). If your SPS administrator has changed this option, use the appropriate suffix instead of -AUTO.

- Enter your username (only the username, without the domain, for example, `Administrator`) into the **Password** field.

4. If the authentication is successful, the desktop of the remote server is displayed.

Configuring password-protected Credential Stores

The following describes how to configure a local Credential Store that stores the credentials used to login to the target host. The Credential Store will be protected by custom passwords. This password must be entered every time One Identity Safeguard for Privileged Sessions (SPS) is rebooted to make the Credential Store available.

Prerequisites

NOTE:

Users accessing connections that use Credential Stores to authenticate on the target server must authenticate on One Identity Safeguard for Privileged Sessions (SPS) using gateway authentication or an AA plugin. Therefore gateway authentication or an AA plugin must be configured for these connections. For details, see "Configuring gateway authentication" in the Administration Guide and "Integrating external authentication and authorization systems" in the Administration Guide.

To configure a local Credential Store that stores the credentials used to login to the target host

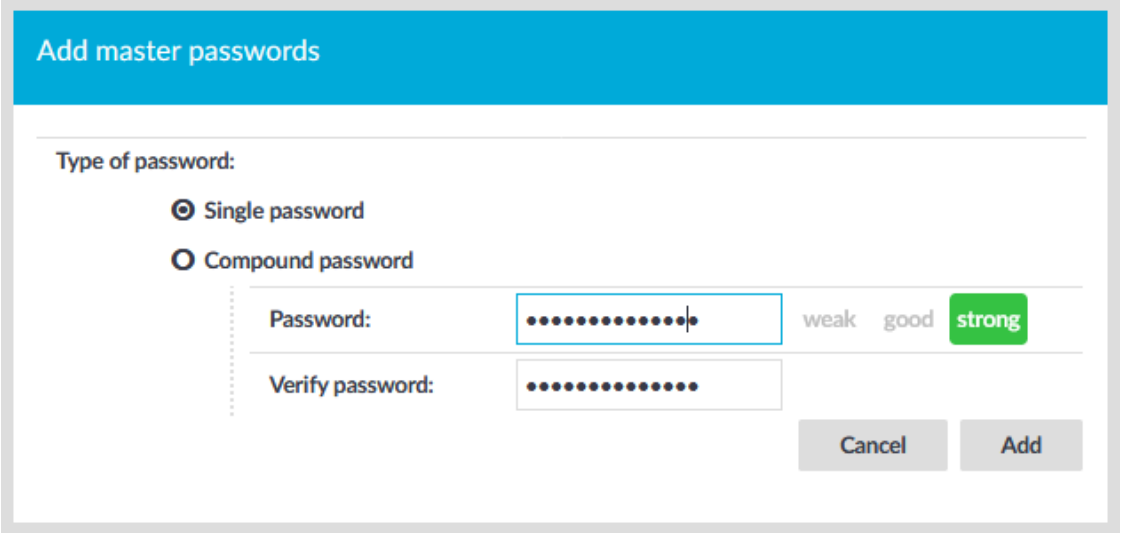
1. Navigate to **Policies > Credential Stores**.
2. Click + and enter a name for the Credential Store.
3. Select **Local**.
4. Select **Encryption key > Password protected**.

   NOTE:

   The contents of the Credential Store, as well as the passwords are included in the configuration backups of SPS. Make sure to encrypt the configuration backups.

5. Select **Master passwords** and click +.

Figure 260: Policies > Credential Stores > Local > Password protected — Set Master Password



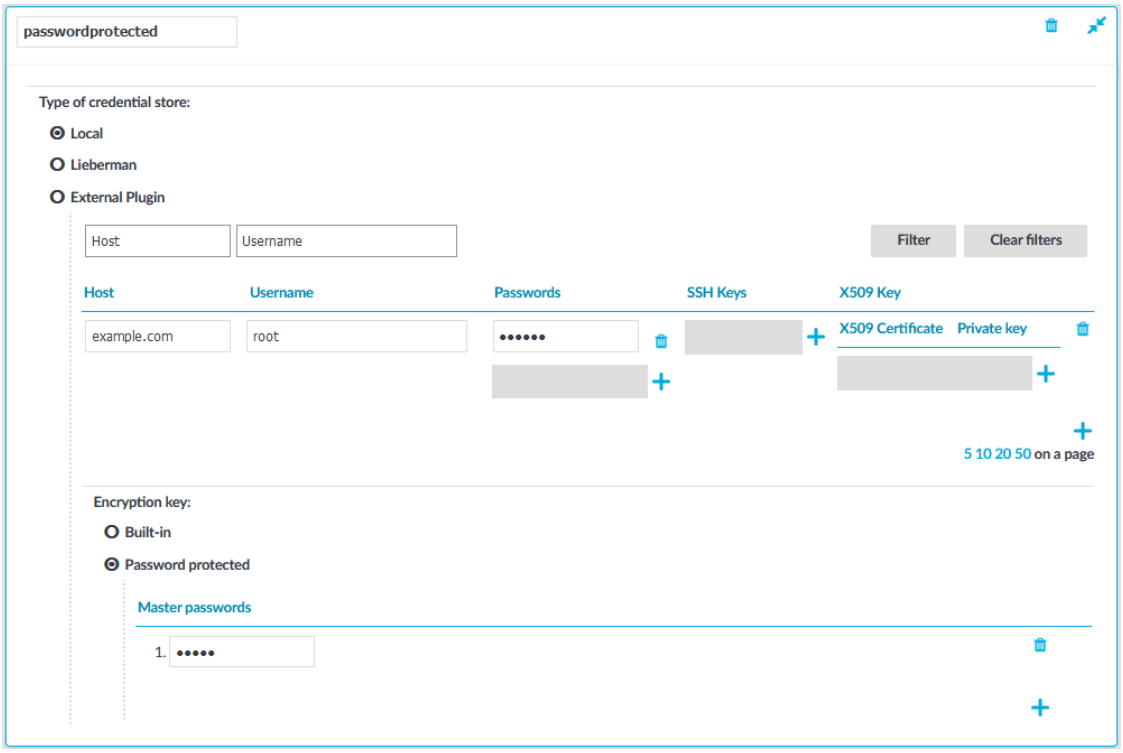
- To protect the Credential Store with a single password, select **Single password** and enter the password into the **Password** and **Verify password** fields. Anyone who knows this password and has the Unlock Credential Store privilege will be able to open the Credential Store. Password-protected Credential Stores must be unlocked on the SPS web interface or console after every SPS reboot.
- To protect the Credential Store with multiple passwords, select **Compound password**, click + and enter a password. Click + to add additional passwords. After finishing listing every password, click **Add**. All of these passwords will be needed to unlock the Credential Store.

SPS encrypts the master passwords using an `aes-256-cbc` cipher, and stores them in a local database.

NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

Figure 261: Policies > Credential Stores > Local > Password protected — Configure Credential Store



6. Repeat the previous step to add another single or compound password. That way, different password sets can be defined for the Credential Store. For example, if a single and a compound password is configured, the chief administrator can unlock the Credential Store with a single password, and two of his subordinates can open the Credential Store together if they know one element each of the compound password.

   TIP:

   To change the password, just click to delete the old password. Then add new passwords as needed.

7. Add credentials to the Credential Store.
   a. Click and enter the destination host and the username. For the destination host, you can use hostname, IP address, or subnet as well. To use the same credentials for every destination host, enter the `0.0.0.0/0` subnet. To use the credentials only on the hosts of a specific domain, enter `*.domain`. Note that:
      - Usernames are case sensitive.
      - To authenticate users of a Windows domain, enter the name of the domain into the **Host** field.

      Use an IPv4 address.

b. Set the credentials. SPS will use these credentials to login to the destination host if the credential store is selected in a Connection policy. If more than one credential is specified to a host-username pair, SPS will attempt to use the credentials as the destination host requests it.

- To add a password, click **Passwords** > +, then enter the password corresponding to the username.
- To upload a private key, click **SSH Keys** > + > ✎, then paste or upload a private key.

  NOTE:

  If the private key is protected by a passphrase, enter the passphrase. The passphrase is needed only once during the upload, it is not required for the later operation of the Credential Store.

- To generate a keypair on SPS click **SSH Keys** > + > ✎, set the length and type of the key, then click **Generate**. After that, click the fingerprint of the key to download the public part of the keypair. There is no way to download the private key from the SPS web interface.

  NOTE:

  One Identity recommends using 2048-bit RSA keys (or stronger).

- To upload a certificate and the corresponding private key, click **X509 Keys** > + > ✎, then paste or upload a certificate and the private key.

  NOTE:

  If the private key is protected by a passphrase, enter the passphrase. The passphrase is needed only once during the upload, it is not required for the later operation of the Credential Store.

NOTE:

One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

c. Repeat the previous step to add further credentials to the username as necessary.

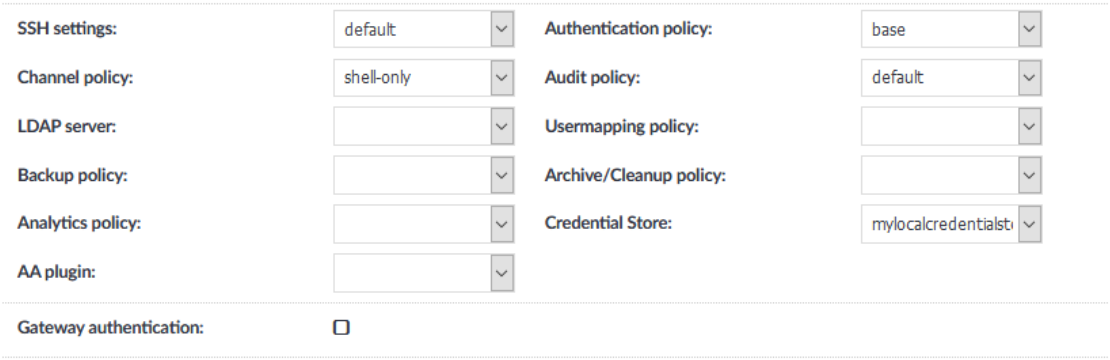
8. Repeat the previous step to add further hosts or usernames as necessary.

NOTE:

Credential Stores can be used together with usermapping policies to simplify the administration of users on the target hosts. For details, see Configuring usermapping policies on page 740.

9. Click Commit.
10. Navigate to the Connection policy where you want to use the Credential Store (for example, to **SSH Control > Connections**), select the Credential Store to use in the **Credential Store** field, then click Commit.

Figure 262: Control > Connections — Select a Credential Store to use



11. Navigate to **Basic Settings > Alerting & Monitoring > Traffic related traps** and enable the **Decryption of a credential failed** (`scbCredStoreDecrpytError`) and **The requested credential store is closed** (`scbCredStoreClosed`) events. That way SPS sends automatic alerts if a Credential Store needs to be unlocked.

> **CAUTION:**
>
> **Password-protected Credential Stores must be unlocked every time after SPS is rebooted. Connections using a password-protected Credential Store will automatically fail until the Credential Store is locked.**
>
> **To unlock a Credential Store, users must have the Unlock Credential Store privilege, or editing (read and write) privileges to the particular Credential Store.**

Unlocking Credential Stores

To unlock a Credential Store and make it available for use, complete the following steps.

Prerequisites

To unlock a Credential Store, users must have the **Unlock Credential Store** privilege, or editing (read and write) privileges to the particular Credential Store.

Steps

1. Login to the One Identity Safeguard for Privileged Sessions (SPS) web interface.
2. Navigate to **Unlock Credential Store** and select the Credential Store to unlock.
3. Enter the password(s) for the Credential Store. For compound passwords, enter every element of the compound password in the correct order.

   NOTE:

   One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`
4. Click **Unlock**.
5. Repeat the previous steps for other Credential Stores as needed.

   NOTE:

   Alternatively, Credential Stores can be unlocked also from the SPS Console Menu.

Using a custom Credential Store plugin to authenticate on the target hosts

The following describes how to configure One Identity Safeguard for Privileged Sessions (SPS) to retrieve the credentials used to login to the target host using a custom plugin.

Prerequisites

To use a custom Credential Store plugin, you have to upload a working Credential Store plugin to SPS. This plugin is a script that can be used to access an external Credential Store or Password Manager. If you want to create such a custom Credential Store plugin, contact our Support Team or see or see the documentation about custom Credential Store plugins.

NOTE:

Users accessing connections that use Credential Stores to authenticate on the target server must authenticate on SPS using gateway authentication. Therefore, gateway authentication must be configured for these connections. For details, see "Configuring gateway authentication" in the Administration Guide.

To upload the custom Credential Store plugin you received, navigate to **Basic Settings > Plugins > Upload/Update Plugins**, browse for the file and click **Upload**.

NOTE:

It is not possible to upload or delete Credential Store plugins if SPS is in sealed mode.

Your plugin `.zip` file may contain an optional sample configuration file. This file serves to provide an example configuration that you can use as a basis for customization if you wish to adapt the plugin to your site's needs.

To configure SPS to retrieve the credentials used to login to the target host using a custom plugin

1. Navigate to **Policies > Credential Stores**.
2. Click + and enter a name for the Credential Store.
3. Select **External Plugin**, then select the plugin to use from the **Plugin** list.
4. If your plugin supports configuration, then you can create multiple customized configuration instances of the plugin for your site. The **Configuration** textbox displays the example configuration of the plugin you selected. If you wish to create a customized configuration instance of the plugin for your site, then edit the configuration here.

   NOTE:

   Plugins created and issued before the release of SPS 5 F1 do not support configuration. If you create a configuration for a plugin that does not support this, the affected connection will stop with an error message.

5. Click Commit.
6. Navigate to the Connection policy where you want to use the Credential Store (for example, to **SSH Control > Connections**), select the Credential Store configuration instance to use in the **Credential Store** field, then click Commit.

Integrating external authentication and authorization systems

One Identity Safeguard for Privileged Sessions (SPS) provides a plugin framework to integrate SPS to external systems to authenticate or authorize the user before authenticating on the target server. Such plugins can also be used to request additional information from the users, for example, to perform multi-factor authentication.

You can use an Authentication and Authorization plugin (aa-plugin) in the following protocols:

- Remote Desktop (RDP)
- Secure Shell (SSH)
- TELNET

- To request a plugin that interoperates with your authentication or authorization system, contact our Support Team.
- For details on configuring SPS to use a plugin, see "Using a custom Authentication and Authorization plugin to authenticate on the target hosts" in the Administration Guide.

How Authentication and Authorization plugins work

If a Connection Policy has an Authentication and Authorization plugin (**AA plugin**) configured, One Identity Safeguard for Privileged Sessions (SPS) executes the plugin as the last step of the connection authorization phase. SPS can request the client to perform other types of authentication before executing the plugin. Using an **AA plugin** in a Connection Policy is treated as gateway authentication if:

- the plugin authenticates the user
- authentication is successful
- the plugin returns the `gateway_user` and `gateway_groups` elements, identifying the user it has authenticated

Other types of gateway authentication will come before authentication by the **AA plugin**, so information from any other type of gateway authentication (for example, the username and usergroups of this authentication) will already be available and therefore can be used by the plugin. If the Authentication and Authorization plugin does perform gateway authentication, you can use a Credential Store as well.

However, for technical reasons, the web-based gateway authentication (that is, authenticating on the SPS web interface if the **Require Gatweay Authentication on the SPS Web Interface** option is selected in the Connection Policy) is performed after the **AA plugin**, so using **AA plugin** and ticking **Require Gateway Authentication on the SPS Web Interface** at the same time is not a valid configuration.

The plugin can interactively request additional information from the client in the SSH, Telnet, and RDP protocols.

NOTE: In SPS 5.8, a user's group membership is determined by querying only the relevant groups configured for the connection from the LDAP/AD server, instead of retrieving all groups of a given user.

This may cause problems when using AD/LDAP-based gateway authentication together with an AA plugin. The AA plugin `authorize()` hook may be called with only a subset of groups as group membership lookup does not consider groups referenced in the AA plugin code.

As a possible workaround, you can add a rule to the channel policy assigned to the connection that never matches (for example, set the **From** address to 0.0.0.0/32), but contains all the gateway groups that the plugin requires. This channel rule will never match, but it will cause SPS to evaluate if a user is a member of those groups, and will make them available for the plugin if so.

Note that only groups queried by SPS are affected. Gateway groups returned by the AA plugin `authenticate()` hook are passed to the `authorize()` hook unchanged.

SPS executes the `authorize` method after the `authentication` method and any inband gateway authentication or inband destination selection steps. As a result, the `authorize` method already has access to the IP address of the target server and the remote username (the username used in the server-side connection).

Optionally, the plugin can return the `gateway_user` and `gateway_groups` values. SPS will only update the gateway username and gateway groups fields in the connection database if the plugin returns the `gateway_user` and `gateway_groups` values. The returned `gateway_user` and `gateway_groups` values override any such attributes already available on SPS about the connection (that means that channel policy evaluations will be affected), so make sure that the plugin uses the original values appropriately.

If the plugin returns the `gateway_user` and `gateway_groups` values, you may have to configure an appropriate **Usermapping policy** in the **Connection Policy**. If the plugin returns a `gateway_user` that is different from the remote user, the connection will fail without a usermapping policy. For details on usermapping policies, see "Configuring usermapping policies" in the Administration Guide.

Prerequisites

- SPS supports Authentication and Authorization plugins in the RDP, SSH, and TELNET protocols.
- In RDP, using an AA plugin together with Network Level Authentication in a Connection Policy has the same limitations as using Network Level Authentication without domain membership. For details, see "Network Level Authentication without domain membership" in the Administration Guide.
- In RDP, using an AA plugin requires TLS-encrypted RDP connections. For details, see "Enabling TLS-encryption for RDP connections" in the Administration Guide.

Optionally, the plugin can return the `gateway_user` and `gateway_groups` elements. SPS will only update the gateway username and gateway groups fields in the connection database if the plugin returns the `gateway_user` and `gateway_groups` elements. The returned gateway username and gateway groups override any such attributes already available on SPS about the connection, so make sure that the plugin uses the original values appropriately.

If the plugin returns the `gateway_user` and `gateway_groups` elements, you may have to configure an appropriate **Usermapping Policy** in the Connection Policy. If the plugin returns a `gateway_user` that is different from the remote user, the connection will fail without a Usermapping Policy. For details on Usermapping Policies, see "Configuring usermapping policies" in the Administration Guide.

Using a custom Authentication and Authorization plugin to authenticate on the target hosts

The following describes how to configure One Identity Safeguard for Privileged Sessions (SPS) to use an Authentication and Authorization plugin (AA plugin) before accessing the target host.

Prerequisites

- To use a custom plugin, you need to upload a working **AA plugin** to SPS. This plugin is a script that uses the SPS API to access an external system. If you want to create such a plugin, contact our Support Team for details and instructions or see the documentation about custom Authentication and Authorization plugins.
- SPS supports AA plugins in the RDP, SSH, and Telnet protocols.
- In RDP, using an **AA plugin** together with Network Level Authentication in a Connection Policy has the same limitations as using Network Level Authentication without domain membership. For details, see "Network Level Authentication without domain membership" in the Administration Guide.
- In RDP, using an **AA plugin** requires TLS-encrypted RDP connections. For details, see "Enabling TLS-encryption for RDP connections" in the Administration Guide.

To configure SPS to use an Authentication and Authorization plugin before accessing the target host

1. To upload the custom plugin you received, navigate to **Basic Settings > Plugins**, browse for the file and click **Upload**.

   > NOTE:
   >
   > It is not possible to upload or delete plugins if SPS is in "Sealed mode" in the Administration Guide.

   Your plugin `.zip` file may contain an optional sample configuration file. This file serves to provide an example configuration that you can use as a basis for customization if you wish to adapt the plugin to your site's needs.
2. If your plugin supports configuration, then you can create multiple customized configuration instances of the plugin for your site. Create an instance by completing the following steps:

a. Go to **Policies > AA Plugin Configurations**. Select the plugin to use from the **Plugin** list.

b. The **Configuration** textbox displays the example configuration of the plugin you selected. You can edit the configuration here if you wish to create a customized instance of the plugin.

> NOTE:
>
> Plugins created and issued before the release of SPS 5 F1 do not support configuration. If you create a configuration for a plugin that does not support this, the affected connection will stop with an error message.

Figure 263: Policies > AA Plugin Configurations — Creating a customized plugin configuration instance



3. Navigate to the Connection policy where you want to use the plugin (for example, to **RDP Control > Connections**), select the plugin configuration instance to use in the **AA plugin** field, then click Commit.
4. If the plugin sets or overrides the gateway username of the connection, configure a **Usermapping policy** and use it in the Connection policy. For details, see "Configuring usermapping policies" in the Administration Guide.
5. Verify that the configuration works properly: try to establish a test connection. For details, see "Performing authentication with AA plugin in Remote Desktop connections" in the Administration Guide. If the plugin is configured to store any metadata about the connection, these data will be available in the **Additional metadata** field of the SPS Search interface.

Performing authentication with AA plugin in terminal connections

The following describes how to establish a terminal connection (SSH, TELNET, or TN3270) to a server.

To establish a terminal connection (SSH, TELNET, or TN3270) to a server

1. Connect to the server.

   To encode additional data as part of the username, you can use the `@` as a field separator, for example:

   ```
   ssh token_id=id@user@server
   ```

   Replace `id` with your actual token ID.
2. If One Identity Safeguard for Privileged Sessions (SPS) prompts you for further information (for example, a one-time password), enter the requested information.
3. Authenticate on the server.
4. If authentication is successful, you can access the server.

Performing authentication with AA plugin in Remote Desktop connections

The following describes how to establish a Remote Desktop (RDP) connection to a server when the **AA plugin** is configured.

To establish an RDP connection to a server when the AA plugin is configured

1. Open your Remote Desktop client application.
2. If you have to provide additional information to authenticate on the server, you must enter this information in your Remote Desktop client application into the *User name* field, before the regular content (for example, your username) of the field.

   To encode additional data, you can use the following special characters:

   - `%` as a field separator
   - `~` as the equal sign
   - `^` as a colon (for example, to specify the port number or an IPv6 IP address)

   For example, to add a token ID before your username, use the following format:

   ```
   domain\token_id~12345%Administrator
   ```

   Note how domain information is provided. If your server is in a domain, make sure that you specify the domain in this format: putting it in front, followed by a backslash (`\`).
3. Connect to the server.

4. If One Identity Safeguard for Privileged Sessions (SPS) prompts you for further information (for example, a one-time password), enter the requested information.
5. Authenticate on the server.
6. If authentication is successful, you can access the server.

Integrating ticketing systems

The plugin framework provided by One Identity Safeguard for Privileged Sessions (SPS) can also be used to integrate SPS to external ticketing (or issue tracking) systems, allowing you to request a ticket ID from the user before authenticating on the target server. That way, SPS can verify that the user has a valid reason to access the server — and optionally terminate the connection if he does not. Requesting a ticket ID currently supports the following protocols:

- Remote Desktop (RDP)
- Secure Shell (SSH)
- TELNET
- TN3270
- To request a plugin that interoperates with your ticketing system, contact our Support Team.
- For details on configuring SPS to use a plugin, see "Using a custom Authentication and Authorization plugin to authenticate on the target hosts" in the Administration Guide.

Performing authentication with ticketing integration in terminal connections

The following describes how to establish a terminal connection (SSH, TELNET, or TN3270) to a server that requires you to enter a ticket ID.

To establish a terminal connection (SSH, TELNET, or TN3270) to a server that requires you to enter a ticket ID

1. Connect to the server.

   You have the option to use the ID of the ticket you are working on as part of the username (replace `id` with the ticket ID):

   ```
   ssh ticket_id=id@user@server
   ```

NOTE:

Your plugin may use a different name for the key `ticket_id` shown in the example. Plugins work with key-value pairs and the names of keys are entirely up to individual plugins.

2. If you did not provide a ticket ID, One Identity Safeguard for Privileged Sessions (SPS) now prompts you to enter it.
3. Authenticate on the server.
4. If the authentication is successful, you can access the server.

Performing authentication with ticketing integration in Remote Desktop connections

The following describes how to establish a Remote Desktop (RDP) connection to a server that requires you to enter a ticket ID.

To establish an RDP connection to a server that requires you to enter a ticket ID

1. Open your Remote Desktop client application.
2. Enter the ticket ID into your Remote Desktop client application into the *User name* field, before or after the regular content (for example, your username) of the field. You must provide the ticket ID in the following format:

   ```
   ticket_id~<your-ticket-id>%
   ```

   Replace `<your-ticket-id>` with your actual ticket number. For example:

   ```
   ticket_id~12345%Administrator
   ```

   NOTE:

   Your plugin may use a different name for the key `ticket_id` shown in the example. Plugins work with key-value pairs and the names of keys are entirely up to individual plugins.

   To encode additional data, you can use the following special characters:

   - `%` as a field separator
   - `~` as the equal sign
   - `^` as a colon (for example, to specify the port number or an IPv6 IP address)

   For example, to add a token ID before your username, use the following format:

   ```
   domain\token_id~12345%Administrator
   ```

   Note how domain information is provided. If your server is in a domain, make sure that you specify the domain in this format: putting it in front, followed by a backslash (`\`).

3. Connect to the server.
4. Authenticate on the server.
5. If the authentication is successful, you can access the server.

Creating a custom plugin

Creating a custom Authentication and Authorization plugin

For more information, see Creating custom Authentication and Authorization plugins.

Creating a custom Credential Store plugin

For more information, see Creating custom Credential Store plugins.

Plugin troubleshooting

On the default log level, One Identity Safeguard for Privileged Sessions (SPS) logs everything that the plugin writes to `stdout` and `stderr`. Log message lines are prefixed with the session ID of the proxy, which makes it easier to find correlating messages.

To transfer information between the methods of a plugin (for example, to include data in a log message when the session is closed), you can use a cookie.

If an error occurs while executing the plugin, SPS automatically terminates the session.

NOTE:

This error is not visible in the verdict of the session. To find out why the session was terminated, you have to check the logs.

Reports

One Identity Safeguard for Privileged Sessions (SPS) periodically creates reports on the activity of the administrators, its system information, as well as the processed traffic. In addition, you can use the connection database for creating custom reports from connection statistics.

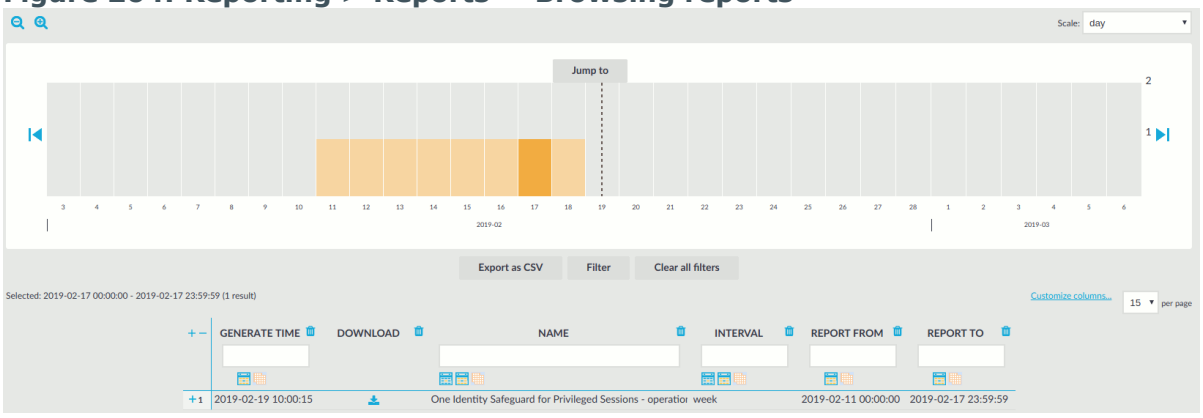
These reports are available in Portable Document (PDF) format by selecting **Reporting > Reports** from the Main Menu. The reports are displayed on a search interface. For more information on using and customizing this interface, see Using the internal search interface on page 334.

The reports are also sent to the e-mail address set at **Basic Settings > Management > Mail settings > Send reports to**, unless specified otherwise in the configuration of the report.

NOTE:

If the **Basic Settings > Management > Mail settings > Send reports to** address is not set, the system report is sent to the SPS administrator's e-mail address.

Figure 264: Reporting > Reports — Browsing reports



Reports can be generated for fixed periods:

- **Daily reports** are generated every day at 00:01.
- **Weekly reports** are generated every week on Monday at 00:01.
- **Monthly reports** are generated on the first day of every month at 00:01.

You can also generate a partial report if necessary, for details, see Generating partial reports on page 818.

To access the reports from the SPS web interface, the user must have the appropriate privileges (for custom reports, the default requirement is membership in the `search` group). In addition, individual reports might have different access requirements configured. For more information on configuring user rights, see Managing user rights and usergroups on page 320.

The following information is available about the reports:

- **Download**: A link to download the report.
- **Name**: Name of the report.
- **Interval**: The length of the reported period, for example week, month, and so on.
- **Report from**: The start of the reported interval.
- **Report to**: The end of the reported interval.
- **Generate time**: The date when the report was created.

TIP:

Use the time bar to find reports that contain a particular period. If you select a period (for example click on a bar), only those reports will be displayed that contain information about the selected period.

Contents of the operational reports

The operational reports of One Identity Safeguard for Privileged Sessions (SPS) are available in Adobe Portable Document Format (PDF), and contain the following information:

- **Configuration changes**: Lists the number of SPS configuration changes per page and per user. The frequency of the configuration changes is also displayed on a chart.
- **Main reports**: Contains statistics about the total traffic that passed SPS, including the number of sessions that passed for every connection policy, the used usernames, clients, and servers, and so on.

  NOTE:

  Connections that are still in progress when the report is generated are excluded from the report. Sessions that are being indexed and reporting jobs are listed in the *Sessions with in progress indexing or reporting jobs* section of the report.
- **Reports by connection**: Contains separate statistics about every connection policy configured on SPS.
- **System health information**: Displays information about the filesystem and network use of SPS, as well as the average load.

Configuring custom reports

To configure a report, create a chapter and assign any of the existing subchapters to it. The following sources (statistics or other queries) are available as reporting subchapters:

- The indexed contents of audit trails, as described in Indexing audit trails on page 592.
- The statistics of an audit trail search, as described in Displaying statistics on search results on page 970.
- Custom queries of the connection database, as described in Creating statistics from custom database queries on page 790. The list of tables and fields you can query are described in Database tables available for custom queries on page 794.

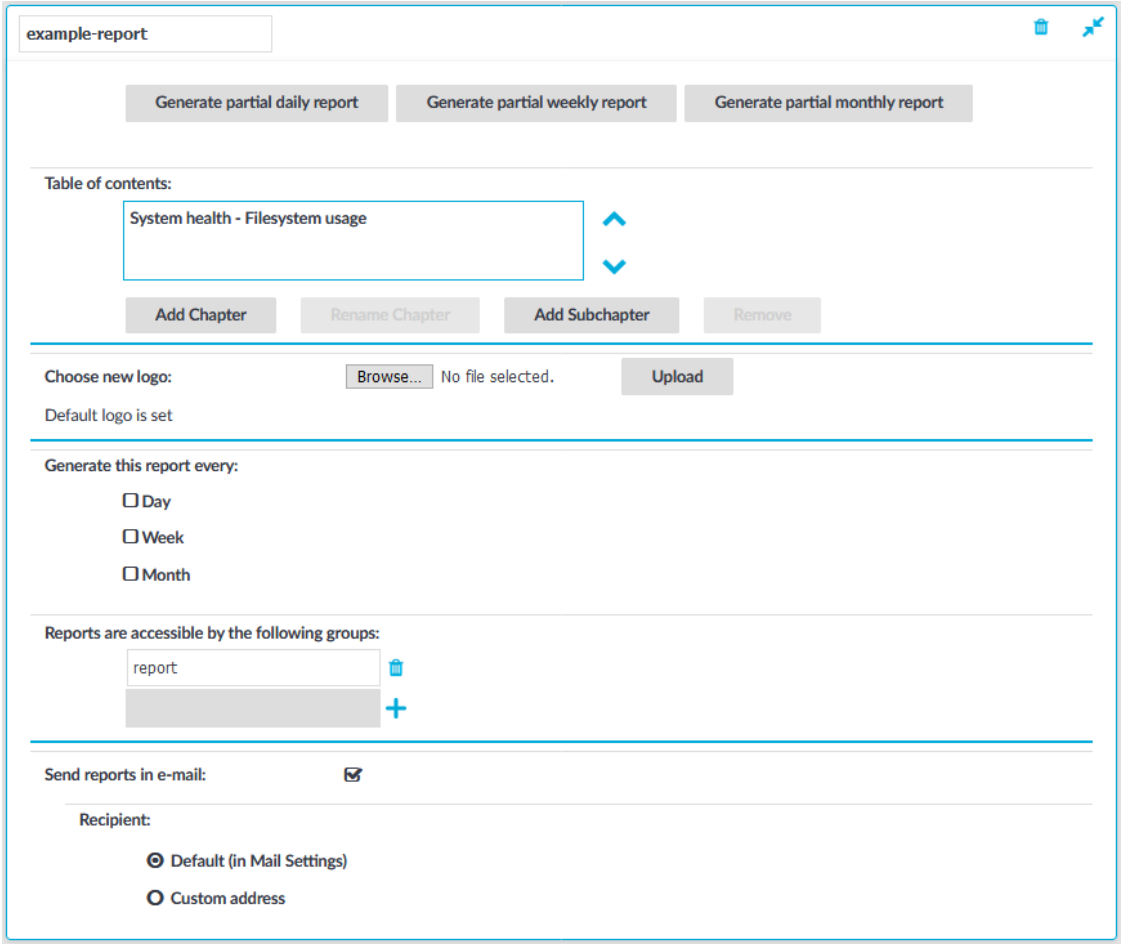
To configure One Identity Safeguard for Privileged Sessions (SPS) to create custom reports, complete the following steps with a user that has read & write/perform access to the **Reporting > Content subchapters** privilege.

For more information on configuring user rights, see Managing user rights and usergroups on page 320.

To configure SPS to create custom reports

1. Login to the SPS web interface, and navigate to **Reporting > Configuration**.

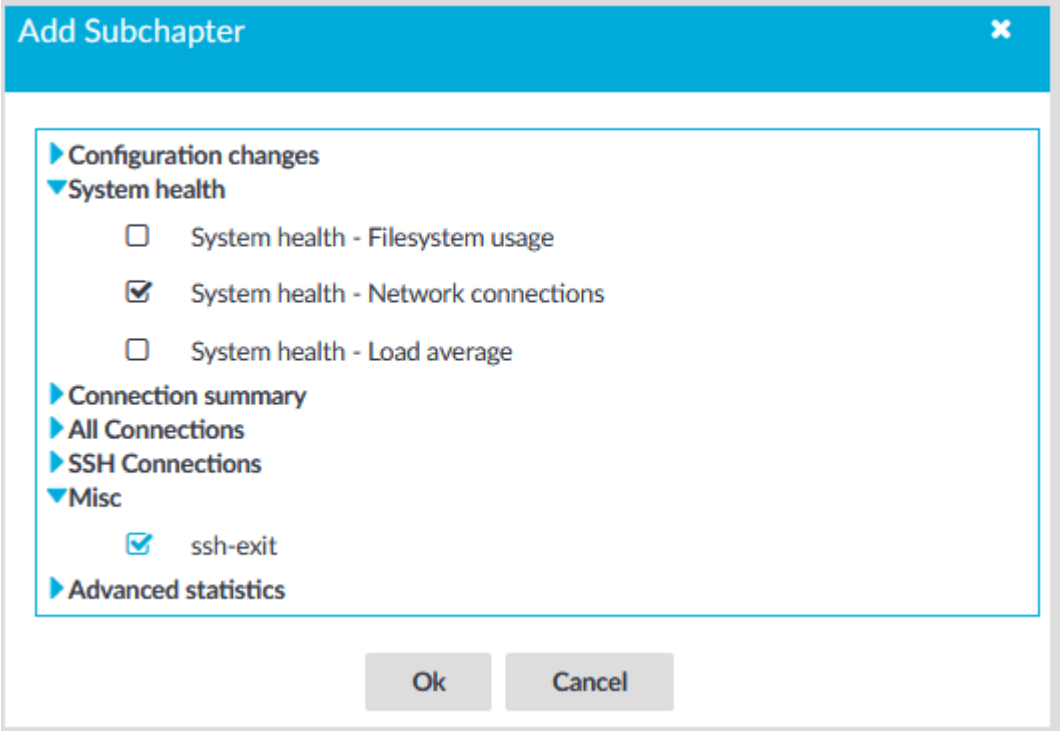
   Figure 265: Reporting > Configuration — Configuring custom reports



2. Click + and enter a name for the custom report.
3. Select **Table of contents > Add Chapter**, enter a name for the chapter, then click **OK**. Repeat this step to create further chapters if needed.
4. If you want to have the logo of your organization on the cover page of the report (instead of the One Identity logo), select **Choose new logo > Browse**, select your logo file, then click **Upload**. You can upload GIF, JPEG, and PNG images. SPS will automatically resize the image to fit on the cover page.
5. Select **Add Subchapter** to add various reports and statistics to the chapter. The available reports will be displayed in a pop-up window.
   - Subchapters created from indexed contents of audit trails are listed under **Misc**.

- Subchapters created from custom statistics are listed under **Search statistics**.
- Subchapters created from custom database queries are listed under **Advanced statistics**.
- Subchapters created from search queries are listed under **Search-based subchapters**.

Figure 266: Reporting > Configuration > Add Subchapter — Adding subchapters to reports



NOTE:

When creating a subchapter that searches for keywords in HTTP, only the timestamp of the results will be visible in the report, without data.

6. Use the arrows to change the order of the subchapters if needed.
7. Select how often shall SPS create the report from the **Generate this report every** field. Weekly reports are created on Mondays, while monthly reports on the first day of the month. If you want to generate the report only manually, leave this field empty.
8. By default, members of the `report` group can access the custom reports via the SPS web interface. To change this, enter the name of a different group into the **Reports are accessible by the following groups** field, or click + to grant access to other groups.

NOTE:

Members of the listed groups will be able to access only these custom reports even if their groups does not have read access to the **Reporting > Reports** page. However, only those reports will be listed, to which their group has access to.

9. By default, SPS sends out the reports in e-mail to the address set in the **Basic Settings > Management > Mail settings > Send reports to** field.

   NOTE:

   If this address is not set, the report is sent to the SPS administrator's e-mail address.

   - To disable e-mail sending, unselect the **Send reports in e-mail** option.
   - To e-mail the reports to a different address, select **Recipient > Custom address**, and enter the e-mail address where the reports should be sent. Click + to list multiple e-mail addresses if needed.

10. Click Commit.

Creating reports from audit trail content

One Identity Safeguard for Privileged Sessions (SPS) can index the contents of audit trails using its own indexer service or external indexers. Indexing extracts the text from the audit trails and segments it to tokens. A token is a segment of the text that does not contain whitespace: for example words, dates (`2009-03-14`), MAC or IP addresses, and so on. The indexer returns the extracted tokens to SPS, which builds a comprehensive index from the tokens of the processed audit trails.

Once indexed, the contents of the audit trails can be searched from the web interface. SPS can extract the commands typed and the texts seen by the user in terminal sessions, and text from graphical protocols like RDP, Citrix ICA, and VNC. Window titles are also detected.

SPS has an internal indexer, which runs on the SPS appliance. In addition to the internal indexer, external indexers can run on Linux hosts.

Processing and indexing audit trails requires significant computing resources. If you have to audit lots of connections, or have a large number of custom reports configured, consider using an external indexer to decrease the load on SPS. For sizing recommendations, ask your One Identity partner or contact our Support Team.

SPS also creates statistics of the occurrences of the search keywords, as well as screenshots from the audit trail. These statistics and screenshots can be included in custom reports as subchapters.

NOTE:

- The screenshot generated from the search results contains the first occurrence of the search keywords. If your search keywords are visible in the audit trail for a longer period, it is possible that the first occurrence is not the most relevant.
- For technical reasons, trail data in terminal connections (SSH and Telnet) is aggregated for each second. The screenshot generated for the report reflects the terminal buffer, as it was visible at the end of that second. If data that contains the search keyword was pushed off-screen during this second, the search still finds it, but it will not be visible on the generated screenshot. Similarly, if you search for multiple keywords, it is possible that you will receive results that do not contain every keyword on the same screen (but they were separately visible within the one-second interval).

NOTE:

Only audit trails created after the content subchapter has been configured will be processed. It is not possible to create reports from already existing audit trails.

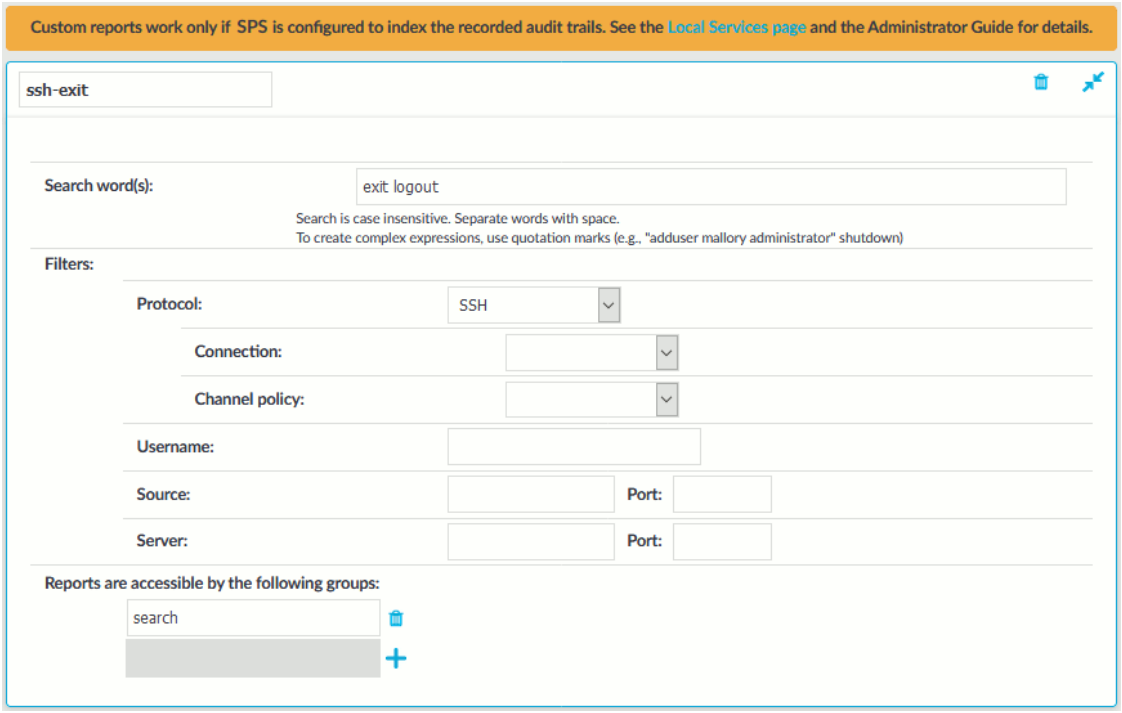
Prerequisites for the indexer service

If you are indexing the audit trails with the indexer service, make sure to enable indexing for the connections you want to include in the report. Otherwise, reporting on audit trail content will not work. For details, see Configuring the internal indexer.

To configure SPS to create reports from the contents of audit trails

1. Login to the SPS web interface, and navigate to **Reporting > Content subchapters**.

Figure 267: Reporting > Content subchapters — Configuring audit-trail reports



2. Click + and enter a name for the subchapter.
3. Enter the search keywords (or parts of the words) into the **Search word(s)** field. Note the following points.
   - Your search expression must be shorter than 255 characters.
   - The search is not case sensitive.
   - Wildcards and regular expressions are not supported.
   - To search for an exact phrase or expression, enclose the keywords in double quotes, for example `"program files"`.
4. Configure filters to select the audit trails to index. The following filters are available:
   - **Protocol**: Process only audit trails of the specified traffic type (for example SSH).
   - **Connection**: Process only audit trails of the specified connection policy.
   - **Channel policy**: Process only audit trails of the specified channel policy.
   - **Username**: Process only audit trails where the specified username was used in the connection. Available only for protocols where the username is known (for example SSH).
   - **Source**: Process only audit trails where the specified client IP address or port was used.

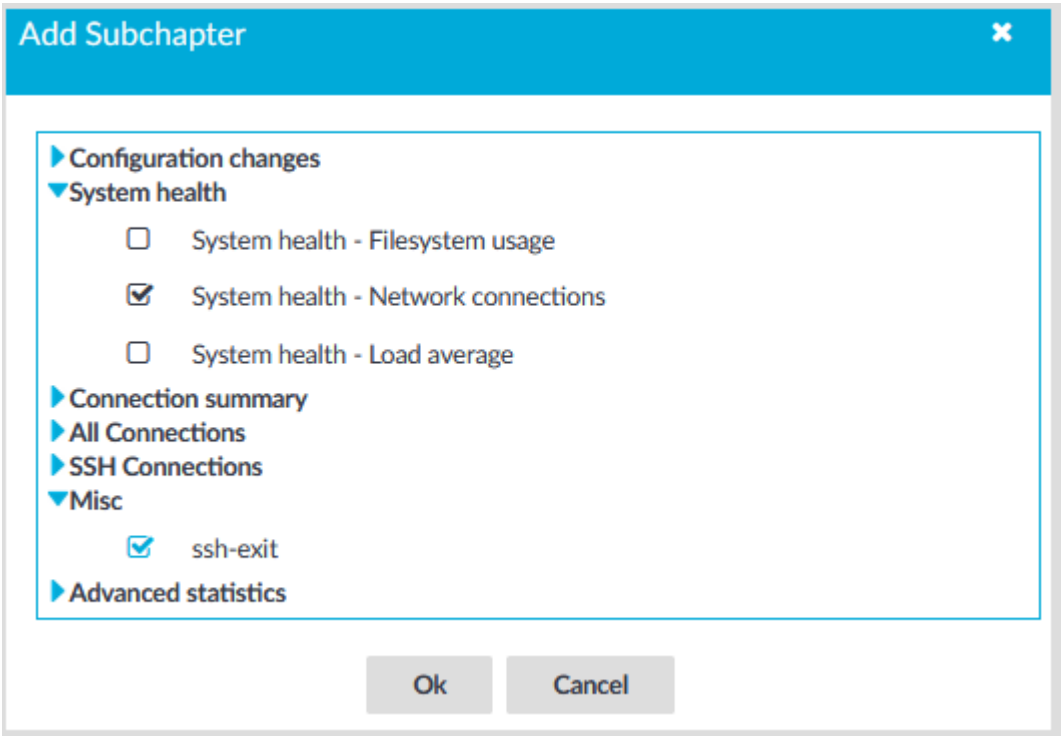
- **Server**: Process only audit trails where the specified server IP address or port was used.

NOTE:

If you do not configure any filters, every available audit trail will be processed. Audit trails are created only for channels where the **Record audit trail** option is enabled for the particular channel in the channel policy.

5. Click Commit.
6. Navigate to **Reporting > Configuration**, and add the new subchapter to an existing report, or create a new report. Subchapters created from searching the contents of audit trails are listed under **Misc**. For details, see Configuring custom reports.

Figure 268: Reporting > Configuration > Add Subchapter — Adding subchapters to reports



Creating report subchapters from search queries

NOTE:

Creating report subchapters from search queries is currently an experimental feature of One Identity Safeguard for Privileged Sessions (SPS), therefore One Identity recommends that only administrators use this feature and only at their own risk.

You can turn any search query or statistics into a subchapter to add to your reports. This is an easy and flexible way of creating reports to monitor traffic, track certain parameters, or get alerted about particular events. The Search interface allows you to:

- Create search-based report subchapters from search results.
- Create search-based report subchapters from scratch.

Creating search-based report subchapters from search results

NOTE:

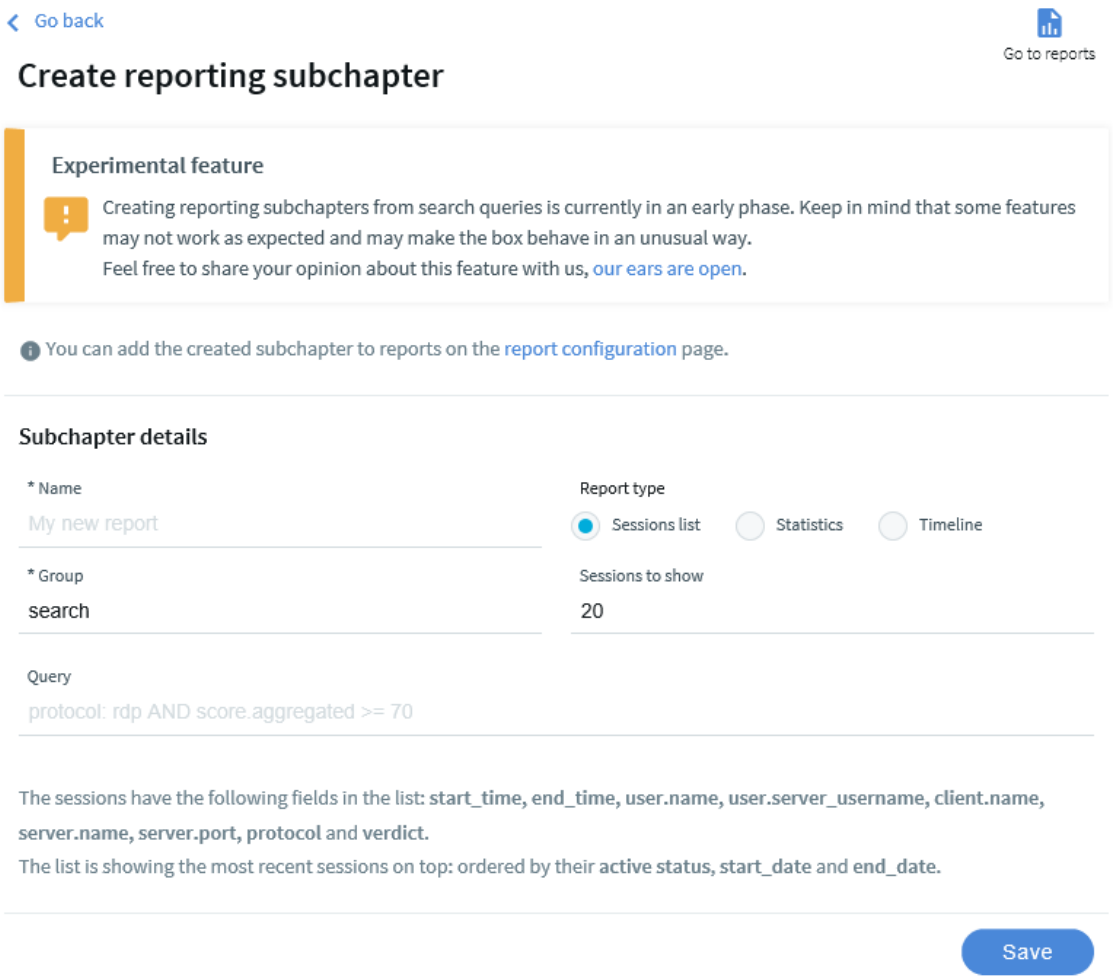
Creating report subchapters from search queries is currently an experimental feature of One Identity Safeguard for Privileged Sessions (SPS), therefore One Identity recommends that only administrators use this feature and only at their own risk.

The following describes how to create a search-based report subchapter from search results.

To create a search-based report subchapter from search results

1. Navigate to **Search**, and perform a query of your choice.
2. Click **Search**. Search results are displayed.
3. Click 

. The **Create reporting subchapter** page is displayed, with the **query** field populated with your query.



4. In the **name** field, add a name to your report.
5. In **Report type**, select the type that fits your query. You can choose from the following types:
   - **Sessions list**: Displays a list of sessions.

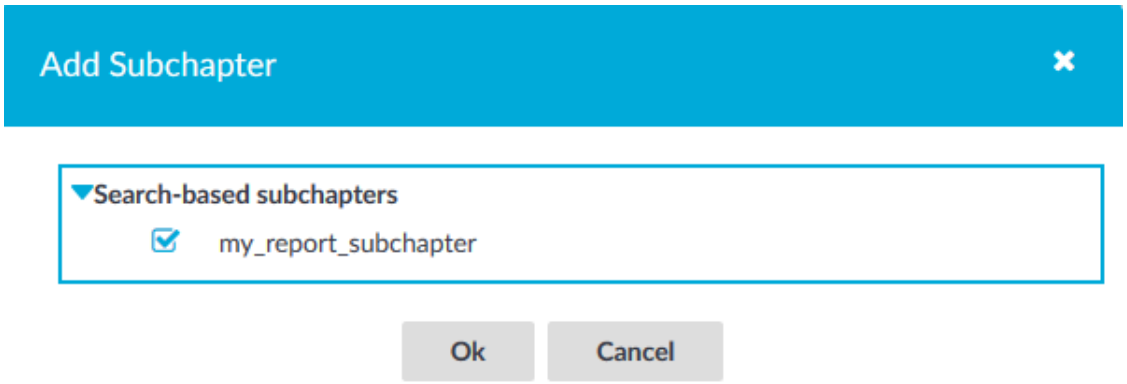
     Set the number of **sessions to show** in the report as required.
   - **Statistics**: Visualizes the distribution of sessions based on the selected metadata.

     Select a **Statistic presentation** for your report, such as **Pie chart**, **List**, **Bar chart**. Select the field (the metadata) to create your statistics on.
   - **Timeline**: Visualizes the distribution of sessions within a day/week/month, depending on the time range chosen for the report under **Reporting > Configuration > Generate this report every > Day/Week/Month**.
6. Click **Save**.
7. Click 

. Alternatively, navigate to **Reporting > Configuration**.

8. Configure a custom report from scratch, or add the subchapter to an existing report. For details, see Configuring custom reports on page 778.

   When adding the subchapter you created, look for it under **Search-based subchapters**.



Creating search-based report subchapters from scratch

> NOTE:
>
> Creating report subchapters from search queries is currently an experimental feature of One Identity Safeguard for Privileged Sessions (SPS), therefore One Identity recommends that only administrators use this feature and only at their own risk.

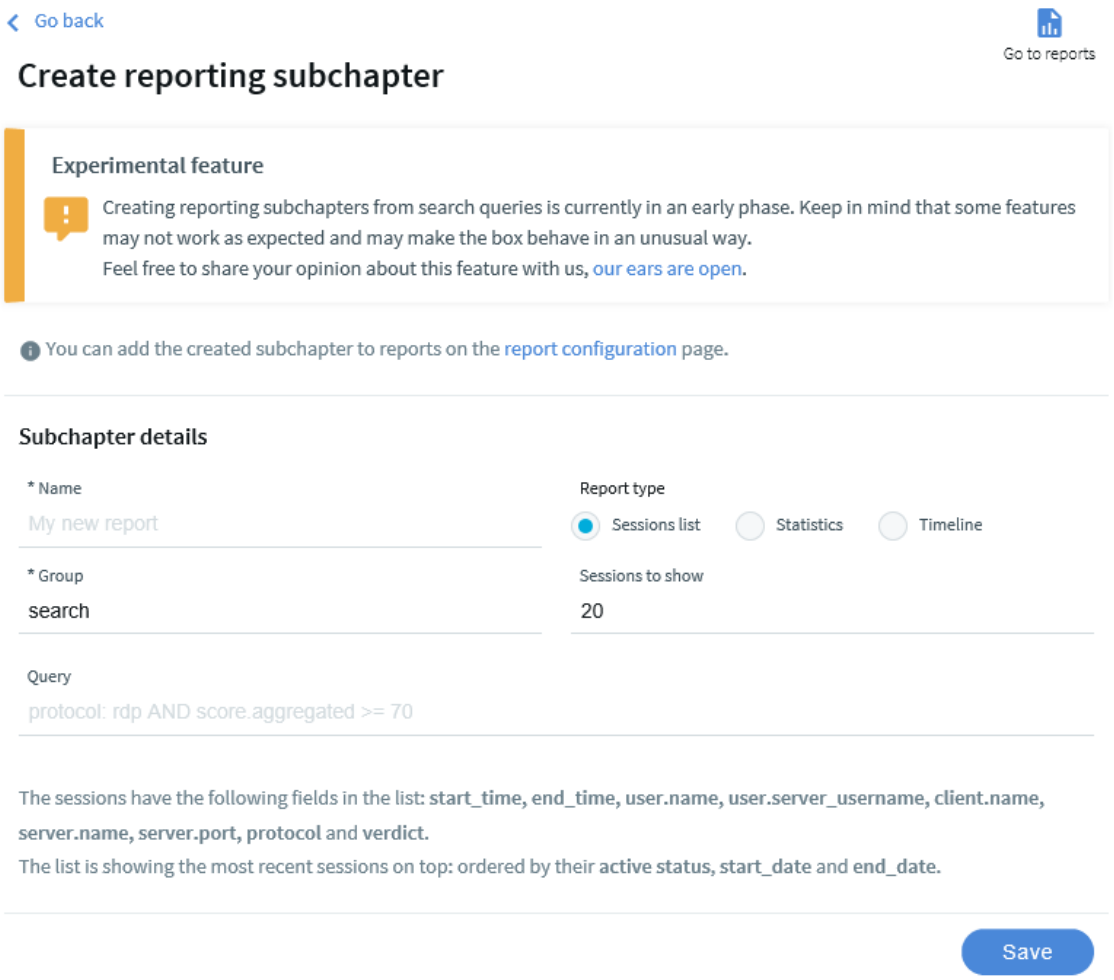
The following describes how to create a search-based report subchapter from scratch.

To create a search-based report subchapter from scratch

1. If you have multiple SPS appliances and they are organized into a cluster where one of the nodes is the Search Master (or Central Search) node, log in to that node.
2. Navigate to **Reporting > Search subchapters**.

3. Click + Add new. The **Create reporting subchapter** page is displayed.

Figure 269: Reporting > Search subchapters — Create reporting subchapter



4. In the **name** field, add a name to your report.
5. In the **query** field, type the query that you want to create a report from.
6. In **Report type**, select the type that fits your query. You can choose from the following types:
   - **Sessions list**: Displays a list of sessions.

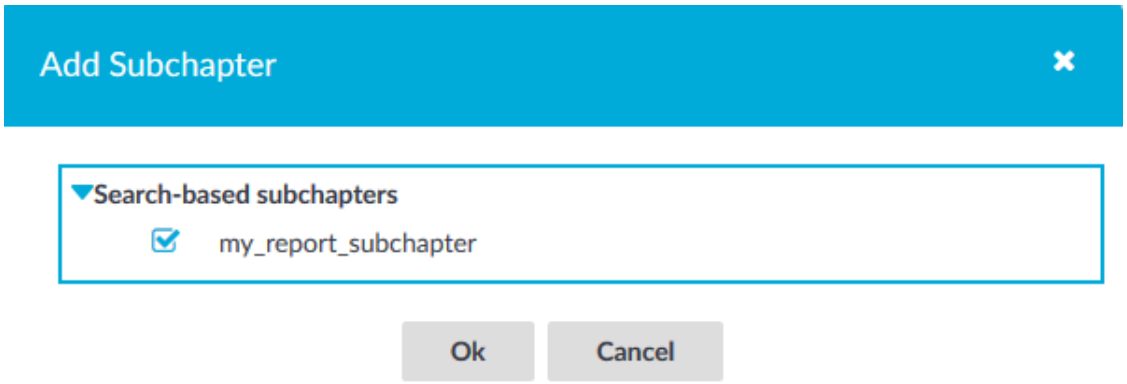
     Set the number of **sessions to show** in the report as required.
   - **Statistics**: Visualizes the distribution of sessions based on the selected metadata.

     Select a **Statistic presentation** for your report, such as **Pie chart**, **List**, **Bar chart**. Select the field (the metadata) to create your statistics on.

- **Timeline**: Visualizes the distribution of sessions within a day/week/month, depending on the time range chosen for the report under **Reporting > Configuration > Generate this report every > Day/Week/Month**.

7. Click **Save**.
8. Click Go to reports . Alternatively, navigate to **Reporting > Configuration**.
9. Configure a custom report from scratch, or add the subchapter to an existing report. For details, see Configuring custom reports on page 778.

   When adding the subchapter you created, look for it under **Search-based subchapters**.



Creating statistics from custom database queries

The following describes how to create statistics from any custom queries from the One Identity Safeguard for Privileged Sessions (SPS) connection database. These custom statistics can be added to regular reports.

CAUTION:

Hazard of denial of service (DoS). This feature of SPS allows the user to execute read-only queries on the database of SPS. If the database is large (stores the data of many connections), and the query is not optimal, executing the query can consume significant CPU and memory resources, severely degrading the performance of SPS. Use this feature only if you possess the required knowledge about SQL queries.

To create statistics from any custom queries from the SPS connection database

1. Navigate to the **Reporting > Advanced statistics** page and click +.
2. Enter a name for the statistics. The created statistics will be available for reports under this name as a subchapter.
3. Enter the SQL query that returns the data you need into the **Query** field. Note the following important points:
   - The query must be a full PostgreSQL query.
   - SQL queries used for pie and bar charts must return a `title` and a `cnt` column, in this order. For example:

     ```
     select
       remote_username as title,
       count(*) as cnt
     from channels
     group by title
     ```

   - The query can be executed on the database tables and views that contain metadata about the audited connections, as well as the content of the audited connections (for example, the commands executed in a session) if indexing is used. Note that these tables do not contain any data from the upstream traffic, that is, passwords entered by the users are not available in the database.
   - Limit the query to avoid unnecessarily long results, for example, `LIMIT = 5000`. Note that SPS automatically limits the results to 10000 entries (this is a hard limit, you cannot increase it).

     CAUTION:

     Generating a report that includes an Advanced statistics chapter that returns several thousands of entries requires significant CPU and memory resources from One Identity Safeguard for Privileged Sessions (SPS). While generating such a partial report, the web interface of SPS can become slow or unresponsive.

   - The structure of the accessible tables may change in future versions of SPS. For details about the tables and their contents, see Database tables available for custom queries on page 794.

   The query can include the following macros: `:range_start`, `:range_end`. When including the statistics in a report, these macros will refer to the beginning and end dates of the reported interval. When clicking **Preview**, the macros will refer to the start and end of the current day.

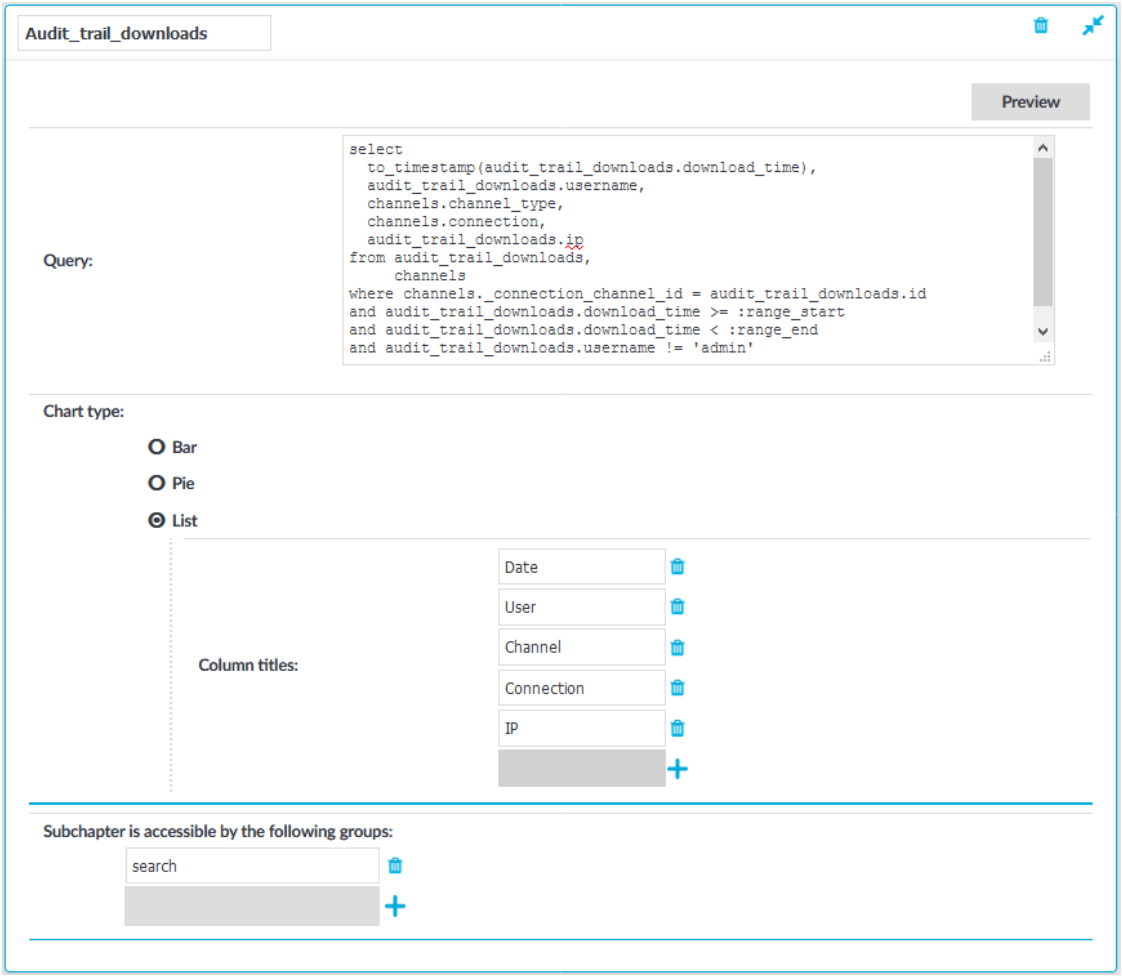
Example

The following query generates a list of audit trail downloads within the reported interval (using standard date formatting), excluding administrator downloads:

Example

```
select
  to_timestamp(audit_trail_downloads.download_time),
  audit_trail_downloads.username,
  channels.channel_type,
  channels.connection,
from audit_trail_downloads,
     channels
where channels._connection_channel_id = audit_trail_downloads.id
and audit_trail_downloads.download_time >= :range_start
and audit_trail_downloads.download_time < :range_end
and audit_trail_downloads.username != 'admin'
order by audit_trail_downloads.download_time;
```

Figure 270: Reporting > Advanced statistics — Creating custom database queries



4. Select the type of chart to display, that is, **Bar**, **Pie** or **List**.
   - For bar charts, enter the name of the Y axis into the **Y axis title** field.
   - For lists, you can customize the name of the columns in the list by clicking + and entering the name of the column into the **Column titles** field.
5. Click **Preview** to test the query.
6. (Optional) By default, users of the **search** group can add these statistics to reports. To specify other groups, select **Subchapter is accessible by the following groups** and click +.

   NOTE:

   Accessing advanced statistics subchapters requires the *Reporting > Advanced statistics* privilege.

7. Click 

to save the query.
8. Add this new subchapter to a report. For details on how to add this subchapter to a selected report, see Configuring custom reports on page 778

Database tables available for custom queries

This section describes the database tables, views, and functions of One Identity Safeguard for Privileged Sessions (SPS) that can be used in the custom queries of the **Reporting > Advanced statistics** page. Generally, views contain a more organized dataset, while tables contain the raw data.

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 13: Database tables and views for custom queries

Database table	Type	Description
`alerting`	table	The list of alerting events. For details, see The alerting table on page 796.
`aps`	table	[OBSOLETE] The list of Audit Player indexing services that are available for SPS. For details, see The aps table on page 796.
`archives`	table	Data about the archiving processes. For details, see The archives table on page 797.
`audit_trail_downloads`	table	Data about the audit trail downloads. For details, see The audit_trail_downloads table on page 797.
`channels`	table	Contains metadata about the channel-opening requests and opened channels. This is the main table storing data about the connections. For details, see The channels table on page 798.
`closed_connection_audit_channels`	view	This view returns all audited channels whose connection have been closed. For details, see The closed_connection_audit_channels view on page 803.
`closed_not_indexed_audit_channels`	view	This view returns all audited channels whose connection have been closed, but have not been indexed yet. For details, see The closed_not_indexed_audit_channels view on page 803.
`connection_`	view	List of commands or window titles detected in the connections.

Database table	Type	Description
events		For details, see The connection_events view on page 804.
connection_occurrences	view	Contains the tokens that are used as search keywords in Content subchapter reports (reports from audit-trail content) and where these tokens appear in the audit trails. For details, see The connection_occurrences view on page 805.
connections	view	A view containing data of the connections. This data is identical to the information available on the **Search** page. For details, see The connections view on page 805.
events	table	The commands or events extracted from the indexed audit trails. For details, see The events table on page 808.
file_xfer	table	Data about the files transfered in the audited connections (SCP, SFTP). For details, see The file_xfer table on page 809.
http_req_resp_pair	table	Information about the requests and responses in HTTP and HTTPS sessions. For details, see The http_req_resp_pair table on page 809.
indexer_jobs	table	Information and statistics about indexer jobs. For details, see The indexer_jobs table on page 810.
occurrences	table	Contains the tokens that are used as search keywords in Content subchapter reports (reports from audit-trail content) and where these tokens appear in the audit trails. For details, see The occurrences table on page 811.
progresses	table	[OBSOLETE] Which audit trail is assigned to which Audit Player for processing. For details, see The progresses table on page 811.
results	table	Contains the tokens that are used as search keywords in Content subchapter reports (reports from audit-trail content) and in which audit trails were these tokens found. For details, see The results table on page 812.
skipped_connections	table	List of errors encountered when processing audit trails. For details, see The skipped_connections table on page 812.
usermapped_channels	view	Information about sessions where usermapping was performed in the connection. For details, see The usermapped_channels view on page 812.

To search the content of audit trails that were processed using indexing, you can use the `lucene` SQL function. For details, see Querying trail content with the lucene-search function on page 817.

The alerting table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 14: Columns of the alerting table

Column	Type	Description
`alerting_time`	timestamp	The timestamp of the alert.
`alerting_type`	text	The type of the alert.
`channel_id`	integer	This value is a reference to the ID of the channels table where the event occurred.
`matched_content`	text	The matched content.
`matched_regexp`	text	The matched regular expression.
`rule_name`	text	The name of the content policy rule.

The aps table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

This table contains information only about Audit Player indexers. It does not contain any information about the indexer service.

Table 15: Columns of the aps table

Column	Type	Description
`ap_id`	integer	[OBSOLETE] The ID of the Audit Player indexer service that is processing the audit trail.
`dead`	boolean	Set to 1 if the Audit Player indexer service on this host is considered to be unavailable.
`id`	integer	The unique ID number of the entry.

Column	Type	Description
last_poll	integer	The timestamp of the last time when the Audit Player indexer service on this host requested an audit trail from SPS.
remote_addr	text	[OBSOLETE] The address of the host running the Audit Player indexer service.

The archives table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 16: Columns of the archives table

Column	Type	Description
id	integer	The unique ID number of the entry.
orig_filename	text	The original name of the file, as stored on SPS.
policy_id	text	The ID of the archiving policy that archived the file.
saved_filename	text	The name of the archive file containing the file.
server	text	The address of the server where the file was archived.
type	text	Indicates the type of the file: `audit` or `index`.

The audit_trail_downloads table

This table contains information about the downloaded audit trails.

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 17: Columns of the audit_trail_downloads table

Column	Type	Description
channel_id	integer	This value is a reference to the ID of the channels table where the download occurred.
download_	integer	Time when the download was started.

Column	Type	Description
time		
from_api	boolean	The value is true, if it has been downloaded through (RPC) API, and false if through the web user interface.
id	integer	The unique ID of the entry.
ip	text	The client IP address.
username	text	The username of the downloader on the web user interface. If indexer is used, then the login username.

The channels table

For details of the different columns, see Connection metadata on page 958.

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 18: Columns of the channels table

Column	Type	Description
application	text	**Application**: The name of the application accessed in a seamless Citrix ICA connection.
_archive_date	date	**Archive date**: The date when the connection was archived or cleaned up.
_archive_path	text	**Archive path**: The path where the audit trail was archived on the remote server.
_archive_policy	text	
_archive_server	text	**Archive server**: The hostname or IP address of the remote server where the audit trail was archived.
audit	text	**Audit-trail**: Name and ID of the audit file storing the traffic of the channel. If the session has an audit trail, a icon is displayed. Note that a the following letters may appear on the download icon:
auth_method	text	**Authentication method**: The authentication method used in the connection. For example, password
channel_policy	text	**Channel policy**: The Channel policy applied to connection. The Channel policy lists the channels (for example, terminal

Column	Type	Description
		session and SCP in SSH, or Drawing and Clipboard in RDP) that can be used in the connection, and also determines if the channel is audited or not. The Channel policy can also restrict access to each channel based on the IP address of the client or the server, a user list, user group, or a time policy.
channel_type	text	**Channel type**: Type of the channel.
client_x509_subject	text	**Client X.509 Subject**: The client's certificate in TELNET or VNC sessions. Available only if the **Client-side transport security settings > Peer certificate validation** option is enabled in SPS.
_close_cleanup	boolean	
command_extracted	boolean	The value is `true` if the window title and the command has been extracted real-time (during alerting) so that the indexer service does not have to extract them again. It is `false` if they will be extracted only during indexing.
_connection_channel_id	integer	**Unique connection ID**: The unique identifier of the connection.
connection_id	text	**Connection policy ID**: The identifier of the connection policy.
connection	text	**Connection policy**: The connection policy that handled the client's connection request.
device_name	text	**Device name**: The name or ID of the shared device (redirect) used in the RDP connection.
dst_ip	text	**Destination IP**: The IP address of the server as requested by the client.
dst_port	integer	**Destination port**: The port number of the server as requested by the client.
dynamic_channel	text	**Dynamic channel**: The name or ID of the dynamic channel opened in the RDP session.
exec_cmd	text	**Exec command**: The command executed in a **Session exec** channel.
four_eyes_authorizer	text	**Four-eyes authorizer**: The username of the user who authorized the session. Available only if 4-eyes authorization is required for the channel. For details on 4-eyes authorization, see Configuring four-eyes authorization on page 752.

Column	Type	Description
four_eyes_description	text	**Four-eyes description**: The description submitted by the authorizer of the session.
id	integer	The unique ID of the channel.
index_status	integer	**Channel's indexing status**: Shows if the channel has been indexed. The following values are possible: • CHANNEL_OPEN (0): The connection of the channel is still open (indexer is waiting for the connection to close). • NOT_INDEXED (1): All channels of the connection have been closed which belong to the connection. The channel is ready for indexing, unless the audit trail was placed in the skipped_connections queue. • INDEXING_IN_PROGRESS (2): The channel is being indexed (indexing in progress). Note that SPS will return search results for the parts of the channel are already indexed. • INDEXED (3): Indexing the channel is complete. • INDEXING_NOT_REQUIRED (4): Indexing not required (indexing is not enabled for the connection). • INDEXING_FAILED (5): Indexing failed. The indexer service writes the corresponding error message in the error_message column of the indexer_jobs table. Note that SPS will return search results for the parts of the channel that were successfully indexed before the error occurred. For example, if the error occurred at the end of a long audit trail, you can still search for content from the first part of the audit trail. • NO_TRAIL (6): Auditing is not enabled for the channel.
local_ip	text	**Server-local IP**: The IP address of SPS used in the server-side connection.
local_port	integer	**Server-local port**: The port number of SPS used in the server-side connection.
originator_addr	text	**Port/X11 forward originator IP**: The IP address of the host initiating the channel in **Remote Forward** and **Local Forward** channels. Note that this host is not necessarily the client or the server of the SSH connection.
originator_port	integer	**Port/X11 forward originator port**: The number of the forwarded port in **Remote Forward** and **Local Forward** channels.

Column	Type	Description
protocol	text	**Protocol**: The protocol used in the connection (Citrix ICA, HTTP, RDP, SSH, Telnet, or VNC).
remote_username	text	**Username on server**: The username used to log in to the remote server. This username can differ from the client-side username if usermapping is used in the connection. For details on usermapping, see Configuring usermapping policies on page 740.
rule_num	text	**Rule number**: The number of the line in the Channel policy applied to the channel.
scp_path	text	**SCP path**: Name and path of the file copied via SCP. Available only for SCP sessions (**Session exec SCP** SSH channels) if the **Log file transfers to database** option is enabled in the Channel Policy of the connection.
server_ip	text	**Server IP**: The IP address of the server connected by SPS.
server_port	integer	**Server port**: The port number of the server connected by SPS.
session_end	integer	**End time**: Date when the channel was closed.
session_id	text	**Session ID**: A globally unique string that identifies the session. This session ID has the following format: *svc/<unique-random-hash>/<name-of-the-connection-policy>:<session-number-since-service-started>/<protocol>*, for example, *svc/5tmEaM7xdNi1oscgVWpbZx/ssh_console:1/ssh*. Log messages related to the session also contain this ID. For example: `2015-03-20T14:29:15+01:00 demo.example zorp/scb_ssh[5594]: scb.audit(4): (svc/5tmEaM7xdNi1oscgVWpbZx/ssh_console:0/ssh): Closing connection; connection='ssh_console', protocol='ssh', connection_id='409829754550c1c7a27e7d', src_ip='10.40.0.28', src_port='39183', server_ip='10.10.20.35', server_port='22', gateway_username='', remote_username='example-username', verdict='ZV_ACCEPT'`
session_start	integer	**Start time**: Date when the channel was started.

Column	Type	Description
src_ip	text	**Source IP**: The IP address of the client.
src_port	integer	**Source port**: The port number of the client.
subsystem_name	text	**Subsystem name**: Name of the SSH subsystem used in the channel.
target_addr	text	**Port-forward target IP**: The traffic was forwarded to this IP address in **Remote Forward** and **Local Forward** channels.
target_port	integer	**Port-forward target port**: The traffic was forwarded to this port in **Remote Forward** and **Local Forward** channels.
username	text	**Username**: The username used in the session. • If the user performed inband gateway authentication in the connection, the field contains the username from the gateway authentication (gateway username). • Otherwise, the field contains the username used on the remote server.
verdict	text	**Verdict**: Indicates what SPS decided about the channel. • ACCEPT: Accepted. • ACCEPT-TERMINATED: Connection was accepted and established, but a content policy terminated the connection. For details on content policies, see Real-time content monitoring with Content Policies on page 451. • CONN-AUTH-FAIL: User authentication failed. • CONN-DENY: Connection rejected. • CONN-FAIL: Connection failed, that is, it was allowed to pass SPS but timed out on the server. • CONN-GW-AUTH-FAIL: Gatway authentication failed. • CONN-KEY-ERROR: Hostkey mismatch. • CONN-USER-MAPPING-FAIL: Usermapping failed. • DENY: Denied. • FOUR-EYES-DEFERRED: Waiting for remote username. • FOUR-EYES-ERROR: Internal error during four-eyes authorization. • FOUR-EYES-REJECT: Four-eyes authorization rejected. • FOUR-EYES-TIMEOUT: Four-eyes authorization timed out.
window_	boolean	The value is true if the window title and the command has

Column	Type	Description
title_extracted		been extracted real-time (during alerting) so that the indexer service does not have to extract them again. It is false if they will be extracted only during indexing.

The closed_connection_audit_channels view

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

This view returns all audited channels whose connection have been closed. The view is defined as follows:

```
create view closed_connection_audit_channels
as
select *
from channels
where audit is not null
and index_status = 1;
```

For details on the returned columns, see The channels table on page 798.

The closed_not_indexed_audit_channels view

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

This view returns all audited channels whose connection have been closed, but have not been indexed yet. The view is defined as follows:

```
create view closed_not_indexed_audit_channels
as
select *
from channels
where audit is not null
and (index_status = 1
or index_status = 2);
```

For details on the returned columns, see The channels table on page 798.

The connection_events view

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

For terminal connections, this view collects the commands issued in a connection. For graphical connections, this view collects the window titles detected in the connection. The view is defined as follows:

```
select
    channels._connection_channel_id as id,
    events.event,
    events.printable
from channels,
    events
where channels.id = events.channel_id;
```

Querying the table (for example, **select * from connection_events limit 10;**) will return results similar to the following:

```
 id |                                          event              | printable
----+-------------------------------------------------------------+-----------
1  | [user@exampleserver ~]$ ls                                   | t
1  | [user@exampleserver ~]$ exit                                 | t
2  | [user@exampleserver ~]$ su -                                 | t
2  | Password:                                                    | t
2  | [root@exampleserver ~]#                                      | t
2  | [root@exampleserver ~]# ifconfig                             | t
2  | [root@exampleserver ~]# ifconfig                             | t
2  | [root@exampleserver ~]# ifconfig                             | t
4  | [user@exampleserver ~]$                                      | t
4  | [user@exampleserver ~]$                                      | t
```

The connection_events view has the following columns.

Table 19: Columns of the connection_events table

Column	Type	Description
`event`	text	The command executed, or the window title detected in the channel (for example, **ls**, **exit**, or **Firefox**).
`id`	integer	The unique ID number of the entry.
`printable`	boolean	Set to 1 if every character of the command can be displayed.

The connection_occurrences view

The view is defined as follows:

```
select
    channels._connection_channel_id as id,
    results.token,
    occurrences.start_time,
    occurrences.end_time,
    occurrences.screenshot
from channels,
     results,
     occurrences
where channels.id = results.channel_id
and results.id = occurrences.result_id;
```

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 20: Columns of the connection_occurrences table

Column	Type	Description
end_time	integer	**End time**: Date when the channel was closed.
id	text	The unique id of the entry.
screenshot	text	The filename of the PNG screenshot (as stored on SPS) about the occurrence of the search token.
start_time	integer	**Start time**: Date when the channel was started.
token	text	The search token visible on the screenshot.

The connections view

This view collects the metadata of the connections. The view is defined as follows:

```
select
    channels."connection",
    channels.protocol,
    channels._connection_channel_id as id,
    channels.connection_id,
    min(channels.session_start) as session_start,
    max(channels.session_end) as session_end,
```

```
    max(channels.src_ip) as src_ip,
    max(channels.src_port) as src_port,
    max(channels.server_ip) as server_ip,
    max(channels.server_port) as server_port,
    max(channels.username) as username,
    max(channels.remote_username) as remote_username,
    max(channels.channel_policy) as channel_policy,
    sum(case
      when channels.session_end is null then 1
      else 0
    end) as active
from channels
group by channels._connection_channel_id,
         channels.protocol,
         channels."connection",
         channels.connection_id;
```

Querying the table (for example, **select * from connections limit 10;**) will return results similar to the following:

```
connection  | protocol |   id   |      connection_id      | session_start |
session_end |    src_ip     | src_port |  server_ip  | server_port | username  |
remote_username | channel_policy | active
------------+----------+--------+-------------------------+---------------+----
---------+--------------+----------+-------------+-------------+-----------+---
-------------+----------------+--------
SSH_Access2 | ssh      | 1      | 5516465814bc36d5570ec8  |    1271098736 |
1271099582 | 192.168.0.62   |     4312 | 192.168.0.20 |          22 | joe     |
joe             | shell-only     |      0
SSH_Access  | ssh      | 10     | 20790868454bc33027964a0 |    1271258787 |
1271259645 | 10.100.58.27   |     2298 | 192.168.0.20 |          22 | joe     |
joe             | shell-only     |      0
SSH_Access  | ssh      | 100    | 20790868454bc33027964a0 |    1272391671 |
1272396886 | 10.100.58.14   |    51342 | 192.168.0.20 |          22 | phil    |
phil            | shell-only     |      0
SSH_Access  | ssh      | 1000   | 20790868454bc33027964a0 |    1274450541 |
1274475742 | 10.100.56.14   |     4633 | 192.168.0.20 |          22 | rick    |
rick            | all            |      0
SSH_Access2 | ssh      | 10000  | 5516465814bc36d5570ec8  |    1282753195 |
1282764804 | 192.168.40.34  |    53097 | 192.168.0.20 |          22 | vivian  |
vivian          | shell-only     |      0
SSH_Access2 | ssh      | 100000 | 5516465814bc36d5570ec8  |    1314979916 |
1314986038 | 192.168.40.85  |    34743 | 192.168.0.20 |          22 | elliot  |
elliot          | Shell-SCP      |      0
SSH_Access2 | ssh      | 100001 | 5516465814bc36d5570ec8  |    1314979917 |
1314984561 | 192.168.40.65  |    56405 | 192.168.0.20 |          22 | root    |
root            | Shell-SCP      |      0
SSH_Access2 | ssh      | 100002 | 5516465814bc36d5570ec8  |    1314979940 |
1314984171 | 192.168.40.100 |     1082 | 192.168.0.20 |          22 | allen   |
```

```
allen           | Shell-SCP      |      0
SSH_Access2 | ssh      | 100003 | 5516465814bc36d5570ec8  |    1314979955 |
1314981233 | 192.168.40.10  |    34263 | 192.168.0.20 |          22 | steve   |
steve           | Shell-SCP      |      0
SSH_Access2 | ssh      | 100004 | 5516465814bc36d5570ec8  |    1314980025 |
1314991838 | 192.168.40.33  |    58500 | 192.168.0.20 |          22 | clark   |
clark           | Shell-SCP      |      0
(10 rows)
```

The connections view has the following columns. For details of the different columns, see Connection metadata on page 958.

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 21: Columns of the connections view

Column	Type	Description
active	bigint	
channel_policy	text	The name of the Channel policy that applied to the particular channel of the connection.
connection	text	The name of the Connection Policy, as configured on the SPS web interface.
connection_id	text	The unique ID of the TCP connection.
id	text	The ID of the channel within the connection.
protocol	text	**Protocol**: The protocol used in the connection (Citrix ICA, HTTP, RDP, SSH, Telnet, or VNC).
remote_username	text	**Username on server**: The username used to log in to the remote server. This username can differ from the client-side username if usermapping is used in the connection. For details on usermapping, see Configuring usermapping policies on page 740.
session_end	integer	**End time**: Date when the channel was closed.
session_start	integer	**Start time**: Date when the channel was started.
src_ip	text	**Source IP**: The IP address of the client.
src_port	integer	**Source port**: The port number of the client.
username	text	**Username**: The username used in the session.

Column	Type	Description
		• If the user performed inband gateway authentication in the connection, the field contains the username from the gateway authentication (gateway username). • Otherwise, the field contains the username used on the remote server.

The events table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 22: Columns of the events table

Column	Type	Description
`id`	integer	The unique ID number of the entry.
`channel_id`	integer	The ID of the channel. This value is actually a reference to the id column of the channels table.
`printable`	boolean	Returns FALSE if text contains control characters or characters that do not have any output or control function at all.
`time`	timestamp	The time when the command was executed.
`record_id`	bigint	The identifier of the event within the audit trail (.zat file).
`type`	adp_type	The type of the event. Possible values: • `adp.event.command`: The command entered in SSH or Telnet. • `adp.event.screen.content`: Screen content. • `adp.event.screen.creditcard`: Credit card numbers detected. Displayed only in the alert table, not visible in the Event field. • `adp.event.screen.windowtitle`: The title of the window in graphic protocols (RDP only).
`event`	text	The screen content, command, or window title that was detected.

The file_xfer table

This table contains information about the files transferred the connections.

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 23: Columns of the file_xfer table

Column	Type	Description
`channel_id`	integer	This value is a reference to the ID of the channels table where the file transfer occurred.
`details`	text	The detailed description of the file transfer. The exact contents of this field depend on the protocol used for the file transfer.
`event`	text	The type of the file operation that occurred, for example, `Create file`.
`filename`	text	The name of the file affected by the file operation.
`path`	text	**SCP path**: Name and path of the file copied via SCP. Available only for SCP sessions (**Session exec SCP** SSH channels) if the **Log file transfers to database** option is enabled in the Channel Policy of the connection.
`id`	integer	The unique ID of the entry
`start_time`	integer	**Start time**: Date when the channel was started.

The http_req_resp_pair table

This table contains information about the requests and responses in HTTP and HTTPS sessions.

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 24: Columns of the http_req_resp_pair table

Column	Type	Description
`id`	integer	The unique ID of the entry.

Column	Type	Description
`url`	text	The URL of the HTTP request.
`channel_id`	integer	The ID of the channel. This value is actually a reference to the id column of the channels table.
`response_code`	text	The status code of the HTTP response.
`request_time`	integer	Unix timestamp indicating when the request has been received.

The indexer_jobs table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 25: Columns of the indexer_jobs table

Column	Type	Description
`audit_trail_id`	integer	Identifies the audit trail using the `_connection_channel_id` of the channel for which the indexing job was created.
`id`	integer	The unique ID of the entry.
`indexer_adp_version`	text	The version number of the ADP component of the indexer service.
`indexer_cpu_time`	integer	The duration of the indexing (CPU time), in millisecond.
`indexer_duration`	integer	The duration of the indexing (actual time), in millisecond.
`indexer_start_time`	timestamp	Time when the indexing started.
`indexer_version`	text	The version number of the indexer service.
`job_id`	text	The unique ID of the indexing job, used by components of the indexing service during indexing only.
`error_message`	text	The error message of the indexer job.
`trail_is_archived`	boolean	The value is `true` if the trail is already archived.

The occurrences table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 26: Columns of the occurrences table

Column	Type	Description
`end_time`	integer	The time when the token (keyword) disappears in the audit trail.
`id`	integer	The unique ID number of the entry.
`result_id`	integer	An ID identifying the occurrence of the token. This value is a reference to the id column of the results table.
`screenshot`	text	A hash of the screenshot used in the report. The actual screen-shot is not stored in the database.
`start_time`	integer	The time when the token (keyword) appears in the audit trail.

The progresses table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

This table contains information only about Audit Player indexers. It does not contain any information about the indexer service.

Table 27: Columns of the progresses table

Column	Type	Description
`audit`	text	**Audit-trail**: Name and ID of the audit file storing the traffic of the channel. If the session has an audit trail, a icon is displayed. Note that a the following letters may appear on the download icon:
`ap_id`	integer	[OBSOLETE] The ID of the Audit Player indexer service that is processing the audit trail.
`id`	integer	The unique ID number of the entry.

The results table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 28: Columns of the results table

Column	Type	Description
channel_id	integer	The ID of the channel where a token was found. This value is actually a reference to the id column of the channels table.
id	integer	The unique ID number of the entry.
token	text	The token (search keyword).

The skipped_connections table

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 29: Columns of the skipped_connections table

Column	Type	Description
channel_id	integer	The ID of the channel. This value is actually a reference to the id column of the channels table.
id	integer	The unique ID number of the entry.
occasions	integer	

The usermapped_channels view

This view collects data about the connections which used a usermapping policy. The view is defined as follows:

```
select
    channels.id,
    channels.scp_path,
    channels.subsystem_name,
    channels.protocol,
```

```
    channels.originator_port,
    channels.channel_policy,
    channels.target_addr,
    channels._archive_path,
    channels._close_cleanup,
    channels.dynamic_channel,
    channels.environment,
    channels.session_start,
    channels.remote_username,
    channels.local_port,
    channels.four_eyes_authorizer,
    channels.src_ip,
    channels.session_end,
    channels._archive_server,
    channels.server_port,
    channels.username,
    channels._archive_date,
    channels.server_ip,
    channels.exec_cmd,
    channels._archive_policy,
    channels.four_eyes_description,
    channels.connection_id,
    channels._connection_channel_id,
    channels.rule_num,
    channels.target_port,
    channels.src_port,
    channels.originator_addr,
    channels.auth_method,
    channels.audit,
    channels.local_ip,
    channels.session_id,
    channels.device_name,
    channels.channel_type,
    channels."connection",
    channels.verdict,
    channels.dst_port,
    channels.dst_ip
from channels
where channels.remote_username is not null
and channels.username <> channels.remote_username;
```

The the usermapped_channels view has the following columns. For details of the different columns, see Connection metadata on page 958.

NOTE:

The structure of these database tables may change in future One Identity Safeguard for Privileged Sessions (SPS) versions.

Table 30: Columns of the usermapped_channels table

Column	Type	Description
_archive_date	date	**Archive date**: The date when the connection was archived or cleaned up.
_archive_path	text	**Archive path**: The path where the audit trail was archived on the remote server.
_archive_policy	text	
_archive_server	text	**Archive server**: The hostname or IP address of the remote server where the audit trail was archived.
audit	text	**Audit-trail**: Name and ID of the audit file storing the traffic of the channel. If the session has an audit trail, a icon is displayed. Note that a the following letters may appear on the download icon:
auth_method	text	**Authentication method**: The authentication method used in the connection. For example, password
channel_policy	text	**Channel policy**: The Channel policy applied to connection. The Channel policy lists the channels (for example, terminal session and SCP in SSH, or Drawing and Clipboard in RDP) that can be used in the connection, and also determines if the channel is audited or not. The Channel policy can also restrict access to each channel based on the IP address of the client or the server, a user list, user group, or a time policy.
channel_type	text	**Channel type**: Type of the channel.
_close_cleanup	boolean	
_connection_channel_id	integer	**Unique connection ID**: The unique identifier of the connection.
connection_id	text	**Connection policy ID**: The identifier of the connection policy.
connection	text	**Connection policy**: The connection policy that handled the client's connection request.
device_name	text	**Device name**: The name or ID of the shared device (redirect) used in the RDP connection.
dst_ip	text	**Destination IP**: The IP address of the server as requested by the client.
dst_port	integer	**Destination port**: The port number of the server as requested by the client.

Column	Type	Description
dynamic_channel	text	**Dynamic channel**: The name or ID of the dynamic channel opened in the RDP session.
exec_cmd	text	**Exec command**: The command executed in a **Session exec** channel.
four_eyes_authorizer	text	**Four-eyes authorizer**: The username of the user who authorized the session. Available only if 4-eyes authorization is required for the channel. For details on 4-eyes authorization, see Configuring four-eyes authorization on page 752.
four_eyes_description	text	**Four-eyes description**: The description submitted by the authorizer of the session.
id	integer	
local_ip	text	**Server-local IP**: The IP address of SPS used in the server-side connection.
local_port	integer	**Server-local port**: The port number of SPS used in the server-side connection.
originator_addr	text	**Port/X11 forward originator IP**: The IP address of the host initiating the channel in **Remote Forward** and **Local Forward** channels. Note that this host is not necessarily the client or the server of the SSH connection.
originator_port	integer	**Port/X11 forward originator port**: The number of the forwarded port in **Remote Forward** and **Local Forward** channels.
protocol	text	**Protocol**: The protocol used in the connection (Citrix ICA, HTTP, RDP, SSH, Telnet, or VNC).
remote_username	text	**Username on server**: The username used to log in to the remote server. This username can differ from the client-side username if usermapping is used in the connection. For details on usermapping, see Configuring usermapping policies on page 740.
rule_num	text	**Rule number**: The number of the line in the Channel policy applied to the channel.
scp_path	text	**SCP path**: Name and path of the file copied via SCP. Available only for SCP sessions (**Session exec SCP** SSH channels) if the **Log file transfers to database** option is enabled in the Channel Policy of the connection.
server_ip	text	**Server IP**: The IP address of the server connected by SPS.

Column	Type	Description
server_port	integer	**Server port**: The port number of the server connected by SPS.
session_end	integer	**End time**: Date when the channel was closed.
session_id	text	**Session ID**: A globally unique string that identifies the session. This session ID has the following format: *svc/<unique-random-hash>/<name-of-the-connection-policy>:<session-number-since-service-started>/<protocol>*, for example, *svc/5tmEaM7xdNi1oscgVWpbZx/ssh_console:1/ssh*. Log messages related to the session also contain this ID. For example: `2015-03-20T14:29:15+01:00 demo.example zorp/scb_ssh[5594]: scb.audit(4): (svc/5tmEaM7xdNi1oscgVWpbZx/ssh_console:0/ssh): Closing connection; connection='ssh_console', protocol='ssh', connection_ id='409829754550c1c7a27e7d', src_ip='10.40.0.28', src_port='39183', server_ip='10.10.20.35', server_port='22', gateway_username='', remote_username='example-username', verdict='ZV_ACCEPT'`
session_ start	integer	**Start time**: Date when the channel was started.
src_ip	text	**Source IP**: The IP address of the client.
src_port	integer	**Source port**: The port number of the client.
subsystem_ name	text	**Subsystem name**: Name of the SSH subsystem used in the channel.
target_addr	text	**Port-forward target IP**: The traffic was forwarded to this IP address in **Remote Forward** and **Local Forward** channels.
target_port	integer	**Port-forward target port**: The traffic was forwarded to this port in **Remote Forward** and **Local Forward** channels.
username	text	**Username**: The username used in the session. • If the user performed inband gateway authentication in the connection, the field contains the username from the gateway authentication (gateway username).

Column	Type	Description
		• Otherwise, the field contains the username used on the remote server.
verdict	text	**Verdict**: Indicates what SPS decided about the channel. • ACCEPT: Accepted. • ACCEPT-TERMINATED: Connection was accepted and established, but a content policy terminated the connection. For details on content policies, see Real-time content monitoring with Content Policies on page 451. • CONN-AUTH-FAIL: User authentication failed. • CONN-DENY: Connection rejected. • CONN-FAIL: Connection failed, that is, it was allowed to pass SPS but timed out on the server. • CONN-GW-AUTH-FAIL: Gatway authentication failed. • CONN-KEY-ERROR: Hostkey mismatch. • CONN-USER-MAPPING-FAIL: Usermapping failed. • DENY: Denied. • FOUR-EYES-DEFERRED: Waiting for remote username. • FOUR-EYES-ERROR: Internal error during four-eyes authorization. • FOUR-EYES-REJECT: Four-eyes authorization rejected. • FOUR-EYES-TIMEOUT: Four-eyes authorization timed out.

Querying trail content with the lucene-search function

CAUTION:

This function works only if you have enabled indexing for the audit trails.

The `lucene_search` function allows you to search the content of indexed audit trails for a specific keyword and return the IDs of the channels that contain the search keyword. The `lucene_search` function requires four parameters:

- *search phrase*: The keyword or keyphrase you are looking for, for example, a command issued in an SSH session (`exit`). The keyphrase can contain the following special operators to be used & (AND), | (OR), ! (NOT). Brackets can be used to group parts of the keyphrase.

- *beginning_timestamp*: The date in UNIX-timestamp format. Only audit trails created after this date will be queried.
- *ending_timestamp*: The date in UNIX-timestamp format. Only audit trails created before this date will be queried.

For example:

```
select lucene_search
from lucene_search('root', 1287402232, 1318938150);

# Sample output:
(1,2,3,1)
(1 row)
```

The output of this query is a formatted as the following:

```
(<channel_id>,<trail_id>,<hits_count>,<rank>)
```

Alternatively, you can use the following query format that returns a header of the displayed columns, and uses the pipe (|) character for separator:

For example:

```
select * from lucene_search('tmp',1496885628,1496954028);
channel_id | trail_id | hits_count | rank
----------------------------+-----
1      | 2      | 3      | 1
(1 row)
```

The output contains the following columns:

- `channel_id`: The ID of the channel within the audit trail (an audit trail file can contain audit trails of multiple channels).
- `trail_id`: Identifies the audit trail using the unique identifier of the session (the _ `connection_channel_id` of the channel for which the audit trail was created).
- `hits_count`: The number of hits in the audit trail.
- `rank`: Shows the relevance of the search result on a 0-1 scale, where 1 is the most relevant. Note that on the SPS search page, this information is scaled to 0-5 (and shown graphically with stars).

For details on how to use more complex keyphrases, see the Apache Lucene documentation.

For details of content indexing, see Configuring the internal indexer on page 593.

Generating partial reports

The following describes how to generate a report manually for a period that has not been already covered in an automatic report.

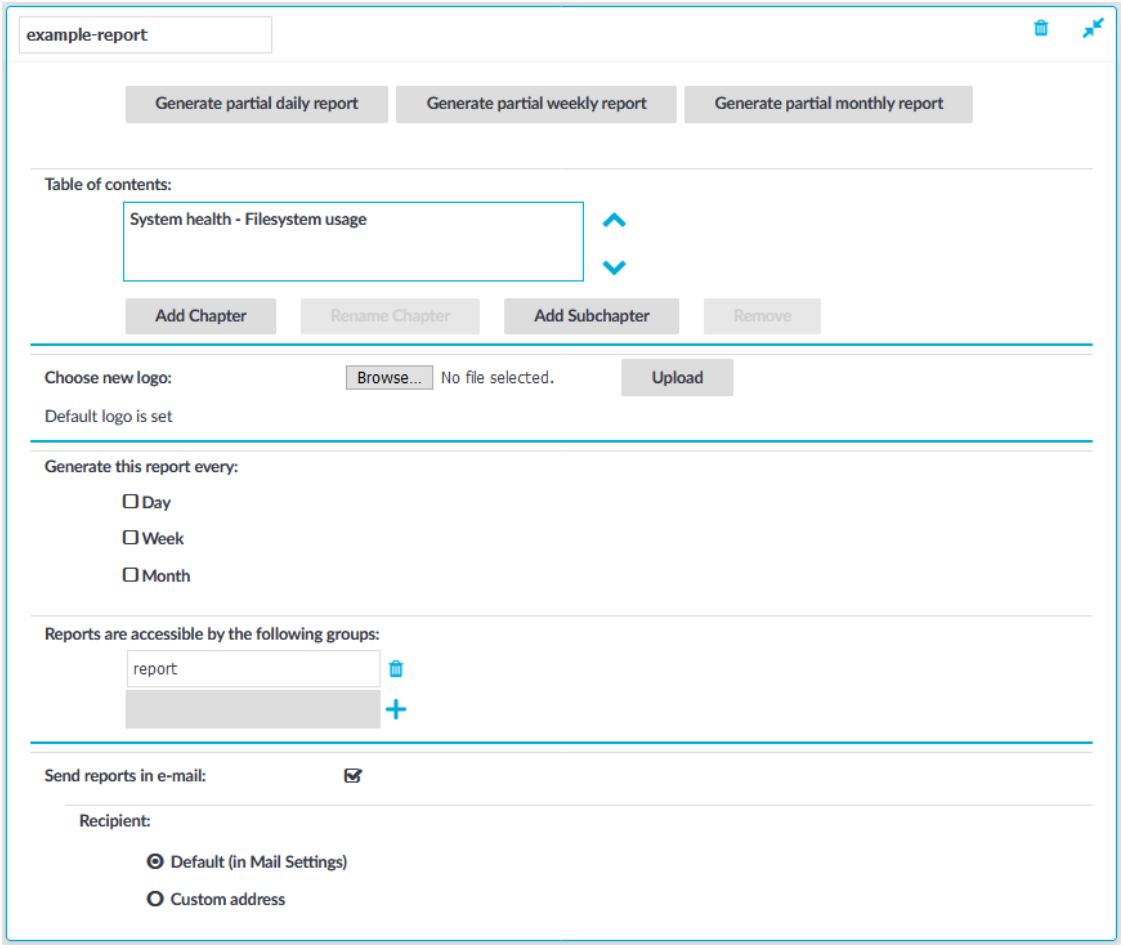
CAUTION:

Generating a report that includes an Advanced statistics chapter that returns several thousands of entries requires significant CPU and memory resources from One Identity Safeguard for Privileged Sessions (SPS). While generating such a partial report, the web interface of SPS can become slow or unresponsive.

To generate a report manually for a period that has not been already covered in an automatic report

1. Log in to the SPS web interface, and navigate to **Reporting > Configuration**.

Figure 271: Reporting > Configuration — Configuring custom reports



2. Select the report you want to generate.
3. Click any of the following buttons:
   - To create a report from the last daily report till now, click **Generate partial daily report**. For example, if you click this button at 11:30 AM, the report will

include the period from 00:01 to 11:30.

- To create a report from the last weekly report till now, click **Generate partial weekly report**. For example, if you click this button on Wednesday at 11:30 AM, the report will include the period from Monday 00:01 to Wednesday 11:30.
- To create a report from the last monthly report till now, click **Generate partial monthly report**. For example, if you click this button at 11:30 AM, December 13, the report will include the period from December 1, 00:01 to December 13, 11:30.

The report will be automatically added in the list of reports (**Reporting > Reports**), and also sent in an e-mail to the regular recipients of the report.

4. Click 

.

Creating PCI DSS reports

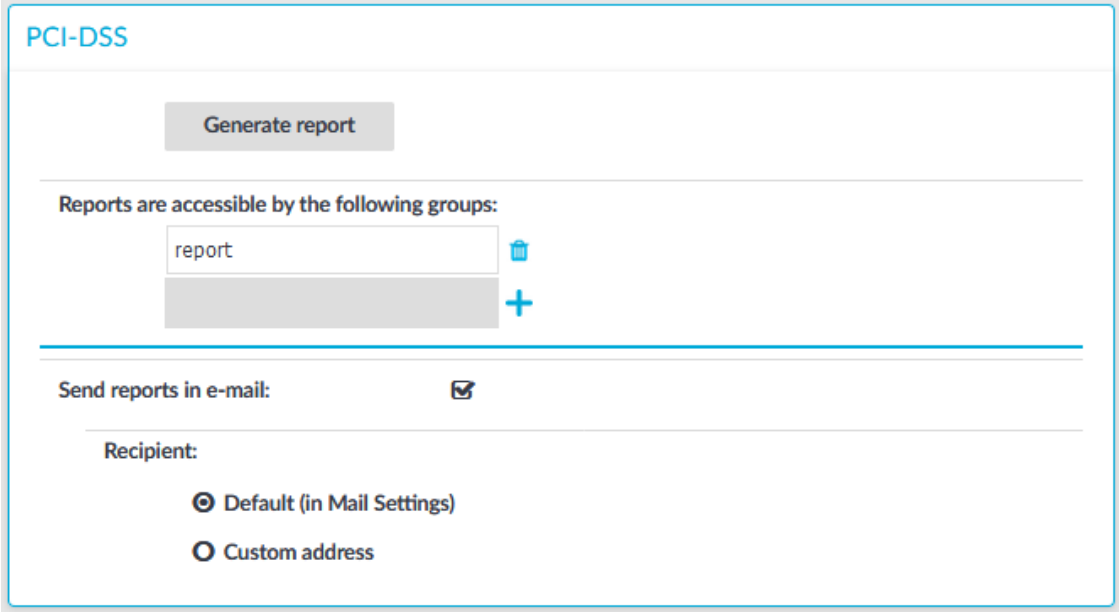
To help you comply with the regulations of the Payment Card Industry Data Security Standard (PCI DSS), One Identity Safeguard for Privileged Sessions (SPS) can generate reports on the compliance status of SPS. Note that this is not a fully-featured compliance report: it is a tool to enhance and complement your compliance report by providing information available in SPS. The report corresponds with the document *Payment Card Industry (PCI) Data Security Standard, Requirements and Security Assessment Procedures, Version 3.0*, published by the PCI Security Standards Council.

For details on the contents of the report, see Contents of PCI DSS reports on page 822.

To create PCI DSS reports

1. Log in to the SPS web interface, and navigate to **Reporting > PCI DSS**.

   Figure 272: Reporting > PCI DSS — Generating PCI DSS reports

2. By default, members of the `report` group can access the custom reports via the SPS web interface. To change this, enter the name of a different group into the **Reports are accessible by the following groups** field, or click + to grant access to other groups.

   NOTE:

   Members of the listed groups will be able to access only these custom reports even if their groups does not have read access to the **Reporting > Reports** page. However, only those reports will be listed, to which their group has access to.

3. By default, SPS sends out the reports in e-mail to the address set in the **Basic Settings > Management > Mail settings > Send reports to** field.

   NOTE:

   If this address is not set, the report is sent to the SPS administrator's e-mail address.

   - To disable e-mail sending, unselect the **Send reports in e-mail** option.
   - To e-mail the reports to a different address, select **Recipient > Custom address**, and enter the e-mail address where the reports should be sent. Click + to list multiple e-mail addresses if needed.

4. Click Commit.

5. Click **Generate report**.

   The report will be automatically added in the list of reports (**Reporting > Reports**), and also sent in an e-mail to the regular recipients of the report.

Contents of PCI DSS reports

To help you comply with the regulations of the Payment Card Industry Data Security Standard (PCI DSS), One Identity Safeguard for Privileged Sessions (SPS) can generate reports on the compliance status of SPS. Note that this is not a fully-featured compliance report: it is a tool to enhance and complement your compliance report by providing information available in SPS. The report corresponds with the document *Payment Card Industry (PCI) Data Security Standard, Requirements and Security Assessment Procedures, Version 3.0*, published by the PCI Security Standards Council.

For details on creating PCI DSS reports, see Creating PCI DSS reports on page 820. The following table details the information included in the SPS PCI DSS reports, and the relevant PCI compliance requirement.

Table 31: Contents of PCI DSS reports

PCI DSS Requirement	Compliance details
Requirement 1.1.6 Documentation and business justification for use of all services, protocols, and ports allowed, including documentation of security features implemented for those protocols considered to be insecure. Examples of insecure services, protocols, or ports include but are not limited to FTP, Telnet, POP3, IMAP, and SNMP v1 and v2.	The report lists the insecure connection policies configured in SPS, including SNMP server and agent settings, and the list of connection policies that permit unencrypted HTTP and Telnet. This list does not include any insecure connections that can be used to access SPS itself.
Requirement 2.1 Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network. This applies to ALL default passwords, including but not limited to those used by operating systems, software that provides security services, application and system accounts, point-of-sale (POS) terminals, Simple Network Management Protocol (SNMP)community strings, etc.).	SPS can be accessed as "root" via the local management console, or - if explicitly enabled - remotely using a Secure Shell (SSH v2) connection. The report lists the local web user accounts that can access SPS. For details on configuring these accounts, see Managing One Identity Safeguard for Privileged Sessions (SPS) users locally on page 303.
Requirement 2.2.2 Enable only necessary services, protocols,	The report includes the list of services running on SPS.

PCI DSS Requirement	Compliance details
daemons, etc., as required for the function of the system.	
Requirement 2.2.3 Implement additional security features for any required services, protocols, or daemons that are considered to be insecure, for example, use secured technologies such as SSH, S-FTP, SSL, or IPSec VPN to protect insecure services such as NetBIOS, file-sharing, Telnet, FTP, etc.	The report lists the connection policies enabling unencrypted HTTP and Telnet access, and any such session that was active when the report was generated.
Requirement 2.3 Encrypt all non-console administrative access using strong cryptography. Use technologies such as SSH, VPN, or SSL/TLS for web-based management and other non-console administrative access.	Use HTTPS to connect to SPS. HTTP connections are forbidden.
Requirement 3.5.2 Store secret and private keys used to encrypt/decrypt cardholder data in one (or more) of the following forms at all times: • Encrypted with a key-encrypting key that is at least as strong as the dataencrypting key, and that is stored separately from the data-encrypting key • Within a secure cryptographic device (such as a host security module (HSM) or PTS-approved point-of-interaction device) • As at least two full-length key components or key shares, in accordance with an industry-accepted method Note: It is not required that public keys be stored in one of these forms.	Audit trails are encrypted with AES128-GCM (audit trails recorded with SPS 5 F3 and earlier are encrypted with AES128-CBC). The master key is encrypted with the key you provided.
Requirement 5.1.2 For systems considered to be not commonly affected by malicious software, perform	SPS is an appliance running minimal services and using a hardened operating system. One Identity, the vendor of SPS,

PCI DSS Requirement	Compliance details
periodic evaluations to identify and evaluate evolving malware threats in order to confirm whether such systems continue to not require anti-virus software.	continuously monitors vulnerabilities and CVEs that might affect the components of SPS, and publishes security updates and announcements as needed. Using an up-to-date SPS version should keep the risk of SPS being affected by malicious software at a minimum level.
Requirement 6.2 Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release.	The report includes the firmware version running on SPS. You can check which is the latest version at the Downloads page.
Requirement 8.1.8 If a session has been idle for more than 15 minutes, require the user to reauthenticate to re-activate the terminal or session.	The report includes the timeout value to the SPS web interface (10 minutes by default). To change this value, see Web interface timeout on page 112.
Requirement 8.2.1 Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.	The report lists where SPS stores passwords, and the hash used to secure them.
Requirement 8.2.4 Change user passwords/passphrases at least every 90 days.	The report lists the password expiry settings of local web users of SPS, and also the last time the password of each user was changed. For details on configuring these accounts, see Managing One Identity Safeguard for Privileged Sessions (SPS) users locally on page 303. For details on configuring password expiry for these accounts, see Setting password policies for local users on page 306.
Requirement 8.2.5 Do not allow an individual to submit a new password/phrase that is the same as any of the last four passwords/phrases he or she has used.	The report includes the password history settings of local web users of SPS. For details on configuring password history for these accounts, see Setting password policies for local users on page 306.

PCI DSS Requirement	Compliance details
Requirement 10.5.3 Promptly back up audit trail files to a centralized log server or media that is difficult to alter.	The report lists the addresses of the logservers where SPS forwards its log messages. For details on forwarding log messages, see Configuring system logging on page 126.
Requirement 10.5.4 Write logs for external-facing technologies onto a secure, centralized, internal log server or media device.	The report lists the security settings of the communication between SPS and the logservers where SPS forwards its log messages. For details on forwarding log messages, see Configuring system logging on page 126.
Requirement 10.7 Retain audit trail history for at least one year, with a minimum of three months immediately available for analysis.	The retention time for local logs of SPS is seven days. To retain them longer, forward them to a remote logserver.

The One Identity Safeguard for Privileged Sessions (SPS) RPC API

NOTE:

The RPC API is deprecated as of One Identity Safeguard for Privileged Sessions (SPS) 5 F7 and will be removed in an upcoming feature release. One Identity recommends using the REST API instead.

Version 3 F3 and later of One Identity Safeguard for Privileged Sessions can be accessed using a Remote-Procedure Call Application Programming Interface (RPC API).

The SPS RPC API allows you to access, query, and manage SPS from remote applications. You can access the API using the Simple Object Access Protocol (SOAP) protocol over HTTPS, meaning that you can use any programming language that has access to a SOAP client to integrate SPS to your environment. You can download simple, proof-of-concept clients for Python and other languages from the SPS web interface.

Accessing SPS with the RPC API offers several advantages:

- Integration into custom applications and environments
- Flexible, dynamic search queries and management

Requirements for using the RPC API

RPC client requirements

Locking One Identity Safeguard for Privileged Sessions (SPS) configuration from the RPC API

Documentation of the RPC API

Enabling RPC API access to One Identity Safeguard for Privileged Sessions (SPS)

Requirements for using the RPC API

To access One Identity Safeguard for Privileged Sessions (SPS) using the RPC API, the following requirements must be met:

- Accessing the appliance via the RPC API must be enabled on the web interface. For details, see Enabling RPC API access to One Identity Safeguard for Privileged Sessions (SPS) on page 829.

- The appliance can be accessed using the SOAP protocol over authenticated HTTPS connections. The WSDL describing the available services is available at *https://<ip-address-of-SPS>/rpc.php/<techversion>?wsdl*. For details on the client libraries tested with SPS see RPC client requirements on page 827.
- The user account used to access SPS via RPC must have *read and write/perform* rights for the *Access RPC API* privilege. This is required for every type of RPC access, even for read-only operations. Members of the `api` group automatically have this privilege. For details on managing user privileges, see Modifying group privileges on page 324.

CAUTION:

Each SPS release provides a separate API with a new API version number. You are recommended to use the SPS version 6.0.13 with the corresponding API version. Earlier versions are not supported

RPC client requirements

The client application used to access One Identity Safeguard for Privileged Sessions (SPS) must meet the following criteria:

- Support SOAP version 1.1 or later.
- Support WSDL version 1.1.
- Properly handle complex object types.
- Include a JSON decoder for interpreting the results of search operations.

The following client libraries have been tested with SPS.

Table 32: SOAP libraries tested with SPS

Client name	Program-ming language	Status	Comments
Apache Axis 1	Java	Working	
Built-in .NET library	.NET	Working	SPS does not support the Expect HTTP Header feature, and must be disabled, for example, using `System.Net.ServicePointManager.Expect100Continue = false;`
Scio	Python	Partially working	Does not handle complex object types, so it cannot perform search queries.
SOAP::Lite	Perl	Working	• Simple types can be used with the following format: `$service->$method(@params)`

Client name	Program-ming language	Status	Comments
			• Complex types work only with the following format: `$service->call ($method, @params)` • Calls using the `$service->call()` format seem to work after doing at least one `$service->$method(@params)` call, for example, a login.
SOAP::WSD-L	Perl	Not working	
Suds	Python	Working	

Locking One Identity Safeguard for Privileged Sessions (SPS) configuration from the RPC API

Accessing One Identity Safeguard for Privileged Sessions (SPS) using the RPC API locks certain components of SPS from other users, just like accessing SPS using the web interface or the console. Locking SPS via RPC can be performed either explicitly by calling the `lockAcquire` function, or implicitly when an operation requires the lock. In either case, ensure that your application verifies that the lock is received and properly handles if the component is locked by someone else (for example, because a user is accessing the component from the web interface).

For details on how locking works in SPS, see "Multiple users and locking" in the Administration Guide.

Documentation of the RPC API

The documentation of the One Identity Safeguard for Privileged Sessions (SPS) RPC API is available online from the SPS web interface: select **Basic Settings > Management > RPC API settings > Open documentation**, or directly from the following URL: *https://<ip-address-of-SPS>/rpc-api-doc/*. This documentation contains the detailed description of the available services and classes.

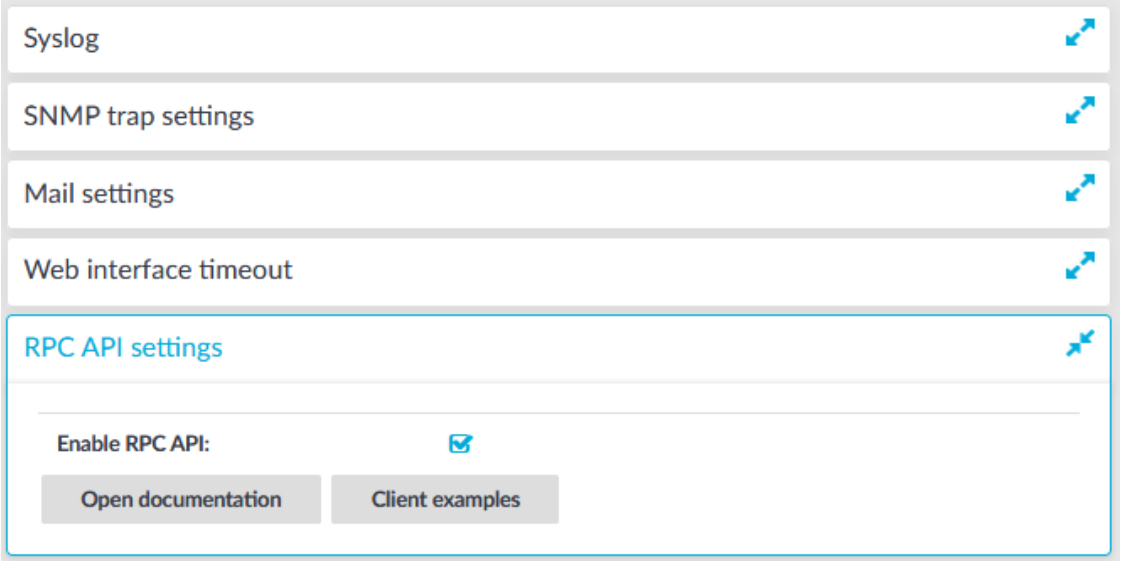
Enabling RPC API access to One Identity Safeguard for Privileged Sessions (SPS)

The following describes how to configure One Identity Safeguard for Privileged Sessions (SPS) to accept RPC API connections.

To configure SPS to accept RPC API connections

1. Log in to the SPS web interface.
2. Select **Basic Settings > Management > RPC API settings > Enable RPC API**.

   Figure 273: Basic Settings > Management > RPC API settings — Enabling RPC API access to SPS

3. Click Commit.

Expected result

Users accounts belonging to a usergroup that have `read and write/perform` rights to the `Access RPC API` privilege can access SPS via the RPC API.

The One Identity Safeguard for Privileged Sessions (SPS) REST API

Starting with One Identity Safeguard for Privileged Sessions (SPS) version 4 F2, certain parts and features of SPS can be configured using a REST API (Representational State Transfer Application Programming Interface). The REST server conforms to the Hypermedia as the Engine of Application State (HATEOAS).

The SPS REST API uses JSON over HTTPS. The REST server has a single entry point and all resources are available at paths (URLs) returned in the response for a request sent to the entry point. The only path that is guaranteed not to change is `/api/authentication`. Every other path should be reached by navigating the links returned.

The SPS REST API allows you to create, read, update and delete (CRUD) the configuration resources of SPS.

The user accessing the SPS REST API must have the **REST server** privilege. For details, see "Modifying group privileges" in the Administration Guide. Note that the built-in **api** usergroup does not have this privilege by default, it is used to access the SOAP RPC API of SPS.

For details on using the REST API, see REST API Reference Guide.

23

One Identity Safeguard for Privileged Sessions (SPS) scenarios

This section discusses common scenarios for One Identity Safeguard for Privileged Sessions (SPS).

Configuring public-key authentication on One Identity Safeguard for Privileged Sessions (SPS)

Organizing connections in non-transparent mode

Using inband destination selection in SSH connections

SSH usermapping and keymapping in AD with public key

Configuring public-key authentication on One Identity Safeguard for Privileged Sessions (SPS)

If a protected server requires public-key authentication from the users, complete one of the following procedures.

- In Configuring public-key authentication using local keys on page 832, One Identity Safeguard for Privileged Sessions (SPS) stores the public keys of the users and the private-public keypair used in the server-side connection locally on SPS.
- In Configuring public-key authentication using an LDAP server and a fixed key on page 833, SPS receives the public keys of the users from an LDAP server and uses a locally-stored private-public keypair in the server-side connection.
- In Configuring public-key authentication using an LDAP server and generated keys on page 834, SPS receives the public keys of the users from an LDAP server. SPS generates a keypair that is used in the server-side connection on-the-fly, then uploads the public key of this pair to the LDAP database. That way the server can authenticate SPS to the (newly generated) public key of the user.

Configuring public-key authentication using local keys

The following describes how to store the public keys of the users and the private-public keypair used in the server-side connection locally on One Identity Safeguard for Privileged Sessions (SPS).

To configure public-key authentication using local keys

1. Navigate to **Policies > Local User Databases** and create a Local User Database. Add the users and their public keys to the database. SPS will authenticate the clients to this database. For details on creating and maintaining local user databases, see Creating a Local User Database on page 486.
2. Navigate to **Policies > Credential Stores** and create a Local Credential Store. Add hostnames and the users to the database. SPS will use these credentials to authenticate on the target server. For details on creating local credential stores, see Configuring local Credential Stores on page 758.
3. Navigate to **SSH Control > Authentication Policies** and create a new Authentication Policy.
4. Select **Authenticate the client to SPS using > Local > Public key**, clear all other options.
5. Select the appropriate usergroup from the **Local User Database** field. SPS will authenticate the users to this local database.
6. Select **Relayed authentication methods > Public key > Fix**, clear all other options.
7. Click **> Generate**. This will generate a private key that is needed only for the configuration, it will not be used in any connection.

   NOTE:

   The Connection Policy will ignore the settings for server-side authentication (set under **Relayed authentication methods**) if a Credential Store is used in the Connection Policy.

8. Click Commit.
9. Navigate to **SSH Control > Connections** and create a new Connection.
10. Enter the IP addresses of the clients and the servers into the **From** and **To** fields.
11. Select the authentication policy created in Step 1 in the **Authentication Policy** field.
12. Configure the other options of the connection as necessary.
13. Click Commit.
14. To test the above settings, initiate a connection from the client machine to the server.

Configuring public-key authentication using an LDAP server and a fixed key

The following describes how to fetch the public keys of the users from an LDAP server and use a locally-stored private-public keypair in the server-side connection.

NOTE:

One Identity recommends using 2048-bit RSA keys (or stronger).

To configure public-key authentication using an LDAP server and a fixed key

1. Navigate to **SSH Control > Authentication Policies** and create a new Authentication Policy.
2. Select **Authenticate the client to SPS using > LDAP > Public key**, deselect all other options.
3. Select **Relayed authentication methods > Public key > Fix**, deselect all other options.
4. Select **Private key** and click . A pop-up window is displayed.
5. Click **Browse** and select the private key of the user, or paste the key into the **Copy-paste** field. Enter the password for the private key into the **Password** field and click **Upload**.

   NOTE:

   One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

   If the private key of the user is not available, click **Generate** to create a new private key. You can set the size of the key in the **Generate key** field. In this case, do not forget to export the public key from SPS and import it to the server. To export the key from SPS, just click on the key and save it to your local computer.
6. Click on the fingerprint of the key in the **Server side private and public key > Private key** field and save the public key. Do not forget to import this public key to the server: all connections that use this new authentication policy will use this keypair on the server side.
7. Click Commit.
8. Navigate to **Policies > LDAP Servers** and click + to create a new LDAP policy.
9. Enter the parameters of the LDAP server. For details, see Authenticating users to an LDAP server on page 459.
10. If different from `sshPublicKey`, enter the name of the LDAP attribute that stores the public keys of the users into the **Publickey attribute name** field.

CAUTION:

The public keys stored in the LDAP database must be in OpenSSH format.

11. Navigate to **SSH Control > Connections** and create a new Connection.
12. Enter the IP addresses of the clients and the servers into the **From** and **To** fields.
13. Select the authentication policy created in Step 1 from the **Authentication Policy** field.
14. Select the LDAP policy created in Step 7 from the **LDAP Server** field.
15. If the server accepts a user only from a specific IP address, select the **Use original IP address of the client** radiobutton from the **SNAT** field.
16. Configure the other options of the connection as necessary.
17. Click Commit.
18. To test the above settings, initiate a connection from the client machine to the server.

Configuring public-key authentication using an LDAP server and generated keys

The following describes how to fetch the public keys of the users from an LDAP server and have One Identity Safeguard for Privileged Sessions (SPS) generate a keypair that is used in the server-side connection on-the-fly, and upload the public key of this pair to the LDAP database.

To configure public-key authentication using an LDAP server and generated keys

1. Navigate to **SSH Control > Authentication Policies** and create a new Authentication Policy.
2. Select **Authenticate the client to SPS using > LDAP > Public key**, deselect all other options.
3. Select **Relayed authentication methods > Public key > Publish to LDAP**, deselect all other options.
4. Click Commit.
5. Navigate to **Policies > LDAP Servers** and click + to create a new LDAP policy.
6. Enter the parameters of the LDAP server. For details, see Authenticating users to an LDAP server on page 459.
7. If different from `sshPublicKey`, enter the name of the LDAP attribute that stores the public keys of the users into the **Publickey attribute name** field.

CAUTION:

The public keys stored in the LDAP database must be in OpenSSH format.

8. Enter the name of the LDAP attribute where SPS shall upload the generated keys into the **Generated publickey attribute name** field.
9. Click Commit.
10. Navigate to **SSH Control > Connections** and create a new Connection.
11. Enter the IP addresses of the clients and the servers into the **From** and **To** fields.
12. Select the authentication policy created in Step 1 from the **Authentication Policy** field.
13. Select the LDAP policy created in Step 7 from the **LDAP Server** field.
14. If the server accepts a user only from a specific IP address, select the **Use original IP address of the client** radiobutton from the **SNAT** field.
15. Configure the other options of the connection as necessary.
16. Click Commit.
17. To test the above settings, initiate a connection from the client machine to the server.

Organizing connections in non-transparent mode

When using One Identity Safeguard for Privileged Sessions (SPS) in non-transparent mode, the administrators must address SPS to access the protected servers. If an administrator has access to more than one protected server, SPS must be able to determine which server the administrator wants to access. For each protected server, the administrators must address either different ports of the configured interface, or different alias IP addresses.

Organizing connections based on port numbers

To allow the administrators to access protected servers by connecting to the IP address of One Identity Safeguard for Privileged Sessions (SPS), and use the port number to select which server they want to access. Organizing connections based on port numbers is advantageous if SPS has a public IP address and the protected servers must be administered from the Internet.

NOTE:

Do not use the listening addresses configured for web login. For more details, see Configuring user and administrator login addresses on page 118.

For details on configuring alias IP addresses, see Managing logical interfaces on page 120.

To organize connections based on port numbers

1. Navigate to the **Connections** tab of the **SSH Control** menu.
2. Add a new connection. Enter the IP address of the administrators into the **From** fields, and the IP address and port number of the server into the **Target** field.
3. Enter the IP address of the logical interface of SPS into the **To** field, and enter a port number into the **Port** field.
4. Repeat Steps 2-3 for every protected server, but every time use a different port number in Step 3.
5. Click Commit.

Organizing connections based on alias IP addresses

To allow the administrators to access protected servers by connecting to an alias IP address of One Identity Safeguard for Privileged Sessions (SPS). The alias IP address determines which server they will access. Organizing connections based on alias IP addresses is advantageous if SPS is connected to a private network and many private IP addresses are available.

NOTE:

Do not use the listening addresses configured for web login. For more details, see Configuring user and administrator login addresses on page 118.

To organize connections based on alias IP addresses

1. Navigate to **Basic Settings > Network**.
2. Set up a logical interface: click + and configure a new logical interface. Add alias IP addresses for every protected server. (Use a different IP address for each.)

   For more information on configuring logical interfaces and alias IP addresses, see Managing logical interfaces on page 120.
3. Navigate to **SSH Control > Connections**.
4. Add a new connection. Enter the IP address of the administrators into the **From** fields, and the IP address and port number of the target server into the **Target** field.
5. Enter an alias IP address of the configured logical interface of SPS into the **To** field.

6. Repeat Steps 4-5 for every protected server, but every time use a different alias IP address in Step 5.
7. Click 

.

Using inband destination selection in SSH connections

The following sections provide examples for using inband destination selection to establish an SSH connection, including scenarios where nonstandard ports or gateway authentication is used.

Since some client applications do not permit the `@` and `:` characters in the username, alternative characters can be used as well:

- To separate the username and the target server, use the `@` or `%` characters, for example: `username%targetserver@scb_address`
- To separate the target server and the port number, use the `:`, `+`, or `/` characters, for example: `username%targetserver+port@scb_address`

For detailed instructions on configuring inband authentication, see Configuring inband destination selection on page 443.

Using inband destination selection with PuTTY

To establish an SSH connection through One Identity Safeguard for Privileged Sessions (SPS) with PuTTY, follow one of the methods:

Common method

To establish the SSH-connection using the most common method, enter the username, the target server's hostname (or IP address), and the hostname (or IP address) of SPS using the `<username>@<server>@<scb>` format in PuTTY.

Example

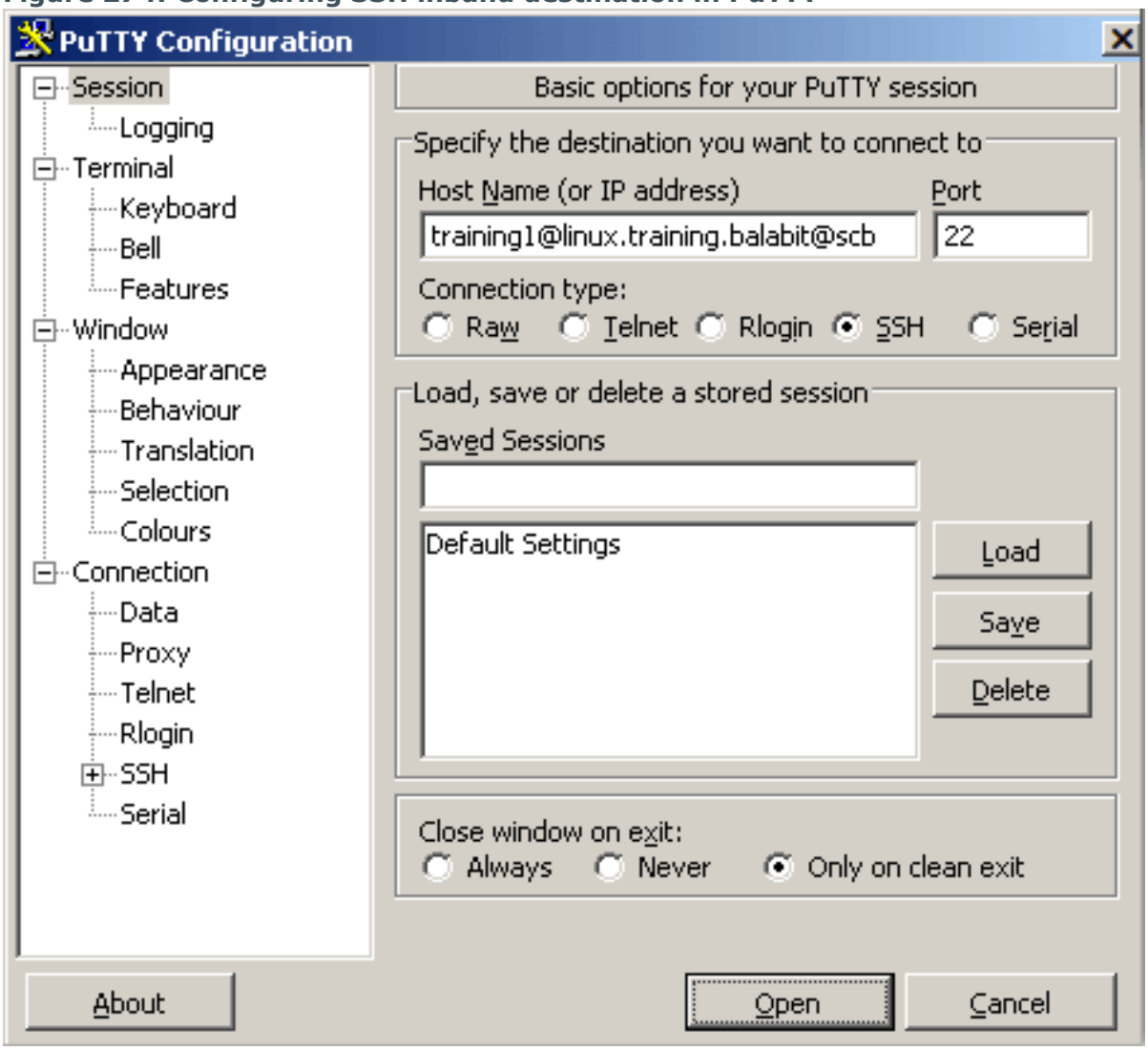
Assuming the following values:

- The username is `training1`
- The target server is `linux.training.example`
- The SPS server is `scb`

You can enter the following destination in PuTTY:

`training1@linux.training.example@scb`

Figure 274: Configuring SSH inband destination in PuTTY



Alternative method

To establish the SSH-connection using a different method,

1. Enter only the hostname (or IP address, depending on your configuration) of SPS in PuTTY.
2. At the login prompt, provide the username on the target server, and the target server's hostname (or IP address) using the `<username>@<server>` format.

Using inband destination selection with OpenSSH

To establish an SSH connection through One Identity Safeguard for Privileged Sessions (SPS), follow these steps:

1. Enter the following command:

   # ssh <username>@<server>@<scb>

   ...where `<username>` is the username, `<server>` is the target server's hostname (or IP address), and `<scb>` is the hostname (or IP address) of SPS

   #### Example

   Assuming the following values:

   - The username is `training1`
   - The target server is `linux.training.example`
   - The SPS server is `scb`

   You can enter the following command:

   # ssh training1@linux.training.example@scb
2. Alternative approach:
   a. Enter only the hostname (or IP address, depending on your configuration) of SPS:

      # ssh <scb>

   b. At the login prompt, provide the username on the target server, and the target server's hostname (or IP address) using the `<username>@<server>` format

Using inband selection and nonstandard ports with PuTTY

The following steps provide instructions for establishing SSH connections with servers that are listening on a non-standard port (the **Inband destination selection > Targets > Port** option is not 22), and the port number targeted by the clients is also a non-standard port (the **To > Port** option of the Connection Policy).

1. Enter the following in PuTTY:
    a. In the **Host Name** field, enter the username on the target server, the target server's hostname (or IP address) and port number, and the hostname (or IP address) of One Identity Safeguard for Privileged Sessions (SPS) in the `<username>@<server>:<port>@<scb>` format
    b. In the **Port** field, enter the port number of the SPS server

Example

Assuming the following values:

- The username is `training1`
- The target server is `192.168.60.100`
- The target server is listening on port `2121`
- The SPS server is `scb`
- The SPS server is listening on port `4444`

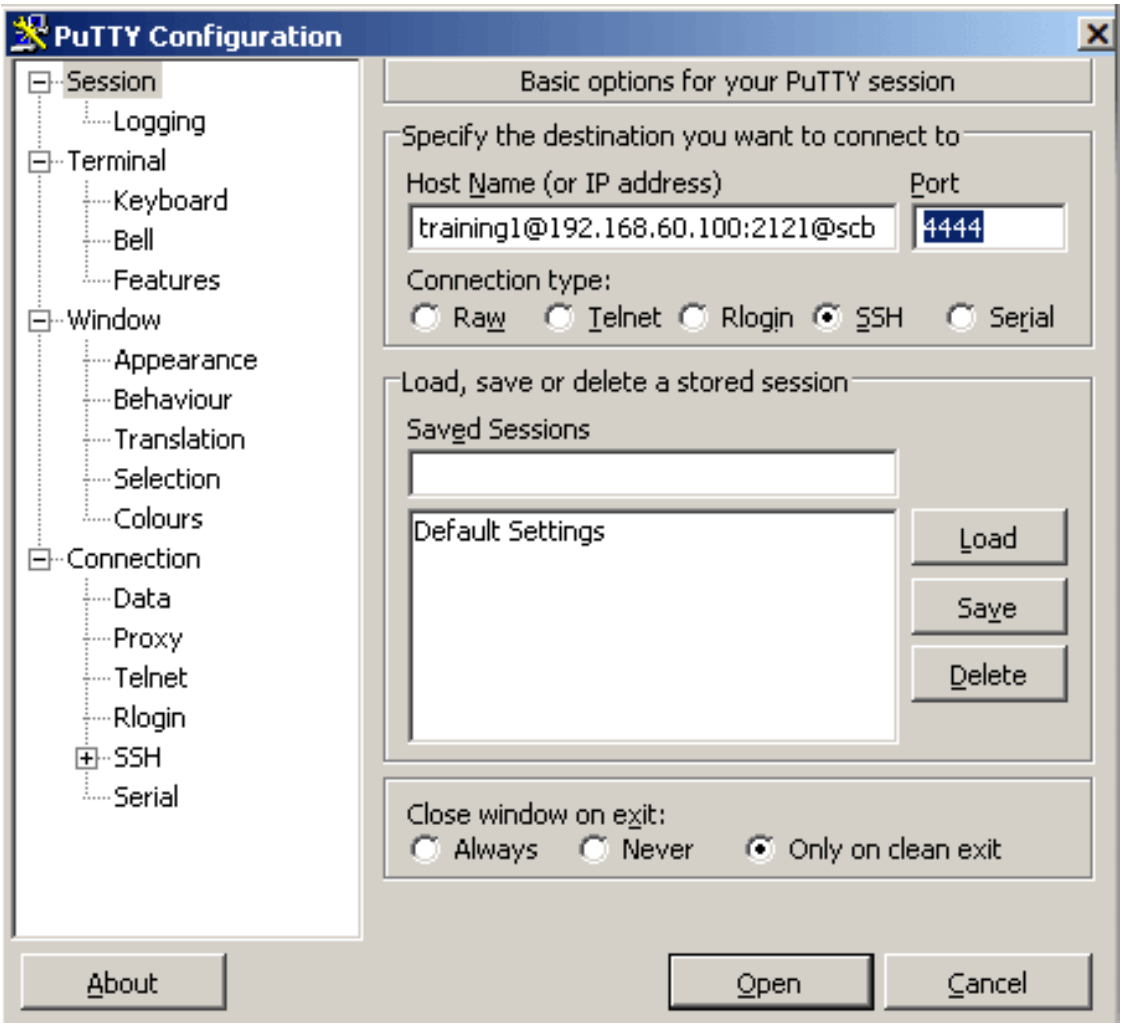
You can enter the following destination hostname in PuTTY:

`training1@192.168.60.100:2121@scb`

Also change the destination port to the SPS server's port number:

`4444`

Figure 275: Configuring SSH inband destination for nonstandard ports in PuTTY



2. Alternative approach:
    a. Enter only the hostname (or IP address, depending on your configuration) and port number of SPS in PuTTY.
    b. At the login prompt, provide the username on the target server, and the target server's hostname (or IP address) and port number using the `<username>@<server>:<port>` format.

Using inband selection and nonstandard ports with OpenSSH

The following steps provide instructions for establishing SSH connections with servers that are listening on a non-standard port (the **Inband destination selection > Targets >**

Port option is not 22), and the port number targeted by the clients is also a non-standard port (the **To > Port** option of the Connection Policy).

1. Enter the following command:

   # ssh -p <scb_port> <username>@<server>:<port>@<scb>

   ...where `<scb_port>` is the port number of One Identity Safeguard for Privileged Sessions (SPS), `<username>` is the username on the target server, `<server:port>` is the target server's hostname (or IP address), `<port>` is the target server's port number, and `<scb>` is the hostname (or IP address) of SPS.

   ### Example

   Assuming the following values:

   - The username is `training1`
   - The target server is `192.168.60.100`
   - The target server is listening on port `2121`
   - The SPS server is `scb`
   - The SPS server is listening on port `4444`

   You can enter the following command:

   # ssh -p 4444 training1@192.168.60.100:2121@scb

2. Alternative approach:

   a. Enter only the hostname (or IP address, depending on your configuration) and port number of SPS with the following command:

      # ssh -p <scb_port> <scb>

   b. At the login prompt, provide the username on the target server, and the target server's hostname (or IP address) and port number using the `<username>@<server>:<port>` format.

Using inband destination selection and gateway authentication with PuTTY

SPS can authenticate users attempting to establish an SSH connection against a gateway (see Configuring gateway authentication on page 742 for more details). You can provide the gateway login credentials in PuTTY:

1. Enter the gateway username, the username on the target server, the target server's hostname (or IP address), and the hostname (or IP address) of One Identity Safeguard for Privileged Sessions (SPS) in the `gu=<gatewayusername>@<username>@<server>@<scb>` format in PuTTY

Example

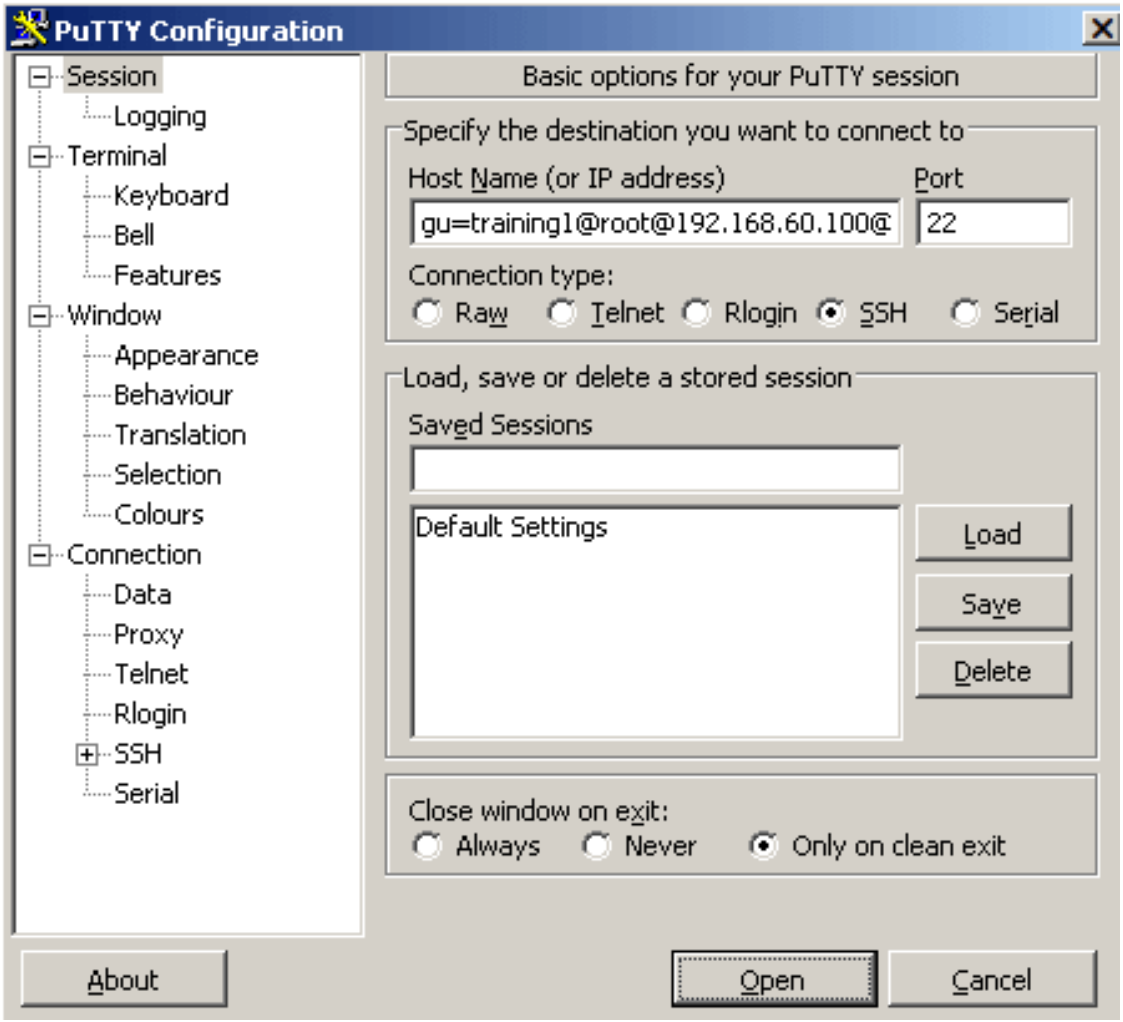
Assuming the following values:

- The gateway username is `training1`
- The username on the target server is `root`
- The target server is `192.168.60.100`
- The SPS server is `scb`

You can enter the following destination in PuTTY:

```
gu=training1@root@192.168.60.100@scb
```

Figure 276: Configuring SSH inband destination and gateway authentication in PuTTY



2. Alternative approach:

a. Enter only the hostname (or IP address, depending on your configuration) of SPS in PuTTY.

b. At the login prompt, provide the username on the target server, and the target server's hostname (or IP address) using the `<username>@<server>` format.

c. When prompted, provide the gateway username.

Using inband destination selection and gateway authentication with OpenSSH

One Identity Safeguard for Privileged Sessions (SPS) can authenticate users attempting to establish an SSH connection against a gateway (see Configuring gateway authentication on page 742 for more details). The following steps explain how you can provide the gateway login credentials:

1. Enter the following command:

   # ssh gu=<gatewayusername>@<username>@<server>@<scb>

   ...where `<gatewayusername>` is the gateway username, `<username>` is the username on the target server, `<server>` is the target server's hostname (or IP address), and `<scb>` is the hostname (or IP address) of SPS.

   ### Example

   Assuming the following values:

   - The gateway username is `training1`
   - The username on the target server is `root`
   - The target server is `192.168.60.100`
   - The SPS server is `scb`

   You can enter the following command:

   # ssh gu=training1@root@192.168.60.100@scb

2. Alternative approach:

   a. Enter only the hostname (or IP address, depending on your configuration) of SPS with the following command:

      # ssh <scb>

   b. At the login prompt, provide the username on the target server, and the target server's hostname (or IP address) using the `<username>@<server>` format.

   c. When prompted, provide the gateway username.

SSH usermapping and keymapping in AD with public key

A customer wants to be able to disable password authentication in SSH for `admin` users on the UNIX servers. However, the customer uses Active Directory, and would not like to enter the username/password at gateway authentication for every login over and over again. Therefore, the customer needs a quasi SSO-like system, with only one group logging in as `root` and another group as `XY` user.

To perform SSH usermapping and keymapping in AD with public key

1. Create an LDAP authentication policy. For details on creating a new authentication policy, see Authentication Policies on page 552. In this scenario, only a few important details will be highlighted.
    a. In the **Authenticate the client to SPS using** field, set the authentication method used on the client-side to **LDAP**. This will be the Active Directory where the gateway will get the public key from, for authentication. Enable **Publickey** only from the **Allowed authentication methods** and disable all other methods.
    b. In the **Relayed authentication methods** field, enable **Public key** and select **Agent**. Disable all other methods.
2. Create a Credential Store that can return a private key for server-side authentication. It is local Credential Stores and external Credential Stores (with a suitable plugin) that can return a private key.

    For detailed step-by-step instructions, see Configuring local Credential Stores on page 758. In this scenario, only a few important details will be highlighted.
    a. Navigate to the bottom of the policy, and click + to add a new user.
    b. Enter the username in the **Username** field (for example: `root`). Generate a **Private key** and upload its public counterpart to the server.
3. Set an LDAP server policy where you set up the Active Directory. For details on authenticating users to an LDAP server, see Authenticating users to an LDAP server on page 459.

    Make sure that the **Publickey attribute name** field in this Active Directory LDAP policy is set to `sshPublicKey`.
4. By default, the Active Directory does not have any attribute that could store the SSH public key. To solve this, add an OpenSSH-LPK compatible schema to the Active Directory by doing any of the following:
    - Create an `sshPublicKey` attribute, and add that directly to one of the objectClasses of the user in question.
    - Create an `sshPublicKey` attribute and an `ldapPublicKey` auxiliary objectClass,

and add the `ldapPublicKey` auxiliary objectClass to one of the objectClasses of the user in question.

The `sshPublicKey` attribute must be compliant with the OpenSSH-LPK schema and have the following properties:

- Name: sshPublicKey
- Object ID: 1.3.6.1.4.1.24552.500.1.1.1.13
- Syntax: Octet String
- Multi-Valued

The `ldapPublicKey` auxiliary objectClass must be compliant with the OpenSSH-LPK schema and have the following properties:

- Name: ldapPublicKey
- OID: 1.3.6.1.4.1.24552.500.1.1.2.0

The OpenSSH-LPK schema is available on the openssh-lpk Google Code page.

The following steps describe how to create an `sshPublicKey` attribute and an `ldapPublicKey` auxiliary objectClass, and then add the `ldapPublicKey` auxiliary objectClass to one of the objectClasses of the user.

a. Enable Schema updates using the registry:
   i. Click **Start**, click **Run**, and then in the **Open** box, type: `regedit`. Press`Enter`.
   ii. Locate and click the following registry key: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters`.
   iii. On the **Edit** menu, click **New**, and then click **DWORD Value**.
   iv. Enter the value data when the following registry value is displayed:

```
Value Name: Schema Update Allowed
Data Type: REG_DWORD
Base: Binary
Value Data: 1
```

   > NOTE:
   >
   > Type `1` to enable this feature, or `0` (zero) to disable it.

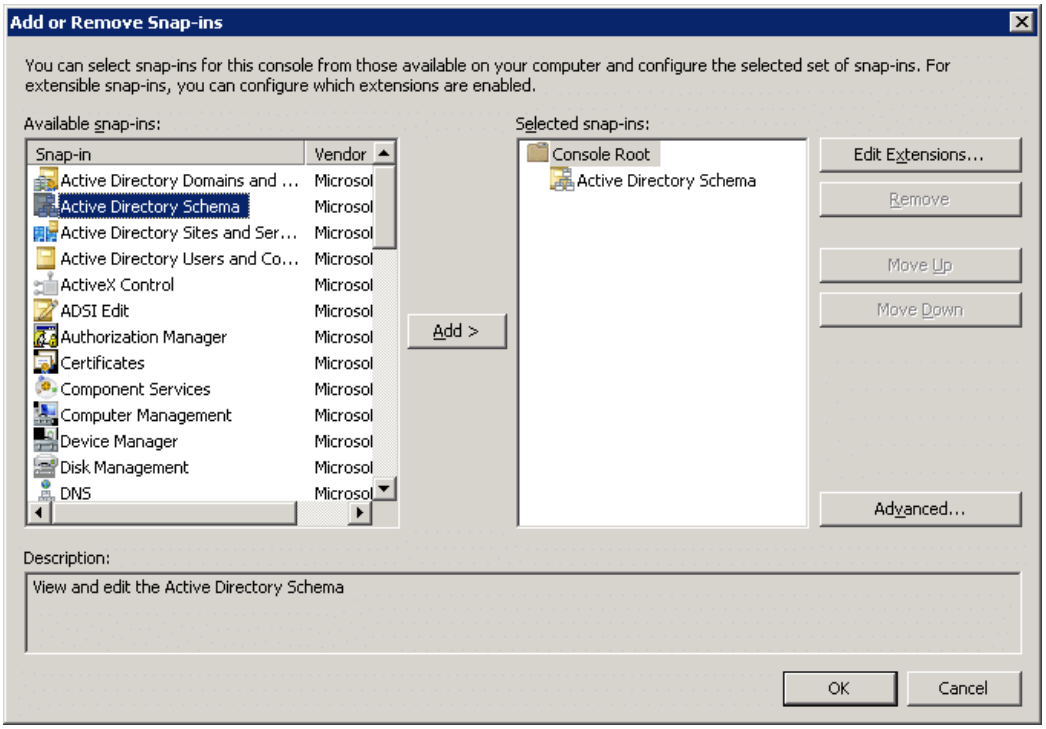
   v. Quit Registry Editor.

b. Install the Schema snap-in. For details, see the Microsoft Documentation. Note that you must have Administrator privileges to install the Schema snap-in.

c. Click **Start**, click **Run**, and then in the **Open** box, type: `MMC`. Press **Enter**.

d. Navigate to **File > Add or Remove Snap-in**, select **Active Directory Schema** and click **Add**. Note that you must have Schema Administrator privileges to complete the following steps.

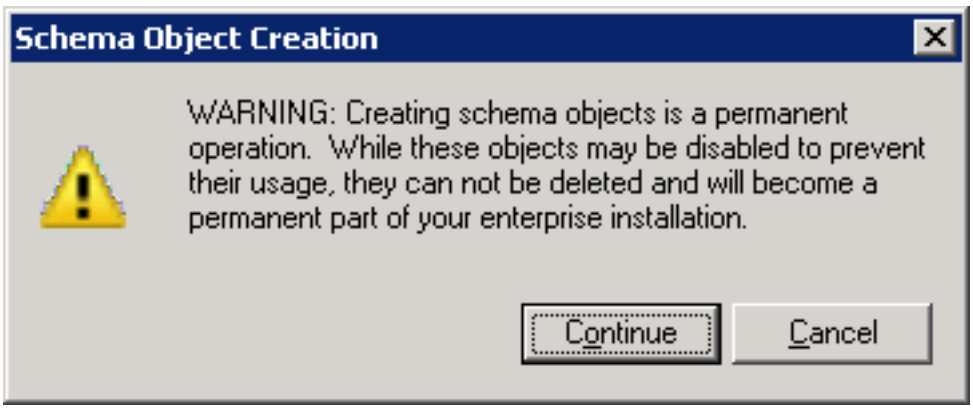
Figure 277: Add or Remove Snap-in



e. Expand the Active Directory schema and right-click **Attributes**.

f. Click **Create Attribute**. If a warning appears, click **Continue**.

Figure 278: Creating Attribute - Warning



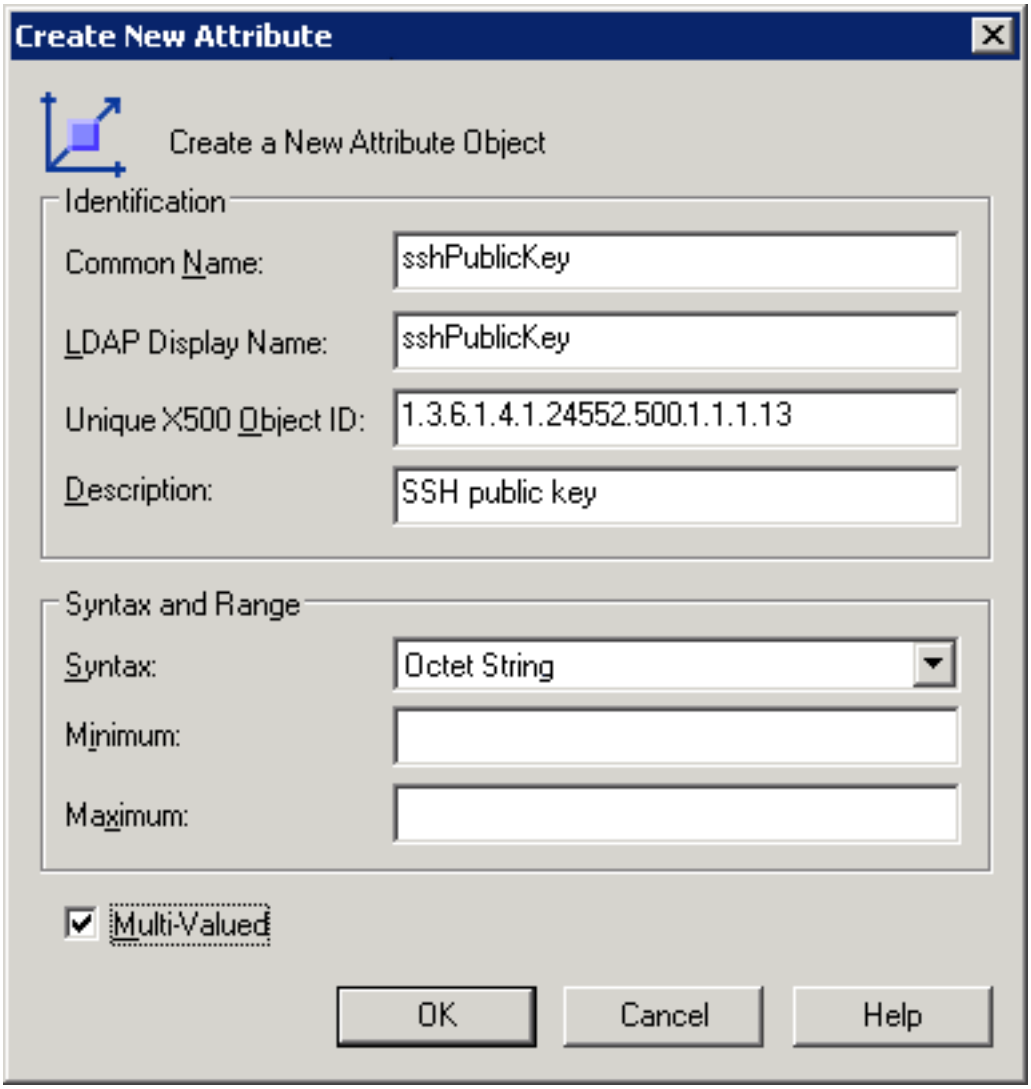
g. In **Common Name** and **LDAP Display Name**, enter `sshPublicKey`.

h. In **Unique X500 Object ID**, enter `1.3.6.1.4.1.24552.500.1.1.1.13`.

i. For **Syntax**, select `Octet String`.

j. Enable **Multi-Valued**. Click **OK**.

Figure 279: Create New Attribute



k. Right-click **Classes** and click **Create class**. If a warning appears, click **Continue**.

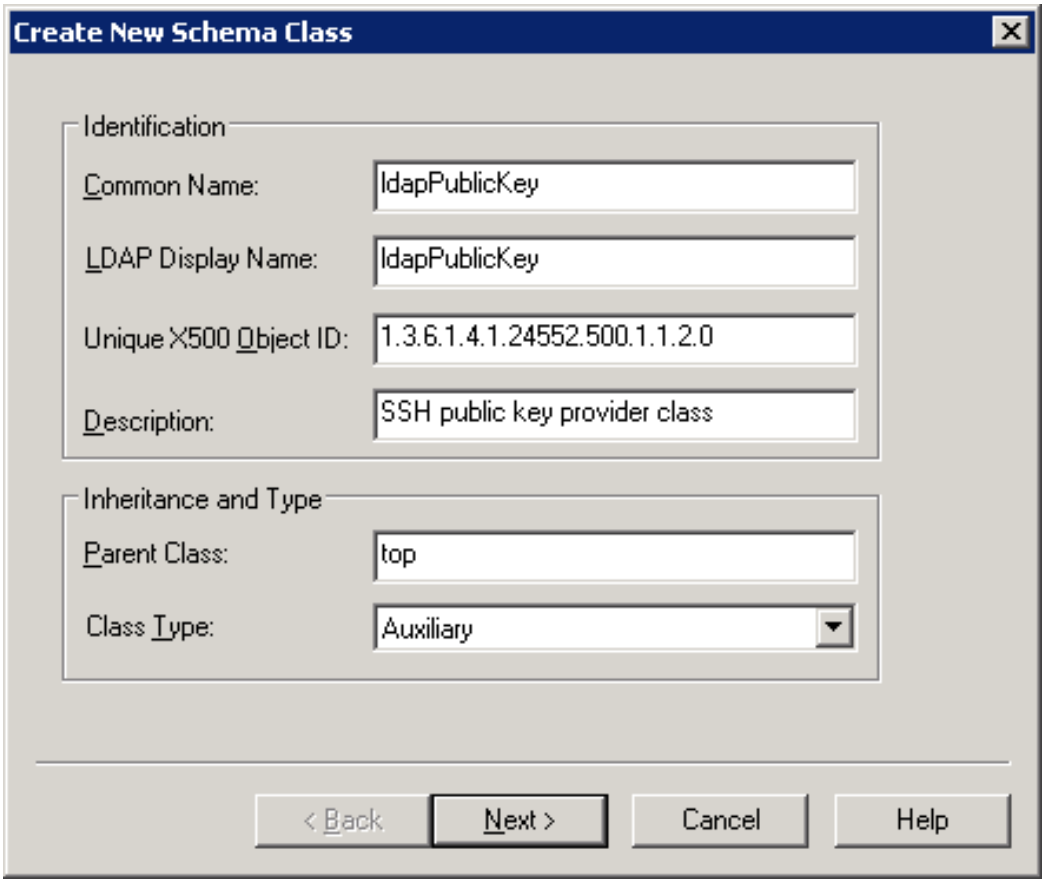
l. In **Common Name** and **LDAP Display Name**, enter `ldapPublicKey`.

m. In **Unique X500 Object ID**, enter `1.3.6.1.4.1.24552.500.1.1.2.0`

n. Create a new schema class.

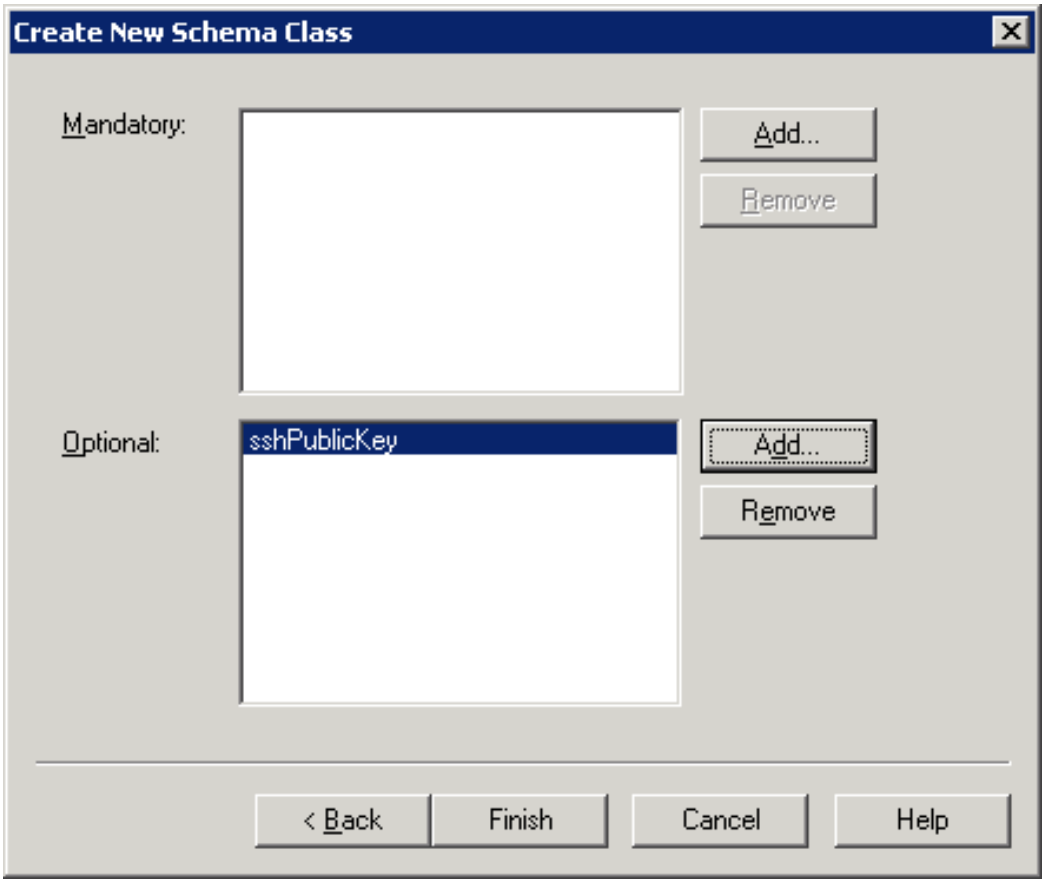
   In **Parent Class**, enter `top`, and in **Class Type**, select `Auxiliary`. Click **Next**.

Figure 280: Create New Schema Class — screen 2



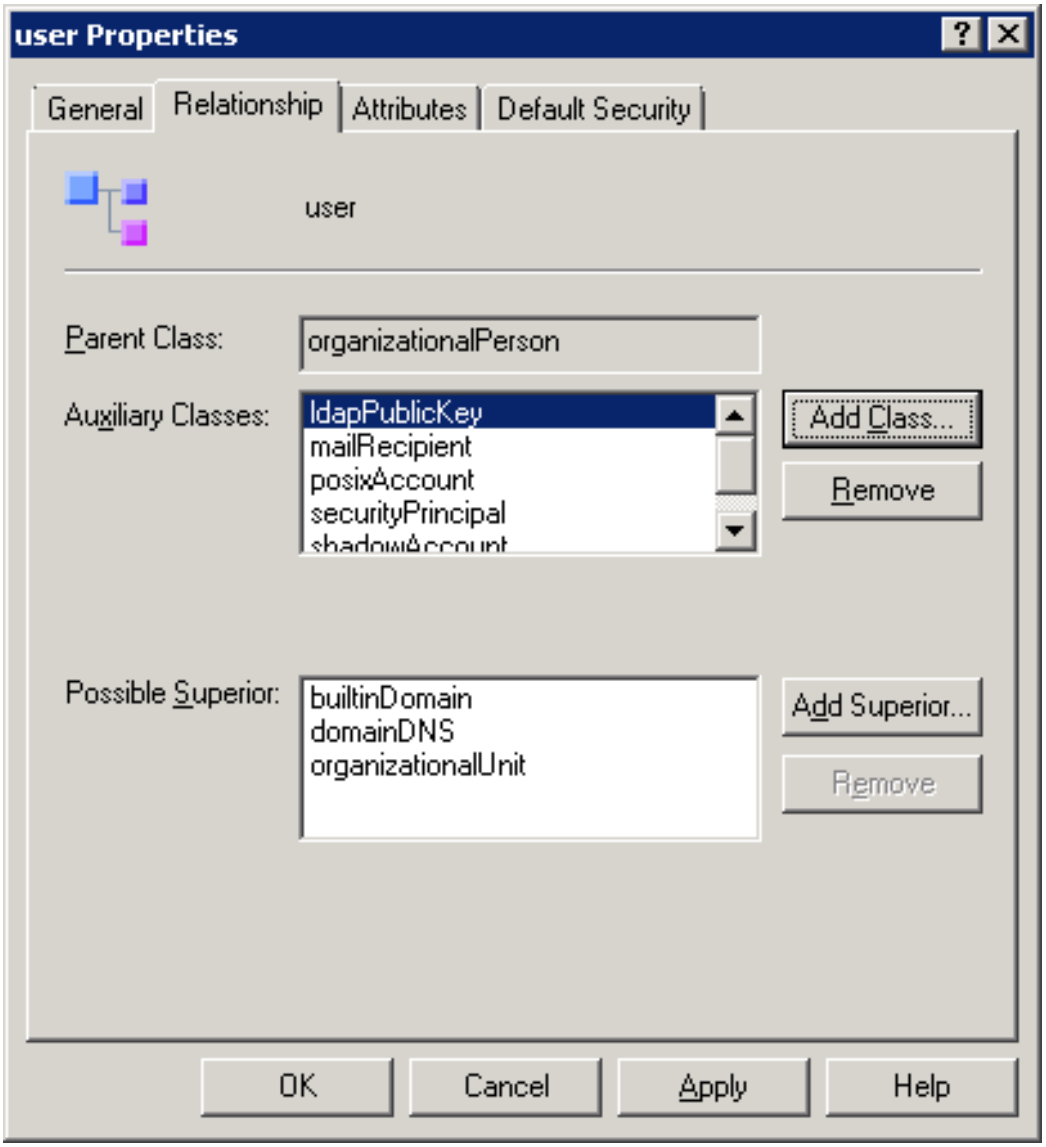
Add `sshPublicKey` to the **Optional** field. Click **Finish**.

Figure 281: Create New Schema Class — screen 1



o. Expand **Classes** and select **User**. Right-click **User** and select **Properties**. Navigate to **Relationship > Auxiliary Classes**, click **Add Class** and add the **ldapPublicKey** class. Click **Apply**.

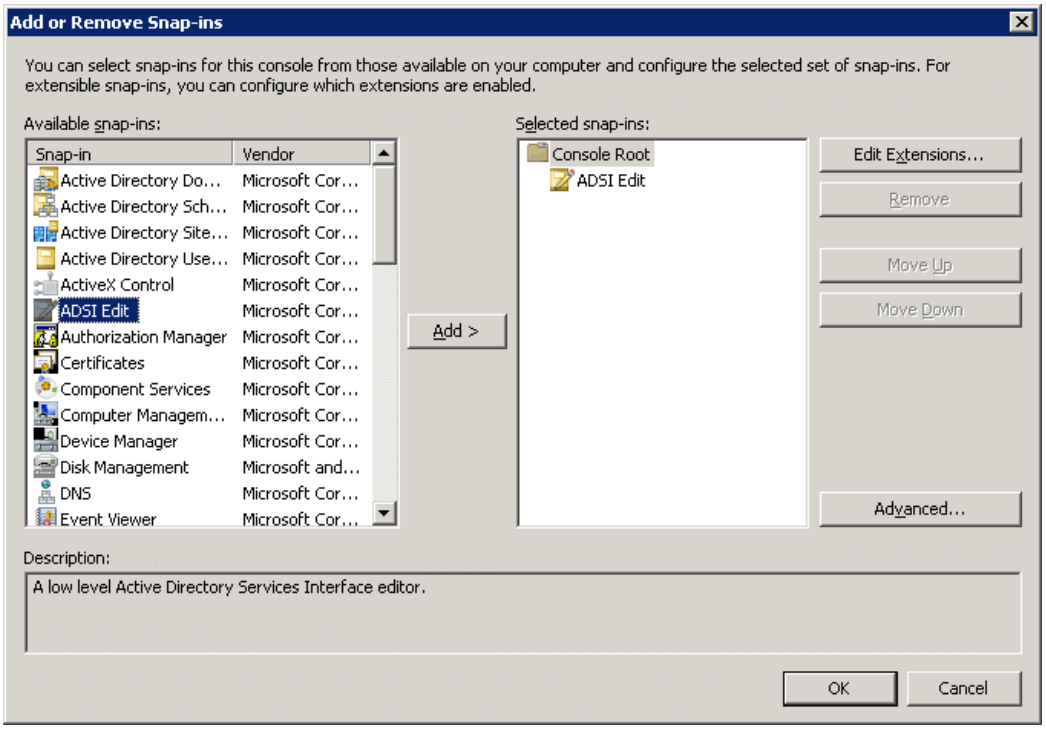
Figure 282: User Properties



5. The next step is to map the public keys to users. This is not possible in a user editor, use a low-level LDAP utility instead.

a. Add ADSI Edit as a snap-in to MMC.

Figure 283: Adding ADSI Edit



b. Right-click on the node and press **Enter**.

c. Search for the user in the tree, right-click on it and select *Properties.* All attributes can be edited there, so `sshPublicKey` too. Add the public keys to the Active Directory users.

> NOTE:
>
> It may happen that `sshPublicKey` is not visible in ADSI Edit. To make `sshPublicKey` visible, complete the procedure described in section Extending the Partial Attribute Set in https://blogs.technet.microsoft.com/scotts-it-blog/2015/02/28/ad-ds-global-catalogs-and-the-partial-attribute-set/.

6. Create a usermapping policy where you will set those groups from the Active Directory who can become root. For details on creating usermapping policies, see Configuring usermapping policies on page 740. In this scenario, only a few important details will be highlighted.

   a. Set **Username on the server** to `root` and select the group you intend to give these rights to.

   b. If you intend to allow other users in without usermapping, enable **Allow other unmapped usernames**.

7. Navigate to the relevant connection on the **SSH Control > Connections** page, and do the following:
    a. In the **Authentication policy** field, add the LDAP authentication policy you created in Step 1.
    b. In the **LDAP Server** field, add the LDAP server policy you created in Step 3.
    c. In the **Credential Store** field, add the Credential Store you created in Step 2.
    d. In the **Usermapping Policy** field, add the usermapping policy you created in Step 6.
    e. Click 

to save the change.

24

Troubleshooting One Identity Safeguard for Privileged Sessions (SPS)

This section describes the tools to detect networking problems, and also how to collect core dump files and view the system logs of One Identity Safeguard for Privileged Sessions (SPS).

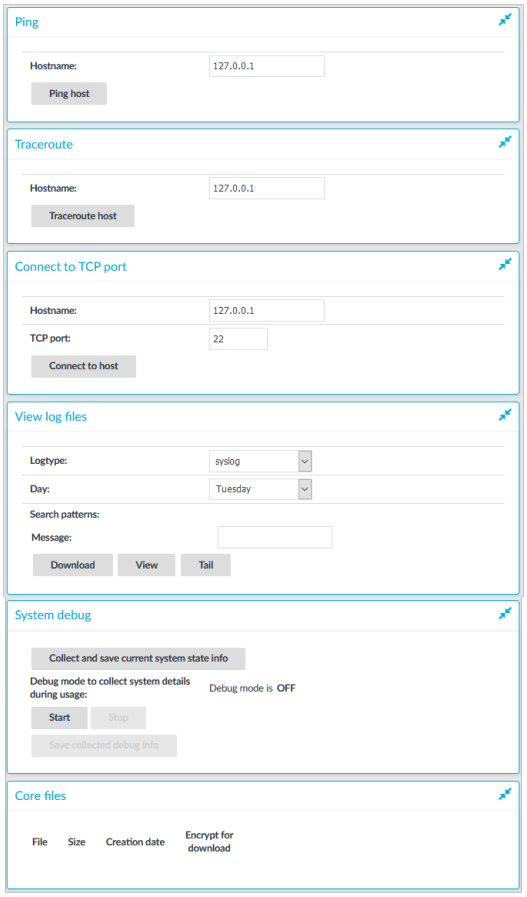
If you need to find the SPS appliance in the server room, you can use IPMI to control the front panel identify light. On One Identity Safeguard for Privileged Sessions N10000, navigate to **Basic Settings > System > Hardware information > Blink identification lights** and click **On** to blink the LEDs of hard disk trays on the front of the SPS appliance in red.

- Network troubleshooting
- Gathering data about system problems
- Viewing logs on One Identity Safeguard for Privileged Sessions (SPS)
- Changing log verbosity level of One Identity Safeguard for Privileged Sessions (SPS)
- Collecting logs and system information for error reporting
- Support hotfixes
- Status history and statistics
- Troubleshooting a One Identity Safeguard for Privileged Sessions (SPS) cluster
- Understanding One Identity Safeguard for Privileged Sessions (SPS) RAID status
- Restoring One Identity Safeguard for Privileged Sessions (SPS) configuration and data
- VNC is not working with TLS
- Configuring the IPMI from the BIOS after losing IPMI password
- Incomplete TSA response received
- Using UPN usernames in audited SSH connections

Network troubleshooting

The **Basic Settings > Troubleshooting** menu provides a number of diagnostic commands to resolve networking issues. Logfiles of One Identity Safeguard for Privileged Sessions (SPS) can also be displayed here — for details, see Viewing logs on One Identity Safeguard for Privileged Sessions (SPS) on page 856.

Figure 284: Basic Settings > Troubleshooting — Network troubleshooting with SPS



The following commands are available:

- **ping**: Sends a simple message to the specified host to test network connectivity.
- **traceroute**: Sends a simple message from SPS to the specified host and displays all hosts on the path of the message. It is used to trace the path the message travels between the hosts.
- **connect**: Attempts to connect the specified host using the specified port. It is used to test the availability or status of an application on the target host.

To execute one of the above commands

1. Navigate to **Basic Settings > Troubleshooting**.
2. Enter the IP address or the hostname of the target host into the **Hostname** field of the respective command. For the **Connect** command, enter the target port into the **TCP port** field.

   Use an IPv4 address.
3. Click the respective action button to execute the command.
4. Check the results in the pop-up window. Log files are displayed in a separate browser window.

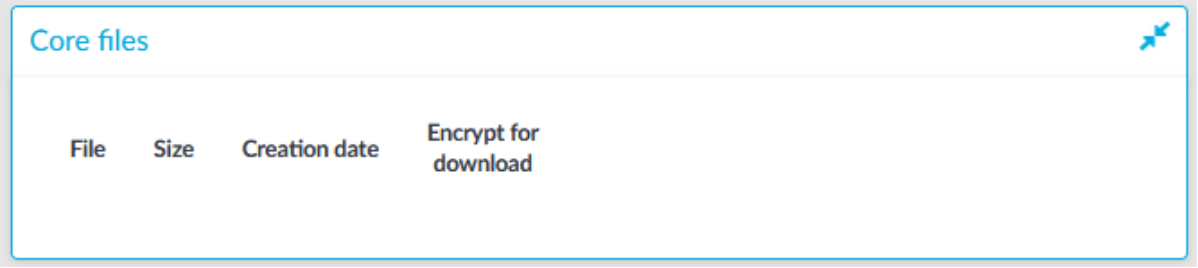
Gathering data about system problems

One Identity Safeguard for Privileged Sessions (SPS) automatically generates core dump files if an important software component (for example, Zorp) of the system crashes for some reason. These core dump files can be of great help to the One Identity Support Team to identify problems. When a core dump file is generated, the SPS administrator receives an alerting e-mail, and an SNMP trap is generated if alerting is properly configured (for details, see Configuring system monitoring on SPS on page 137 and System logging, SNMP and e-mail alerts on page 126).

To list and download the generated core dump files, navigate to **Basic Settings > Troubleshooting > Core files**.

By default, core dump files are deleted after 14 days. To change the deletion timeframe, navigate to **Basic Settings > Management > Core files**.

Figure 285: Basic Settings > Troubleshooting — System troubleshooting with SPS



Viewing logs on One Identity Safeguard for Privileged Sessions (SPS)

The **Troubleshooting** menu provides an interface to view the logs generated by the various components of One Identity Safeguard for Privileged Sessions (SPS).

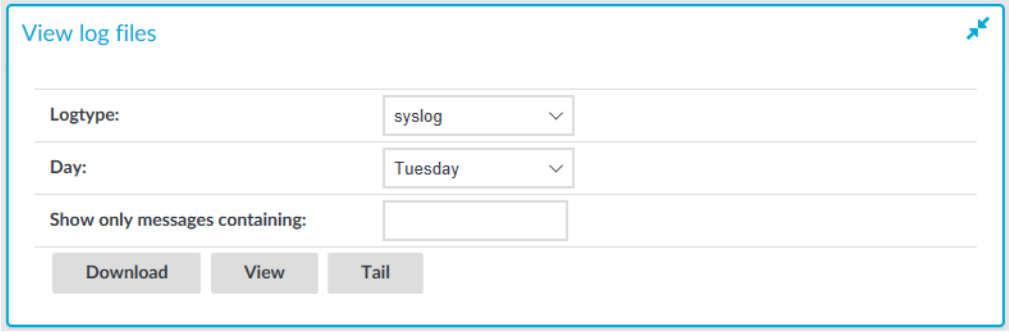
NOTE:

Because of performance reasons, log files larger than 2 Megabytes are not displayed in the web interface. To access these logs, download the file instead.

To view logs on SPS

1. Navigate to **Basic Settings > Troubleshooting > View log files**.

   Figure 286: Basic Settings > Troubleshooting — Viewing logs on SPS

2. Use the **Logtype** roll-down menu to select the message type.
   - `syslog`: All system logs of the SPS host.
   - `scb`: Logs of the SPS web interface.
   - `paa`: Logs related to the workings of the One Identity Safeguard for Privileged Analytics module.
   - `logadapter`: Logs of the log adapter plugin(s) and syslog instance(s) configured for ingesting logs from an external source.
   - `http`: Logs of the HTTP connections passing through SPS.
   - `ica`: Logs of the ICA connections passing through SPS.
   - `rdp`: Logs of the RDP connections passing through SPS.
   - `ssh`: Logs of the SSH connections passing through SPS.
   - `telnet`: Logs of the Telnet connections passing through SPS.
   - `vnc`: Logs of the VNC connections passing through SPS.
3. Use the buttons at the bottom of the dialog to perform the following tasks:
   - To download the log file, click **Download**.
   - To follow the current log messages real-time, click **Tail**.
   - To display the log messages, click **View**.
4. To display log messages of the last seven days, select the desired day from the **Day** field and click **View**.

TIP:

To display only the messages of a selected host or process, enter the name of the host or process into the **Show only messages containing** field.

The **Show only messages containing** field acts as a generic filter: enter a keyword or a regular expression to display only messages that contain the keyword or match the expression.

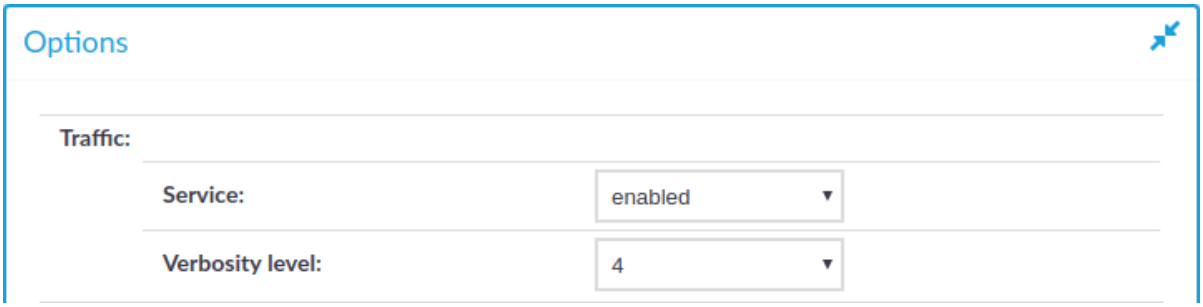
Changing log verbosity level of One Identity Safeguard for Privileged Sessions (SPS)

The logging level of One Identity Safeguard for Privileged Sessions (SPS) can be set separately for every protocol.

NOTE:

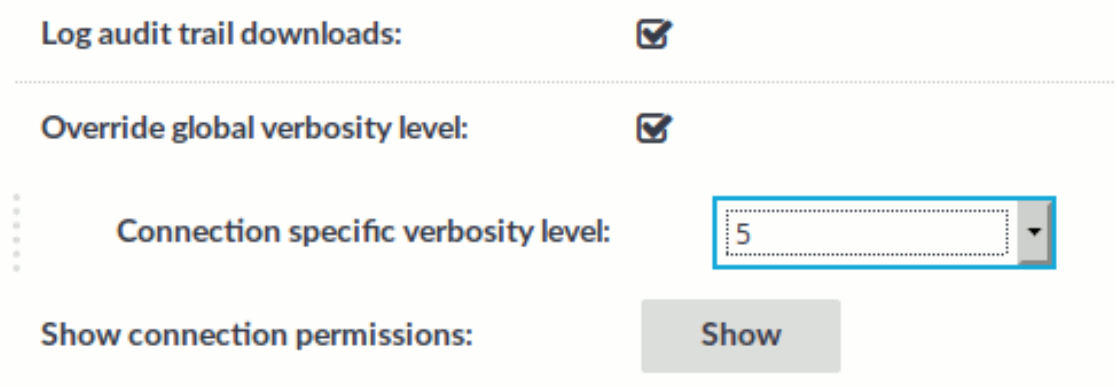
The **Basic Settings > Management > Verbose system logs > Enable** option is not related to the verbosity of traffic logs: it increases the log level of the non-network-related events, for example adds the commands executed by the SPS web interface to the logs, and so on.

Figure 287: <Protocol name> Control > Global Options — Changing the verbosity level



To change the verbosity level of SPS

1. Navigate to the **Global Options** page of the traffic you want to change the log level of, for example, to **SSH Control > Global Options** to change the log level of SSH traffic, **RDP Control > Global Options** for remote desktop traffic, and so on.



2. Select the desired log level from the **Verbosity level** field. Note that the new verbosity level applies only to new sessions started after committing the change. The verbosity level of active sessions will not change.

   NOTE:

   The verbosity level ranges from 1 (no logging) to 10 (extremely detailed), with level 4 being the default normal level. To debug complex problems, you might have to increase the verbosity level to 7. Higher level is needed only in extreme cases.

   CAUTION:

   High verbosity levels generate very large amount of log messages and might result in a very high load on the machine.

   For log levels 8-10, the logs contain highly sensitive data for all connections, as well as passwords and private keys in plain text format.

3. Click Commit.
4. *Optional:* To set a different verbosity level for sessions that belong to a specific Connection Policy, complete the following steps:
   - Navigate to the Connection Policy you want to modify.
   - Select **Override global verbosity level**.
   - Select the desired log level from the **Connection specific verbosity level** field. Note that the new verbosity level applies only to new sessions started after committing the change. The verbosity level of active sessions will not change.
   - Click Commit.

Collecting logs and system information for error reporting

To track down support requests, the One Identity Support Team might request you to collect system-state and debugging information. This information is collected automatically, and contains log files, the configuration file of One Identity Safeguard for Privileged Sessions (SPS), and various system-statistics.

NOTE:

Sensitive data like key files and passwords are automatically removed from the files, that is, configuration files do not contain passwords or keys. However, if you increase the proxy verbosity level to 8-10 in the Global Options, then for troubleshooting purposes, the logs can contain highly sensitive data, for example, passwords and keys in plain text format. If you are concerned about the presence of sensitive data, check the collected log files before submitting to the Support Portal.

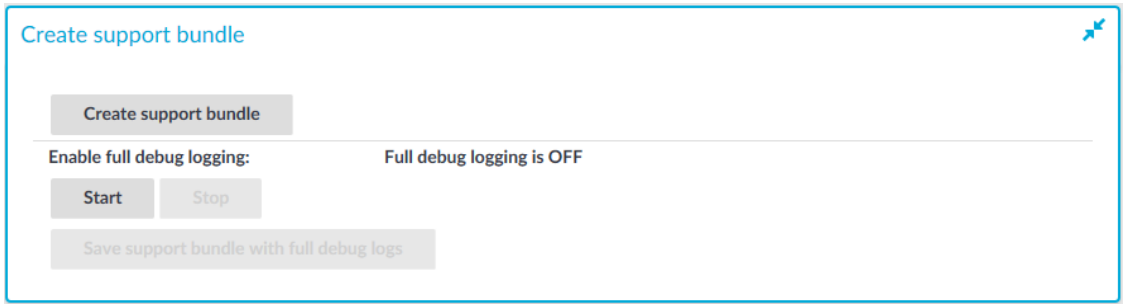
The **Basic Settings > Management > Verbose system logs > Enable** option is not related to the verbosity of log messages: it adds the commands executed by the SPS web interface to the log.

To collect system-state information (also known as a support bundle), navigate to **Basic Settings > Troubleshooting > Create support bundle** and click **Create support bundle**, then save the created zip file. The name of the file uses the `debug_info-<hostname>YYYYMMDDHHMM` format.

To collect information for a specific error

1. Navigate to **Basic Settings > Troubleshooting > Create support bundle**.

   Figure 288: Basic Settings > Troubleshooting > Create support bundle — Collecting debug information

2. Click **Start**.

NOTE:

Starting a full debug logging increases the log level of SPS, and might cause performance problems if the system is under a high load.

For troubleshooting purposes, the logs can contain highly sensitive data, for example, passwords and keys in plain text format. If you are concerned about the presence of sensitive data, check the collected log files before submitting to the Support Portal.

3. Reproduce the event that causes the error, for example connect to a server.
4. Click **Stop**.
5. Click **Save support bundle with full debug logs** and save the created zip file. The name of the file uses the `debug_info-<hostname>YYYYMMDDHHMM` format.

   SPS includes the configuration files of any plugins installed. Note that depending on the plugin, these configuration files can contain sensitive information, such as passwords or API keys. In this case, edit the plugin-related files in the `plugins` directory of the support bundle and delete the sensitive information.
6. Attach the file to your support ticket.

Support hotfixes

This section describes support hotfixes and their installation in One Identity Safeguard for Privileged Sessions (SPS).

Support hotfixes are official additions (signed .deb packages created by the Support Team) to a specific SPS release. By uploading a hotfix to an SPS appliance, it is possible to apply a modification (for example, a bugfix) quickly and without making the firmware Tainted. The hotfix files only work with the version of SPS they are created for.

You can upload the hotfix file you received from our Support Team in the SPS user interface.

CAUTION:

Consider the following:

- **Clicking Upload immediately installs the hotfix to SPS.**
- **Installing multiple hotfix files to a single configuration of SPS is possible, but you cannot delete an individual hotfix file from SPS without the Support Team's assistance.**
- **Installing a new firmware will delete all hotfix files installed on the previous version of SPS.**

If you have to delete an individual hotfix file from SPS without installing a new firmware first, contact our Support Team.

Installing support hotfixes

This section describes the most important requirements and information regarding the installation procedure of support hotfixes.

Prerequisites

The hotfix files are normally not publicly accessible for download (unless attached to Knowledgebase Articles). As a result, if you want to install them to your SPS, you must first contact our Support Team for a hotfix file specifically created for your request. Consider that you cannot delete the installed hotfix file from SPS without the Support Team's assistance. In addition, rebooting the SPS appliance after deleting an installed hotfix is necessary. We strongly recommend that you only install hotfixes to SPS if you contact our Support Team for instructions beforehand.

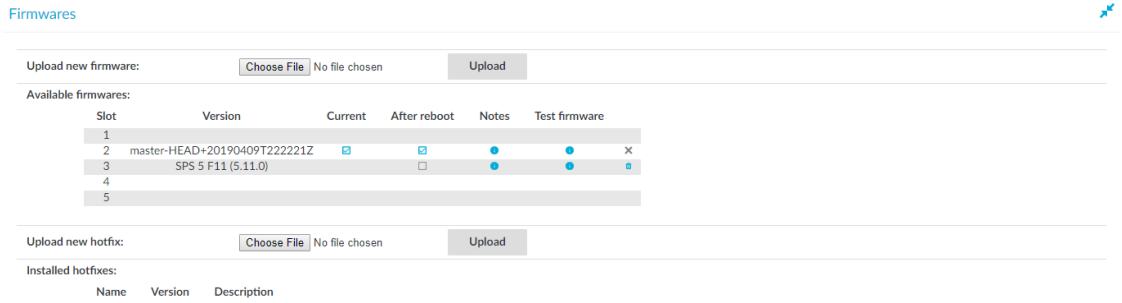
NOTE:

The hotfix files only work with the version of SPS they are created for. SPS automatically checks their version during upload.

To install the support hotfix file

1. Navigate to **Basic Settings > System > Firmwares**.

Figure 289: Uploading a hotfix file in the SPS user interface



2. Under the **Upload new hotfix:** section, click **Choose File** and select the hotfix file you want to upload.
3. Click **Upload**.

CAUTION:

Consider the following:

- **Clicking Upload immediately installs the hotfix to SPS.**
- **Installing multiple hotfix files to a single configuration of SPS is possible, but you cannot delete an individual hotfix file from SPS without the Support Team's assistance.**
- **Installing a new firmware will delete all hotfix files installed on the previous version of SPS.**

If you have to delete an individual hotfix file from SPS without installing a new firmware first, contact our Support Team.

4. If installation is successful, SPS will list information about the hotfix under **Installed hotfixes:**, such as **Name**, **Version** and **Description**.

NOTE:

Upload will fail in the following cases:

- The hotfix file version does not pass the version check.
- The hotfix file package is not properly signed by our Support Team.
- The file you want to upload is not an appropriate .deb package or the file is corrupted.

If upload fails, SPS will revert to its previous state automatically.

Status history and statistics

SPS displays various statistics and status history of system data and performance on the dashboard at **Basic Settings > Dashboard**. The dashboard is essentially an extension of the system monitor: the system monitor displays only the current values, while the dashboard creates graphs and statistics of the system parameters.

The dashboard consists of different modules. Every module displays the history of a system parameter for the current day. To display the graph for a longer period (last week, last month, or last year), select the **Week**, **Month**, or **Year** options, respectively. Hovering the mouse over a module enlarges the graph and displays the color code used on the graph.

All types of data is collected every five minutes. This means that if changes are more frequent, it might not be represented in the graphs.

NOTE:

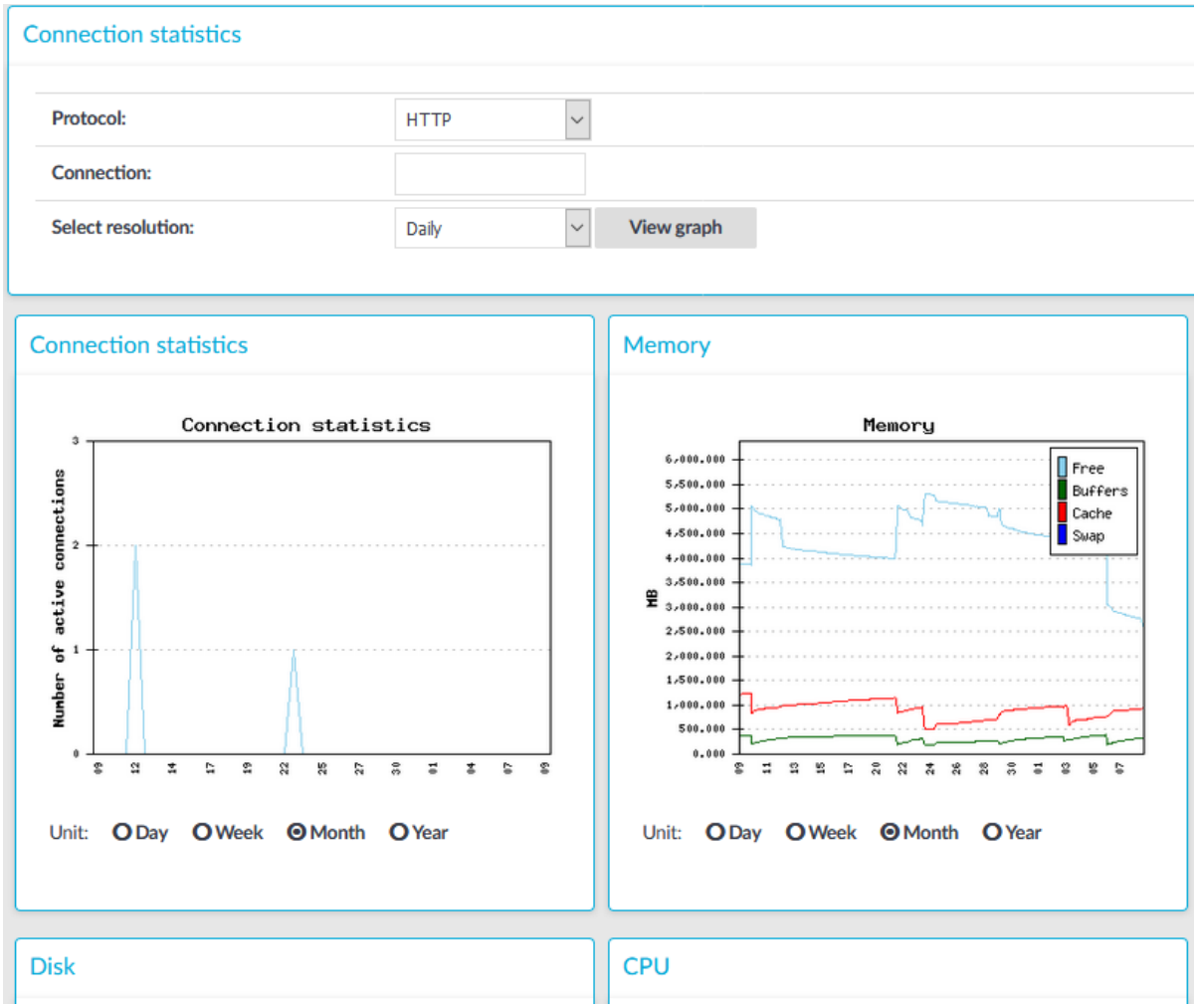
If all parameters displayed are 0 at a certain point in time, it might mean that at that time One Identity Safeguard for Privileged Sessions (SPS) was not functional (for example, turned off or unresponsive). Or, in certain cases it might also mean that there was no information at that time.

NOTE:

If you want to compare data displayed on the Dashboard to data displayed on the System Monitor, they might be different, because data on System Monitor is based on SNMP values, whereas data on the related Dashboard modules are based on the output of different commands.

To display statistics of a module as a table for the selected period, click on the graph.

Figure 290: Basic Settings > Dashboard — The dashboard

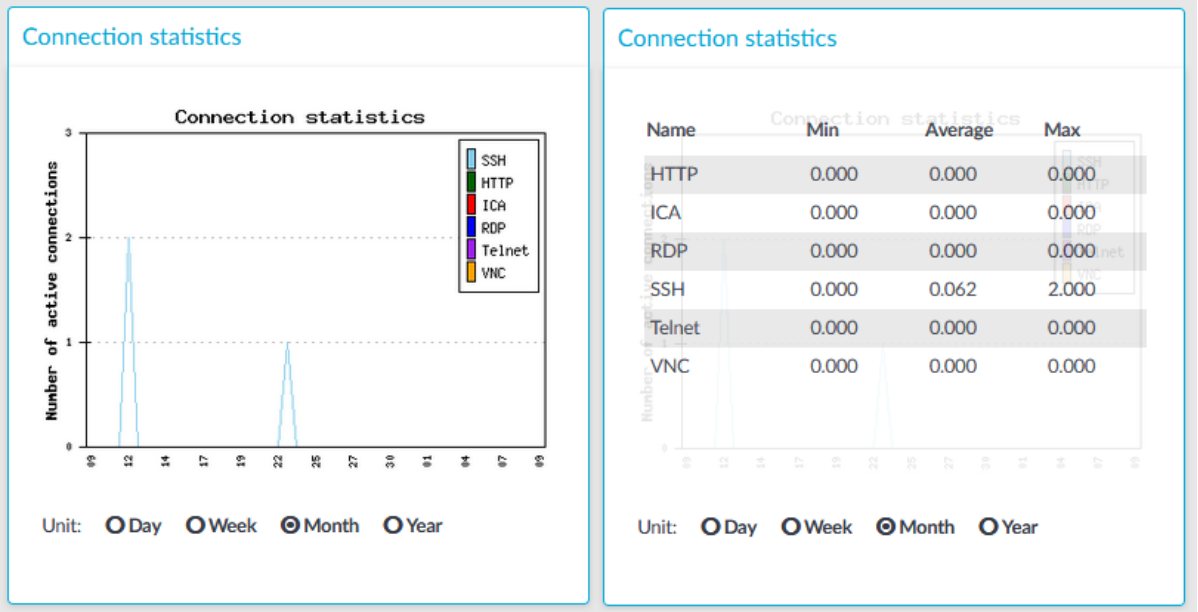


The following modules are displayed on the dashboard of SPS:

- **Connection statistics**: Number of active connections per protocol.
- **Memory**: The memory used by the system.
- **Disk**: Filesystem usage for the different partitions.
- **CPU**: CPU usage.
- **Network connections**: Number of network connections.
- **Physical interface 1 (eth0)**: Traffic on physical interface 1.
- **Physical interface 2 (eth1)**: Traffic on physical interface 2.
- **Physical interface 3 (eth2)**: Traffic on physical interface 3.
- **Load average**: Average load of the system.
- **Number of processes**: The number of running processes.

Connection statistics

Figure 291: Basic Settings > Dashboard > Connection statistics



The **Connection statistics** module on the **Dashboard** is based on statistics of high-level proxy-service protocols (SSH, RDP, VNC, ICA, and so on). These numbers display all active high-level proxy-service protocols, but these numbers are counted by all service connections too, which are connected to some protocols. Because of this, these numbers can differ from the numbers displayed on the **Active Connections** page.

For example, if there are several active ICA connections in your system, it means that there are approximately the same number of CGP connections that are opened and counted in the **Connection statistics** module under the ICA label. If these CGP or ICA high-level proxy-service protocols are opening more than one TCP connections, these connections will be counted in the **Network connetion** module as different TCP connections, but these will count as only one connection on the **Active Connections** page.

Statistics

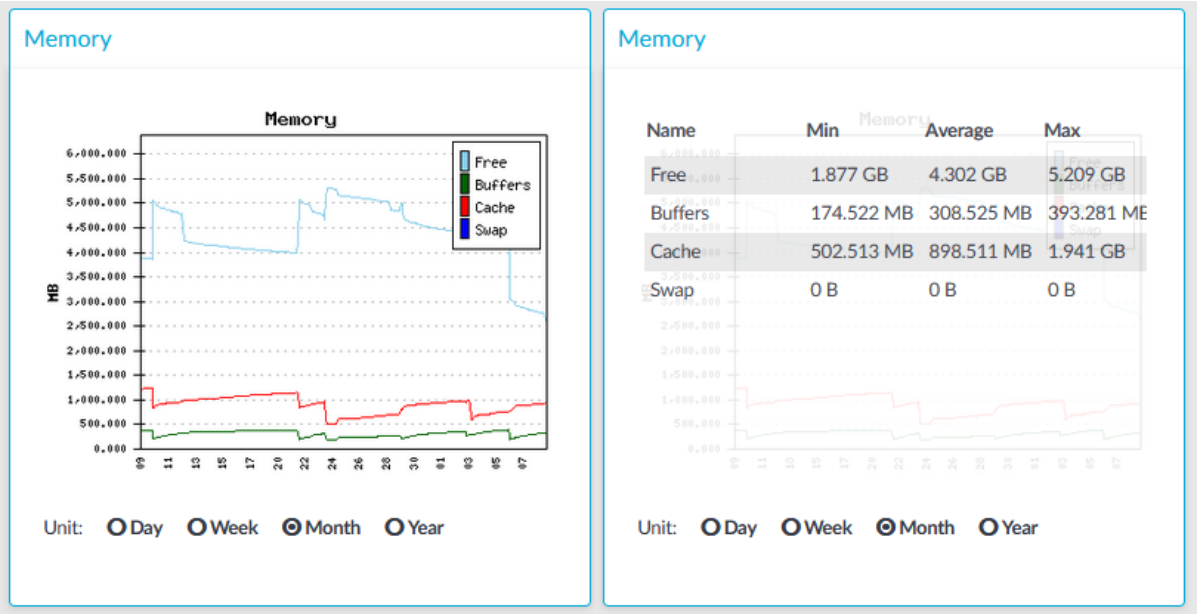
The connection types displayed can be the following:

- **RDP**: The number of RDP connections.
- **SSH**:The number of SSH connections.
- **HTTP**: The number of HTTP connections.
- **ICA**: The number of Citrix connections.
- **Telnet**: The number of Telnet connections.
- **VNC**: The number of VNC connections.

The **Min**, **Average** and **Max** values are displayed as a whole number if the value is constant for the statistics interval (the statistics are stored every 5 minutes). If minor changes occur in the actual values (for example, new connections are established), these changes can be displayed as fractions.

Memory

Figure 292: Basic Settings > Dashboard > Memory



The **Memory** module on the **Dashboard** is based on data provided by the Linux kernel (`/proc` and `/sys` directories). The standard Munin plugins query this information from these locations and they are displayed on the GUI.

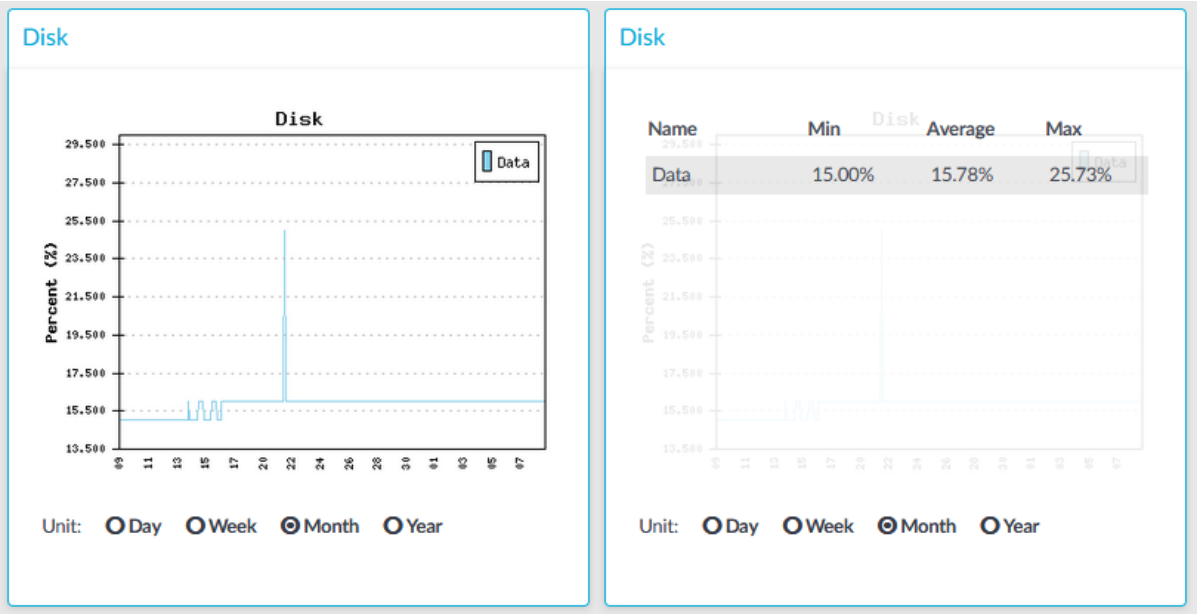
Statistics

The memory types displayed are the following:

- **Free**: Free memory
- **Buffers**: In-memory block I/O buffers.
- **Cache**: Memory used for disk caching. This does not count as "used" memory, because it is freed when it is required.
- **Swap**: Swap space usage (memory contents that have been temporarily moved to disk).This value might be high in case of lack of memory.

Disk

Figure 293: Basic Settings > Dashboard > Disk



The **Disk** module on the **Dashboard** is based on the output of the **df** command.

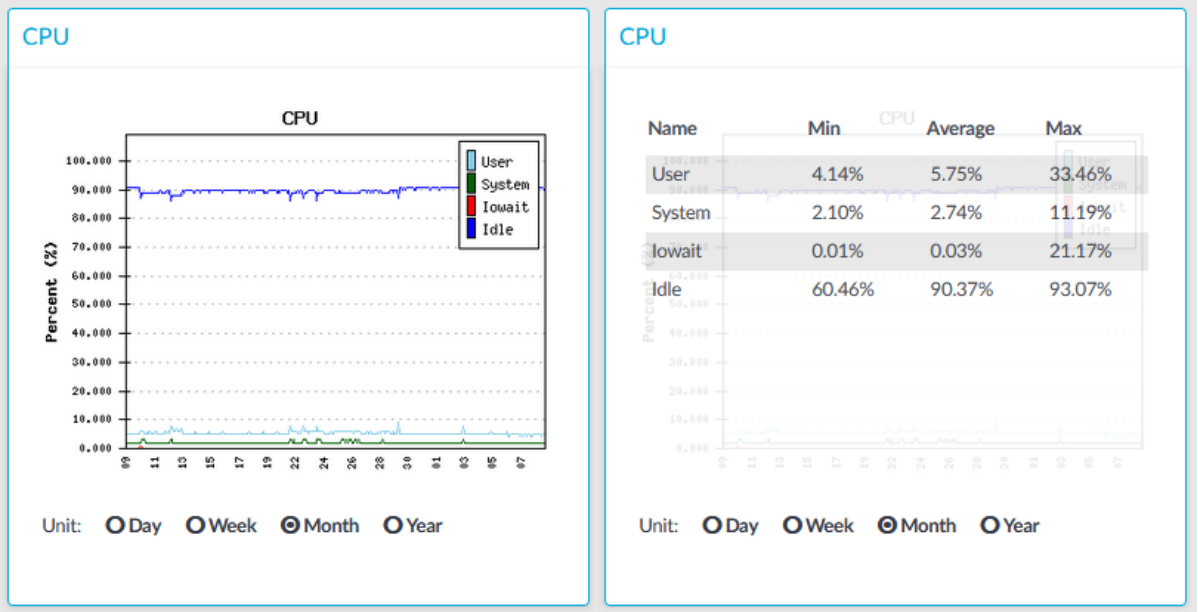
Statistics

The information displayed is the following:

- **Data**: The percent of disk that the core firmware uses.

CPU

Figure 294: Basic Settings > Dashboard > CPU



The **CPU** module on the **Dashboard** is based on data provided by the Linux kernel (`/proc` and `/sys` directories). The standard Munin plugins query this information from these locations and they are displayed on the GUI.

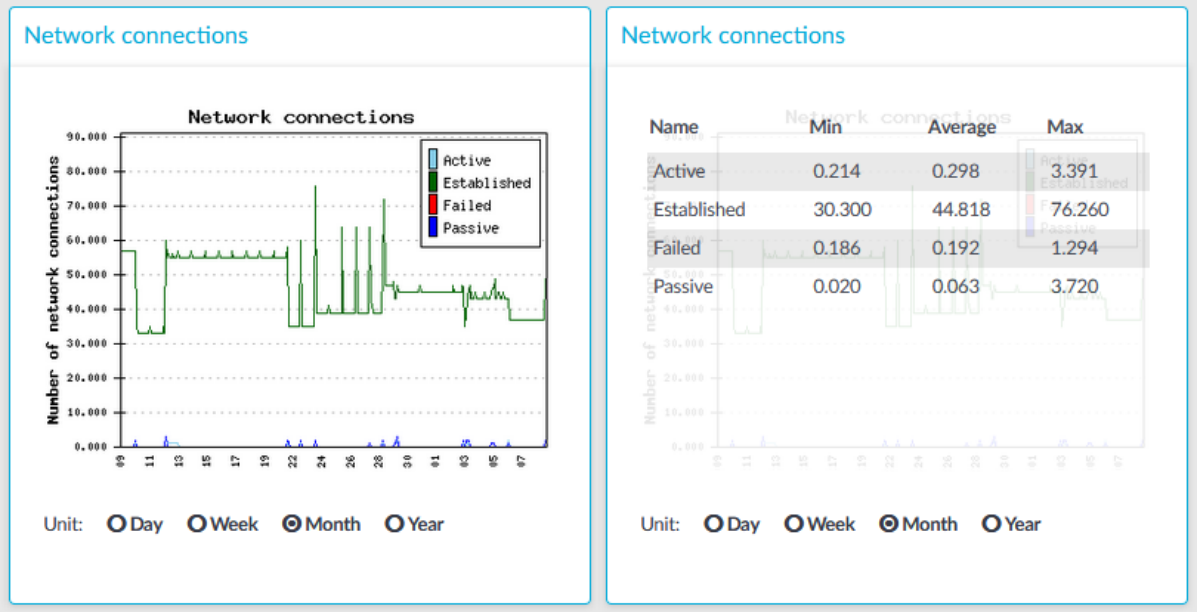
Statistics

The following details are displayed about CPU usage:

- **Idle**: Idle time of the processors. If there are more than one processors, they all add up to x100%, for example in case of 2 processors it adds up to 200% maximum.
- **Iowait**: Time spent receiving and handling hardware interrupts as a percentage of processor ticks. That is, waiting for IO.
- **System**: Kernel CPU usage.
- **User**: CPU usage of everything other than kernel.

Network connections

Figure 295: Basic Settings > Dashboard > Network connections



The **Network connetion** module on the **Dashboard** is based on the output of the **netstat -s** command. This command generates statistical information from all interfaces of all TCP connections. This means that in addition to the high-level proxy-service protocols (SSH, RDP, VNC, ICA, and so on), but all types of TCP connections are counted as well. The standard Munin plugins query this information and then it is displayed on the GUI. The graph itself displays the TCP activity of all network interfaces combined.

Statistics

The connection types displayed can be the following:

- **Active**: The number of active TCP openings per second.
- **Established**: The number of currently open connections.
- **Failed**: The number of failed TCP connection attempts per second.
- **Passive**: The number of passive TCP openings per second.
- **Resets**: The number of TCP connection resets.

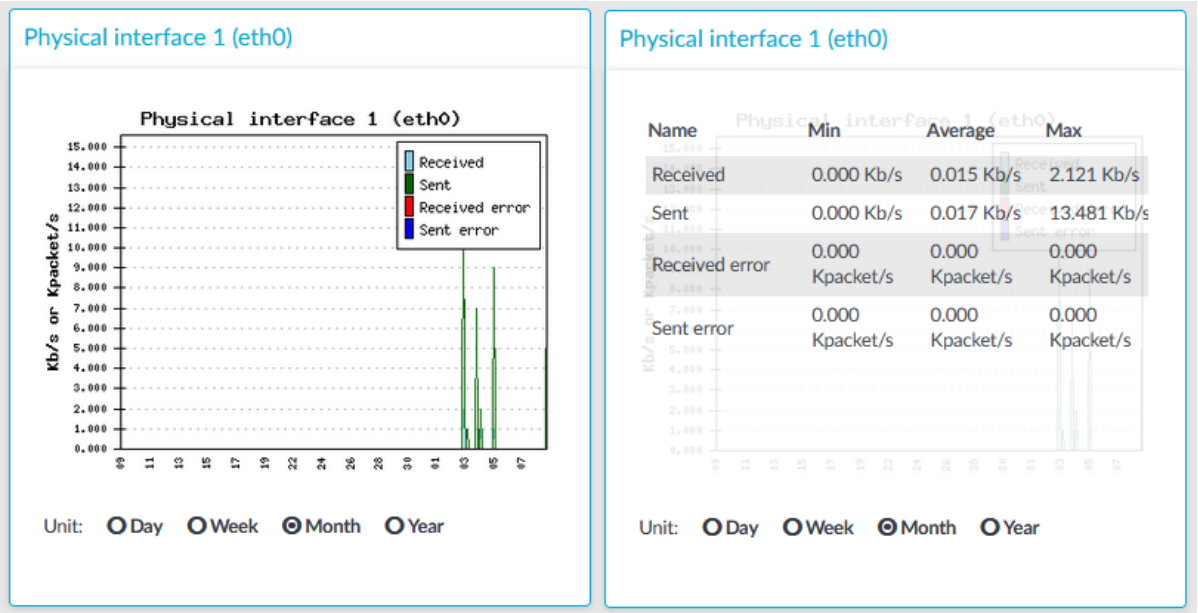
The **Min**, **Average** and **Max** values are displayed as a whole number if the value is constant for the statistics interval (the statistics are stored every 5 minutes). If minor changes occur in the actual values (for example, new connections are established), these changes can be displayed as fractions.

To determine the maximum values that the system can handle, consider the following:

- The type of machine that you run One Identity Safeguard for Privileged Sessions (SPS) on.
- The type of connections that are established and the content of these connections. For example:
    - If the users of RDP or ICA connections are watching videos, that can greatly reduce the amount of parallel connections that can pass through without experiencing speed reduction.
    - If the users mostly generate text-based content (for example, Excel, Word), then more connections can be used.
    - If a connection is not used actively, then it has minimal impact on SPS: only the memory allocation remains. In case of RDP, if the RDP client window is minimized, there is no network traffic at all.

Interface

Figure 296: Basic Settings > Dashboard > Interface



The **Interface** module on the **Dashboard** is based on data provided by the Linux kernel (`/proc` and `/sys` directories). The standard Munin plugins query this information from these locations and they are displayed on the GUI.

Statistics

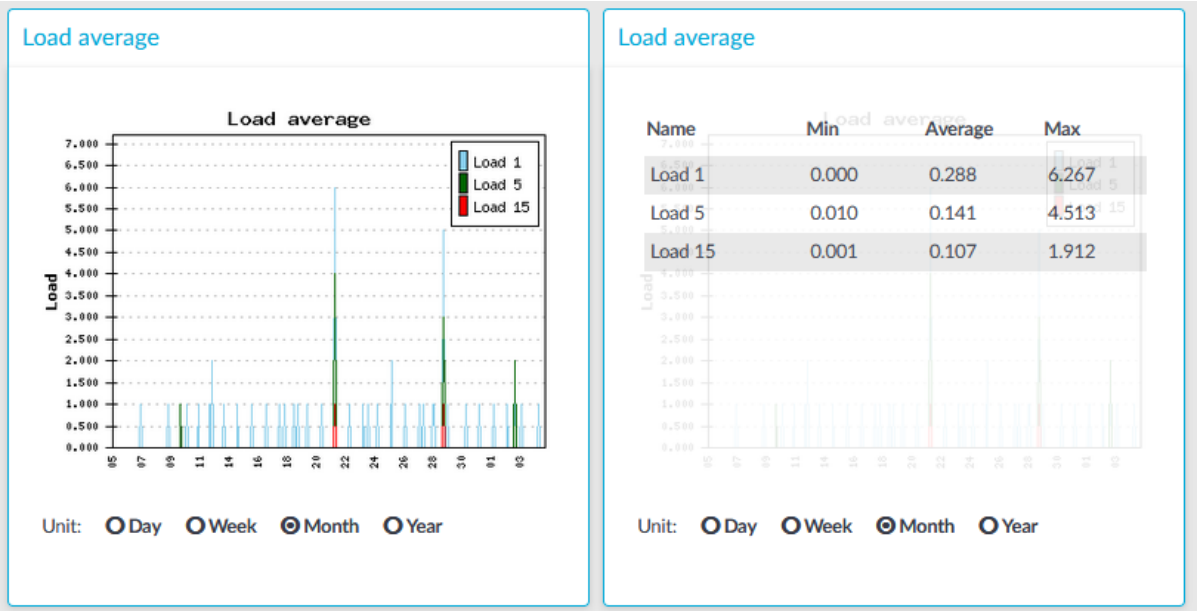
The memory types displayed are the following:

- **Received**: The network interface has received x Kilobytes per second.
- **Sent**: The network interface has sent x Kilobytes per second.

- **Received error**: The amount of errors, packet drops, and collisions on the network interface (in Kilopackets per second).
- **Sent error**: The amount of errors, packet drops, and collisions on the network interface (in Kilopackets per second).

Load average

Figure 297: Basic Settings > Dashboard > Load average



The **Load average** module on the **Dashboard** is based on data provided by the Linux kernel (`/proc` and `/sys` directories). The standard Munin plugins query this information from these locations and they are displayed on the GUI.

A measure of the amount of computational work that a computer system performs. The load average represents the average system load over a period of time. It conventionally appears in the form of three numbers which represent the system load during the last one-, five-, and fifteen-minute periods.

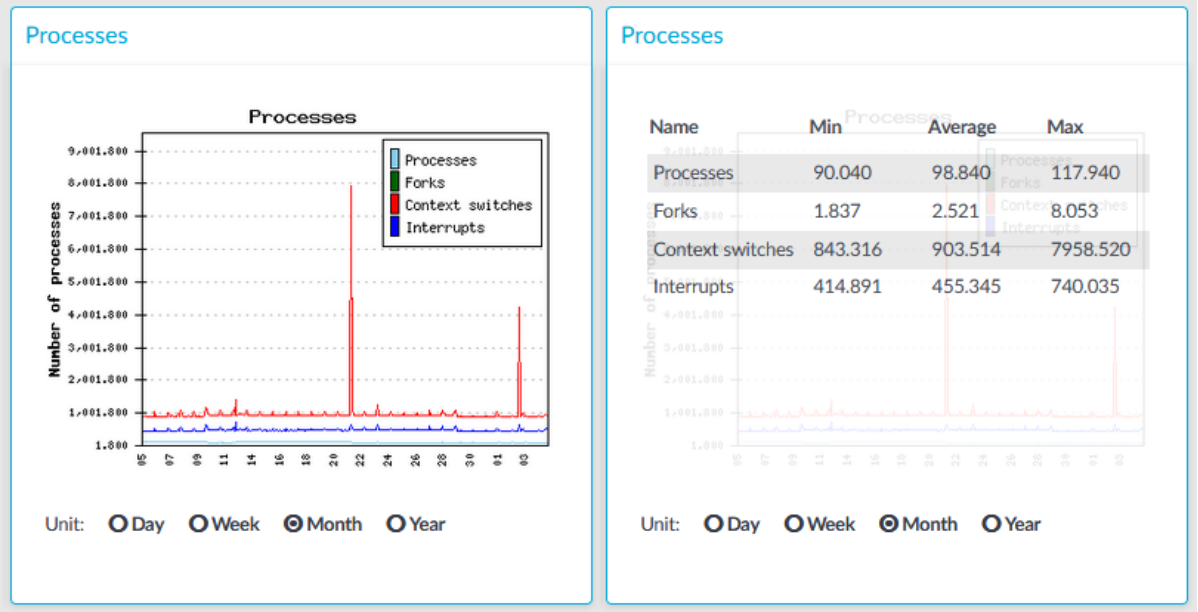
Statistics

The load average types displayed are the following:

- **Load 1**: Average load in 1 minute.
- **Load 5**: Average load in 5 minute.
- **Load 15**: Average load in 15 minute.

Number of processes

Figure 298: Basic Settings > Dashboard > Number of processes



The **Number of processes** module on the **Dashboard** is based on the output of the **ps** command.

Statistics

The process types displayed are the following:

- **Processes**: Number of running processes.
- **Forks**: Number of forks (system calls). It is an operation where a process creates a copy of itself.
- **Context switches**: Number of context switches. It is the switching of the CPU from one processor thread to another.
- **Interrupts**: Number of interrupts.

Displaying custom connection statistics

The following describes how to display statistics of a specific connection policy.

To display statistics of a specific connection policy

1. Navigate to **Basic Settings > Dashboard > Connection statistics**.
2. To display the statistics of a connection policy, enter the name of the policy into the **Connection**.
3. Select the time period to display from the **Select resolution** field.
4. Click **View graph**.

Troubleshooting a One Identity Safeguard for Privileged Sessions (SPS) cluster

The following sections help you to solve problems related to high availability clusters.

- For a description of the possible statuses of the One Identity Safeguard for Privileged Sessions (SPS) cluster and its nodes, the DRBD data storage system, and the heartbeat interfaces (if configured), see Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses on page 873.
- To recover a cluster that has broken down, see Recovering One Identity Safeguard for Privileged Sessions (SPS) if both nodes broke down on page 876.
- To resolve a split-bran situation when the nodes of the cluster were simultaneously active for a time, see Recovering from a split brain situation on page 876.
- To replace a broken node with a new appliance, see Replacing a HA node in a One Identity Safeguard for Privileged Sessions (SPS) cluster on page 879.

Understanding One Identity Safeguard for Privileged Sessions (SPS) cluster statuses

This section explains the possible statuses of the One Identity Safeguard for Privileged Sessions (SPS) cluster and its nodes, the DRBD data storage system, and the heartbeat interfaces (if configured). SPS displays this information on the **Basic Settings > High Availability** page.

The **Status** field indicates whether the SPS nodes recognize each other properly and whether those are configured to operate in high availability mode. The status of the individual SPS nodes is indicated in the **Node HA state** field of the each node. The following statuses can occur:

- **Standalone**: There is only one SPS unit running in `standalone` mode, or the units have not been converted to a cluster (the **Node HA state** of both nodes is `standalone`). Click **Convert to Cluster** to enable High Availability mode.

- **HA**: The two SPS nodes are running in High Availability mode. **Node HA state** is `HA` on both nodes, and the **Node HA UUID** is the same on both nodes.
- **Half**: High Availability mode is not configured properly, one node is in `standalone`, the other one in `HA` mode. Connect to the node in `HA` mode, and click **Join HA** to enable High Availability mode.
- **Broken**: The two SPS nodes are running in High Availability mode. **Node HA state** is `HA` on both nodes, but the **Node HA UUID** is different. For assistance, contact our Support Team.
- **Degraded**: SPS was running in high availability mode, but one of the nodes has disappeared (for example broken down, or removed from the network). Power on, reconnect, or repair the missing node.
- **Degraded (Disk Failure)**: A hard disk of the secondary node is not functioning properly and must be replaced. To request a replacement hard disk and for details on replacing the hard disk, contact our Support Team.
- **Degraded Sync**: Two SPS units were joined to High Availability mode, and the first-time synchronization of the disks is currently in progress. Wait for the synchronization to complete. Note that in case of large disks with lots of stored data, synchronizing the disks can take several hours.
- **Split brain**: The two nodes lost the connection to each other, with the possibility of both nodes being active nodes (that is, primary nodes) for a time.

  CAUTION:

  Hazard of data loss In this case, valuable audit trails might be available on both SPS nodes, so special care must be taken to avoid data loss. For details on solving this problem, see Recovering from a split brain situation on page 876.

  Do NOT reboot or shut down the nodes.
- **Invalidated**: The data on one of the nodes is considered out-of-sync and should be updated with data from the other node. This state usually occurs during the recovery of a split-brain situation when the DRBD is manually invalidated.
- **Converted**: After converting nodes to a cluster (clicking **Convert to Cluster**) or enabling High Availability mode (clicking **Join HA**) and before rebooting the node(s).

NOTE:

If you experience problems because the nodes of the HA cluster do not find each other during system startup, navigate to **Basic Settings > High Availability** and select **HA (Fix current)**. That way the IP address of the HA interfaces of the nodes will be fix, which helps if the HA connection between the nodes is slow.

The **DRBD status** field indicates whether the latest data (including SPS configuration, audit trails, log files, and so on) is available on both SPS nodes. The primary node (this node) must always be in **consistent** status to prevent data loss. Inconsistent status means that the data on the node is not up-to-date, and should be synchronized from the node having the latest data.

The **DRBD status** field also indicates the connection between the disk system of the SPS nodes. The following statuses are possible:

- **Connected**: Both nodes are functioning properly.
- **Connected (Disk Failure)**: A hard disk of the secondary node is not functioning properly and must be replaced. To request a replacement hard disk and for details on replacing the hard disk, contact our Support Team.
- **Invalidated**: The data on one of the nodes is considered out-of-sync and should be updated with data from the other node. This state usually occurs during the recovery of a split-brain situation when the DRBD is manually invalidated.
- **Sync source** or **Sync target**: One node (**Sync target**) is downloading data from the other node (**Sync source**).

  When synchronizing data, the progress and the remaining time is displayed in the **System monitor**.

  > **CAUTION:**
  >
  > **When the two nodes are synchronizing data, do not reboot or shutdown the primary node. If you absolutely must shutdown the primary node during synchronization, shutdown the secondary node first, and then the primary node.**

- **Split brain**: The two nodes lost the connection to each other, with the possibility of both nodes being active nodes (that is, primary nodes) for a time.

  > **CAUTION:**
  >
  > **Hazard of data loss In this case, valuable audit trails might be available on both SPS nodes, so special care must be taken to avoid data loss. For details on solving this problem, see Recovering from a split brain situation on page 876.**

- **WFConnection**: One node is waiting for the other node, the connection between the nodes has not been established yet.

If a redundant heartbeat interface is configured, its status is also displayed in the **Redundant Heartbeat status** field, and also in the **HA > Redundant** field of the System monitor. For a description of redundant heartbeat interfaces, see Redundant heartbeat interfaces on page 374.

The possible status messages are explained below.

- **NOT USED**: There are no redundant heartbeat interfaces configured.
- **OK**: Normal operation, every redundant heartbeat interface is working properly.
- **DEGRADED-WORKING**: Two or more redundant heartbeat interfaces are configured, and at least one of them is functioning properly. This status is displayed also when a new redundant heartbeat interface has been configured, but the nodes of the SPS cluster has not been restarted yet.

- **DEGRADED**: The connection between the redundant heartbeat interfaces has been lost. Investigate the problem to restore the connection.
- **INVALID**: An error occurred with the redundant heartbeat interfaces. Contact the One Identity Support Team for help. For assistance, contact our Support Team.

Recovering One Identity Safeguard for Privileged Sessions (SPS) if both nodes broke down

It can happen that both nodes break down simultaneously (for example because of a power failure), or the secondary node breaks down before the original primary node recovers.

NOTE:

As of One Identity Safeguard for Privileged Sessions (SPS) version 2.0.2, when both nodes of a cluster boot up in parallel, the node with the `1.2.4.1` HA IP address will become the primary node.

To properly recover SPS

1. Power off both nodes by pressing and releasing the power button.

   CAUTION:

   Hazard of data loss If SPS does not shut off, press and hold the power button for approximately 4 seconds. This method terminates connections passing SPS and might result in data loss.

2. Power on the node that was the primary node before SPS broke down. Consult the system logs to find out which node was the primary node before the incident: when a node boots as primary node, or when a takeover occurs, SPS sends a log message identifying the primary node.

   TIP:

   Configure remote logging to send the log messages of SPS to a remote server where the messages are available even if the logs stored on SPS become unaccessible. For details on configuring remote logging, see System logging, SNMP and e-mail alerts on page 126.

3. Wait until this node finishes the boot process.
4. Power on the other node.

Recovering from a split brain situation

A split brain situation is caused by a temporary failure of the network link between the cluster nodes, resulting in both nodes switching to the active (that is, primary node) role while disconnected. This might cause new data (for example, audit trails) to be created on

both nodes without being replicated to the other node. Thus, it is likely in this situation that two diverging sets of data have been created, which cannot be trivially merged.

CAUTION:

Hazard of data loss In a split brain situation, valuable audit trails might be available on both One Identity Safeguard for Privileged Sessions (SPS) nodes, so special care must be taken to avoid data loss.

The nodes of the SPS cluster automatically recognize the split brain situation once the connection between the nodes is reestablished, and do not perform any data synchronization to prevent data loss. When a split brain situation is detected, it is visible on the SPS system monitor, in the system logs (`Split-Brain detected, dropping connection!`), on the **Basic Settings > High Availability** page, and SPS sends an alert as well.

Once the network connection between the nodes has been re-established, one of the nodes will become the active (that is, primary) node, while the other one will be the backup node (that is, the secondary node). This means that one node is providing services similar to normal operation, and the other one is kept passive (as a backup) to avoid network interferences. Note that there is no synchronization between the nodes at this stage.

To recover a SPS cluster from a split brain situation, complete the following steps.

CAUTION:

Do NOT shut down the nodes.

Data recovery

In the procedure described here, data will be saved from the host currently acting as the secondary node host. This is required because data on this host will later be overwritten by the data available on the current primary node.

NOTE:

During data recovery, there will be no service provided by SPS.

To recover from a split brain situation

1. Log in to the primary node. If no Console menu is showing up after login, then this is the secondary node. In this case, try the other node.
2. Select **Shells > Boot Shell**.
3. Enter `/usr/share/heartbeat/hb_standby`. This will change the current secondary node to primary node and the current primary node to secondary node (HA failover).
4. Exit the console.
5. Wait a few seconds for the HA failover to complete.
6. Log in on the other host. If no Console menu is showing up, the HA failover has not completed yet. Wait a few seconds and try logging in again.
7. Select **Shells > Core Shell**.

8. Issue the **systemctl stop zorp-core.service** command to disable all traffic going through SPS.
9. Save the files from `/var/lib/zorp/audit` that you want to keep. Use **scp** or **rsync** to copy data to your remote host.

   TIP:

   To find the files modified in the last *n*24* hours, use `find . -mtime -n`.

   To find the files modified in the last *n* minutes, use `find . -mmin -n .`
10. Enter:

    ```
    pg_dump -U scb -f /root/database.sql
    ```

    Back up the `/root/database.sql` file.
11. Exit the console.
12. Log in again, and select **Shells > Boot Shell**.
13. Enter `/usr/share/heartbeat/hb_standby`. This will change the current secondary node to primary node and the current primary node to secondary node (HA failover).
14. Exit the console.
15. Wait a few minutes to let the failover happen, so the node you were using will become the secondary node and the other node will become the primary node.

    The nodes are still in a split-brain state but now you have all the data backed up from the secondary node, and you can synchronize the data from the primary node to the secondary node, which will turn the HA state from "Split-brain" to "HA". For details on how to do that, see HA state recovery on page 878.

HA state recovery

In the procedure described here, the "Split-brain" state will be turned to the "HA" state. Keep in mind that the data on the current primary node will be copied to the current secondary node and *data that is available only on the secondary node will be lost* (as that data will be overwritten).

Steps: Swapping the nodes (optional)

NOTE:

If you completed the procedure described in Data recovery on page 877, you do not have to swap the nodes. You can proceed to the steps about data synchronization.

If you want to swap the two nodes to make the primary node the secondary node and the secondary node the primary node, perform the following steps:

1. Log in to the primary node. If no Console menu is showing up after login, then this is the secondary node. In this case, try the other node.
2. Select **Shells > Boot Shell**.

3. Enter `/usr/share/heartbeat/hb_standby`. This will output:

   ```
   Going standby [all]
   ```

4. Exit the console.
5. Wait a few minutes to let the failover happen, so the node you were using will become the secondary node and the other node will be the primary node.

Steps: Initializing data synchronization

To initialize data synchronization, complete the following steps:

1. Log in to the secondary node. If the Console menu is showing up, then this is the primary node. In this case, try logging in to the other node.
2. Enter the following commands. These commands will make the secondary node discard the data available only here, on this node.

   ```
   drbdadm secondary r0
   drbdadm connect --discard-my-data r0
   ```

3. Log out of the secondary node.
4. Log in to the primary node.
5. Select **Shells > Boot Shell**.
6. Enter:

   ```
   drbdadm connect r0
   ```

7. Exit the console.
8. Check the High Availability state on the web interface of SPS, in the **Basic Settings > High Availability > Status** field. During synchronization, the status will say **Degraded Sync**, and after the synchronization completes, it will say **HA**.

Replacing a HA node in a One Identity Safeguard for Privileged Sessions (SPS) cluster

The following describes how to replace a unit in a One Identity Safeguard for Privileged Sessions (SPS) cluster with a new appliance.

To replace a unit in a SPS cluster with a new appliance

1. Verify the HA status on the working node. Select **Basic Settings > High Availability**. If one of the nodes has broken down or is missing, the **Status** field displays `DEGRADED`.

2. Note down the **Gateway IP** addresses, and the IP addresses of the **Heartbeat** and the **Next hop monitoring** interfaces.
3. Perform a full system backup. Before replacing the node, create a complete system backup of the working node. For details, see Data and configuration backups on page 146.
4. Check which firmware version is running on the working node. Select **Basic Settings > System > Version details** and write down the exact version numbers.
5. Log in to your support portal and download the CD ISO for the same SPS version that is running on your working node.
6. Without connecting the replacement unit to the network, install the replacement unit from the ISO file. Use the IPMI if needed.
7. When the installation is finished, connect the two SPS units with an Ethernet cable via the Ethernet connectors labeled as `4` or `HA`.
8. Reboot the replacement unit and wait until it finishes booting.
9. Login to the working node and verify the HA state. Select **Basic Settings > High Availability**. The **Status** field should display `HALF`.
10. Reconfigure the **Gateway IP** addresses, and the IP addresses of the **Heartbeat** and the **Next hop monitoring** interfaces. Click 

.
11. Click **Other node > Join HA**.
12. Click **Other node > Reboot**.
13. The replacement unit will reboot and start synchronizing data from the working node. The **Basic Settings > High Availability > Status** field will display `DEGRADED SYNC` until the synchronization finishes. Depending on the size of the hard disks and the amount of data stored, this can take several hours.
14. After the synchronization is finished, connect the other Ethernet cables to their respective interfaces (external to `1` or `EXT`, internal to `3` or `INT`, management to `2` or `MGMT`) as needed for your environment.

Expected result

A node of the SPS cluster is replaced with a new appliance.

Resolving an IP conflict between cluster nodes

The IP addresses of the HA interfaces connecting the two nodes are detected automatically, during boot. When a node comes online, it attempts to connect to the IP address 1.2.4.1. If no other node responds until timeout, then it sets the IP address of its HA interface to 1.2.4.1, otherwise (if there is a responding node on 1.2.4.1) it sets its own HA interface to 1.2.4.2.

Replaced nodes do not yet know the HA configuration (or any other HA settings), and will attempt to negotiate it automatically in the same way. If the network is, for any reason, too slow to connect the nodes on time, the replacement node boots with the IP address of 1.2.4.1, which can cause an IP conflict if the other node has also set its IP to that same address previously. In this case, the replacement node cannot join the HA cluster.

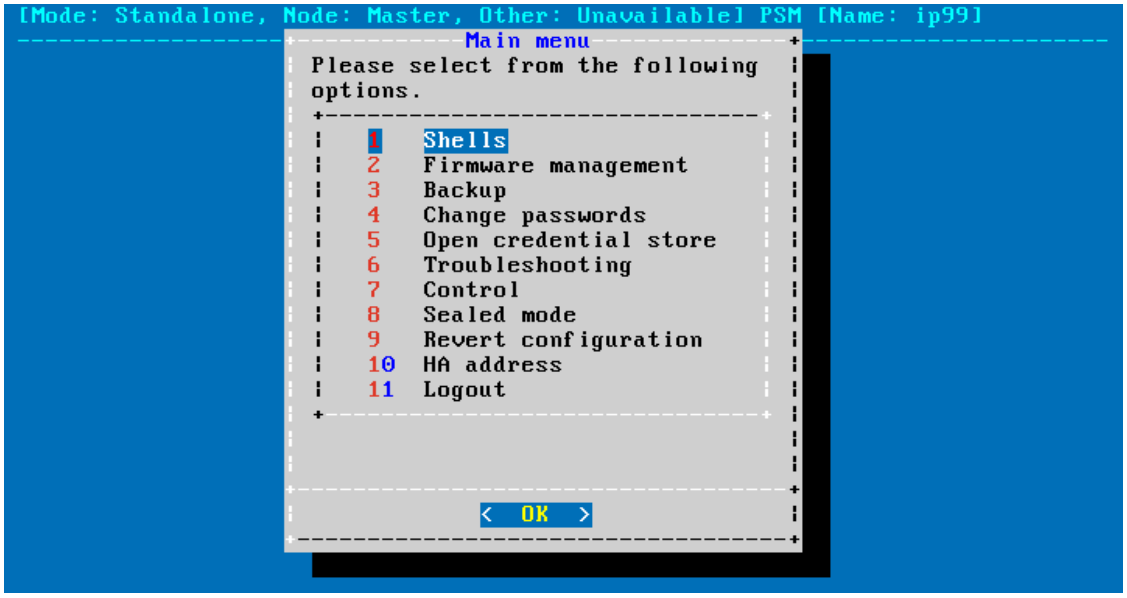
To manually assign the correct IP address to the HA interface of a node, perform the following steps:

1. Log in to the node using the IPMI or the physical console.

   Configuration changes have not been synced to the new (replacement) node, as it could not join the HA cluster. Use the default password of the root user of One Identity Safeguard for Privileged Sessions (SPS), see "Installing the SPS hardware" in the Installation Guide.

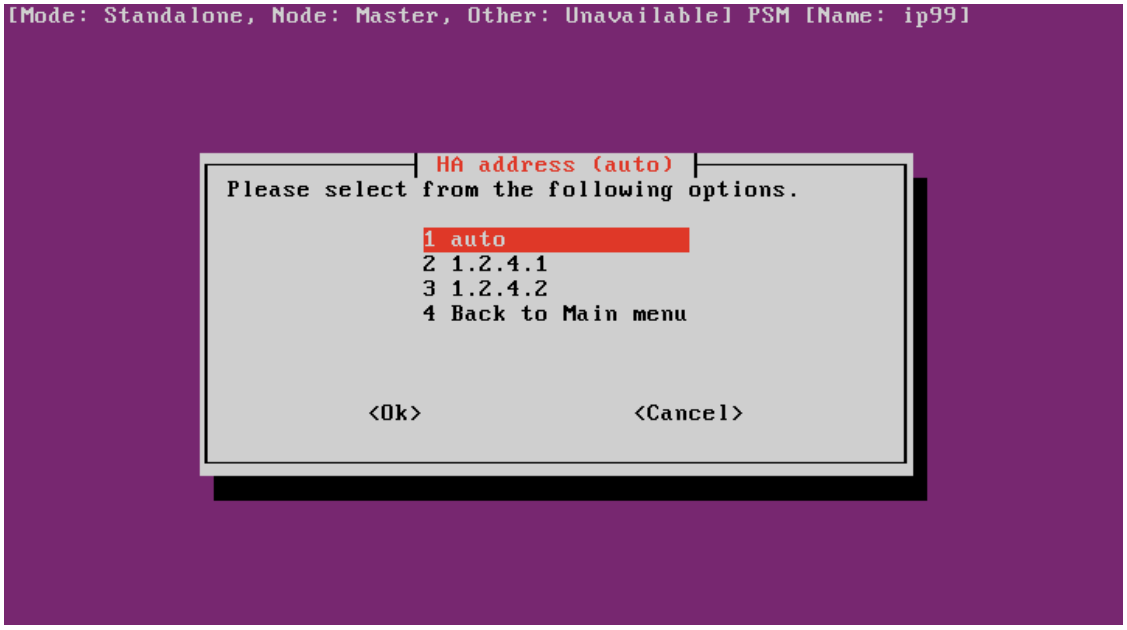
2. From the console menu, choose **10 HA address**.

   Figure 299: The console menu

3. Choose the IP address of the node.

Figure 300: The console menu



4. Reboot the node.

Understanding One Identity Safeguard for Privileged Sessions (SPS) RAID status

This section explains the possible statuses of the One Identity Safeguard for Privileged Sessions (SPS) RAID device and the underlying hard disks. SPS displays this information on the **Basic Settings > High Availability** page. The following statuses can occur:

- **Optimal**: The hard disks are working as expected.
- **Degraded**: One or more hard disk has reported an error, and might have to be replaced. For assistance, contact our Support Team.
- **Failed stripes**: One or more stripes of data failed on the RAID device. It is possible that data loss occurred, but unfortunately there is no way to find out the extent of the data loss (if any).
    - *If you have a single SPS node*: You must reinstall SPS and restore the data from the latest backup. For details, see "One Identity Safeguard for Privileged Sessions Software Installation Guide" in the Installation Guide and Restoring One Identity Safeguard for Privileged Sessions (SPS) configuration and data on page 883. If you do not have backup, contact our Support Team.
    - *If you have a high-availability SPS cluster*: Shut the node down. Do NOT disconnect its HA interface. Reinstall the node (for details, see "One Identity

Safeguard for Privileged Sessions Software Installation Guide" in the Installation Guide), power it on, then navigate to **Basic Settings > High Availability**, and click **Join HA**. For assistance, contact our Support Team.

- **Offline**: The RAID device is not functioning, probably because several disks have broken down. SPS cannot operate properly in this case. For assistance, contact our Support Team.

Restoring One Identity Safeguard for Privileged Sessions (SPS) configuration and data

The following procedure describes how to restore the configuration and data of One Identity Safeguard for Privileged Sessions (SPS) from a complete backup, for example, after a hardware replacement.

CAUTION:

Do not enable audited traffic to SPS restoring the system backup is complete.

During the restore process, the REST-based search might not function properly, since the data to search in might still be incomplete.

To restore the configuration and data of SPS from a complete backup

1. Connect to your backup server and locate the directory where SPS saves the backups. The configuration backups are stored in the `config` subdirectory in timestamped files. Find the latest configuration file (the configuration files are called `PSM-timestamp.config`).
2. Connect to SPS.

   If you have not yet completed the Welcome Wizard, click **Browse**, select the configuration file, and click **Import**.

   If you have already completed the Welcome Wizard, navigate to **Basic Settings > System > Import configuration > Browse**, select the configuration file, and click **Import**.
3. Navigate to **Policies > Backup & Archive/Cleanup**. Verify that the settings of the target servers and the backup protocols are correct.
4. Navigate to **Basic Settings > Management > System backup**, click **Restore now** and wait for the process to finish. Depending on the amount of data stored in the backup, and the speed of the connection to the backup server, this may take a long time.
5. Navigate to **SSH Control > Connections**, and click **Restore ALL**. Repeat this step for other traffic types. Depending on the amount of data stored in the backup, and the speed of the connection to the backup server, this may take a long time.

VNC is not working with TLS

Some vendors may use custom protocol elements and TLS-encryption that do not have available documentation. As a result, these cannot be audited by One Identity Safeguard for Privileged Sessions (SPS). Regardless of vendors, only the custom features described in the RFC 6143 are supported. As for encryptions, only those completely TLS-encapsulated streams can be processed where the TLS encryption process was started before the VNC protocol handshake.

Configuring the IPMI from the BIOS after losing IPMI password

It may happen that you inadvertently lose the IPMI password of your One Identity Safeguard for Privileged Sessions (SPS). The following procedure describes how you can re-configure your SPS if you lose your IPMI password.

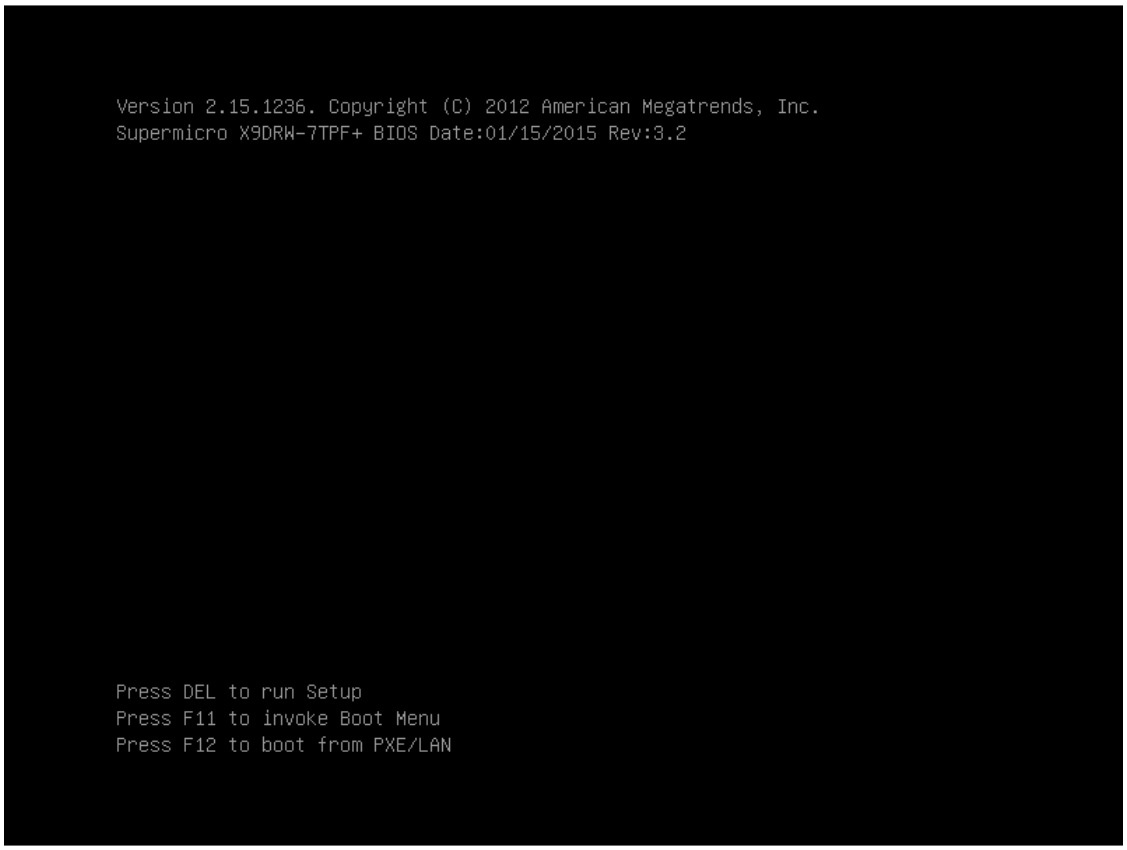
Prerequisites

To apply the procedure outlined here, you will need physical access to a monitor and keyboard.

To configure the IPMI from the BIOS after losing your IPMI password

1. Shut down SPS.
2. Unplug the SPS physical appliance's power cord.
3. Wait 30 seconds.
4. Replug the power cord.
5. Restart the appliance.
6. Press the DEL button when the POST screen comes up while the appliance is booting.

Figure 301: POST screen during booting



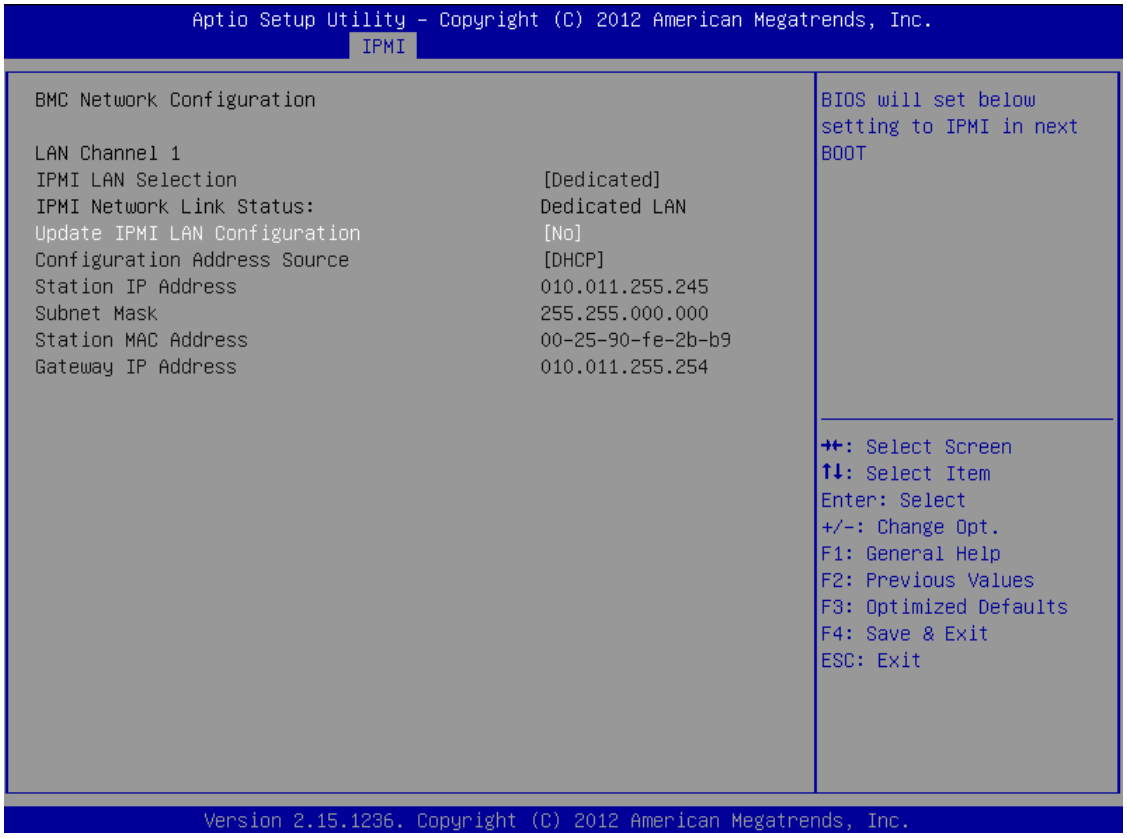
7. In the BIOS, navigate to the **IPMI** page.
8. On the **IPMI** page, select **BMC Network Configuration**, and press `Enter`.

Figure 302: IPMI page > BMC Network Configuration option



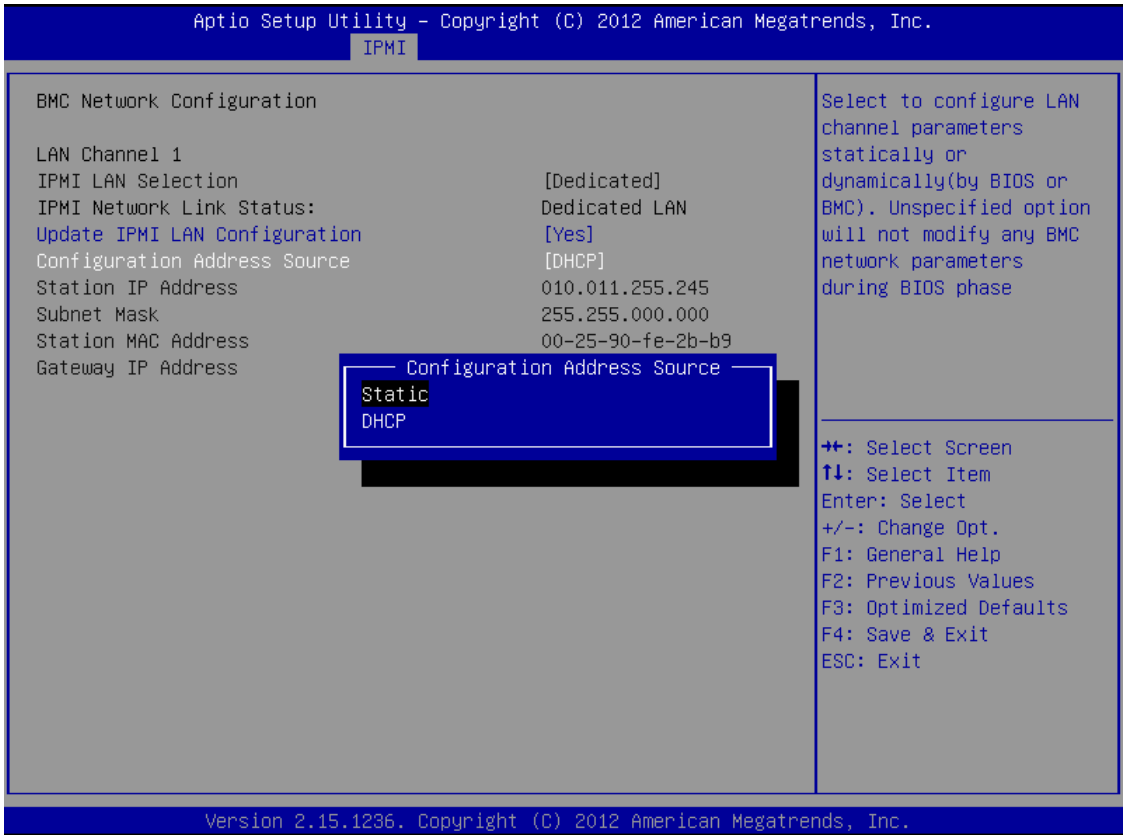
9. On the **BMC Network Configuration** page, select **Update IPMI LAN Configuration**, press `Enter`, and select **Yes**.

Figure 303: BMC Network Configuration page > Update IPMI LAN Configuration



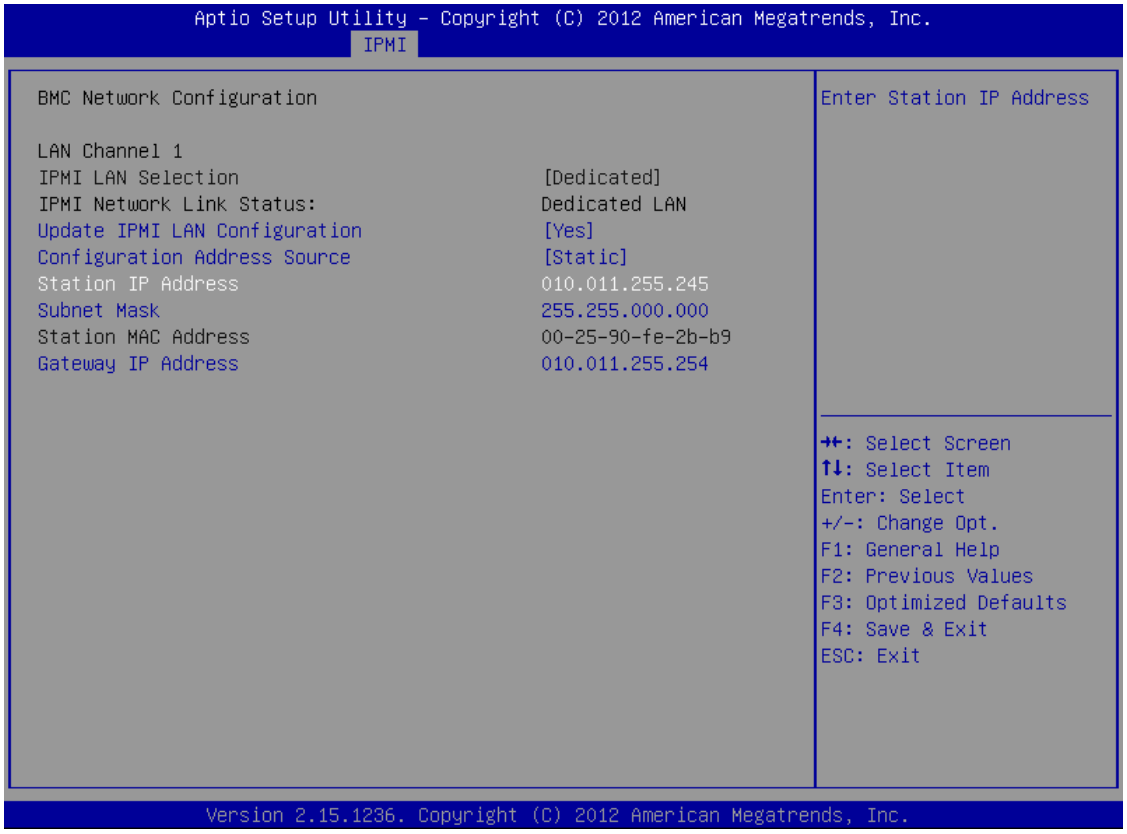
10. Stay on the **BMC Network Configuration** page, select **Configuration Address Source**, press **Enter**, and select **Static**.

Figure 304: BMC Network Configuration page > Configuration Address Source



11. Still on the **BMC Network Configuration** page, configure the **Station IP Address**, **Subnet Mask**, and **Gateway IP Address** individually.

Figure 305: BMC Network Configuration page > Station IP Address, Subnet Mask, Gateway IP Address



12. Press F4 to save the settings, and exit from the BIOS.

    About a minute later, you will be able to log in on the IPMI web interface.

Incomplete TSA response received

When using a TSA certificate generated with Windows Certificate Authority, you might see a similar error message:

```
Incomplete TSA response received, TSA HTTP server may be responding slowly;
errno='Success (0)', timeout_seconds='30'
```

When generating the certificate, make sure that you do the following:

Optional Key Usage: If **Key Usage** is present, it must be `digitalSignature` and/or `nonRepudiation`. Other values are not permitted. Make sure that in **Encryption**, **Allow key exchange without key encryption (key agreement)** is selected.

CAUTION:

In Encryption, do NOT select Allow key exchange only with key encryption (key encipherment), because it will result in errors.

For details, see Generating TSA certificate with Windows Certificate Authority on Windows Server 2008 on page 414 or Generating TSA certificate with Windows Certificate Authority on Windows Server 2012 on page 419.

Using UPN usernames in audited SSH connections

When you specify user names in a User Principal Name (UPN) format (e-mail address as username) for an SPS-audited SSH connection, the connection is unsuccessful.

The connection is unsuccessful because SPS uses the '@' character in the username as inband destination selection. If this happens, the username is stripped from the domain part and the UPN suffix is interpreted as inband target. For example, if using test@ema.il as username, the username for the connection will be 'test' and the inband destination is 'ema.il'. SPS interprets the last two '@' characters from the connection string, for example, username@my-inband-target@SPS.

To avoid this, you must use inband destination selection. By specifying the target host explicitly, you can prevent SPS to misinterpret the '@' character from UPN usernames.

- For more information, see the How to use UPN usernames in audited SSH connections Knowledge Base article.
- For more information about inband destination selection, see Using inband destination selection in SSH connections.

25

Using SPS with SPP

You can join your One Identity Safeguard for Privileged Sessions (SPS) deployment to your One Identity Safeguard for Privileged Passwords (SPP) deployment. That way you can jointly use the features of the two deployments.

Both appliances provide different functionality. You can use them together or independently from each other.

SPP provides:

- Machine and account discovery
- Password rotation and management
- Advanced access request and approval workflows
- A user portal and desktop application to initiate connections

SPS provides:

- Transparent or non-transparent interception of remote admin protocols (SSH, RDP, Telnet, Citrix ICA, and VNC)
- Audit recording and video-like playback of sessions
- Inband authentication of the monitored users independently from the target servers
- Basic access control policy enforcement
- Advanced search and reporting capabilities in the audit records
- Built-in user behavior analytics for the recorded sessions (One Identity Safeguard for Privileged Analytics)

Passwords-initiated (SPP-initiated) workflow

In the Passwords-initiated workflow, the users initiate sessions from SPP. In this workflow SPP uses SPS as a session-recording device.

You can use your browser or the One Identity Safeguard desktop client to request access from SPP and initiate the connection to the target server via SPS. SPP creates an access string for the user's SSH or RDP client that allows these clients to connect to the target server via SPS, so SPS can audit and record the session. In this sense this workflow is nontransparent, the user must use a browser or the One Identity Safeguard desktop client.

This is what all SPS users who bought the Sessions Module use before SPP version 2.7.

For details on configuring this workflow, see Configuring SPP for Passwords-initiated workflow.

Sessions-initiated (SPS-initiated) workflow

In the Sessions-initiated workflow, the users initiate sessions from SPS. In this workflow SPS uses SPP as a credential store.

This workflow is transparent in the sense that you can connect to the target server or to SPS directly using your SSH or RDP client application. SPS authenticates these clients and communicates with SPP to get the password for the target server. It then uses that password to open the connection. Authentication happens on SPS, while authorization happens on SPP based on the user's entitlements.

This is what old and new users of standalone SPS are likely to prefer.

The usual SPP Access Requests workflows that SPP provides are supported:

- Auto-approved access request
- Approved/denied access request (similar to the four-eyes authorization feature of SPS)

Configuring the Passwords-initiated workflow

Passwords-initiated (SPP-initiated) workflow

In the Passwords-initiated workflow, the users initiate sessions from SPP. In this workflow SPP uses SPS as a session-recording device.

You can use your browser or the One Identity Safeguard desktop client to request access from SPP and initiate the connection to the target server via SPS. SPP creates an access string for the user’s SSH or RDP client that allows these clients to connect to the target server via SPS, so SPS can audit and record the session. In this sense this workflow is nontransparent, the user must use a browser or the One Identity Safeguard desktop client.

This is what all SPS users who bought the Sessions Module use before SPP version 2.7.

For details on configuring this workflow, see Configuring SPP for Passwords-initiated workflow.

Prerequisites

- Minimum versions:
  - SPP version 2.7
  - SPS version 6.0
- You must have built an SPS cluster by promoting an SPS node to the role of the

Central Management node, even if it is a single node. For more information, see Building a cluster.

To configure the Passwords-initiated (SPP-initiated) workflow

1. On SPS, join SPP and SPS as described in Joining SPS to SPP.
2. Configure SPP to use the joined SPS as described in Configuring SPP for Passwords-initiated workflow.
3. Optionally, customize monitoring settings as follows:
    - To make use of the more advanced features of SPS, you can change the **safeguard_default** Connection Policy or create a new Connection Policy and select that in SPP.
    - The critical setting is the AA plugin – make sure you use the same one as what the auto-generated one uses.

Configuring SPP for Passwords-initiated workflow

To configure SPP to use the joined SPS in Passwords-initiated (SPP-initiated) workflows, complete the following steps. For details on the workflow, see Using SPS with SPP.

Prerequisites

- Minimum SPP version 2.7
- You have joined SPP and SPS as described in Joining SPS to SPP.

To configure SPP for Passwords-initiated workflow

1. On SPP, assign the managed networks for sessions management.
    a. Navigate to **Administrative Tools > Settings > Cluster > Managed Networks**.
    b. Add the network you want to monitor with SPS and choose the SPS appliance for the **Sessions Managed By** field.
2. Select the SPS for the access request policy.
    a. Navigate to **Administrative Tools > Entitlements > Access Request Policies > (create or edit a policy)**.
    b. On the **Session Settings** tab of the selected policy, select the SPS Connection Policy. The IP address of the cluster master is displayed first followed by the SPS Connection Policy name (**safeguard_default** by default).

Configuring the Sessions-initiated workflow

Sessions-initiated (SPS-initiated) workflow

In the Sessions-initiated workflow, the users initiate sessions from SPS. In this workflow SPS uses SPP as a credential store.

This workflow is transparent in the sense that you can connect to the target server or to SPS directly using your SSH or RDP client application. SPS authenticates these clients and communicates with SPP to get the password for the target server. It then uses that password to open the connection. Authentication happens on SPS, while authorization happens on SPP based on the user's entitlements.

This is what old and new users of standalone SPS are likely to prefer.

The usual SPP Access Requests workflows that SPP provides are supported:

- Auto-approved access request
- Approved/denied access request (similar to the four-eyes authorization feature of SPS)

To configure the Sessions-initiated (SPS-initiated) workflow

1. On SPS, link SPP and SPS as described in Joining SPS to SPP.
2. Configure SPP to allow SPS to request passwords from SPP as described in Configuring SPP for Sessions-initiated workflow.
3. Configure SPS to use the linked SPP as a Credential Store as described in Configuring SPS for Sessions-initiated workflow.
4. Optionally, customize monitoring settings as follows:

Configuring SPP for Sessions-initiated workflow

To configure SPP to use the linked SPS in Sessions-initiated (SPS-initiated) workflows, complete the following steps. For details on the workflow, see Using SPS with SPP.

Prerequisites

- Minimum versions:
  - SPP version 2.8
  - SPS version 6.2 and newer, including 6.0.2 and newer versions of the 6.0.x branch, but excluding 6.1.x
- You must have built an SPS cluster by promoting an SPS node to the role of the Central Management node, even if it is a single node. For more information, see Building a cluster.

CAUTION: When linking your One Identity Safeguard for Privileged Sessions (SPS) deployment to your One Identity Safeguard for Privileged Passwords (SPP) deployment, ensure that the SPS and SPP versions match exactly, and you keep the versions synchronized during an upgrade. For example, you can only link SPS version 6.6 to SPP version 6.6, and if you upgrade SPS to version 6.7, you must also upgrade SPP to 6.7.

Make sure that you do not mix Long Term Supported (LTS) and feature releases. For example, do not link an SPS version 6.0 to an SPP version 6.1.

To configure SPP for Sessions-initiated workflow

1. On SPP, navigate to **Administrative Tools > Settings > Session Management**.
2. Click **Session Module Password Access Enabled** to enable this feature. A dialog opens.
3. Type `Enable`.
4. Ensure that SPP can access the usernames that will authenticate on the SPS gateway either from a local user database, or from an Active Directory.
5. Ensure that these users have the proper Entitlements to access the passwords to their target servers. Otherwise, SPS will reject their sessions.

Configuring SPS for Sessions-initiated workflow

To configure SPS to use the linked SPP in Sessions-initiated (SPS-initiated) workflows, complete the following steps. For details on the workflow, see Using SPS with SPP.

Prerequisites

- Minimum versions:
  - SPP version 2.8
  - SPS version 6.2 and newer, including 6.0.2 and newer versions of the 6.0.x branch, but excluding 6.1.x
- You must have built an SPS cluster by promoting an SPS node to the role of the Central Management node, even if it is a single node. For more information, see Building a cluster.

CAUTION: When linking your One Identity Safeguard for Privileged Sessions (SPS) deployment to your One Identity Safeguard for Privileged Passwords (SPP) deployment, ensure that the SPS and SPP versions match exactly, and you keep the versions synchronized during an upgrade. For example, you can only link SPS version 6.6 to SPP version 6.6, and if you upgrade SPS to version 6.7, you must also upgrade SPP to 6.7.

Make sure that you do not mix Long Term Supported (LTS) and feature releases. For example, do not link an SPS version 6.0 to an SPP version 6.1.

Limitations

- Only SSH and RDP sessions are supported.
- Users must perform gateway authentication on SPS with the same username they have Entitlements for in SPP.
  - For SSH sessions, the gateway authentication can use a Local User Database, an LDAP server, or an Active Directory server as authentication backend.

    Note that SPP does not support every type of LDAP and Active Directory settings that SPS does. Verify that you can configure both appliances to access and retrieve data from the LDAP or Active Directory server.
  - For RDP sessions, SPS must be configured to act as a Remote Desktop Gateway. For details, see Using One Identity Safeguard for Privileged Sessions (SPS) as a Remote Desktop Gateway.

    The gateway authentication can use a Local User Database or an Active Directory server as authentication backend. When using an Active Directory server, note the following points.
    - Both SPS and SPP must use the same server, and be the member of the same domain as the Active Directory server.
    - SPP does not support every type of Active Directory settings that SPS does. Verify that you can configure both appliances to access and retrieve data from the Active Directory server.

- SPS does not receive the domain of the authenticated user from the Domain Controller. SPS assumes that the user belongs to the same domain that SPS is joined into. (Configuring trust between the Domain Controller of SPS and the Domain Controller of the user does not solve this problem.)

- You must use a uniform address for the target server. Use either its IPv4 address or its hostname everywhere: when configuring the Assets in SPP, the Connection Policies and Channel Policies in SPS, and also when the user sets the target address in the SSH/RDP application. Otherwise, the authentication will fail.

To configure SPS for Sessions-initiated workflow

Configure Connection Policies on SPS to audit your sessions. Note that you have to complete these steps for each Connection Policy that uses SPP as a Credential Store. For the general steps on configuring Connection Policies, see Configuring connections.

1. Select **Credential Store > safeguard_default**
2. Select **AA plugin > safeguard_default**
3. Select **Usermapping policy > safeguard_default**
4. Configure gateway authentication. The users must perform gateway authentication on SPS with the same username they have Entitlements for in SPP. For details, see Configuring gateway authentication.
5. When you are using an Approve/deny workflow on SPP, increase the **Idle timeout** setting of the Connection Policy. SPS will wait for an approval from SPP until half the time set in **Idle timeout**. For example, if the authorizer on SPP has 2 minutes to approve the access request, set the **Idle timeout** option on SPS to more than double this value, for example, 5 minutes.

Joining SPS to SPP

You can join your One Identity Safeguard for Privileged Sessions (SPS) deployment to your One Identity Safeguard for Privileged Passwords (SPP) deployment.

> IMPORTANT: Joining your SPS and SPP appliances is an action that you cannot undo.
>
> If the primary IP address of your SPS or SPP changes, you must repeat this procedure to rejoin the clusters.

Prerequisites

- Your SPS deployment must be an SPS cluster (not a high-availability cluster, but a Central Management cluster). Even if your SPS deployment consists of a single, standalone node, you must convert it to the Central Management node of its own single-node cluster. For details, see Managing Safeguard for Privileged Sessions

(SPS) clusters on page 346.

Configuration synchronization must be enabled between the nodes of the SPS cluster. This is required so SPP entitlements work properly for each SPS node.

> NOTE:
>
> If you have multiple standalone SPS appliances, consider joining them to a cluster before joining SPP. In general, One Identity recommends creating a cluster if the nodes can use a common configuration, or later you might want to centrally search the data of every node. Creating a cluster from the SPS nodes after joining SPP is problematic and should be avoided.

- You will need the primary IP address or the hostname of your SPP deployment that SPS can use to access SPP. Only IPv4 addresses are supported.
- You will need the username and password to an SPP account that has "Appliance" and "Operations" permissions.
- Verify that your SPS policies do not contain the `safeguard_default` string in their names. During the join process, SPS automatically creates and configures several policies and plugins. The name of these policies usually contains the string `safeguard_default`. Existing policies with such names will be overwritten.
- The SPP and SPS nodes must be able to communicate on the TCP `8649` port. If needed, update your firewall policies.

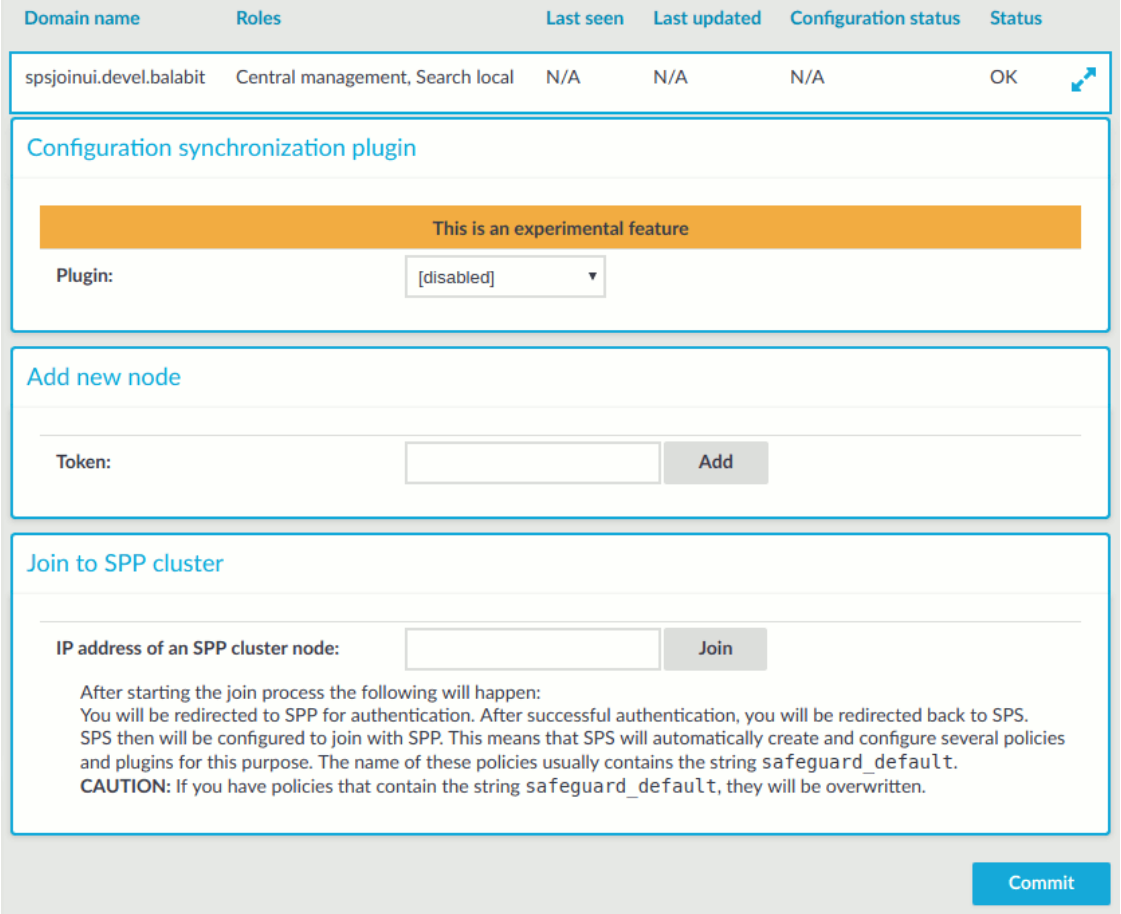
  > NOTE: When updating your firewall policies to enable the connection between SPS and SPP nodes, consider the following:
  >
  > - **Connecting SPS nodes to SPS nodes**: Make sure that between all the SPS nodes, the `500` and `4500` UDP ports are opened bidirectionally. This is required so that every node can initiate and accept connections from every other node through the mentioned ports.
  > - **Connecting SPP nodes to SPP nodes**: For more information, see *Enrolling replicas into a cluster* in the One Identity Safeguard for Privileged Passwords Administration Guide.
  > - **Connecting SPS nodes to SPP nodes**: Make sure that between all the SPS and SPP nodes, the `8649` TCP port is opened bidirectionally. This is required so that every node can initiate and accept connections from every other node through the mentioned port. If there are `M` SPP and `N` SPS nodes in your setup, then create `M × N × 2` firewall rules to link the SPS nodes to the SPP nodes.
- During the join process, SPS must be able to access SPP using HTTPS on the tcp 443 port. This is required only once during the join process. If needed, update your firewall policies.

To join your SPS deployment to SPP

1. (Optional) Create a configuration backup of SPS. For details, see Creating configuration backups on page 157.
2. (Optional) Create a configuration backup of SPP. For details, see the Safeguard for Privileged Passwords Administration Guide, Backup and Retention settings.

3. Login to the Central Management node of your SPS cluster. This node has **Central Management** listed in the **Basic Settings > Cluster management > Roles** field.

Figure 306: Basic Settings > Cluster management — Joining SPS to SPP



4. Navigate to **Basic Settings > Cluster management > Join to SPP cluster** and enter the primary IP address of SPP into the **IPv4 address or hostname of SPP to join** field. Only IPv4 addresses are supported.
5. Click **Join**. Wait until you are redirected to SPP.
6. Login to SPP. Wait until you are redirected to SPS.
7. Wait until SPS creates and configures the policies and plugins required for the joint operation of SPS and SPP. This step can usually take up to a minute.
8. You will receive a message:
    - If the join is unsuccessful, this message displays: `Request failed`. Check the credentials and the IP address you provided. For details on resolving errors, see Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues on page 907 and Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join

error resolution on page 904

- If the join is successful, this message displays: `SPS successfully joined to SPP`. SPP automatically closes any open access requests.

9. Log out from the SPS web interface.

CAUTION:

If the primary IP address of your SPS or SPP changes, you must repeat this procedure to rejoin the clusters.

Troubleshooting the SPS to SPP join

- Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join error resolution
- Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues

Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join error resolution

Common join error resolutions follow which may occur when joining SPS (SPP) to One Identity Safeguard for Privileged Sessions (SPS).

Typo in SPP's address, the address is not reachable

- Error: The browser reports errors when SPS redirects to SPP's login page, for example, `This site can’t be reached`. The exact error message depends on the browser.
- Resolution: Click the Back button of the browser and enter the correct address into the **Pv4 address or hostname of SPP to join** field.

Typo in SPP's address, the address is alive, but not an SPP

- Error: After clicking the Join button, a web site other than SPP's Login interface is displayed.

- Resolution: Click the Back button of the browser and enter the correct address into the **Pv4 address or hostname of SPP to join** field.

SPP's HTTPS certificate does not match its IP address or hostnam

- Error message: `Error joining to SPP: SPS has failed to join to SPP. For more information, see the error details. (JoinFailed)`
- Raw error:

```
{
    "response": "Error sending request: SSLError: HTTPSConnectionPool
(host='examplespp.company', port=443): Max retries exceeded with url:
/service/core/v3/Cluster/SessionModules (Caused by SSLError
(CertificateError(\"hostname 'examplespp.company' doesn't match
'192.0.2.123'\",),))",
    "status": null,
    "url":
"https://examplespp.company/service/core/v3/Cluster/SessionModules"
 }
```

- Resolution:
  - If SPP's certificate contains SPP's IPv4 address in the **Common Name** or **subjectAltName** field, then enter that IP address when joining SPS to SPP.
  - If SPP's certificate contains only its DNS name in the **Common Name** or **subjectAltName** field, then use that hostname when joining SPS to SPP.
  - Otherwise, set up an SSL server certificate for SPP which matches its IP address in the certificate's **Common Name** or **subjectAltName**fields (see SSL Certificates in the Safeguard Administration Guide) and retry joining. Wait about five minutes to let the timeout of the failed join request expire before starting a new join request after a failed incomplete one. (Alternatively, see Reversing the SPP to SPS join in the Safeguard Administration Guide.)

Typo in SPP credentials

- Error: Login to the SPP web interface fails.
- Raw error:

```
{
            "error": "invalid_request",
            "error_description": "Access denied.",
            "success": false
      }
```

- Resolution: Make sure that the correct username and password are entered in the **SPP username**: and **SPP password:** prompts.

SPP user has insufficient permission

- Error message: `Error joining to SPP: SPS has failed to join to SPP. For more information, see the error details. (JoinFailed)`

```
Authorization is required for this request.

      Code: 60108
      URL: https://192.0.2.123/service/core/v3/Cluster/SessionModules
      Status: 403
```

- Resolution: When SPS redirects to SPP's Login interface, then login as an SPP user has "Appliance" and "Operations" permissions.

SPS is already joined to SPP

- Error message: `Error joining to SPP: SPS has failed to join to SPP. For more information, see the error details. (JoinFailed)`

```
The session connection has a missing, invalid, or non-unique
value [ NodeId ].

      Code: 60657
      URL: https://192.0.2.123/service/core/v3/Cluster/SessionModules
      Status: 400
```

- Resolution: See Reversing the SPP to SPS join in the Safeguard Administration Guide.

Joining takes too long (more than five minutes)

- `ERROR: Request to https://192.0.2.123/service/a2a/v2/PsmValidation failed, response (HTTP 403):`

```
      {
            "Code": 60108,
            "Message": "Authorization is required for this request.",
            "InnerError": null
      }
```

- Resolution: Make sure that SPS is not overloaded and try joining again.

Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues

In addition to the monitoring tools in SPP, you can use the Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues on page 907 during the join process. Several SPS tools are described below.

Join process fails and real-time monitoring

If the join process fails for any reason, consult the system logs.

To view the Safeguard for Privileged Sessions logs, navigate to **Basic Settings** | **Troubleshooting** | **View log files**.

To show only the logs for the join process:

1. Select a **Logtype** of **syslog**.
2. Select the **Day**; today is the default.
3. In the **Show only messages containing** text box, enter `SPP-join`.

Use the buttons at the bottom of the dialog to perform the following tasks:

- To download the log file, click **Download**.
- To follow the current log messages real-time, click **Tail**. The latest logs will update in a browser window while you interact with the join process.
- To display the log messages, click **View**.

To increase the level of detail in the log, enable debug level logging at **Basic Settings** | **Management** | **Debug logging** | **Enable debug logs**.

Join successful but connections do not work

When SPP and SPS report a successful join, but the connections don't work, view the SPS connection logs.

In Safeguard for Privileged Sessions, navigate to **Basic Settings** | **Troubleshooting** | **View log files**.

To show only the logs for the join process:

1. Select a **Logtype** of **ssh** or **rdp**.
2. Select the **Day** (today is the default).
3. In the **Show only messages containing** text box, enter `SPP-join`.

To change the verbosity level of SPS, complete the following steps in Safeguard for Privileged Sessions:

1. Navigate to the **Global Options** page of the traffic for which you want to change the log level. For example, go to **SSH Control** | **Global Options** to change the log level of SSH traffic, **RDP Control** | **Global Options** for remote desktop traffic, and so on.
2. Select the desired log level from the **Verbosity level** field. The verbosity level ranges from 1 (no logging) to 10 (extremely detailed), with level 4 being the default normal level.

CAUTION: High verbosity levels generate a very large amount of log messages and might result in a very high load on the machine. Log levels set around 9 to 10, may result in logs with highly sensitive data, for example, passwords in plain text format.

Testing network issues

You can use the Diagnostics tools of SPP and SPS to test network issues. The following commands are available:

- **ping**: Sends a simple message to the specified host to test network connectivity.
- **traceroute**: Sends a simple message from SPS to the specified host and displays all hosts on the path of the message. It is used to trace the path the message travels between the hosts.
- **connect**: Attempts to connect the specified host using the specified port. It is used to test the availability or status of an application on the target host.

To execute one of the above commands on SPS, see Network troubleshooting on page 854.

To execute one of the above commands on SPP, see Diagnostics tools of SPP.

Creating an SPS Support Bundle

If you have an issue which needs Support assistance, you may be asked to provide an SPS Support Bundle. To collect system-state information (also known as a debug bundle) in One Identity Safeguard for Privileged Sessions, see Collecting logs and system information for error reporting on page 860.

Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join error resolution

Common join error resolutions follow which may occur when joining SPS (SPP) to One Identity Safeguard for Privileged Sessions (SPS).

Typo in SPP's address, the address is not reachable

- Error: The browser reports errors when SPS redirects to SPP's login page, for example, `This site can’t be reached`. The exact error message depends on the browser.
- Resolution: Click the Back button of the browser and enter the correct address into the **Pv4 address or hostname of SPP to join** field.

Typo in SPP's address, the address is alive, but not an SPP

- Error: After clicking the Join button, a web site other than SPP's Login interface is displayed.
- Resolution: Click the Back button of the browser and enter the correct address into the **Pv4 address or hostname of SPP to join** field.

SPP's HTTPS certificate does not match its IP address or hostnam

- Error message: `Error joining to SPP: SPS has failed to join to SPP. For more information, see the error details. (JoinFailed)`
- Raw error:

```
{
    "response": "Error sending request: SSLError: HTTPSConnectionPool
(host='examplespp.company', port=443): Max retries exceeded with url:
/service/core/v3/Cluster/SessionModules (Caused by SSLError
(CertificateError(\"hostname 'examplespp.company' doesn't match
'192.0.2.123'\",),))",
    "status": null,
    "url":
"https://examplespp.company/service/core/v3/Cluster/SessionModules"
 }
```

- Resolution:
  - If SPP's certificate contains SPP's IPv4 address in the **Common Name** or **subjectAltName** field, then enter that IP address when joining SPS to SPP.
  - If SPP's certificate contains only its DNS name in the **Common Name** or **subjectAltName** field, then use that hostname when joining SPS to SPP.
  - Otherwise, set up an SSL server certificate for SPP which matches its IP address in the certificate's **Common Name** or **subjectAltName**fields (see SSL Certificates in the Safeguard Administration Guide) and retry joining. Wait about five minutes to let the timeout of the failed join request expire before starting a new join request after a failed incomplete one. (Alternatively, see Reversing the SPP to SPS join in the Safeguard Administration Guide.)

Typo in SPP credentials

- Error: Login to the SPP web interface fails.
- Raw error:

```
{
            "error":  "invalid_request",
            "error_description":  "Access  denied.",
            "success":  false
      }
```

- Resolution: Make sure that the correct username and password are entered in the **SPP username**: and **SPP password:** prompts.

SPP user has insufficient permission

- Error message: `Error joining to SPP: SPS has failed to join to SPP. For more information, see the error details. (JoinFailed)`

```
 Authorization is required for this request.

       Code: 60108
       URL: https://192.0.2.123/service/core/v3/Cluster/SessionModules
       Status: 403
```

- Resolution: When SPS redirects to SPP's Login interface, then login as an SPP user has "Appliance" and "Operations" permissions.

SPS is already joined to SPP

- Error message: `Error joining to SPP: SPS has failed to join to SPP. For more information, see the error details. (JoinFailed)`

```
The session connection has a missing, invalid, or non-unique
value [ NodeId ].

       Code: 60657
       URL: https://192.0.2.123/service/core/v3/Cluster/SessionModules
       Status: 400
```

- Resolution: See Reversing the SPP to SPS join in the Safeguard Administration Guide.

Joining takes too long (more than five minutes)

- `ERROR: Request to https://192.0.2.123/service/a2a/v2/PsmValidation failed, response (HTTP 403):`

```
{
      "Code": 60108,
      "Message": "Authorization is required for this request.",
      "InnerError": null
}
```

- Resolution: Make sure that SPS is not overloaded and try joining again.

Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues

In addition to the monitoring tools in SPP, you can use the Safeguard for Privileged Passwords (SPP) to One Identity Safeguard for Privileged Sessions (SPS) join issues on page 907 during the join process. Several SPS tools are described below.

Join process fails and real-time monitoring

If the join process fails for any reason, consult the system logs.

To view the Safeguard for Privileged Sessions logs, navigate to **Basic Settings** | **Troubleshooting** | **View log files**.

To show only the logs for the join process:

1. Select a **Logtype** of **syslog**.
2. Select the **Day**; today is the default.
3. In the **Show only messages containing** text box, enter `SPP-join`.

Use the buttons at the bottom of the dialog to perform the following tasks:

- To download the log file, click **Download**.
- To follow the current log messages real-time, click **Tail**. The latest logs will update in a browser window while you interact with the join process.
- To display the log messages, click **View**.

To increase the level of detail in the log, enable debug level logging at **Basic Settings** | **Management** | **Debug logging** | **Enable debug logs**.

Join successful but connections do not work

When SPP and SPS report a successful join, but the connections don't work, view the SPS connection logs.

In Safeguard for Privileged Sessions, navigate to **Basic Settings** | **Troubleshooting** | **View log files**.

To show only the logs for the join process:

1. Select a **Logtype** of **ssh** or **rdp**.
2. Select the **Day** (today is the default).
3. In the **Show only messages containing** text box, enter `SPP-join`.

To change the verbosity level of SPS, complete the following steps in Safeguard for Privileged Sessions:

1. Navigate to the **Global Options** page of the traffic for which you want to change the log level. For example, go to **SSH Control** | **Global Options** to change the log level of SSH traffic, **RDP Control** | **Global Options** for remote desktop traffic, and so on.
2. Select the desired log level from the **Verbosity level** field. The verbosity level ranges from 1 (no logging) to 10 (extremely detailed), with level 4 being the default normal level.

CAUTION: High verbosity levels generate a very large amount of log messages and might result in a very high load on the machine. Log levels set around 9 to 10, may result in logs with highly sensitive data, for example, passwords in plain text format.

Testing network issues

You can use the Diagnostics tools of SPP and SPS to test network issues. The following commands are available:

- **ping**: Sends a simple message to the specified host to test network connectivity.
- **traceroute**: Sends a simple message from SPS to the specified host and displays all hosts on the path of the message. It is used to trace the path the message travels between the hosts.
- **connect**: Attempts to connect the specified host using the specified port. It is used to test the availability or status of an application on the target host.

To execute one of the above commands on SPS, see Network troubleshooting on page 854.

To execute one of the above commands on SPP, see Diagnostics tools of SPP.

Creating an SPS Support Bundle

If you have an issue which needs Support assistance, you may be asked to provide an SPS Support Bundle. To collect system-state information (also known as a debug bundle) in One Identity Safeguard for Privileged Sessions, see Collecting logs and system information for error reporting on page 860.

Configuring external devices

This section describes scenarios about configuring external devices to redirect selected traffic to One Identity Safeguard for Privileged Sessions (SPS).

Configuring advanced routing on Linux

Configuring advanced routing on Cisco routers

Configuring advanced routing on Sophos UTM (formerly Astaro Security Gateway) firewalls

Configuring advanced routing on Linux

The following describes how to configure a Linux-based router to redirect selected traffic to One Identity Safeguard for Privileged Sessions (SPS) instead of its original destination. This procedure should work on most modern Linux-based routers, including Check Point firewalls.

Prerequisites

The router must have the `iptables` and `ip` tools installed.

To configure a Linux-based router to redirect selected traffic to SPS instead of its original destination

1. Create the packet filter rules that will mark the connections to be sent to SPS using the CONNMARK feature of iptables. Mark only those connections that must be redirected to SPS.

```
# iptables -t mangle -I PREROUTING -i <interface-facing-the-clients> -p tcp -d <network-of-the-servers> --dport <port-to-access> -j CONNMARK --set-mark 1
```

Example: Setting up a connection mark for Linux policy routing

For example, if the network interface of the router that faces the clients is called `eth0`, the servers are located in the `10.0.0.0/24` subnet, and the clients access the servers using port `3389` (the default port of the RDP protocol), then this command looks like:

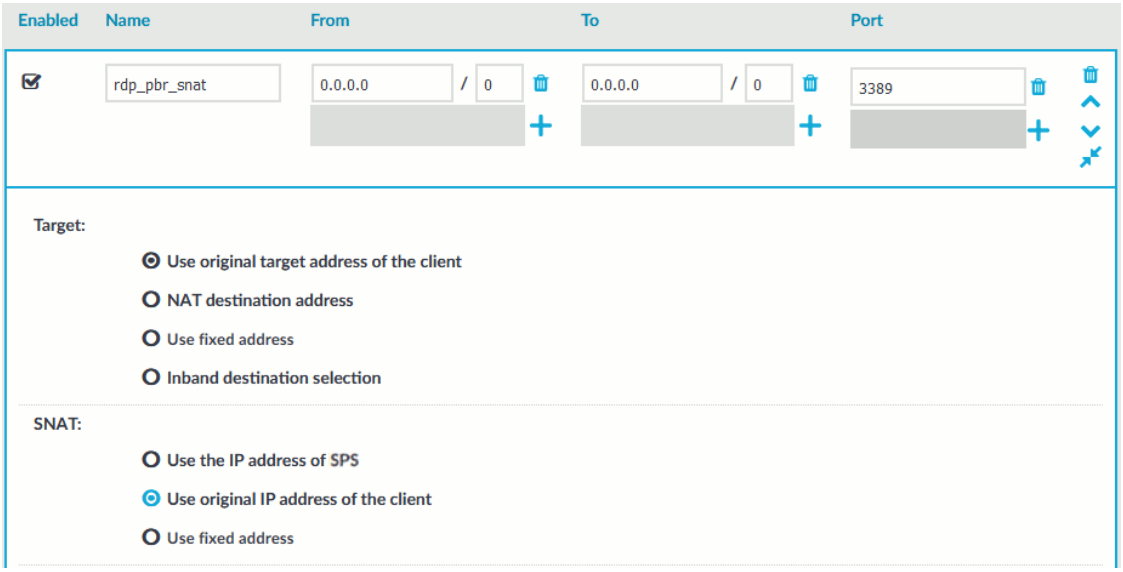
```
# iptables -t mangle -I PREROUTING -i eth0 -p tcp -d 10.0.0.0/24 --
dport 3389 -j CONNMARK --set-mark 1
```

2. Create a rule that redirects the answers of the servers to SPS. That way both the client-to-server and the server-to-client traffic is routed to SPS.

   NOTE:

   This step is only required if you want to use Source NAT (IP Spoofing) instead of SPS's address towards the monitored servers.

Figure 307: Control > Connections — Using SNAT




```
# iptables -t mangle -I PREROUTING -i <interface-facing-the-servers> -
p tcp -s <network-of-the-servers> --sport <port-to-access> -j CONNMARK
--set-mark 1
```

3. Convert the CONNMARK marks to MARK:

```
# iptables -t mangle -A PREROUTING ! -i <interface-facing-the-scb> -m
connmark --mark 1 -j MARK --set-mark 1
```

CAUTION:

This rule must be placed after the CONNMARK rules.

4. Add the table name to the `/etc/iproute2/rt_tables` of the router. Use the following format (for details on routing tables, see for example the Guide to IP Layer Network Administration with Linux):

```
103 scb
```

5. Create a routing table that has a single entry with a default route to SPS:

```
# /sbin/ip route add default via <ip-address-of-SPS> table scb
```

6. Create a routing rule that selects the routing table called `scb`, if the connection is marked.

```
# /sbin/ip rule add from all fwmark 1 table scb
```

7. If SPS is configured to spoof the IP address of the clients on the server side (that is, the **SNAT > Use original IP address of the client** option of the connection policies is selected), enable spoofing on the router for the interface connected to SPS.

```
# echo 0 > /proc/sys/net/ipv4/conf/<interface-facing-SPS>/rp_filter
# echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

Expected result:

The traffic from the clients targeting the specified port of the servers is redirected to SPS. Therefore, SPS can be configured to control and audit this traffic.

Configuring advanced routing on Cisco routers

The following describes how to configure a Cisco router to redirect selected traffic to One Identity Safeguard for Privileged Sessions (SPS) instead of its original destination. This procedure should work on most modern Cisco IOS releases but was specifically tested on IOS version 12.3.

To configure a Cisco router to redirect selected traffic to SPS instead of its original destination

1. Create an ACL (Access Control List) entry that matches the client and server subnets and the to-be-audited port. Keep in mind that whatever is permitted by this ACL is what will be matched, so make sure that the scope of the ACL entry is narrowed down as much as possible.

   ```
   #(config) ip access-list extended ssh-inbound
   #(config-ext-nacl) permit tcp <src net> <src mask> <dst net> <dst mask>
   eq <dst port>
   ```

   Example: Configuring an ACL entry for Cisco policy routing

   For example, if the clients are in the `192.168.0.0/24` subnet, the servers are located in the `10.0.0.0/24` subnet, and the clients access the servers using port `22` (the default port of the SSH protocol), then the permit clause should be:

   ```
   #(config-ext-nacl) permit tcp 192.168.0.0 0.0.0.255 10.0.0.0
   0.0.0.255 eq 22
   ```

   TIP:

   Cisco ACLs use inverse netmasks for defining network addresses. To calculate an inverse mask given a subnet mask, simply subtract each octet value from 255.

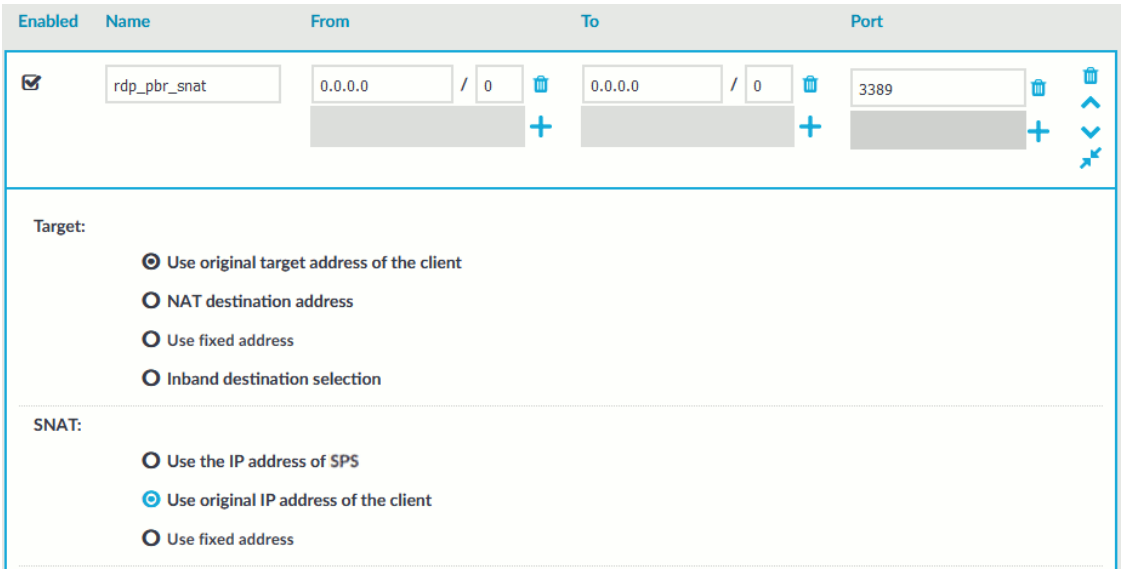
2. Create an ACL entry that matches the reply packets coming from the server zone and targeted at the client zone to make sure that replies are reaching the SPS.

   ```
   #(config) ip access-list extended ssh-outbound
   #(config-ext-nacl) permit tcp <dst net> <dst mask> eq <dst port> <src
   net> <src mask>
   ```

   NOTE:

   This step is only required if you want to use Source NAT (IP Spoofing) instead of SPS's address towards the monitored servers.

Figure 308: Control > Connections — Using SNAT



Example: Configuring an ACL entry for reply packets with Cisco policy routing

In case of the example in step 1, the permit clause should be:

```
#(config-ext-nacl) permit tcp 10.0.0.0 0.0.0.255 eq 22
192.168.0.0 0.0.0.255
```

3. Create a route-map entry. It controls which packets are affected by policy routing and where they should be forwarded to. The `match` commands specify the conditions under which policy routing occurs. The `set` commands specify the routing actions to perform if the criteria enforced by the `match` commands are met. A new route-map can be defined as follows:

   ```
   #(config) route-map scb-inbound
   ```

   a. Set your route-map to match the traffic in ACL `ssh-inbound`:

      ```
      #(config-route-map) match ip address ssh-inbound
      ```

   b. Set an action on the matching traffic. Define a next-hop entry to redirect the traffic to the SPS.

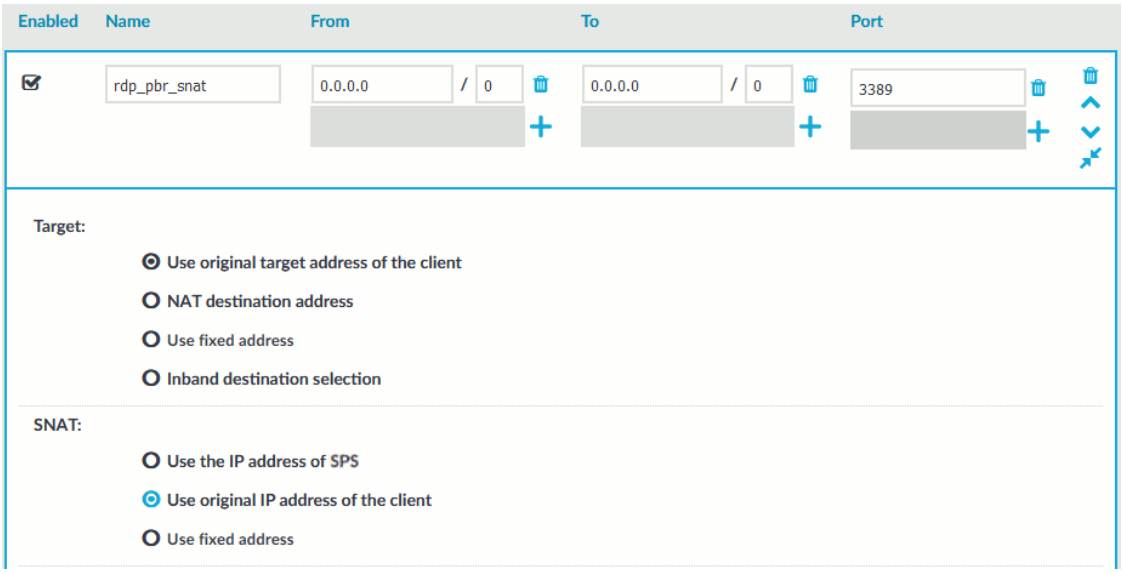
      ```
      #(config-route-map) set ip next-hop <SPS IP address>
      ```

4. Create another route-map that controls the reply packet flow.

```
#(config) route-map scb-outbound
#(config-route-map) match ip address ssh-outbound
#(config-route-map) set ip next-hop <SPS IP address>
```

> NOTE:
>
> This step is only required if you want to use Source NAT (IP Spoofing) instead of SPS's address towards the monitored servers.

Figure 309: Control > Connections — Using SNAT



5. Apply the route-map to the appropriate interfaces.
   a. First, add the ssh-inbound route-map entry to the interface facing the clients:

   ```
   #(config) interface <interface-facing-the-clients>
   #(config-if) ip policy route-map scb-inbound
   ```

   b. Then add the ssh-outbound route-map entry to the interface facing the servers:

   ```
   #(config) interface <interface-facing-the-servers>
   #(config-if) ip policy route-map scb-outbound
   ```

Expected result:

The traffic from the clients targeting the specified port of the servers is redirected to SPS. Therefore, SPS can be configured to control and audit this traffic.

The full configuration for the above topology:

```
! interface facing the clients
interface FastEthernet0/0
 ip address 192.168.0.254 255.255.255.0
 ip policy route-map scb-inbound
 duplex full
 speed auto
 no mop enabled

! interface facing the SCB
interface FastEthernet0/1
 ip address 172.16.0.254 255.255.255.0
 duplex full
 speed auto
 no mop enabled

! interface facing the servers
interface FastEthernet1/0
 ip address 10.0.0.254 255.255.255.0
 ip policy route-map scb-outbound
 duplex full
 speed auto
 no mop enabled

! access lists matching the server and client subnets and the SSH port -
incoming packets
ip access-list extended ssh-inbound
 permit tcp 192.168.0.0 0.0.0.255 10.0.0.0 0.0.0.255 eq 22
! access lists matching the server and client subnets and the SSH port -
reply packets
ip access-list extended ssh-outbound
 permit tcp 10.0.0.0 0.0.0.255 eq 22 192.168.0.0 0.0.0.255

! policy routing entry matching on the incoming SSH connections and
! redirecting them to the SCB external interface
route-map scb-inbound permit 10
 match ip address ssh-inbound
 set ip next-hop 172.16.0.1

! the following part is only required for SNAT-based SCB configuration
! policy routing entry matching on the SSH reply packets and
! redirecting them to the SCB external interface
route-map scb-outbound permit 10
 match ip address ssh-outbound
 set ip next-hop 172.16.0.1
```

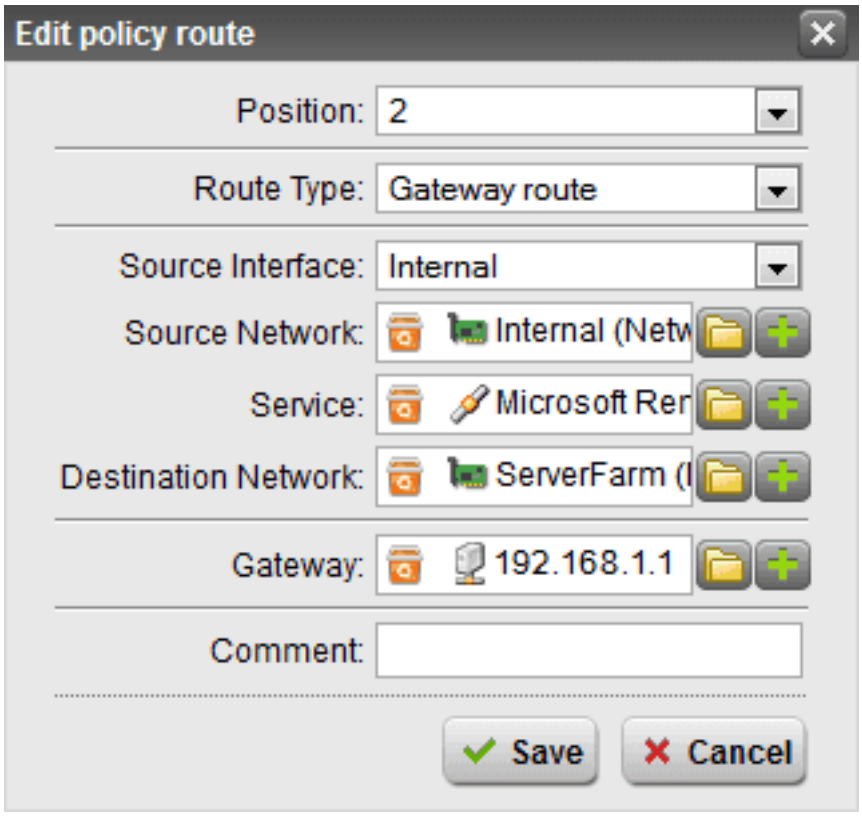
Configuring advanced routing on Sophos UTM (formerly Astaro Security Gateway) firewalls

The following describes how to configure a Sophos UTM firewall to redirect selected traffic to One Identity Safeguard for Privileged Sessions (SPS) instead of its original destination. Interface 1 will be referred to as 'Internal' and Interface 2 will be referred to as 'ServerFarm'.

To configure a Sophos UTM firewall to redirect selected traffic to SPS instead of its original destination

1. On the **Policy Routes** tab of the Sophos UTM firewall, click **New Policy Route**.

2. **Figure 310: New Policy Route**

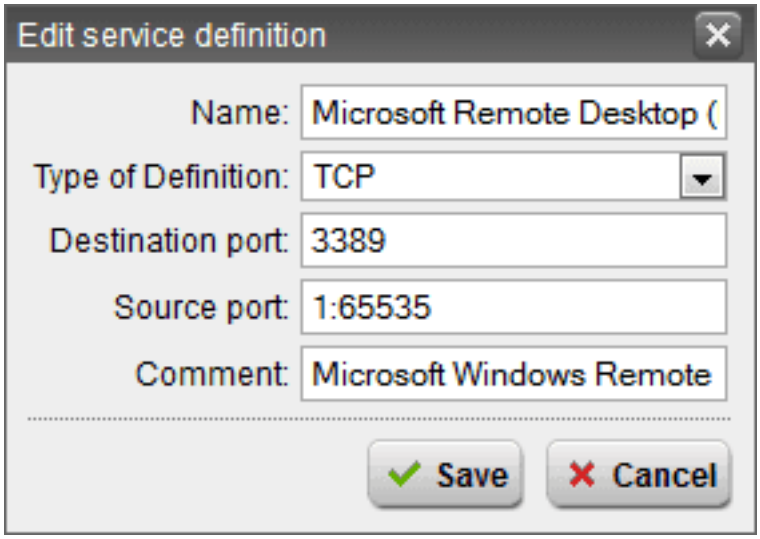


In the dialog box, enter the following settings:

- *Position*: Set the position number, defining the priority of the policy route. Lower numbers have higher priority. Routes are matched in ascending order. Once a route has been matched, routes with a higher number will not be evaluated anymore.

- *Route Type*: Select `Gateway route`. Packets will be sent to a particular host (gateway).
- *Source Interface*: Select `Internal`. This is the interface where the data packet to be routed arrives from.
- *Source Network*: Select `Internal (Network)`. This is the source network of the data packets to be routed.
- *Service*: Select `Microsoft Remote Desktop Protocol`. This is the service definition that matches the data packet to be routed.
- *Destination Network*: Select `ServerFarm (Network)`. This is the destination network of the data packets to be routed.
- *Gateway*: Select the IP address of SPS. This is the router where the gateway will forward data packets to.
- *Comment*: Optionally, enter a description or other information.

3. Click **Save**.
4. Click the status icon to activate the route.
5. Navigate to **Definitions & Users > Service Definitions** and click **New Service Definition**.
6. **Figure 311: New Service Definition**



In the dialog box, enter the following settings. It will ensure that the policy will apply to all TCP/3389:

- *Name*: Enter a descriptive name for the definition (for example Microsoft Remote Desktop Protocol).
- *Type of Definition*: Select `TCP`. This is the service type.

> **NOTE:**
>
> The definition type cannot be changed after saving the definition. To change the definition type, delete the service definition and create a new one with the desired settings.

- *Destination port*: Enter `3389`. This is the destination port that can either be entered as a single port number (for example `80`), or as a port range, using a colon as delimiter (for example 1024:64000).
- *Source port*: Enter `1:65535`. This is the source port that can either be entered as a single port number (for example `80`), or as a port range, using a colon as delimiter (for example 1024:64000).
- *Comment*: Optionally, enter a description or other information.

7. Click **Save**. The new definition appears in the service definition list.

   With this step, the client-server routing is configured.

8. To configure the server-client routing, create another policy route, and In the dialog box, enter the following settings:
   - *Position*: Set the position number, defining the priority of the policy route. Lower numbers have higher priority. Routes are matched in ascending order. Once a route has been matched, routes with a higher number will not be evaluated anymore.
   - *Route Type*: Select `Gateway route`. Packets will be sent to a particular host (gateway).
   - *Source Interface*: Select `ServerFarm`. This is the interface where the data packet to be routed arrives from.
   - *Source Network*: Select `ServerFarm (Network)`. This is the source network of the data packets to be routed.
   - *Service*: Select `3389`. This is the service definition that matches the data packet to be routed.
   - *Destination Network*: Select `Internal (Network)`. This is the destination network of the data packets to be routed.
   - *Gateway*: Select the IP address of SPS. This is the router where the gateway will forward data packets to.
   - *Comment*: Optionally, enter a description or other information.

Using SCP with agent-forwarding

When the client uses SSH to access a target server via One Identity Safeguard for Privileged Sessions (SPS) and authenticates with the public keys, the SPS Authentication Policy has **Public key > Agent** configured on the server-side. If the client supports agent-forwarding, this works well. However, scp does not: it always adds the `-a` option to the command-line to disable agent-forwarding. Explicitly allowing agent-forwarding with the `-A` or the `-oForwardAgent yes` command-line option, or writing `ForwardAgent yes` into the configuration has no effect, because the implicit `-a` at the end of the command-line takes precedence.

Solution 1: Use a wrapper script

The scp application can be started with the `-S` option to use an external application to create the encrypted connection. On Linux and UNIX platforms, this external application can be, for example, the following script that removes the unnecessary option from the scp command line.

```
#!/usr/bin/perl
exec '/usr/bin/ssh', '-A', map {$_ eq '-oForwardAgent=no' ? ( ) : $_} @ARGV
```

If you want your clients to use this script transparently, you can create an alias for it with the following command:

```
alias scp='scp -S <path-to-the-script-on-the-client>'
```

Solution 2: Use ssh master-channels

This solution relies on sending scp through an SSH master-control channel. In this case, scp does not need agent-forwarding, because it is already performed during the ControlMaster setup. The advantage of this solution is that the scp connection is setup quickly, because no authentication is needed, since the connection is already open. The disadvantage is that first a ControlMaster connection must be opened to the target host using the following command:

```
ssh -M -S /tmp/<address-of-the-target-server> <address-of-the-target-server>
```

When staring scp, reference the control path created with the previous command:

```
scp -oControlPath=/tmp/<address-of-the-target-server> [[user@]host1:]file1 ...
[[user@]host2:]file2
```

Solution 3: Patch the scp source

You can simply patch the scp source to overcome the problem, but then you need to recompile and re-install scp on every platform you use in your environment. The following is a sample patch for openssh-5.6p1:

```
--- scp-org.c    2010-07-02 05:37:33.000000000 +0200
+++ scp-new.c    2010-09-08 17:56:33.000000000 +0200
@@ -339,7 +339,6 @@
     args.list = NULL;
     addargs(&args, "%s", ssh_program);
     addargs(&args, "-x");
-    addargs(&args, "-oForwardAgent no");
     addargs(&args, "-oPermitLocalCommand no");
     addargs(&args, "-oClearAllForwardings yes");
```

Solution 4: Use fix or mapped keys on server-side

This is not agent-forwarding anymore, but scp still can use keys. Instead of passing the user-keys to the target server, SPS can authenticate on the server using a fix key, or a separate key for every user. Setting the server-side keys on SPS (or fetching them from LDAP), has the following advantages:

- The user cannot bypass SPS and directly connect to the target server
- Key-handling in the server environment becomes much simpler, because you do not have to import the user-keys to every host (if this is done locally, without a central identity management system)

For details on configuring server-side keys on SPS, see Relayed authentication methods on page 557.

Solution 5: WinSCP and agent-forwarding

WinSCP is a common tool for Windows to transfer files using SFTP/SCP. To use agent-forwarding in WinSCP, enable it in the **SSH > Authentication** options and load your keys.

Security checklist for configuring One Identity Safeguard for Privileged Sessions (SPS)

The following checklist is a set of recommendations and configuration best practices to ensure that your One Identity Safeguard for Privileged Sessions (SPS) is configured securely.

Encryption-related settings

Connection policies

Appliance access

Networking considerations

Encryption-related settings

- One Identity recommends using 2048-bit RSA keys (or stronger).
- Use strong passwords: at least 8 characters that include numbers, letters, special characters, and capital letters. For local One Identity Safeguard for Privileged Sessions (SPS) users, require the use of strong passwords (set **AAA > Settings > Minimal password strength** to `strong`). For details, see "Setting password policies for local users" in the Administration Guide.
- When exporting the configuration of SPS, or creating configuration backups, always use encryption. Handle the exported data with care, as it contains sensitive information, including credentials. For details on encrypting the configuration, see "Encrypting configuration backups with GPG" in the Administration Guide.
- Use every keypair or certificate only for one purpose. Do not reuse cryptographic keys or certificates (for example, do not use the certificate of the One Identity Safeguard for Privileged Sessions (SPS) webserver to encrypt audit trails, or the same keypair for signing and encrypting data).

- Do not use the CBC block cipher mode, or the `diffie-hellman-group1-sha1` key exchange algorithm. For details, see "Supported encryption algorithms" in the Administration Guide.
- Always encrypt your audit trails to protect sensitive data. For details, see "Encrypting audit trails" in the Administration Guide.

Connection policies

- When configuring connection policies, always limit the source of the connection to the client network that requires access to the connection.
- Always use gateway authentication to authenticate clients. Do not trust the source IP address of a connection, or the result of server authentication.
- To prevent Denial of Service (DoS) attacks against One Identity Safeguard for Privileged Sessions (SPS), set the **Connection rate limit** option of your connection policies. For details, see "Configuring connections" in the Administration Guide.
- Configure your RDP connection policies to use strong encryption. To enable SSL-encryption for the RDP protocol, see "Enabling TLS-encryption for RDP connections" in the Administration Guide.
- In RDP connections, if the client uses the Windows login screen to authenticate on the server, the password of the client is visible in the audit trail. To avoid displaying the password when replaying the audit trail, you are recommended to encrypt the upstream traffic in the audit trail using a separate certificate from the downstream traffic. For details, see "Encrypting audit trails" in the Administration Guide.
- Ensure that host key verification is enabled in SSH connection policies. That is, the **Server side hostkey settings > Allow plain host keys** and **Server side hostkey settings > Allow X.509 host certificates** options do not have the **No check required** option selected. For details, see "Setting the SSH host keys of the connection" in the Administration Guide.

Appliance access

- Accessing the One Identity Safeguard for Privileged Sessions (SPS) host directly using SSH is not recommended or supported, except for troubleshooting purposes. In such case, the One Identity Support Team will give you exact instructions on what to do to solve the problem.

  For security reasons, disable SSH access to SPS when it is not needed. For details, see "Enabling SSH access to the One Identity Safeguard for Privileged Sessions (SPS) host" in the Administration Guide.
- Permit administrative access to SPS only from trusted networks. If possible, monitored connections and administrative access to the SPS web interface should

originate from separate networks.

- Configure SPS to send an alert if a user fails to login to SPS. For details, see the **Login failed** alert in "System related traps" in the Administration Guide.
- Configure **Disk space fill-up prevention**, and configure SPS to send an alert if the free space on the disks of SPS is low. For details, see "Preventing disk space fill-up" in the Administration Guide.

Networking considerations

- One Identity Safeguard for Privileged Sessions (SPS) stores sensitive data. Use a firewall and other appropriate controls to ensure that unauthorized connections cannot access it.
- If possible, enable management access to SPS only from trusted networks.
- Make sure that the HA interface of SPS is connected to a trusted network.

Jumplists for in-product help

To find the documentation for a specific UI element, browse the following sections.

Basic Settings > Management

Basic Settings > Local Services

Basic Settings > System

<Protocol name> Control > Global Options

Basic Settings > Management

- **Basic Settings > Management > Syslog**: For details, see Configuring system logging on page 126.
- **Basic Settings > Management > SNMP trap settings**: For details, see Configuring SNMP alerts on page 131.
- **Basic Settings > Management > Mail settings**: For details, see Configuring e-mail alerts on page 129.
- **Basic Settings > Management > Web interface timeout**: For details, see Web interface timeout on page 112.
- **Basic Settings > Management > RPC API settings**: For details, see Enabling RPC API access to One Identity Safeguard for Privileged Sessions (SPS) on page 829.
- **Basic Settings > Management > Change root password**: For details, see Changing the root password of One Identity Safeguard for Privileged Sessions (SPS) on page 394.
- **Basic Settings > Management > System backup**: For details, see:
  - Creating configuration backups on page 157
  - Encrypting configuration backups with GPG on page 158
- **Basic Settings > Management > Verbose system logs**: For details, see Changing log verbosity level of One Identity Safeguard for Privileged Sessions (SPS) on page 858.
- **Basic Settings > Management > SSL certificates**: For details, see Managing the certificates used on One Identity Safeguard for Privileged Sessions (SPS) on page

407.

- **Basic Settings > Management > Core files**: For details, see Gathering data about system problems on page 856.
- **Basic Settings > Management > Disk space fill-up prevention**: For details, see Preventing disk space fill-up on page 139.
- **Basic Settings > Management > Web gateway authentication**: For details, see Configuring out-of-band gateway authentication on page 744.
- **Basic Settings > Management > Splunk forwarder**: For details, see Using the Splunk forwarder on page 171.

Basic Settings > Local Services

- **Basic Settings > Local Services > SSH server**: For details, see Enabling SSH access to the One Identity Safeguard for Privileged Sessions (SPS) host on page 391.
- **Basic Settings > Local Services > Web login (admin and user)**: For details, see Configuring user and administrator login addresses on page 118.
- **Basic Settings > Local Services > Web login (user only)**: For details, see Configuring user and administrator login addresses on page 118.
- **Basic Settings > Local Services > SNMP server settings**: For details, see Querying SPS status information using agents on page 133.
- **Basic Settings > Local Services > Indexer service**: For details, see Configuring the internal indexer on page 593.
- **Basic Settings > Local Services > Privileged Account Analytics**: Select this option only if you are also using One Identity Safeguard for Privileged Analytics.

  To enable One Identity Safeguard for Privileged Analytics and analyze the behavior of your users, One Identity Safeguard for Privileged Sessions (SPS) requires a special license. Also, depending on the number of your users and sessions, the performance and sizing of SPS must be considered. If you are interested in One Identity Safeguard for Privileged Analytics, contact our Sales Team, or your One Identity representative. For details on One Identity Safeguard for Privileged Analytics, see the One Identity One Identity Safeguard for Privileged Analytics website. For details on enabling One Identity Safeguard for Privileged Analytics, see Safeguard for Privileged Analytics Configuration Guide.
- **Basic Settings > Local Services > Cluster Interface**: This option is related to an experimental feature that will allow you to manage and synchronize the configuration of multiple SPS appliances from a central server. If you are interested in this feature, contact our Support Team.

Basic Settings > System

- **Basic Settings > System > System control**: For details, see Controlling One Identity Safeguard for Privileged Sessions (SPS): reboot, shutdown on page 343.
- **Basic Settings > System > Traffic control**: For details, see Disabling controlled traffic on page 344.
- **Basic Settings > System > Version details**: For details, see Upgrading One Identity Safeguard for Privileged Sessions (SPS) (single node) on page 380.
- **Basic Settings > System > Export configuration**: For details, see Exporting the configuration of One Identity Safeguard for Privileged Sessions (SPS) on page 384.
- **Basic Settings > System > Import configuration**: For details, see Importing the configuration of One Identity Safeguard for Privileged Sessions (SPS) on page 386.
- **Basic Settings > System > License**: For details, see Managing the One Identity Safeguard for Privileged Sessions (SPS) license on page 387.
- **Basic Settings > System > Sealed mode**: For details, see Sealed mode on page 397.
- **Basic Settings > System > Firmwares**: For details, see Upgrading One Identity Safeguard for Privileged Sessions (SPS) (single node) on page 380.

<Protocol name> Control > Global Options

- **<Protocol name> Control > Global Options > Traffic**: For details, see Changing log verbosity level of One Identity Safeguard for Privileged Sessions (SPS) on page 858.
- **<Protocol name> Control > Global Options > Audit > Timestamping**: For details, see:
  - Digitally signing audit trails on page 474
  - Timestamping audit trails with external timestamping service on page 472
  - Timestamping audit trails with built-in timestamping service on page 469
- **<Protocol name> Control > Global Options > Audit > Delete search metadata from SPS after**: For details, see Configuring cleanup for the One Identity Safeguard for Privileged Sessions (SPS) connection database on page 488.

Overview

Access control in One Identity Safeguard for Privileged Sessions (SPS) is based on groups. Whenever a user needs to access a protected resource, like navigating to a configuration page on the SPS web interface, or opening a channel in a connection, SPS checks the access control list associated with the resource in question.

The access control lists grant access to groups. Therefore, SPS needs to determine which groups the user is a member of to evaluate the access rules.

When you configure SPS to use an LDAP backend, SPS will:

1. Identify the user. For more information, see *User identification* below.
2. Determine the relevant groups the user is a member of. For more information, see *Group membership resolution* below.

User identification

SPS works with plain usernames, for example, `administrator`. This must be unambiguously resolved to an LDAP user object in order to determine the user's groups. If a user identification returns multiple results, SPS treats this as an error, and access to the user in question is denied.

Only the user object returned in this phase is used for group membership checks, and *not* the original plain username.

User resolution depends on the type of the backend (POSIX or Active Directory).

For more information, see the backend-specific sections below.

Group membership resolution

SPS works with plain group names, for example, `superusers`. For group membership checks, SPS looks up a relevant group object in LDAP and checks if the user object returned during user identification is a member of that group. Since some of the group object's attributes are always used for group membership checks, the group object must also exist in LDAP.

Group membership resolution depends on the LDAP backend type.

For more information, see the backend-specific sections below.

Common to all backends

All backends have configurable parameters relevant for user identification and group membership:

- `bind_dn` and `bind_password`: *Bind DN* and *Bind password* are used for user identification and group membership check during authentication to the LDAP database. If you leave it empty, One Identity Safeguard for Privileged Sessions (SPS) will try to bind anonymously.
- `user_base_dn`: *User Base DN* is where SPS searches for users.
- `group_base_dn`: *Group Base DN* is where SPS searches for groups. Only groups under this base are considered for membership.
- `memberof_check`: the *Enable checking for group DNs in user objects* setting allows checking a configurable attribute in the user object. This attribute contains a list of group DNs the user is additionally a member of. This user attribute is usually **memberOf**. For more information, see the backend-specific sections below.
- `user_dn_in_groups`: *Check the user DN in these groups* is a list of additional group object classes and their respective attributes where SPS will look for member user DNs. For more information, see the backend-specific sections below.

All comparisons and searches are done by SPS in a way that plain user and group names are matched with attribute values by the LDAP server. As a result, user and group names are case insensitive if and only if the matching rule for the attribute in question is case insensitive in the LDAP database.

POSIX LDAP backend

In addition to the common parameters, the POSIX backend has the following configurable parameters:

- `username_attribute`: *Username (user ID) attribute name* is the name of the attribute in the user object, which contains the user's plain username.
- `membership_check`: *Enable POSIX group membership check* enables POSIX *primary* and *supplementary* group membership checking. When enabled, it has the following configurable parameter:
    - `member_uid_attribute`: the optional *POSIX group membership attribute name* is the name of the attribute in a **posixGroup** group object, which lists the plain usernames that are members of the group. These groups are usually referred to as *supplementary groups* of the referred user.

User identification in POSIX

To determine the user entry for a given plain username, One Identity Safeguard for Privileged Sessions (SPS) performs a search under `user_base_dn` for objects having the `username_attribute` equal to the plain username of the user. The **objectClass** of the user object is not restricted.

The user object returned here is used for group membership checks.

Group membership resolution in POSIX

For all group membership checks, only the LDAP user object returned during user identification phase is used.

The plain group name is always compared to the **cn** attribute of the group object.

A user is treated as a member of a group given by its plain group name if the plain group name matches the **cn** attribute of the group object, and *any* of the following is true:

- The group is the user's primary group. That is, the group is a **posixGroup**, and the user's **gidNumber** attribute is equal to the group's **gidNumber** attribute.

  This check is performed only when the `membership_check` option is enabled for POSIX.

  > Note
  >
  > It is OK for the user to have no **gidNumber** attribute, in which case this check will be skipped.

- The group lists the user's short username. That is, the group is a **posixGroup**, and it's `member_uid_attribute` contains the short username from the user object.

  This check is performed only when the `membership_check` option is enabled, and the `member_uid_attribute` is configured.

  > Note
  >
  > For the purpose of this check, the user's short username is retrieved from the user object's `username_attribute`. Currently, this attribute should only contain a single username. A warning will appear in the logs if this is not the case, and the first value of the attribute will be used as returned by the server. This is a known limitation.

- The group lists the user's **dn** in any of the additional group objects configured in `user_dn_in_groups`.

  For example, if a row is added with `objectClass` set to **groupOfNames** and `attribute` set to **member**, SPS will treat the user as a member of all groups where the group is a **groupOfNames**, and the group's **member** attribute contains the user's **dn**.

- The user lists the group's **dn**. That is, the user's `memberof_user_attribute` contains the **dn** of the group, and the **objectClass** of the referred group is `memberof_group_objectclass`.

  This check is performed only when the `memberof_check` option is enabled for POSIX.

> Note
>
> SPS compares the **dn** stored in the `memberof_user_attribute` to the **dn** of the group object itself in a strict stringwise manner. Therefore, the user attribute must contain the group DN exactly as it would be returned by the LDAP server. No case or accent differences are allowed.

Active Directory LDAP backend

In addition to the common parameters, the Active Directory (AD) backend has the following additional configurable parameters:

- `membership_check`: *Enable AD group membership check* enables AD specific non-primary group membership checking.

  > Note
  >
  > The AD user's primary group is always checked regardless of this setting.

  - `nested_groups`: *Enable nested groups* allows AD nested group support. See below for details.

Additionally, AD supports case and accent insensitive matching in many of the user and group name attributes. Since One Identity Safeguard for Privileged Sessions (SPS) relies on the server to perform comparisons, case and accent insensitive user and group name support depends solely on the server configuration.

User identification in AD

To determine the user entry for a given plain username, SPS performs a search under `user_base_dn` for objects having either the **sAMAccountName** or the **userPrincipalName** equal to the plain username of the user. The **objectClass** of the user object is not restricted.

> Note
>
> Although **userPrincipalName** in AD is a Internet-style name like *user@example.com*, it matches simple names like *user*.

Only the user object returned here is used for group membership checks.

Group membership resolution in AD

For all group membership checks, only the LDAP user object returned during user identification phase is used.

The plain group name is always compared to the **cn** attribute of the group object.

A user is treated as a member of a group if both the group object's **objectClass** and **objectCategory** is **group**, and *any* of the following is true:

- The group is the user's primary group. That is, the **objectSID** attribute of the group matches the Security Identifier calculated from the user object's **objectSID** and **primaryGroupID** attributes, as described in the Microsoft Support article How to use the PrimaryGroupID attribute to find the primary group for a user.

  > Note
  >
  > When using the AD backend, this check is always performed, even if the `membership_check` option is disabled. However, it is OK for the user to have no primary group.

- The group lists the user's short username. That is, the group's **memberUid** attribute contains the short username from the user object.

  This check is performed only when the `membership_check` option is enabled for AD.

  > Note
  >
  > For the purpose of this check, the user's short username is retrieved from the user object's **sAMAccountName** attribute only, which is a single-valued attribute in AD. This is a known limitation.
  >
  > It is OK for the **sAMAccountName** attribute to be missing, in which case this check will be skipped.

- The group lists the user's **dn**. That is, the group object's **member** attribute contains the user's **dn**.

  This check is performed only when the `membership_check` option is enabled for AD.

  This is the only place where *nested groups* are supported. When the `nested_groups` setting is enabled in the configuration, SPS will also find groups which do not directly contain the user's **dn** in their **member** attribute, but do contain an intermediate group's **dn**, which in turn contains the user **dn** in its **member** attribute. This nesting can be arbitrarily deep, limited only by AD.

  > Note
  >
  > Due to the nature of the way AD resolves the nested group chain, intermediate groups might be outside the configured `group_base_dn`.

> Note
>
> Although an **objectCategory** in AD is a DN-valued attribute, it does match simple names like **group**.

Additionally, a user is treated as a member of a group if:

- The group lists the user's **dn** in any of the additional group objects configured in `user_dn_in_groups`.

  For example, if a row is added with `objectClass` set to **groupOfNames** and `attribute` set to **member**, SPS will treat the user as a member of all groups where the group is a **groupOfNames**, and the group's **member** attribute contains the user's **dn**.

  > Note
  >
  > There is no additional restriction on the group's **objectClass** in this case.

- The user lists the group's **dn**. That is, the user's `memberof_user_attribute` contains the **dn** of the group, and the **objectClass** of the referred group is **group**.

  This check is performed only when the `memberof_check` option is enabled for AD.

  > Note
  >
  > SPS compares the **dn** stored in the `memberof_user_attribute` to the **dn** of the group object itself in a strict stringwise manner. Therefore, this user attribute must contain the group DN exactly as it would be returned by the LDAP server. No case or accent differences are allowed.

Appendix: Deprecated features

These features have been deprecated and will be removed from the upcoming releases.

Deprecated: Using the Search (classic) interface

IMPORTANT: Support for the Search (classic) interface is deprecated. This feature will be removed from the upcoming releases. One Identity recommends using the supported Search interface instead. For more information, see Using the Search interface.

This section describes how to browse the audit trails stored on One Identity Safeguard for Privileged Sessions (SPS), or archived to a remote server, how to search for a specific audit trail, and also how to replay them from the browser.

- Searching audit trails: the One Identity Safeguard for Privileged Sessions (SPS) connection database on page 934 explains how to use and customize the search interface, describes the connection metadata that is available on SPS, and provides examples of wildcards and boolean search operators you can use.
- Connection details on page 937 describes how to review the details of a connection, and to replay it online from SPS.
- Displaying statistics on search results on page 970 describes how you can create custom statistics from the search results, and how to save them for reports.

Searching audit trails: the One Identity Safeguard for Privileged Sessions (SPS) connection database

One Identity Safeguard for Privileged Sessions (SPS) has a search interface for browsing the audit trails. This connection database also contains the various meta-information about connections and connection-requests. The search queries can include only alphanumerical characters.

To access the search interface, navigate to **Search**. Only users with the following privileges can access the **Search** page:

- Members of groups who are configured as **Authorizers** with the `Search` or `Search&Authorize` permission set in the **Access Control** field of a connection policy. These users can access only the audit trails of the respective connections.

For more information on configuring authorizers for a connection, see Configuring four-eyes authorization on page 752.

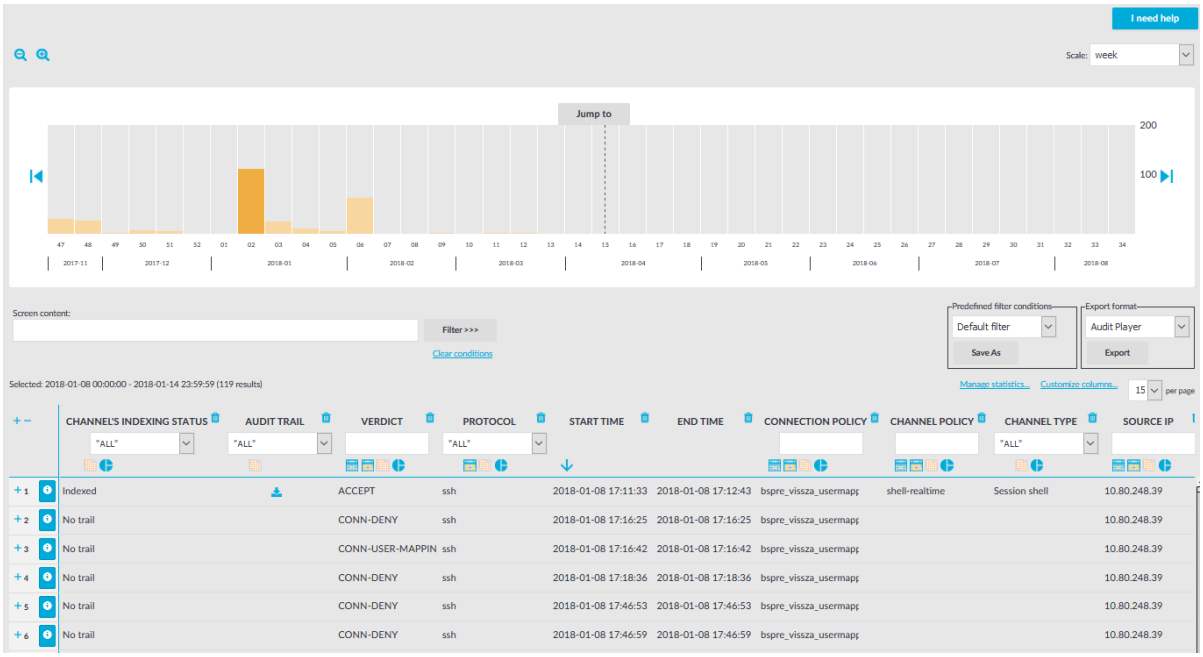
- Members of groups who have the `Search` privilege set.

  Assigning the **Search** privilege to a user on the AAA page automatically enables the **Search in all connections** privilege, and grants the user access to every audit trail, even if the user is not a member of the groups listed in the **Access Control** option of the particular connection policy.

  For more information on configuring user rights, see Managing user rights and usergroups on page 320.

- The `admin` user.

Figure 312: Search — Browse the connections database



Changing the time interval

The bars display the number of results in the selected interval. Use the and icons to zoom, and the arrows to display the previous or the next intervals. To explicitly select a date, select **Jump to** and set the date in the calendar. You can change the length of the displayed interval with the **Scale** option.

Hovering the mouse above a bar displays the number of entries and the start and end date of the period that the bar represents. Click a bar to display the entries of that period in the table. Use Shift+Click to select multiple bars.

Searching connections

> NOTE:
>
> This feature is available only if auditing and content indexing was requested for the connection. For details, see Configuring the internal indexer on page 593.

To search in the content of the indexed audit trails, enter your search keywords in the **Screen content** field, and click **Filter**. Search is case insensitive. You can use complex expressions and boolean operators. For more information, see Using the content search on page 950.

Filtering search results

Connection metadata is displayed in customizable columns that you can filter for any parameter, or a combination of parameters. To filter the list of search results, enter the filter expression in the input field of the appropriate column, and press Enter, or click on an entry in the table.

For the description of the available columns, see Connection metadata on page 958.

For information on using and saving filters, see Using and managing search filters on page 964.

> NOTE:
>
> When you use filters, the bars display the statistics of the filtered results.

Filtering displays also partial matches. You can use the  icon to perform an exact search, and the  icon for inverse filtering ("does not include"). To clear filters from a column, click .

To restore the original table, click **Clear conditions**.

> TIP:
>
> Use the drop-down menu of the **Protocol** column to quickly filter the list for a single protocol.

Exporting the search results

To export the search results as a comma-separated text file, select **Export format > CSV**, and click **Export**.

For instructions on displaying statistics about your search results, see Displaying statistics on search results on page 970.

Viewing the details of a connection

To display the summary of a connection, click  , or use the  shortcuts to view the corresponding connection details (for example, Events). The summary is displayed in the

connection details pop-up window. For more information, see Connection details on page 937.

To download the audit trail of a session, click the icon in the **Audit-trail** column.

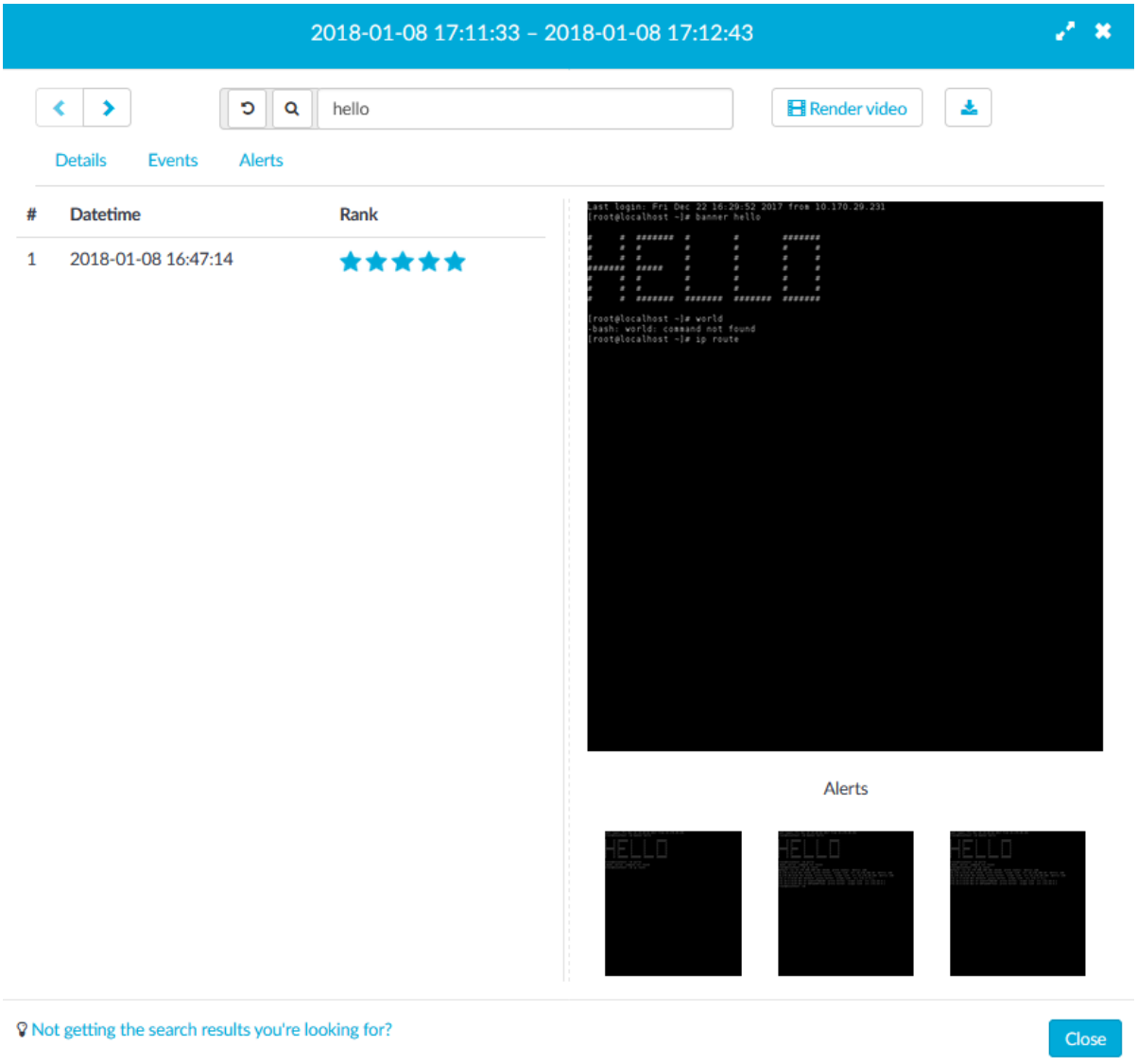
Connection details

The Details pop-up window provides in-depth information on each of the indexed audit trails stored in the connection database. You can use it to gain contextual insight about the indexed session and its events.

The pop-up window consists of two main parts: the header and the trail details. In the header, you can:

- Move to the previous / next trail listed on the Search page with the ‹ and › buttons.
- Search the current trail. Search is performed on the displayed audit trail only. When you move between trails, search is reset to the query you used on the Search page (if you entered one). You can also revert to that query using the button. For details on using search expressions, see Using the content search on page 950.
- Export / follow the trail. Click the button to export the trail, or the button to follow an ongoing connection. The trail data is exported in `.srs` format, which you can open with the Safeguard Desktop Player application.

Figure 313: Audit trail details



Trail details

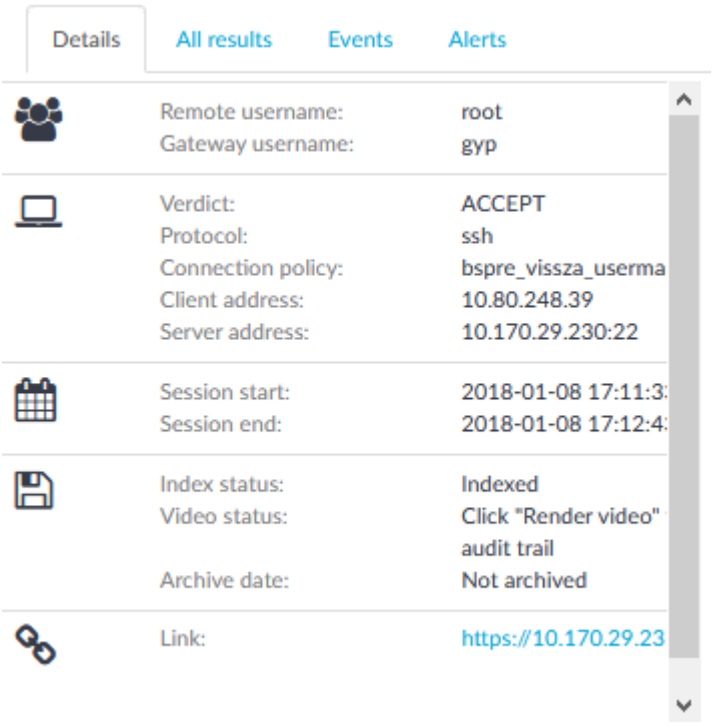
The details section is organized into tabs (left) and screenshots (right). The Details tab is always visible. The All results, Events, and Alerts tabs are displayed dynamically, when there is matching content in the trail.

Details tab: Quick summary of the connection details (user, server, time).

- User information: remote and gateway username. The gateway username corresponds with the `Username` field of the connection metadata database, so note the following:

    - If the user performed inband gateway authentication in the connection, the field contains the username from the gateway authentication (gateway username).
    - Otherwise, the field contains the username used on the remote server.
- Connection information: connection verdict, protocol, connection policy, client and server address.
- Session time: start and end time of the connection.
- Trail information: is the trail indexed, or archived.
- Link: a link that leads to the Search page filtered to show only this connection. Note that if you share this link, other users can access the audit trail only if they have the required privileges, and can access One Identity Safeguard for Privileged Sessions (SPS) using the IP address in the link (SPS can be configured to be accessible using multiple IP addresses).

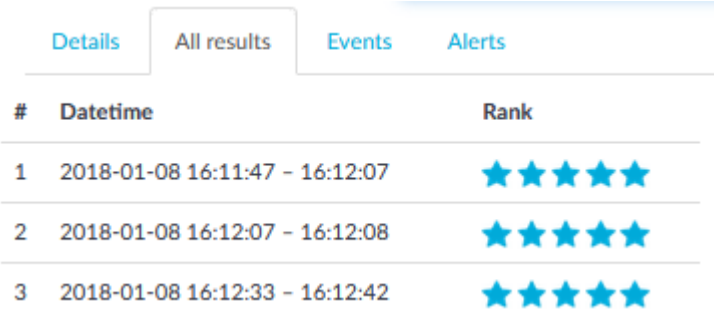
Figure 314: Details tab



All results tab: Matching results for your search on the Search page (or in the trail contents), in chronological order.

- Date and time of the matching event.
- Search rank. The displayed Rank indicates how closely the result matches your search query.
- Screenshots. If screenshots are available for the trail, you can click each search result to view the corresponding screenshot.

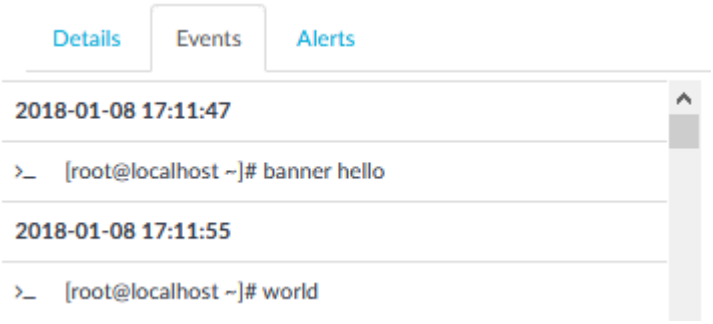
Figure 315: All results tab



Events tab: Connection events, in chronological order.

- Date and time of the event.
- Event type (command, screen content, window title).
- Event details.

Figure 316: Events tab

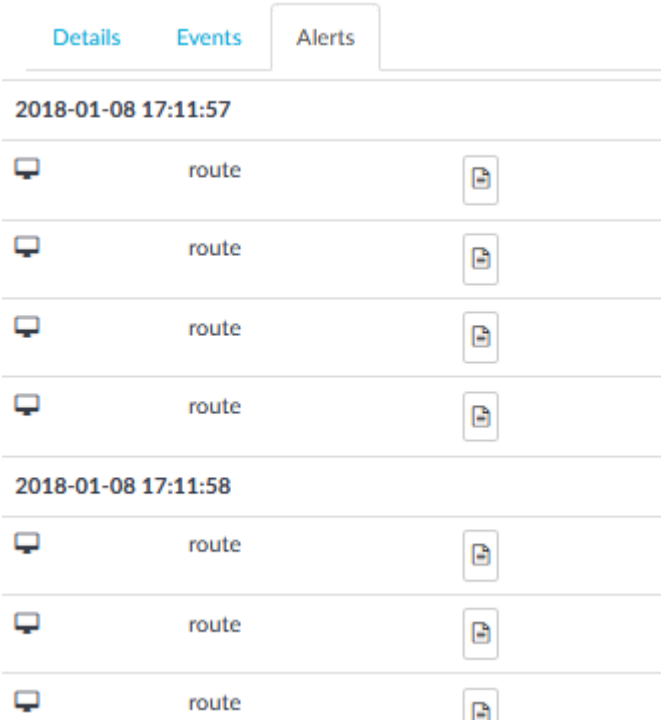


Alerts tab: Content policy alerts triggered in the session, in chronological order.

An event is listed as alert only if the **Actions > Store in Connection Database** option is selected in the **Content Policy** used to handle the session.

- Date and time of the alert.
- The type of the alert (command, screen content, credit card, window title).
- The matching content.
- Terminal buffer contents. If the alert is not visible on the screenshot, you can click the icon to view the contents of the full terminal buffer.
- Screenshots. If screenshots are available for the trail, you can click each alert to view the corresponding screenshot.

Figure 317: Alerts tab



Screenshots are generated for search results and alerts when the trail is opened, and for subsequent searches. You can scroll between screenshots using the carousel, and view each screenshot in full size. Selecting a screenshot highlights the corresponding search result or alert.

Screenshots are not available for:

- Ongoing connections.
- Unindexed trails.
- Trails of HTTP connections.
- Encrypted trails (without the necessary certificate).

NOTE:

For SSH and Telnet trails, trail data is aggregated for each second. The screenshot you see reflects the terminal buffer as it was visible at the end of that second. If data was pushed off-screen during this second, the search still finds it, but it will not be visible on the generated screenshot.

Replaying audit trails in your browser in Search (classic)

CAUTION:

You can replay audit trails in your browser, or using the Safeguard Desktop Player application. Note that there are differences between these solutions.

	Browser	Safeguard Desktop Player
Works without installation	✓	-
Works on any operating system	✓	Windows, Linux, Mac
Can replay audit trails recorded with SPS 5 F4 and newer	✓	✓
Can replay TN5250 sessions	✓	✓
Can extract files from SCP, SFTP, HTTP and RDP sessions	-	✓
Can replay HTTP sessions	-	Only exports raw files from the command line
Can replay X11 sessions	✓	✓
Can start replay while rendering is in progress	-	✓
Can follow 4-eyes connections	-	✓
Can replay live streams in follow mode	-	✓
Can export to PCAP	-	✓
Can display user input	✓	✓
Can display subtitles for video	-	✓
Export audit trail as video	-	✓
Export screen content text	-	✓
Can search in the contents of the audit trails	-	✓

For details on the Safeguard Desktop Player application, see Safeguard Desktop Player User Guide.

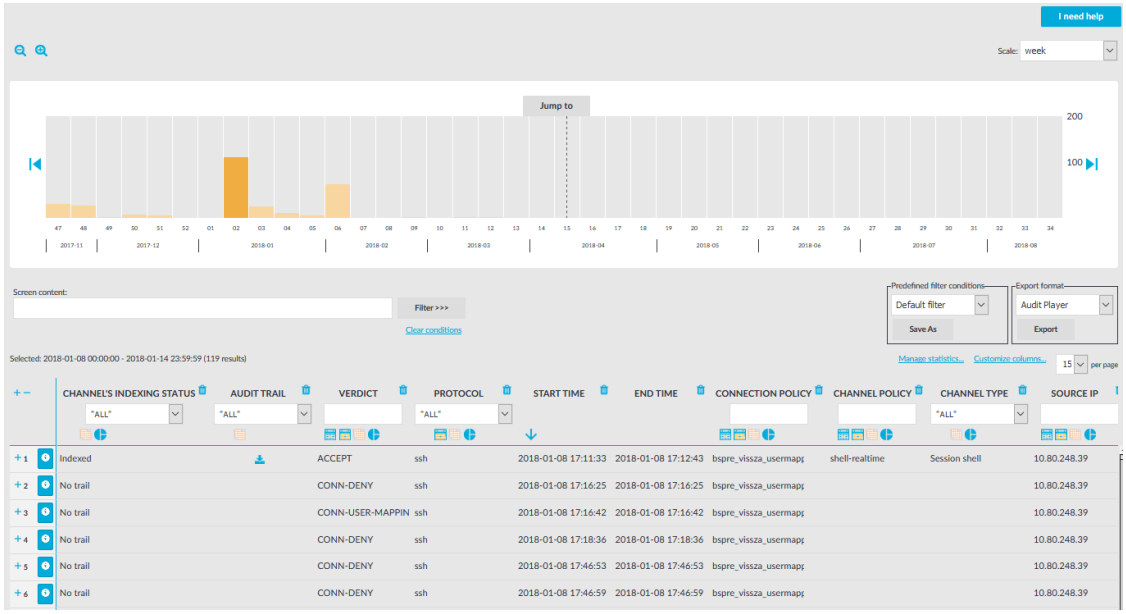
CAUTION:

Even though the One Identity Safeguard for Privileged Sessions (SPS) web interface supports Internet Explorer and Microsoft Edge in general, to replay audit trails you need to use Internet Explorer 11, and install the Google WebM Video for Microsoft Internet Explorer plugin. If you cannot install Internet Explorer 11 or another supported browser on your computer, use the the Safeguard Desktop Player application. For details, see "Replaying audit trails in your browser" in the Administration Guide and Safeguard Desktop Player User Guide.

To replay an audit trail in your browser

1. On the **Search** page, find the audit trail you want to replay.

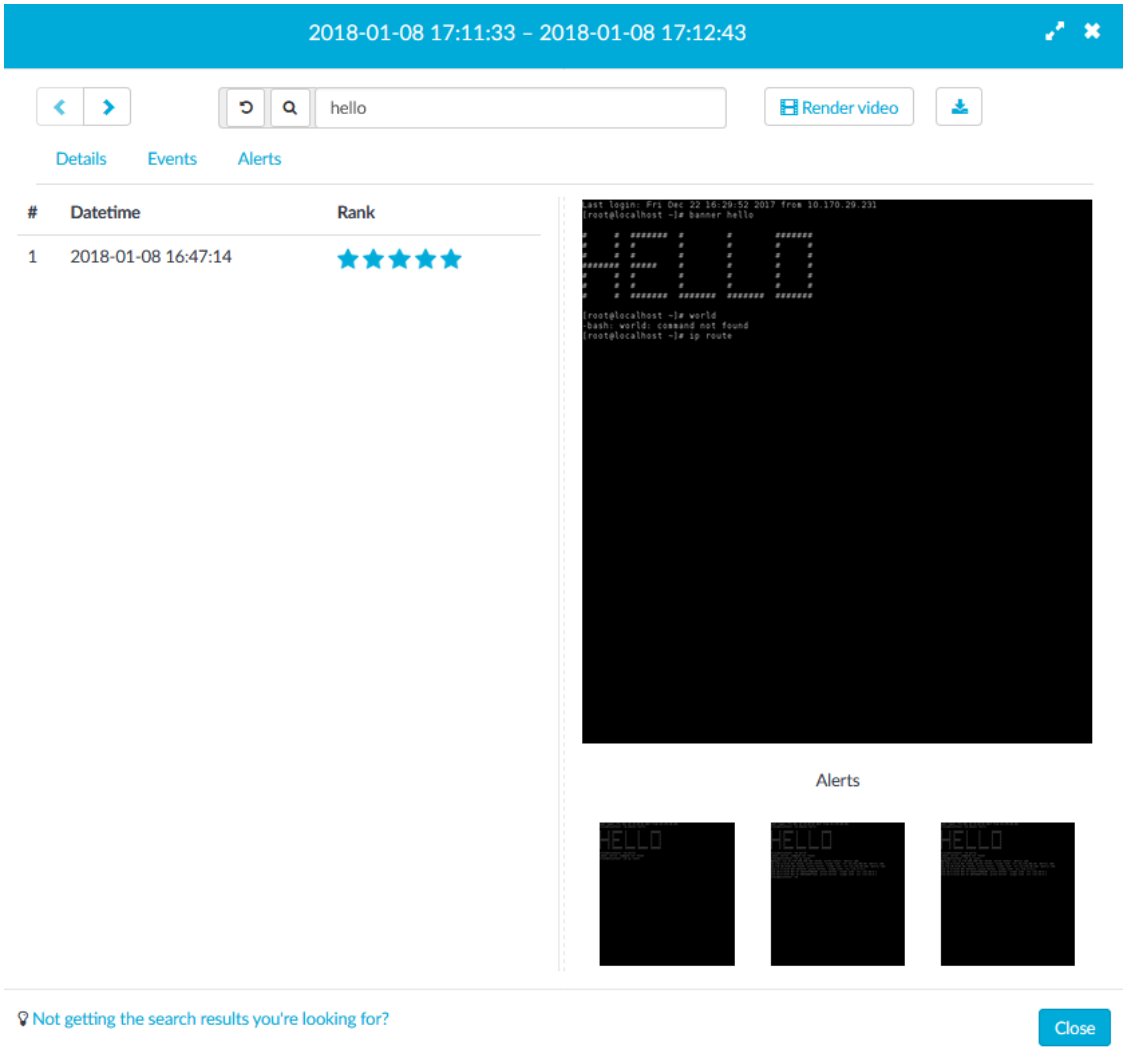
Figure 318: Search — Browse the connections database



2. (Optional) To replay encrypted audit trails, upload your permanent or temporary keys to the **User menu > Private keystore**. For more information, see Replaying encrypted audit trails in your browser on page 724.
3. Click to display the details of the connection.
4. Click Render video to generate a video file from the audit trail that you can replay. Depending on the load of the indexer and the length and type of the audit trail, this can take several minutes (to cancel processing the audit trail, click). The **Video status** field shows the progress of the this process.

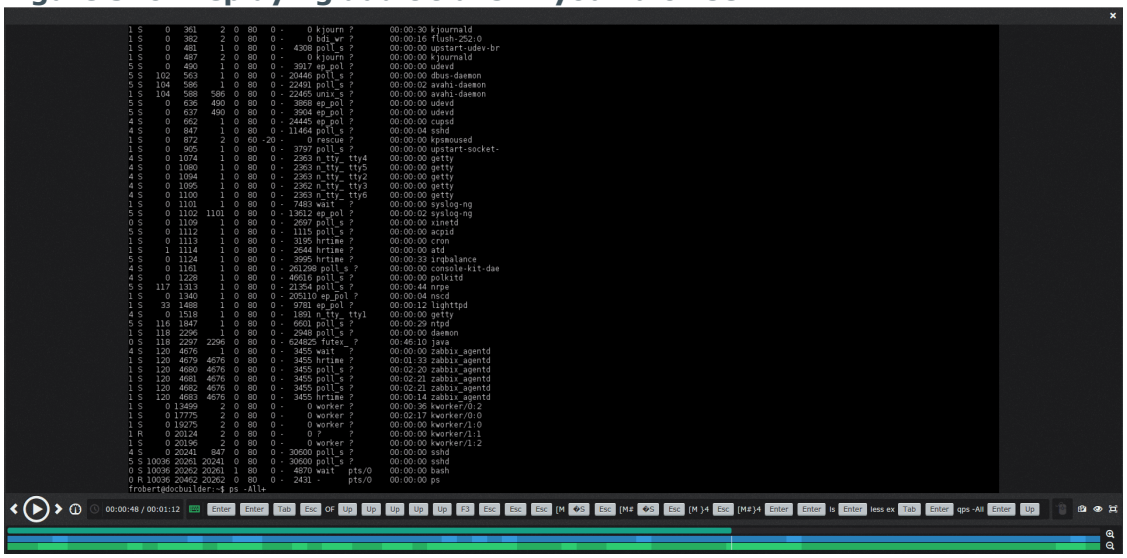
   When the video is available, Render video changes to .

Figure 319: Search — Audit trail details



5. To replay the video, click ▶. The Player window opens.
6. The Player window has the following controls.

Figure 320: Replaying audit trails in your browser



a.  : *Play*, *Pause*

b.  : *Jump to previous event*, *Jump to next event*

c.  : *Adjust replay speed*

d. 

: *Time since the audit trail started / Length of the audit trail*. Click on the time to show the date (timestamp) of the audit trail.

e. 

: *List of keyboard events*. Special characters like ENTER, F1, and so on are displayed as buttons. If the upstream traffic is encrypted, upload your permanent or temporary keys to the **User menu > Private keystore** to display the keyboard events.

f.  : *Active mouse button*

g.  : *Create a screenshot*

h.  : *Show / hide events*. Select the types of events to display. Depending on the protocol used and how the audit trail was processed, SPS can display keyboard events, commands, mouse events, and window titles. Commands and window titles are displayed as subtitles at the top of the screen.

i.  : *Fullscreen mode*

j.  : Progress bar

k.  : Shows the distribution of events. Blue - commands, green - keyboard events, yellow - mouse events, orange - window title.

l.  : *Close the player*, and return to the Connection details page.

Replaying encrypted audit trails in your browser

To view screenshots generated for encrypted audit trails, and replay encrypted audit trails in your browser, you have to upload the necessary certificates and corresponding private keys to your private keystore. Depending on the encryption, decrypting the upstream part of an audit trail might require an additional set of certificates and keys.

Only RSA keys (in PEM-encoded X.509 certificates) can be uploaded to the private keystore.

NOTE:

Certificates are used as a container and delivery mechanism. For encryption and decryption, only the keys are used.

One Identity recommends using 2048-bit RSA keys (or stronger).

For more information on audit trail encryption, see Encrypting audit trails on page 465.

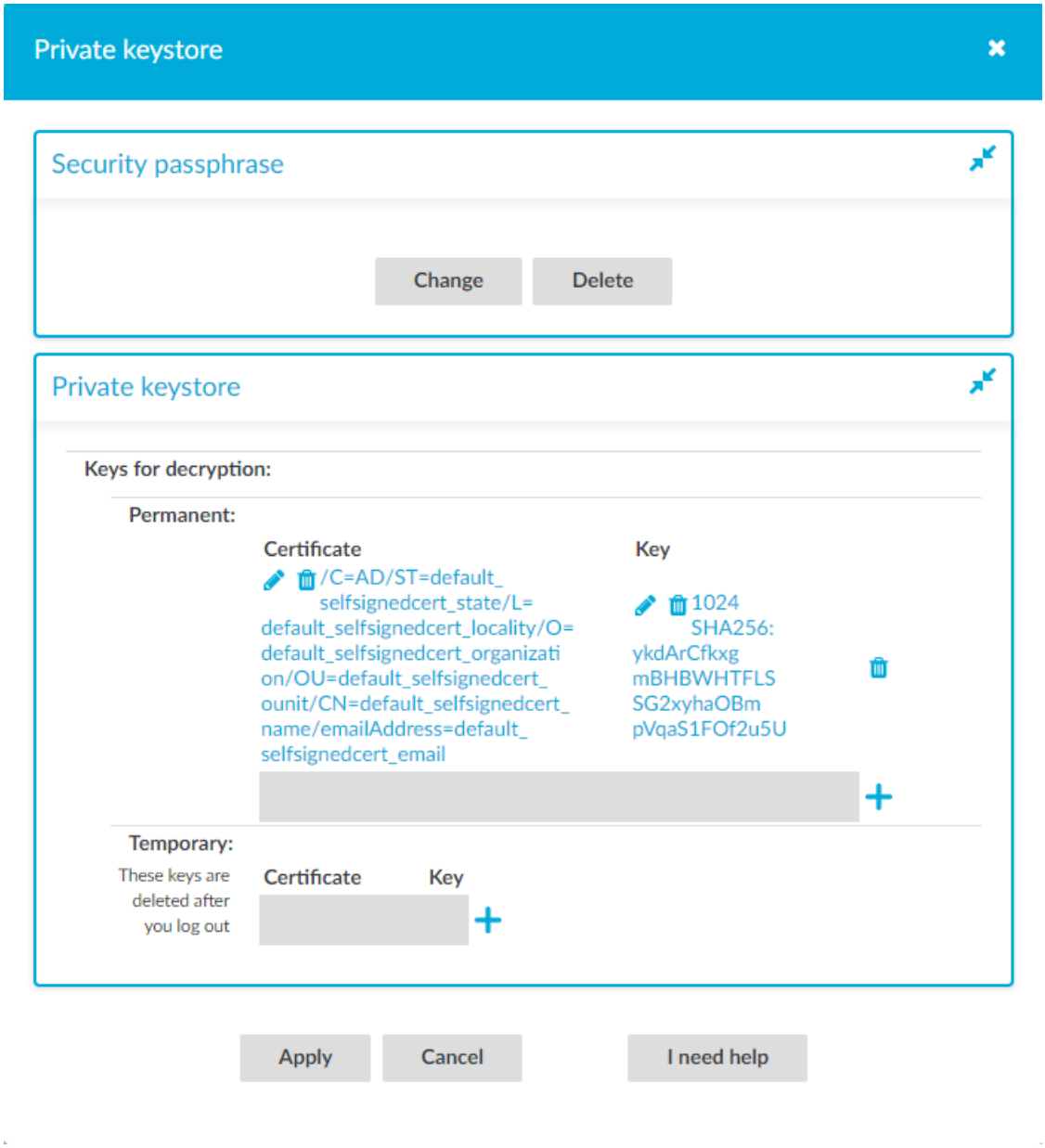
You can upload certificates permanently or temporarily. The temporary certificates are deleted when you log out of One Identity Safeguard for Privileged Sessions (SPS).

The certificates and private keys in your keystore can be protected with a passphrase. To use the certificates and private keys in a passphrase-protected keystore for decrypting audit trails, you have to unlock the keystore first by providing the security passphrase. The keystore then remains unlocked for the duration of your session.

To replay encrypted audit trails in your browser

1. Click on **User menu > Private keystore**.

Figure 321: User menu > Private keystore — The private keystore



2. (Optional) Create a security passphrase, if you have not configured one yet.
   a. In **Security passphrase**, click **Change**.
   b. In the New field, enter your new security passphrase. Repeat the same

passphrase in the **Confirm** field.

> NOTE:
>
> One Identity Safeguard for Privileged Sessions (SPS) accepts passwords that are not longer than 150 characters. The following special characters can be used: `!"#$%&'()*+,-./:;<=>?@[\]^-`{|}`

c. Click **Apply**.

If you forgot your security passphrase, contact our Support Team.

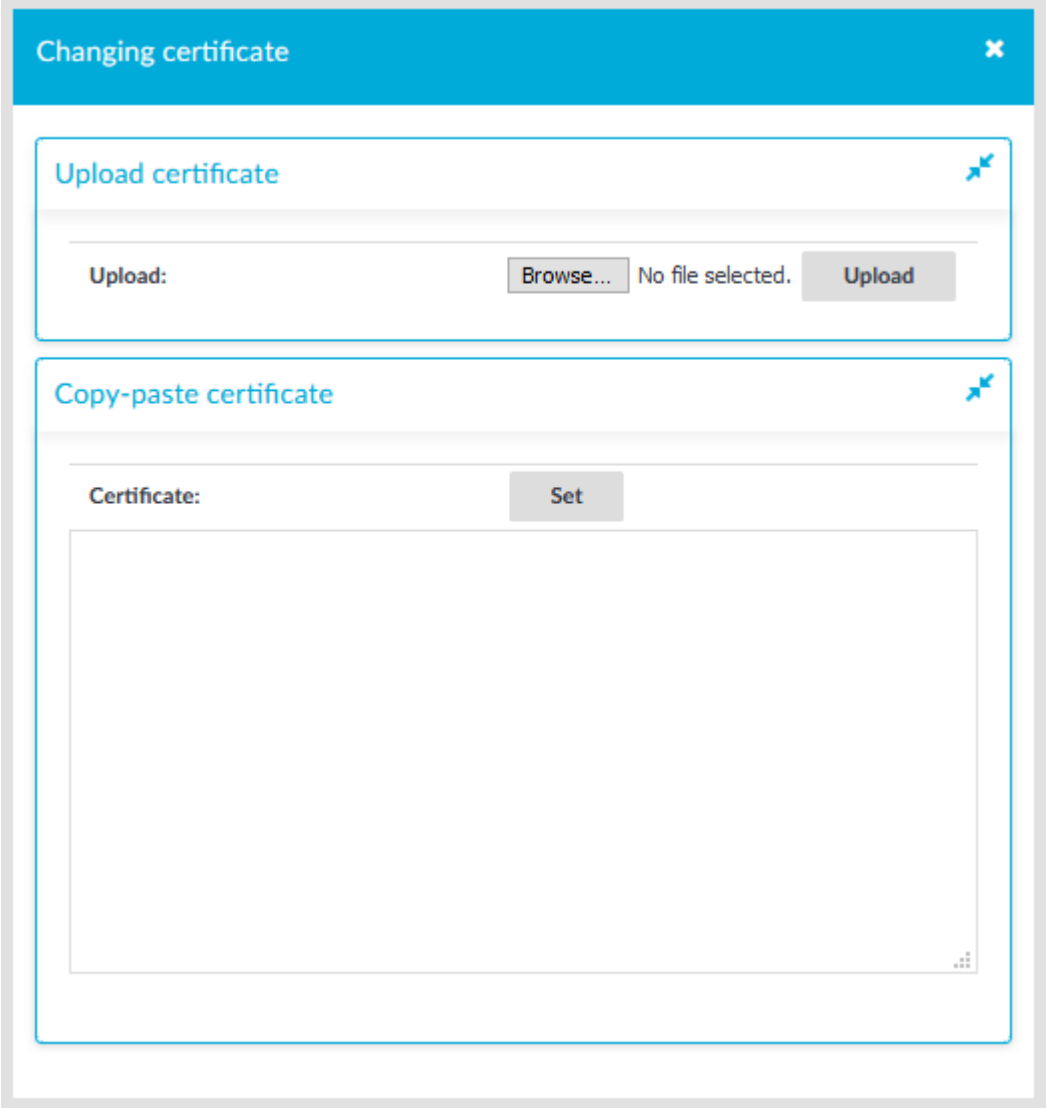
3. Click + to add a new certificate.

Figure 322: Adding certificates



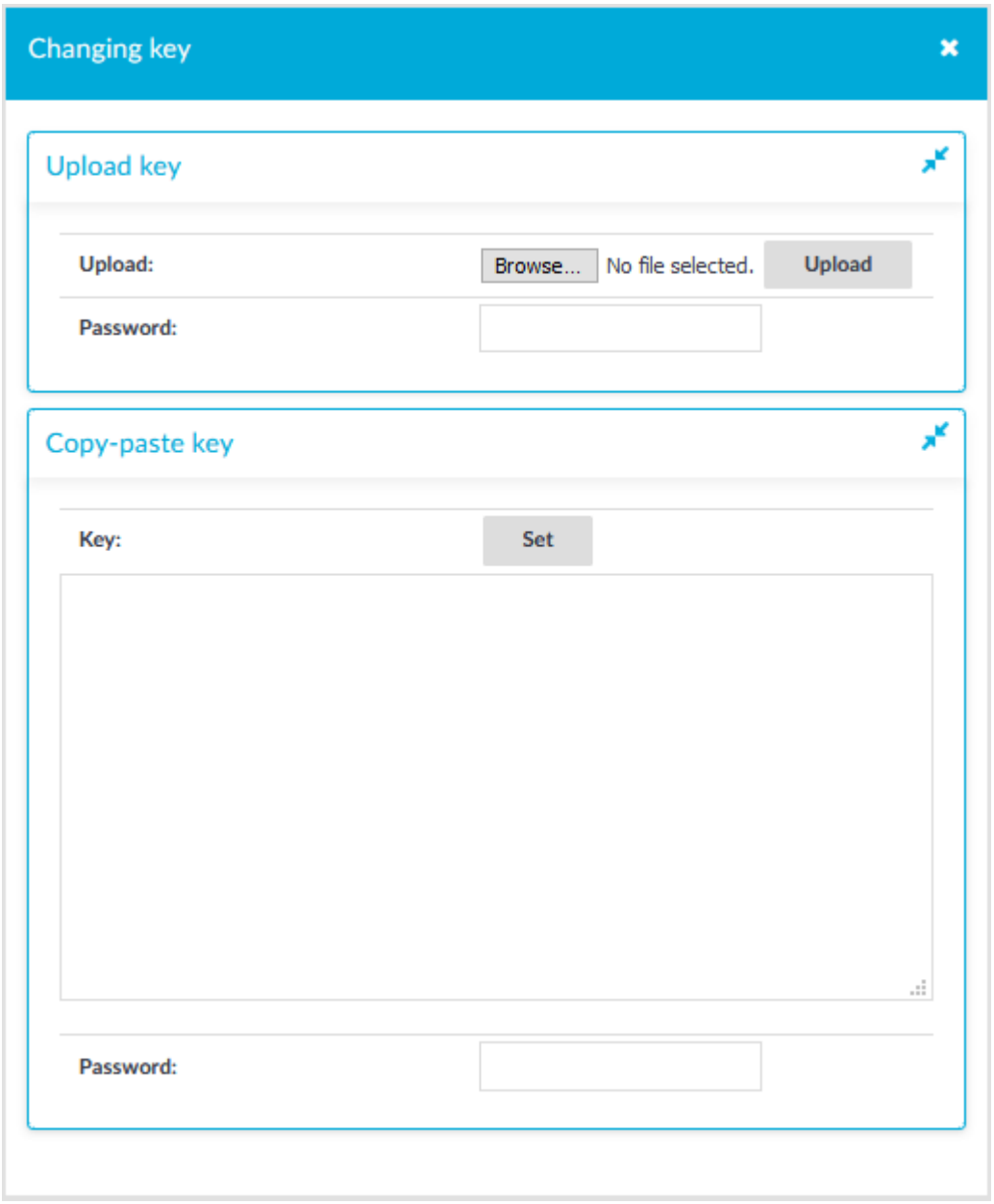
4. Click the first to upload the new certificate. A pop-up window is displayed.

Figure 323: Uploading certificates



5. Select **Browse**, select the file containing the certificate, and click **Upload**. Alternatively, you can also copy-paste the certificate into the **Certificate** field and click **Set**.
6. To upload the private key corresponding to the certificate, click the second icon. A pop-up window is displayed.

Figure 324: Uploading the private key



7. Select **Browse**, select the file containing the private key, provide the **Password** if the key is password-protected, and click **Upload**. Alternatively, you can also copy-paste the private key into the **Key** field, provide the **Password** there, and click **Set**.
8. To add more certificate-key pairs, click + and repeat the steps above.
9. To finish uploading certificates and keys to your private keystore, click **Apply**.

Using the content search

To most effectively search in the contents of the audit trails, make sure that the following prerequisites are met:

- Indexing was enabled in the connection policy related to the audit trail during the session, and
- the audit trail has already been indexed.

You can use the following in content search:

- wildcards
- boolean expressions
- search in the commands of terminal connections (for example, `command:"sudo su"`)
- search in the window titles of graphical connections (for example, `title:settings`)

The following sections provide examples for different search queries.

- For examples of exact matches, see Searching for exact matches on page 951.
- For examples of using boolean operators to combine search keywords, see Combining search keywords on page 952.
- For examples of wildcard searches, see Using wildcard searches on page 953.
- For examples of searching with special characters, see Searching for special characters on page 955.
- For examples of fuzzy search that finds words with similar spelling, see Searching for fuzzy matches on page 957.
- For examples of proximity search to find words that appear within a special distance, see Proximity search on page 957.
- For examples of adjusting the relevance of a search term, see Adjusting the relevance of search terms on page 957.

For details on how to use more complex keyphrases that are not covered in this guide, see the Apache Lucene documentation.

Searching for exact matches

By default, One Identity Safeguard for Privileged Sessions (SPS) searches for keywords as whole words and returns only exact matches. Note that if your search keywords include special characters, you must escape them with a backslash (\) character. For details on special characters, see Searching for special characters on page 955. The following characters are special characters: `+ - & | ! ( ) { } [ ] ^ " ~ * ? : \ /`

Example: Searching for exact matches

Search expression	`example`
Matches	example
Does not match	examples example.com query-by-example exam

To search for an exact phrase, enclose the search keywords in double quotes.

Search expression	`"example command"`
Matches	example command
Does not match	example command example: command

To search for a string that includes a backslash characters, for example, a Windows path, use two backslashes (`\\`).

Search expression	`C\:\\Windows`
Matches	C:\Windows

Combining search keywords

You can use boolean operators – `AND`, `OR`, `NOT`, and + (required), – to combine search keywords. More complex search expressions can also be constructed with parentheses. If you enter multiple keywords,

Example: Combining keywords in search

Search expression	`keyword1 AND keyword2`
Matches	(returns hits that contain both keywords)

Search expression	`keyword1 OR keyword2`
Matches	(returns hits that contain at least one of the keywords)

Search expression	`"keyword1 keyword2" NOT "keyword2 keyword3"`
Matches	(returns hits that contain the first phrase, but not the second)

Search expression	`+keyword1 keyword2`
Matches	(returns hits that contain keyword1, and may contain keyword2)

To search for expressions that can be interpreted as boolean operators (for example: `AND`), use the following format: `"AND"`.

Example: Using parentheses in search

Use parentheses to create more complex search expressions:

Search expression	`(keyword1 OR keyword2) AND keyword3`
Matches	(returns hits that contain either keyword1 and keyword3, or keyword2 and keyword3)

Using wildcard searches

You can use the `?` and `*` wildcards in your search expressions.

Example: Using wildcard ? in search

The `?` (question mark) wildcard means exactly one arbitrary character. Note that it does not work for finding non-UTF-8 or multibyte characters. If you want to search for these characters, the expression `??` might work, or you can use the `*` wildcard instead.

You cannot use a * or ? symbol as the first character of a search.

Search expression	`example?`
Matches	example1 examples example?
Does not match	example.com example12 query-by-example

Search expression	`example??`
Matches	example12
Does not match	example.com example1 query-by-example

Example: Using wildcard * in search

The * wildcard means 0 or more arbitrary characters. It finds non-UTF-8 and multibyte characters as well.

Search expression	`example*`
Matches	example examples example.com
Does not match	query-by-example example*

Example: Using combined wildcards in search

Wildcard characters can be combined.

Search expression	`ex?mple*`
Matches	example1 examples example.com exemple.com example12
Does not match	exmples query-by-example

Searching for special characters

To search for the special characters, for example, question mark (?), asterisk (*), backslash (\) or whitespace ( ) characters, you must prefix these characters with a backslash (\). Any character after a backslash is handled as character to be searched for. The following characters are special characters: `+ - & | ! ( ) { } [ ] ^ " ~ * ? : \ /`

Example: Searching for special characters

To search for a special character, use a backslash (\).

Search expression	`example\?`
Matches	example?
Does not match	examples example1

To search for a string that includes a backslash characters, for example, a Windows path, use two backslashes (\\).

Search expression	`C\:\\Windows`
Matches	C:\Windows

To search for a string that includes a slash character, for example, a UNIX path, you must escape the every slash with a backslash (\/).

Search expression	`\/var\/log\/messages`
Matches	/var/log/messages

Search expression	`\(1\+1\)\:2`
Matches	(1+1):2

Searching in commands and window titles

For terminal connections, use the `command:` prefix to search only in the commands (excluding screen content). For graphical connections, use the `title:` prefix to search only in the window titles (excluding screen content). To exclude search results that are commands or window titles, use the following format: `keyword AND NOT title:[* TO *]`.

You can also combine these search filters with other expressions and wildcards, for example, `title:properties AND gateway`.

Example: Searching in commands and window titles

Search expression	`command:"sudo su"`
Matches	`sudo su` as a terminal command
Does not match	`sudo su` in general screen content

Search expression	`title:settings`
Matches	`settings` appearing in the title of an active window
Does not match	`settings` in general screen content

To find an expression in the screen content and exclude search results from the commands or window titles, see the following example.

Search expression	`properties AND NOT title:[* TO *]`
Matches	`properties` appearing in the screen content, but not as a window title.
Does not match	`properties` in window titles.

You can also combine these search filters with other expressions and wildcards.

Search expression	`title:properties AND gateway`
Matches	A screen where `properties` appears in the window title, and `gateway` in the screen content (or as part of the window title).
Does not match	Screens where both `properties` and `gateway` appear, but `properties` is not in the window title.

Searching for fuzzy matches

Fuzzy search uses the tilde ~ symbol at the end of a single keyword to find hits that contain words with similar spelling to the keyword.

Example: Searching for fuzzy matches

Search expression	`roam~`
Matches	roams foam

Proximity search

Proximity search uses the tilde ~ symbol at the end of a phrase to find keywords from the phrase that are within the specified distance from each other.

Example: Proximity search

Search expression	`"keyword1 keyword2"~10`
Matches	(returns hits that contain keyword1 and keyword2 within 10 words from each other)

Adjusting the relevance of search terms

By default, every keyword or phrase of a search expression is treated as equal. Use the caret ^ symbol to make a keyword or expression more important than the others.

Example: Adjusting the relevance of search terms

Search expression	`keyword1^4 keyword2`
Matches	(returns hits that contain keyword1 and keyword2, but keyword1 is 4-times more relevant)
Search expression	`"keyword1 keyword2"^5 "keyword3 keyword4"`
Matches	(returns hits that contain keyword1 keyword2 and keyword3 keyword4, but keyword1 keyword2 is 5-times more relevant)

Connection metadata

One Identity Safeguard for Privileged Sessions (SPS) stores the following parameters about the connections:

- **Additional metadata**: Data about the connection recorded by the different plugins of SPS, for example, when using an Authentication and Authorization plugin.
- **Alerting**: The list of content policy alerts triggered in the connection. For every alert, the following information is displayed:
  - *Time of alert*: Date and time of the alert
  - *Alerting type*: The type of the event (command, screen content, and so on).
  - *Matched rule value*: The expression that matched the content.
  - *Matched context*: Click this column to display the context of the matched content, for example, the contents of the screen, or the command line. The value that triggered the alert is highlighted.

  For example, a content policy that detects every execution of the `sudo` command in SSH commands, creates the following entry: `2012-10-05 15:46:17.902004: (adp.event.command) 'sudo'`
- **Application**: The name of the application accessed in a seamless Citrix ICA connection.
- **Archive date**: The date when the connection was archived or cleaned up.
- **Archive path**: The path where the audit trail was archived on the remote server.
- **Archive server**: The hostname or IP address of the remote server where the audit trail was archived.
- **Audit trail downloads**: An audit trail has been downloaded.

- **Audit-trail**: Name and ID of the audit file storing the traffic of the channel. If the session has an audit trail, a  icon is displayed. Note that a the following letters may appear on the download icon:
  - *C*: The audit trail has been cleaned up and is not available any more.
  - *A*: The audit trail has been archived. SPS will try to retrieve it from the archive server.
  - *X*: The audit trail is not available for some reason.

  You can filter the **Audit-trail** column for the following values:
  - **no audit trail**: Channels that have no audit trails.
  - **has audit trail**: Channels that have audit trails.
  - **online**: Channels that belong to an active, ongoing connection. If you are auditing every connection, then this list shows the connections also shown on the **Active Connections** page.
  - **archived**: Channels that had their audit trails archived to a remote server, but SPS cannot access the audit trail.
- **Authentication method**: The authentication method used in the connection. For example, `password`
- **Channel policy**: The Channel policy applied to connection. The Channel policy lists the channels (for example, terminal session and SCP in SSH, or Drawing and Clipboard in RDP) that can be used in the connection, and also determines if the channel is audited or not. The Channel policy can also restrict access to each channel based on the IP address of the client or the server, a user list, user group, or a time policy.
- **Channel type**: Type of the channel.
- **Channel's indexing status**: Shows if the channel has been indexed. The following values are possible:
  - `CHANNEL_OPEN (0)`: The connection of the channel is still open (indexer is waiting for the connection to close).
  - `NOT_INDEXED (1)`: All channels of the connection have been closed which belong to the connection. The channel is ready for indexing, unless the audit trail was placed in the `skipped_connections` queue.
  - `INDEXING_IN_PROGRESS (2)`: The channel is being indexed (indexing in progress). Note that SPS will return search results for the parts of the channel are already indexed.
  - `INDEXED (3)`: Indexing the channel is complete.
  - `INDEXING_NOT_REQUIRED (4)`: Indexing not required (indexing is not enabled for the connection).
  - `INDEXING_FAILED (5)`: Indexing failed. The indexer service writes the corresponding error message in the `error_message` column of the `indexer_jobs` table. Note that SPS will return search results for the parts of the channel that

were successfully indexed before the error occurred. For example, if the error occurred at the end of a long audit trail, you can still search for content from the first part of the audit trail.

  - `NO_TRAIL (6)`: Auditing is not enabled for the channel.

- **Client X.509 Subject**: The client's certificate in TELNET or VNC sessions. Available only if the **Client-side transport security settings > Peer certificate validation** option is enabled in SPS.
- **Connection policy ID**: The identifier of the connection policy.
- **Connection policy**: The connection policy that handled the client's connection request.
- **Destination IP**: The IP address of the server as requested by the client.
- **Destination port**: The port number of the server as requested by the client.
- **Device name**: The name or ID of the shared device (redirect) used in the RDP connection.
- **Duration**: The length of the session.
- **Dynamic channel**: The name or ID of the dynamic channel opened in the RDP session.
- **End time**: Date when the channel was closed.
- **Events**: A table that shows the commands that the user issued in a terminal session. These commands are searchable, together with the command prompt itself. Available only for Telnet and SSH session shell connections, if the audit trail has been indexed. For details on configuring indexing, see Indexing audit trails on page 592.
- **Exec command**: The command executed in a **Session exec** channel.
- **File operations**: The list of file operations (for example, file upload, create directory) performed by the client. Available only for SCP and SFTP sessions (**Session exec SCP** and **Session SFTP** SSH channels) if the **Log file transfers to database** option is enabled in the Channel Policy of the connection.

  For both SCP and SFTP connections, the filename is stored in a human-readable way if it only includes UTF-8-encoded characters. If the filename is not a valid UTF-8-encoded filename, the non-ASCII characters are translated into their hexadecimal equivalents. In both cases, the asterisk (*) characters are escaped with another asterisk (*) character.

  ### In case of a valid UTF-8-encoded filename:

  - *Filename:* **ár*víz****
  - *Hexadecimal sequence:* **C3 A1 72 2A 76 C3 AD 7A 2A 2A**
  - *Stored in connection database as:* **ár**víz****

In case of a not valid UTF-8-encoded filename:

- *Filename:* **ár*víz**** in ISO-8859-2
- *Hexadecimal sequence:* **E1 72 2A 76 ED 7A 2A 2A**
- *Stored in connection database as:* ***e1r**v*edz****

NOTE:

For SFTP connections, this field includes the path and the filename. For SCP connections, it includes only the filename, the path is available in the **SCP Path** field.

Windows and UNIX systems use different separator characters in the pathname, backslash (\) and slash (/), respectively. As the SCP and SFTP protocols do not specify the separator character used, SPS uses slash (/), for example, `/var/log/messages`.

TIP:

Use the filter in the header of the column to find sessions containing a specific file (for example, enter `.gz` to list sessions that accessed files with the .gz extension). Note that currently it is possible to search only in the filename and path, and not in the changed privileges.

- **Four-eyes authorizer**: The username of the user who authorized the session. Available only if 4-eyes authorization is required for the channel. For details on 4-eyes authorization, see Configuring four-eyes authorization on page 752.
- **Four-eyes description**: The description submitted by the authorizer of the session.
- **Hits and rank**: Available only for indexed trails, when searching the content of the audit trails. This column is displayed automatically.
  - For channels indexed with the indexer service, displays the number of hits (search results) found in the audit trail, and the rank (relevance) of the audit trail regarding the search keywords. **Rank** is displayed as 1-5 stars. **Hits** is returned as a number for up to 100 results in the audit trail — the exact number of hits is not displayed if it is higher than 100.
  - For channels indexed with the Audit Player indexer service, rank is not available. The exact number of hits is not displayed, only that the search keywords were found in the trail at least once. In this case, the **Hits and rank** column displays **1+**.

  Note that because of performance reasons, the number of hits can be inaccurate and is only an approximation. In this case, SPS displays a tilde (~) sign to mark approximated hit counts.
- **Port-forward target IP**: The traffic was forwarded to this IP address in **Remote Forward** and **Local Forward** channels.
- **Port-forward target port**: The traffic was forwarded to this port in **Remote Forward** and **Local Forward** channels.

- **Port/X11 forward originator IP**: The IP address of the host initiating the channel in **Remote Forward** and **Local Forward** channels. Note that this host is not necessarily the client or the server of the SSH connection.
- **Port/X11 forward originator port**: The number of the forwarded port in **Remote Forward** and **Local Forward** channels.
- **Protocol**: The protocol used in the connection (Citrix ICA, HTTP, RDP, SSH, Telnet, or VNC).
- **Rule number**: The number of the line in the Channel policy applied to the channel.
- **SCP path**: Name and path of the file copied via SCP. Available only for SCP sessions (**Session exec SCP** SSH channels) if the **Log file transfers to database** option is enabled in the Channel Policy of the connection.

  NOTE:

  This field includes only the path, the filename is available in the **File Operations** field.

  Windows and UNIX systems use different separator characters in the pathname, backslash (\) and slash (/), respectively. As the SCP and SFTP protocols do not specify the separator character used, SPS uses slash (/), for example, `/var/log/messages`.
- **Server IP**: The IP address of the server connected by SPS.

  NOTE:

  In case of HTTP, this is the target IP of the first request of the session, since it cannot be guaranteed that all page content come from the same server.
- **Server-local IP**: The IP address of SPS used in the server-side connection.
- **Server-local port**: The port number of SPS used in the server-side connection.
- **Server port**: The port number of the server connected by SPS.
- **Session ID**:

  A globally unique string that identifies the session. This session ID has the following format: *svc/<unique-random-hash>/<name-of-the-connection-policy>:<session-number-since-service-started>/<protocol>*, for example, *svc/5tmEaM7xdNi1oscgVWpbZx/ssh_console:1/ssh*.

  Log messages related to the session also contain this ID. For example:

```
2015-03-20T14:29:15+01:00 demo.example
zorp/scb_ssh[5594]: scb.audit(4):
(svc/5tmEaM7xdNi1oscgVWpbZx/ssh_console:0/ssh):
Closing connection; connection='ssh_console',
protocol='ssh', connection_id='409829754550c1c7a27e7d',
```

```
src_ip='10.40.0.28', src_port='39183',
server_ip='10.10.20.35', server_port='22',
gateway_username='', remote_username='example-username',
verdict='ZV_ACCEPT'
```

- **Source IP**: The IP address of the client.
- **Source port**: The port number of the client.
- **Start time**: Date when the channel was started.
- **Subsystem name**: Name of the SSH subsystem used in the channel.
- **URLs**: The list of URLs accessed in an HTTP session (the list is displayed in a pop-up window).
- **Unique connection ID**: The unique identifier of the connection.
- **Username**: The username used in the session.
    - If the user performed inband gateway authentication in the connection, the field contains the username from the gateway authentication (gateway username).
    - Otherwise, the field contains the username used on the remote server.
- **Username on server**: The username used to log in to the remote server. This username can differ from the client-side username if usermapping is used in the connection. For details on usermapping, see Configuring usermapping policies on page 740.
- **Verdict**: Indicates what SPS decided about the channel.
    - `ACCEPT`: Accepted.
    - `ACCEPT-TERMINATED`: Connection was accepted and established, but a content policy terminated the connection. For details on content policies, see Real-time content monitoring with Content Policies on page 451.
    - `CONN-AUTH-FAIL`: User authentication failed.
    - `CONN-DENY`: Connection rejected.
    - `CONN-FAIL`: Connection failed, that is, it was allowed to pass SPS but timed out on the server.
    - `CONN-GW-AUTH-FAIL`: Gatway authentication failed.
    - `CONN-KEY-ERROR`: Hostkey mismatch.
    - `CONN-USER-MAPPING-FAIL`: Usermapping failed.
    - `DENY`: Denied.
    - `FOUR-EYES-DEFERRED`: Waiting for remote username.
    - `FOUR-EYES-ERROR`: Internal error during four-eyes authorization.
    - `FOUR-EYES-REJECT`: Four-eyes authorization rejected.
    - `FOUR-EYES-TIMEOUT`: Four-eyes authorization timed out.

NOTE:

The **Verdict** column only accepts capital letters.

NOTE:

The rejected (`CONN-DENY`) HTTP requests are collected into a session, to avoid having too many entries in the database (for example when the user visits a forbidden page, and reloads the page several times, only one session will be visible instead of all the separate requests). The denied sessions have timeout and session ID.

Using and managing search filters

- To filter the search results, set the filters you need and click **Filter**.
- To apply a predefined filter, select the filter from the **Predefined filter conditions** field.
- To create and save a filter, complete Creating and saving filters for later use on page 964. Note that filters cannot be modified, only deleted.
- To delete a predefined filter, select the filter from the **Predefined filter conditions** field and click **Delete**.

  NOTE:

  You need the **Manage global filters** privilege to delete global filters. For more information on managing user rights, see Managing user rights and usergroups on page 320.

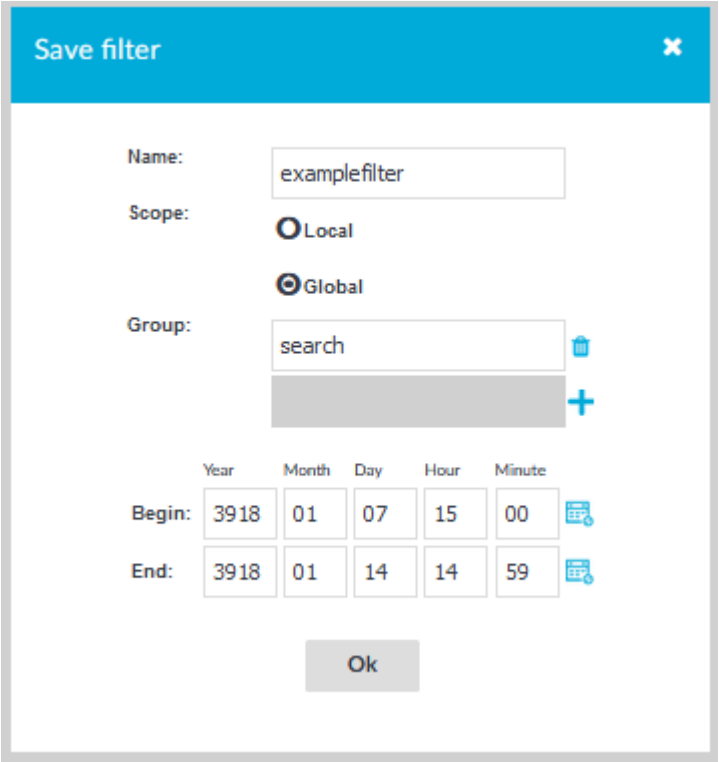
Creating and saving filters for later use

The following describes how to create and save a filter for later use.

To create and save a filter for later use

1. Navigate to the **Search** page.
2. Set the filters you need.
3. Select **Predefined filter conditions > Save As**. A pop-up window is displayed.
4. Enter a name for the filter into the **Name** field.

Figure 325: Search > Save as — Saving filter conditions



5. If you want the filter to be available for other One Identity Safeguard for Privileged Sessions (SPS) users as well, select **Global**. To restrict the availability of the filter to a set of specific users, select **Scope > Global**, click +, and enter the name of the group whose members may use the filter. Repeat this step to add other groups if needed. **Local** filters are visible only for you.

   NOTE:

   Filters cannot be modified later, only deleted. A filter can be deleted by the user who created it, and by users whose group has the **Search > Manage global filters** privilege.

   For more information on managing user rights, see Managing user rights and usergroups on page 320.

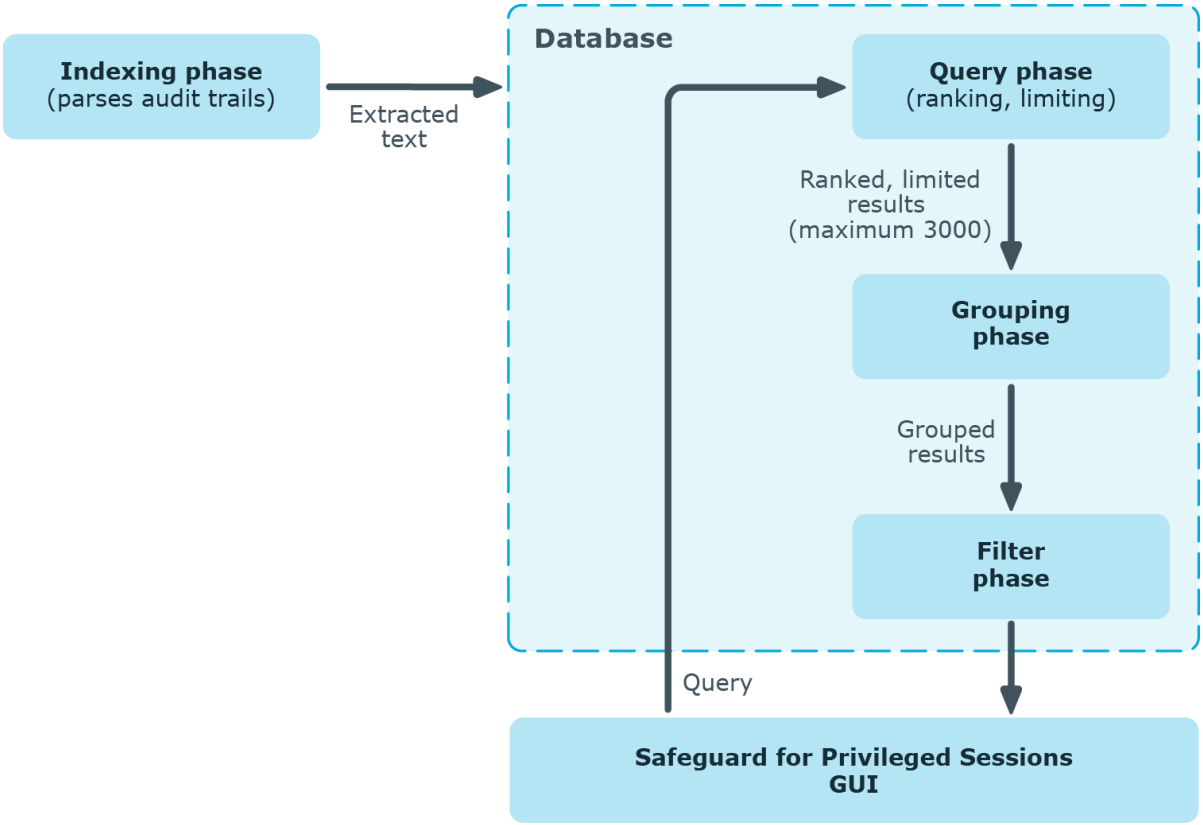
6. To modify the timeframe of the search, select **Interval**, and set the beginning and ending date and time of the search. This is useful when you want to display only the connections of a specific event. Note that you must always set an interval for global filters.
7. Click **OK**.

The search and filter process

The screen content is first indexed, then processed with the search backend, and finally, the filter expressions are applied. This process is described in detail in the following

sections.

Figure 326: The search and filter process



Prerequisites - Indexing phase

First, as a prerequisite of the search process, screen content is indexed. The indexing phase generates a database that the search and filter processes will run on.

The indexer parses the audit trail files, and builds an "inventory" of the privileged user's activity data based on what appeared on their screen.

1. In the case of a terminal session, screen content corresponds to the activity data that is captured in a terminal window. In the case of graphical protocols, screen content is whatever is visible in the graphical user interface of the applications the user is interacting with. In the latter case, the indexer's Optical Character Recognition (OCR) engine extracts text that appeared on the screen (for example, window titles).

   NOTE:

   If a piece of text is displayed for less than 1 second, it is not extracted.

2. The indexer returns the information extracted from the parsed audit trail files to One Identity Safeguard for Privileged Sessions (SPS). In the case of a terminal session, the captured text is put in the backend database as one document per one second of

screen content. Because of this, the content that you have searched for might only partially appear in the screenshot. In the case of graphical protocols, the captured text is put in the backend database as one document per screenshot.

3. The queries will be run on this database during the search process.

For details on indexing, see Indexing audit trails on page 592.

Search and filter process phases

The search and filter process consists of three major phases:

- Query phase
- Grouping phase
- Filter phase

Query phase

In the query phase, the backend ranks and then limits the number of results.

1. The result of one query is the top 3000 documents, ordered by the default ranking system of the backend.

   This means that if there are more than 3000 results, those of the lowest rank will not be passed to the next phase at all.

   The ranking system cannot be modified, so there is no way to "upvote" those results of lower ranks.

   If you want to ensure that all important results are passed to the grouping phase, use a smaller time range that you run the query on. If there are fewer than 3000 results, it is certain that the events you are interested in will be included in the grouping phase.
2. The grouping phase receives the results.

Grouping phase

The grouping phase groups the results that were passed on from the query phase.

1. First, the results with the same trail IDs are grouped together. A trail ID group contains all search hits that are in that trail.
2. The trail ID groups are then further grouped by seach expression and time range. This group is essentially the time range during which the expression is displayed on the screen (for example, if the text `root` is displayed from 00:00:12 to 00:01:45, this will be one group).
3. This grouped result is displayed in the search screen as one row.

Filter phase

The filter phase applies filter expressions to these grouped results.

NOTE:

If there were screen content search results that were excluded during the query phase, the filter expressions will not be applied to them.

Example: Filtering for search results that were excluded in the query phase

For example, if you want to filter for Telnet connections where the text `root` was displayed, the following can happen:

You search for the **Screen content**: `root`. There are 3100 search results that consist of 3050 SSH connections and 50 Telnet connections. In this example, Telnet connections received the lowest ranks for some reason. 100 results that have received the lowest rank are excluded, and in this example it means all Telnet connections.

If you filter for protocol `Telnet` now, you will not see any results.

To remedy this situation, try searching in a smaller time range to make sure that there are less than 3000 search results. If you are unsure about the time range, you might want to attempt fine-tuning the backend search manually. For details, see: Fine-tuning the backend search manually on page 968.

Fine-tuning the backend search manually

You can fine-tune your search manually with the command line utility **lucenectl**. To do this, log on to the core shell. For details, see Accessing the One Identity Safeguard for Privileged Sessions (SPS) console on page 389.

- Specify more exact time ranges (use Unix timestamps).

  For example, to limit the time range to Thursday, June 30, 2016 11:39:51 AM - Thursday, November 3, 2016 2:44:46 PM, enter the following command:

  ```
  lucenectl search --from-to 1467286791 1478184286 --text remote --limit
  3000 --aggregate-by-trail --normalize-rank
  ```

  NOTE:

  For converting timestamps to Unix timestamp, use https://www.epochconverter.com/.

- Increase the query limit of 3000 to a limit of your choice.

  For example, to increase the query limit of 3000 to 4500, enter the following command:

```
lucenectl search --from-to 1467286791 1478184286 --text <your-screen-content-search-expression> --limit 4500 --aggregate-by-trail --normalize-rank
```

```
lucenectl search --from-to 1467286791 1478184286 --text remote --limit 4500 --aggregate-by-trail --normalize-rank
```

> **NOTE:**
>
> If you do not receive more results with a larger query limit, it means that you have found all results with your search expression.

However, the downside of using **lucenectl** to fine-tune your search is that after the cli search, you have to manually extract the trails that you find interesting with the help of the metadb.

The following example shows the output of a lucenectl search:

```
{
  "hits": [
     {
        "hits_count": 1,
        "channel_id": 1,
        "trail_id": "58",
        "rank": 0.4068610216585047
     },
     {
        "hits_count": 7,
        "channel_id": 761,
        "trail_id": "12",
        "rank": 1.0
     },
     {
        "hits_count": 2,
        "channel_id": 1,
        "trail_id": "139",
        "rank": 0.5923645275802537
     }
  ]
}
```

- rank: the larger the number, the higher the rank
- hits_count: the number of times the screen content search expression is displayed in the audit trail
- trail_id: the ID of the trail
- channel_id: the ID of the channel

The most relevant audit trail will probably be the one with the highest rank.

If you have determined which audit trail you are interested in, enter the following command. The value of **_connection_channel_id** will be the value of the **trail_id** from the lucenectl output that you have determined as most relevant.

```
psql -U scb scb -c "select audit from channels where _connection_channel_
id = 12;"
```

The output of this command will be:

```
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:2
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:1
```

From this output, the audit trail file name path is as follows: `/<audittrailpath>/audit-scb_rdp-1467274538-0.zat`

NOTE:

If you cannot find the file at the path, check whether it has been archived and search for the file in the archive path. Use the following command:

```
psql -U scb scb -c "select audit, _archive_path  from channels where _
connection_channel_id = 12;"
```

The output of this command will be:

```
                    audit                                    | _archive_path
-------------------------------------------------------------+---------------
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:2 | /<myarchive/path>/
/<audittrailpath>/audit-scb_rdp-1467274538-0.zat:1 | /<myarchive/path>/
                                                             |
```

If you still cannot find the audit trail, contact our Support Team.

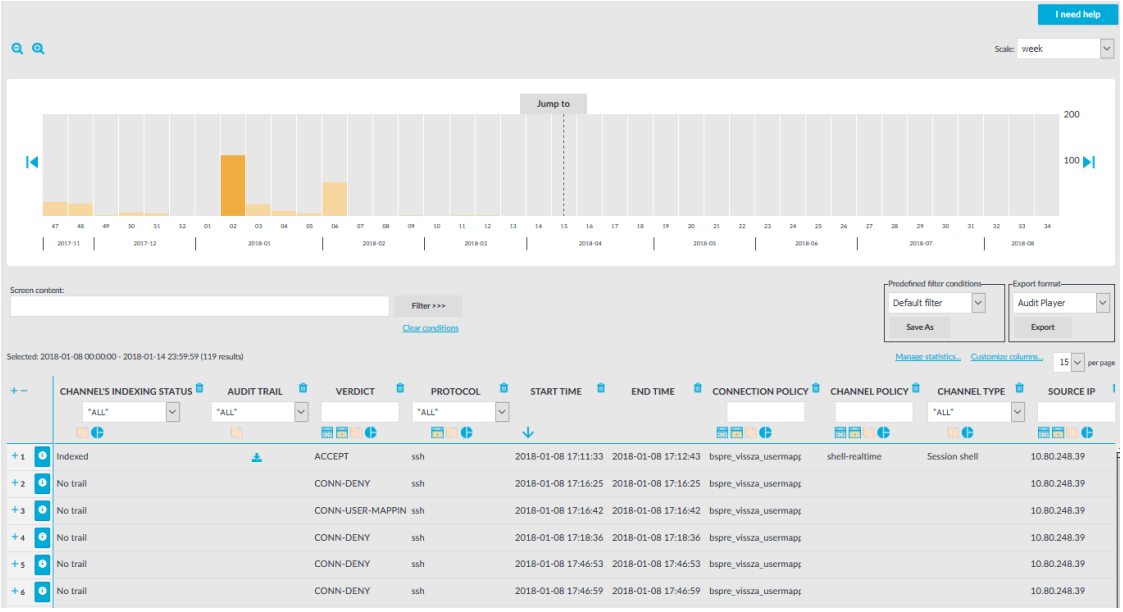
Displaying statistics on search results

One Identity Safeguard for Privileged Sessions (SPS) can create statistics (bar, pie and list) from various information about the search results, for example, the distribution of the target hosts, and so on.

To display statistics about the connections

1. Navigate to the **Search (classic) > Search** page.
2. Set the filters you need.
3. Click the icon in the header of the table. A pop-up window is displayed.

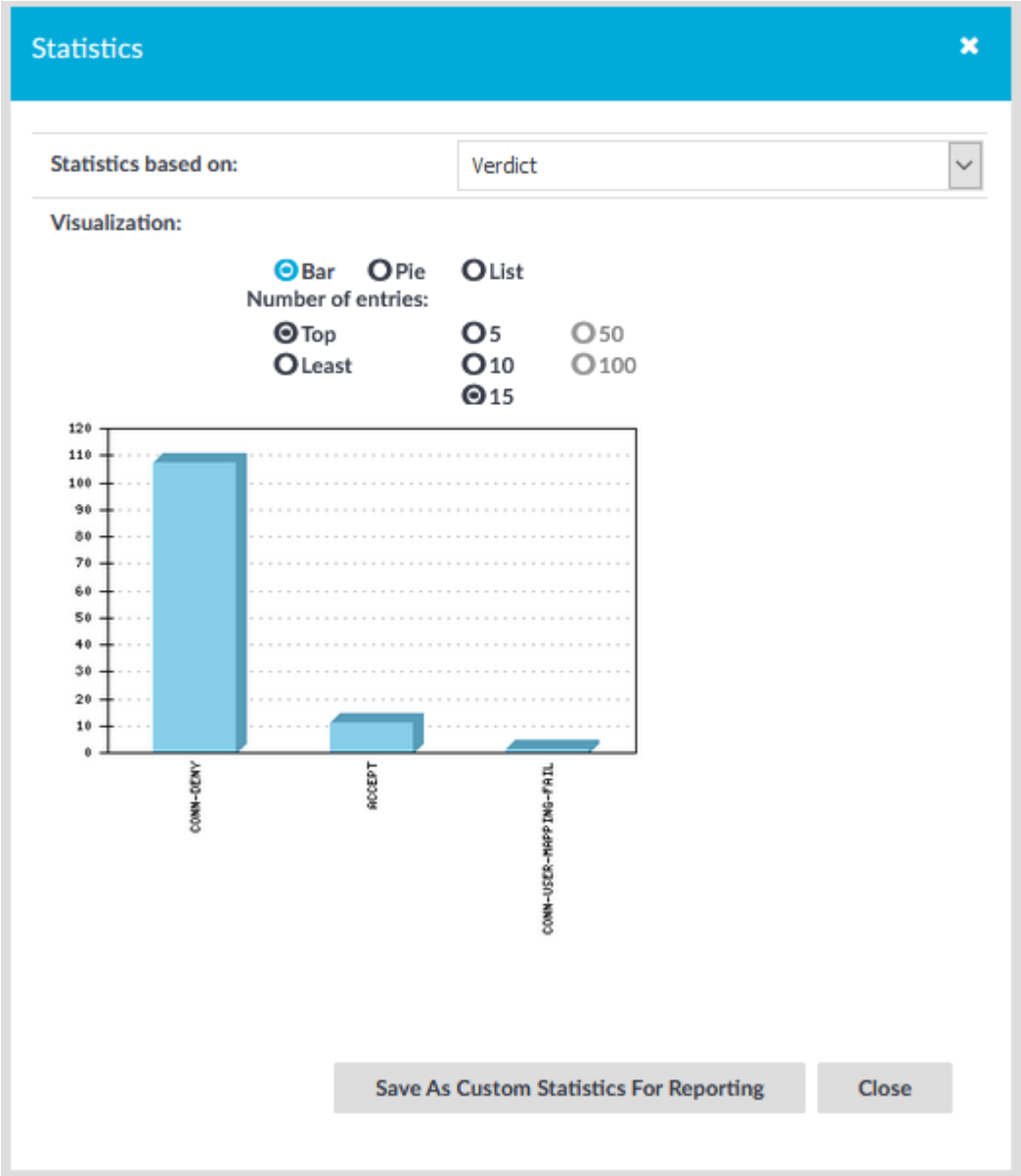
Figure 327: Search (classic) > Search — Displaying statistics



4. Select the type of metadata you want to create statistics on from the **Statistics based on** field, for example, **Source IP**.
5. Select the type of chart to display, that is, **Bar**, **Pie** or **List**. The chart will be displayed in the same pop-up window.
6. By default, the statistics start with the largest number of entries. To start statistics with the least number of entries, select **Least**.
7. Select the number of data groups to display from the **Number of entries** field. For example, if you want to display the statistics of the ten hosts that start the most connections (the "top talkers"), select *10*. That way the top ten talkers will be displayed individually, while the amount of connections started by the other hosts will be aggregated and labeled as **Others**.

> NOTE:
>
> For pie and bar charts you can select *5*, *10* and *15*, for lists *5*, *10*, *15*, *50* and *100*.

Figure 328: Search (classic) > Search — Selecting display type



8. (Optional) To export statistics data to a CSV file, select **List**, set the number of entries and click **Export all to CSV**. SPS compiles the selected data into a `results.csv` file.

   NOTE:

   This action exports all rows, not only the currently displayed ones.

9. (Optional) You can also save these statistics and include them in reports as a report subchapter. You can include these subchapters into your reports in the **Reports > Configuration** menu.

a. To save these statistics as custom statistics for reporting, click **Save As Custom Statistics For Reporting**.

b. Add a name for the statistics in the **Name** field.

c. Select a group from the already existing groups in the **Groups** field. The autocomplete function helps you with the selection.

d. (Optional) The **Add to report as a subchapter** function enables you to instantly add this statistics as a subchapter to the selected report.

e. Click **Save**. This action includes the saved statistics as a selectable subchapter into **Reporting > Configuration**. For details on how to add this subchapter to a selected report, see Configuring custom reports on page 778.

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

4

4-eyes authorization
4-eyes authorization is an advanced authorization method where only two administrators logging in simultaneously are permitted to access the server. These administrators can monitor each other's work, reducing the chance of (accidental or intentional) human errors in the server administration process.

A

access policy
Collection of access policies. Access policies define who can authorize and audit a connection.

alias IP
An additional IP address assigned to an interface that already has an IP address. The normal and alias IP addresses both refer to the same physical interface.

Audit Player
Audit Player is a desktop application that can replay recorded audit trails like movie. The Audit Player is available for the Microsoft Windows and GNU/Linux platforms.

Audit trail
An audit trail is a file storing the recorded activities of the administrators in an encrypted format. Audit trails can be replayed using the Audit Player application.

auditing policy
The auditing policy determines which events are logged on host running Microsoft Windows operating systems.

authentication
The process of verifying the authenticity of a user or client before allowing access to a network system or service.

Authentication Policy
An authentication policy is a list of authentication methods that can be used in a connection. Connection definitions refer to an authentication policy to determine how the client can authenticate to the target server.

B

BOM
The byte order mark (BOM) is a Unicode character used to signal the byte-order of the message text.

BSD-syslog protocol
The old syslog protocol standard described in RFC 3164. Sometimes also referred to as the legacy-syslog protocol.

C

CA
A Certificate Authority (CA) is an institute that issues certificates.

Cadence
[[[Undefined variable TemplateGuideVariables.OneIdentityNameShort]]] font that contains standard icons used in the user interfaces for various [[[Undefined variable TemplateGuideVariables.OneIdentityNameShort]]] products.

certificate
A certificate is a file that uniquely identifies its owner. Certificates contains information identifying the owner of the certificate, a public key itself, the expiration date of the certificate, the name of the CA that signed the certificate, and some other data.

Channel Policy
The channel policy lists the SSH channels (for example terminal session, SCP, and so on) that can be used in a connection. The channel policy can further restrict access to each channel based on the IP address of the client or the server, a user list, or a time policy.

client mode
In client mode, syslog-ng collects the local logs generated by the host and forwards them through a network connection to the central syslog-ng server or to a relay.

Common Gateway Protocol (CGP)
Reliable connection is also known as Common Gateway Protocol (CGP). It makes reconnection possible to the server in case of a network failure. Default port number is 2598.

Connection Policy
Connection policies determine if a server can be accessed from a particular client. Connection policies reference other resources (policies, usergroups, keys) that must be configured and available before creating a connection policy.

controlled traffic
SPS audits and controls only the traffic that is configured in the connection and channel policies, all other traffic is forwarded on the packet level without any inspection.

D

destination
A named collection of configured destination drivers.

destination driver
A communication method used to send log messages.

destination, local
A destination that transfers log messages within the host, for example writes them to a file, or passes them to a log analyzing application.

destination, network
A destination that sends log messages to a remote host (that is, a syslog-ng relay or server) using a network connection.

disk buffer
The Premium Edition of syslog-ng can store messages on the local hard disk if the central log server or the network connection to the server becomes unavailable.

disk queue
See disk buffer.

domain name
The name of a network, for example: balabit.com.

Drop-down
Flare default style, that can be used to group content within a topic. It is a resource to structure and collapse content especially in non-print outputs.

E

embedded log statement
A log statement that is included in another log statement to create a complex log path.

F

filter
An expression to select messages.

firmware
A firmware is a collection of the software components running on SPS. Individual software components cannot be upgraded on SPS, only the entire firmware. SPS contains two firmwares, an external (or boot) firmware and an internal (or core) firmware. These can be upgraded separately.

fully qualified domain name (FQDN)
A domain name that specifies its exact location in the tree hierarchy of the Domain Name System (DNS). For example, given a device with a local hostname myhost and a parent domain name example.com, the fully qualified domain name is myhost.example.com.

G

gateway
A device that connects two or more parts of the network, for example: your local intranet and the external network (the Internet). Gateways act as entrances into other networks.

Glossary
List of short definitions of product specific terms.

H

HA network interface
The HA interface (labeled 4 or HA) is an interface reserved for communication between the nodes of SPS clusters.

High Availability
High Availability (HA) uses a second SPS unit (called secondary node) to ensure that the services are available even if the first unit (called primary node) breaks down.

host
A computer connected to the network.

hostname
A name that identifies a host on the network.

I

ICA
The base protocol of Citrix products (default port tcp/1494). It does desktop or application remoting through TCP or other network protocols. Independent Computing Architecture (ICA) is a proprietary protocol for an application server system, designed by Citrix Systems. The protocol lays down a specification for passing data between server and clients, but is not bound to any one platform. ICA is broadly similar in purpose to window servers such as the X Window System. It also provides for the feedback of user input from the client to the server, and a variety of means for the server to send graphical output, as well as other media such as audio, from the running application to the client.

IETF-syslog protocol
The syslog-protocol standard developed by the Internet Engineering Task Force (IETF), described in RFC 5424-5427.

K

key pair
A private key and its related public key. The private key is known only to the owner, while the public key can be freely distributed. Information encrypted with the private key can only be decrypted using the public key.

L

LDAP
The Lightweight Directory Access Protocol (LDAP), is an application protocol for querying and modifying data using directory services running over TCP/IP.

License
SPS's license determines the number of servers (IP addresses) that SPS protects. The license limits the number of IP addresses accessible.

log path
A combination of sources, filters, parsers, rewrite rules, and destinations: syslog-ng examines all messages arriving to the sources of the logpath and sends the messages matching all filters to the defined destinations.

log source host
A host or network device (including syslog-ng clients and relays) that sends logs to the syslog-ng server. Log source hosts can be servers, routers, desktop computers, or other devices capable of sending syslog messages or running syslog-ng.

log statement
See log path.

logstore
A binary logfile format that can encrypt, compress, and timestamp log messages.

Long Term Supported release
Long Term Supported releases are major releases of that are supported for three years after their original release.

LSH
See log source host.

N

name server
A network computer storing the IP addresses corresponding to domain names.

node
An SPS unit running in High Availability mode.

Note
Circumstance, that needs special attention.

O

Oracle Instant Client
The Oracle Instant Client is a small set of libraries, which allow you to connect to an Oracle Database. A subset of the full Oracle Client, it requires minimal installation but has full functionality.

output buffer
A part of the memory of the host where syslog-ng stores outgoing log messages if the destination cannot accept the messages immediately.

output queue
Messages from the output queue are sent to the target syslog-ng server. The syslog-ng application puts the outgoing messages directly into the output queue, unless the output queue is full. The output queue can hold 64 messages, this is a fixed value and cannot be modified.

overflow queue
See output buffer.

P

parser
A set of rules to segment messages into named fields or columns.

ping
A command that sends a message from a host to another host over a network to test connectivity and packet loss.

port
A number ranging from 1 to 65535 that identifies the destination application of the transmitted data. For example: SSH commonly uses port 22, web servers (HTTP) use port 80, and so on.

primary node
The active SPS unit that is inspecting the traffic when SPS is used in High Availability mode.

PSM
An old abbreviation of Safeguard for Privileged Sessions (SPS).

Public-key authentication
An authentication method that uses encryption key pairs to verify the identity of a user or a client.

R

redundant Heartbeat interface
A redundant Heartbeat interface is a virtual interface that uses an existing interface of the SPS device to detect that the other node of the SPS cluster is still available. The virtual interface is not used to synchronize data between the nodes, only Heartbeat messages are transferred.

regular expression
A regular expression is a string that describes or matches a set of strings.

relay mode
In relay mode, syslog-ng receives logs through the network from syslog-ng clients and forwards them to the central syslog-ng server using a network connection.

Remote Desktop Gateway
Remote Desktop Gateway (RD Gateway) is a role service in the Remote Desktop Services server role that allows authorized remote users to connect to resources located on an internal or private network from any Internet-connected device. The accessible resources can be terminal servers, remote applications, remote desktops, and so on. This service is also called Remote Desktop Gateway or RD Gateway.

rewrite rule
A set of rules to modify selected elements of a log message.

S

SaaS
Software-as-a-Service.

SCB
An old abbreviation of Safeguard for Privileged Sessions (SPS).

secondary node
The passive SPS unit that replaces the active unit (the primary node) if the primary node becomes unavailable.

server mode
In server mode, syslog-ng acts as a central log-collecting server. It receives messages from syslog-ng clients and relays over the network, and stores them locally in files, or passes them to other applications, for example, log analyzers.

Skin
Used to design the online output window.

Snippet
Flare file type that can be used to reuse content. The One Identity SPS contains various default snippets.

SNMP
Simple Network Management Protocol (SNMP) is an industry standard protocol used for network management. SPS can send SNMP alerts to a central SNMP server.

source
A named collection of configured source drivers.

source driver
A communication method used to receive log messages.

source, local
A source that receives log messages from within the host, for example, from a file.

source, network
A source that receives log messages from a remote host using a network connection, for example, network(), syslog().

split brain
A split brain situation occurs when for some reason (for example, the loss of connection between the nodes) both nodes of an SPS cluster become active (primary) nodes. This might cause that new data (for example, audit trails) is created on both nodes without being replicated to the other node. Thus, it is likely in this situation that two diverging sets of data are created, which cannot be trivially merged.

SPS
Safeguard for Privileged Sessions

SSH settings
SSH settings determine the parameters of the connection on the protocol level, including timeout value and greeting message of the connection, as well as the encryption algorithms used.

SSL
See TLS.

syslog-ng
The syslog-ng application is a flexible and highly scalable system logging application, typically used to manage log messages and implement centralized logging.

syslog-ng agent
The syslog-ng Agent for Windows is a commercial log collector and forwarder application for the Microsoft Windows platform. It collects the log messages of the Windows-based host and forwards them to a syslog-ng server using regular or SSL-encrypted TCP connections.

syslog-ng client
A host running syslog-ng in client mode.

syslog-ng Premium Edition
The syslog-ng Premium Edition is the commercial version of the open-source application. It offers additional features, like encrypted message transfer and an agent for Microsoft Windows platforms.

syslog-ng relay
A host running syslog-ng in relay mode.

syslog-ng server
A host running syslog-ng in server mode.

T

template
A user-defined structure that can be used to restructure log messages or automatically generate file names.

Time Policy
The time policy determines which hours of a day can the users access a connection or a channel.

Tip
Additional, usefull information.

TLS
Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols which provide secure communications on the Internet. The application can encrypt the communication between the clients and the server using TLS to prevent unauthorized access to sensitive log messages.

traceroute
A command that shows all routing steps (the path of a message) between two hosts.

U

UNIX domain socket
A UNIX domain socket (UDS) or IPC socket (inter-procedure call socket) is a virtual socket, used for inter-process communication.

User List
User lists are white- or blacklists of usernames that allow fine-control over who can access a connection or a channel.