



One Identity Safeguard for Privileged Sessions 6.0.12

DEPRECATED How to connect One Identity TPAM with One Identity Safeguard for Privileged Sessions

Copyright 2021 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 DEPRECATED How to connect One Identity TPAM with One Identity Safeguard for Privileged Sessions

Updated - 29 November 2021, 09:16

Version - 6.0.12

Contents

Introduction	4
How SPS and TPAM work together	5
Technical requirements	6
How SPS and TPAM work together - in detail	7
Configuring SPS	10
Using a custom Credential Store plugin to authenticate on the target hosts	10
Storing sensitive plugin data securely	11
Configuring gateway authentication	12
Configuring DNS resolution	12
Configuring TPAM	13
Adding an ISA CLI user	13
Assign ISA access policies to ISA CLI user	14
Obtaining the private key of the ISA CLI user	14
Enabling custom attributes in TPAM	15
TPAM plugin parameter reference	16
[tpam]	17
[plugin]	21
About us	23
Contacting us	23
Technical support resources	23

Introduction

NOTE:

This tutorial describes the deprecated version of the plugin.

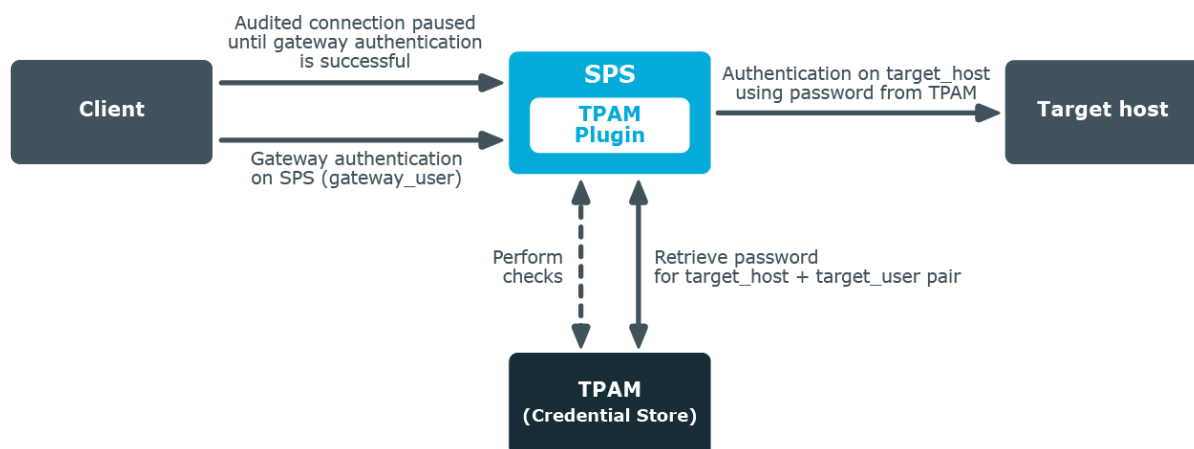
To upgrade your deprecated plugin for One Identity Safeguard for Privileged Sessions 6.0, see [Upgrading plugins for One Identity Safeguard for Privileged Sessions version 6.0](#).

This tutorial describes how to connect your One Identity Safeguard for Privileged Sessions (SPS) with TPAM using a plugin to automatically retrieve passwords.

Users wishing to access a target host are able to authenticate themselves without actually having access to the credentials required to access that host. Passwords are retrieved by SPS from TPAM using a plugin, with SPS impersonating the authenticated user and TPAM acting as the repository of user credentials (a Credential Store in SPS terminology).

This automatic password retrieval is crucial as this method protects the confidentiality of passwords, enabling you to protect critical assets and meet compliance requirements.

How SPS and TPAM work together



1. A client attempts to establish a connection to a protected server (the target host) through SPS as a gateway.
In SPS, TPAM is configured as a Credential Store (together with a TPAM plugin) in the connection policy matching the connection.
2. SPS prompts the client, now considered a gateway user, for credentials.
3. The client provides authentication details.
4. To retrieve the password required to access the target host from the configured Credential Store (that is, TPAM), SPS establishes an SSH connection to TPAM.
5. SPS and TPAM mutually verify each other's identity.
6. The TPAM plugin maps the data received from SPS to corresponding data entries in TPAM so that TPAM receives data that it can process.
7. SPS retrieves information from TPAM through the TPAM plugin. Depending on the authorization method configured, SPS:
 - Performs certain checks about the gateway user.
If the checks are successful and the client is granted access, SPS retrieves from TPAM the password required to establish the connection between the client and the target host.
 - Retrieves the requested password from TPAM without performing any checks.
8. SPS authenticates the client to the target host, and establishes the connection.

Technical requirements

To successfully connect SPS with TPAM, you need the following components.

In SPS:

- A copy of the TPAM plugin, version 1.1.0 or later.
- Gateway authentication must be configured in the connection that uses TPAM as Credential Store so that gateway user details are at hand. For details, see [Configuring gateway authentication](#) on page 12.
- DNS resolution must be configured for the target host. For details, see [Configuring DNS resolution](#) on page 12.

In TPAM:

If you do not want the password to be automatically provided by TPAM following successful gateway authentication, one of the following is required (depending on the chosen [authorization](#) method):

- The gateway user requesting to access the target host must be present in TPAM as a Requestor with approval to view the password.
- The gateway user requesting to access the target host must have an access policy for either the Accounts or Systems that you want to include in the scope of password retrieval.

How SPS and TPAM work together - in detail

1. A client attempts to establish a connection to a protected server (the target host) through SPS as a gateway.

In SPS, TPAM is configured as a Credential Store (together with a TPAM plugin) in the connection policy matching the connection.

For details on setting up gateway authentication on the connection that uses TPAM as a Credential Store, see [Configuring gateway authentication](#) on page 12.

For details on configuring a TPAM plugin, see [Using a custom Credential Store plugin to authenticate on the target hosts](#) on page 10.

2. SPS prompts the client, now considered a gateway user, for credentials.
3. The client provides authentication details.
4. To retrieve the password required to access the target host from the configured Credential Store (that is, TPAM), SPS establishes an SSH connection to TPAM as an Information Security Administrator (ISA) CLI user present in TPAM.

For details on how to add an ISA CLI user in TPAM with ISA permissions at the Account level, see [Adding an ISA CLI user](#) on page 13 and [Assign ISA access policies to ISA CLI user](#) on page 14.

5. SPS and TPAM mutually verify each other's identity. This requires the presence of the following keys:
 1. TPAM's [server_public_key](#)
 2. the private [server_user_key](#) of the ISA CLI user

This key must be stored in a local Credential Store on SPS. For details, see [Storing sensitive plugin data securely](#) on page 11.
6. The TPAM plugin maps the data received from SPS to corresponding data entries in TPAM so that TPAM receives data that it can process. The goal is to match up SPS data with TPAM data as follows:

Table 1: Mapping SPS data to TPAM data

Data in SPS	Data in TPAM
gateway user	Requestor
target user	Account
target host	System

- a. The TPAM plugin calculates the hostname of the target host. TPAM expects the address of the target host as a hostname rather than as an IP address.

If the address of the target host is an IP address, there are two options depending on how the `system_name_resolver` parameter is configured in the TPAM plugin:

- The IP address is retrieved from TPAM itself. This is the default behavior. For this to happen, the `system_name_resolver` parameter must be set to `tpam`, and you must ensure that the ISA CLI user has an ISA access policy assigned for all relevant Systems. For details, see [Assign ISA access policies to ISA CLI user](#) on page 14.
- A DNS lookup is performed. For this to happen, you have to set the `system_name_resolver` parameter to `dns`, and also configure DNS resolution in SPS. For details, see [Configuring DNS resolution](#) on page 12.

If the address of the target host is in FQDN format, then the hostname part of the FQDN is kept.

- b. Optionally, if the relevant setting (`system_prefix`) is enabled, a prefix of your choice is prepended to the hostname.
- c. Optionally, this step might involve an extra round of mapping. If the relevant setting (`system_maptoreal`) is enabled, the TPAM plugin performs a lookup to check whether the Account-System pair is mapped to custom fields set in TPAM. If yes, then it is the password corresponding to the custom data entries that TPAM will retrieve.

Table 2: Mapping Account-System data to custom field data in TPAM

Account-System data in TPAM	Custom data in TPAM
Real Account	ManagedAccount.AccountCustom1
Real System	ManagedAccount.AccountCustom2

For details on how to enable custom fields in TPAM, see [Enabling custom attributes in TPAM](#) on page 15.

7. SPS retrieves information from TPAM through the TPAM plugin. Depending on how the `authorization` parameter is configured, the following happens:

- If authorization is set to approval: SPS checks whether the gateway user is present in TPAM as a Requestor with approval to view the password. If present, SPS retrieves from TPAM the password required to establish the connection between the client and the target host.
 - If authorization is set to policy: SPS checks whether the gateway user has an access policy in TPAM for either the Account or System in question. If present, SPS retrieves from TPAM the password required to establish the connection between the client and the target host.
 - If authorization is set to gateway: The gateway user is granted access and SPS retrieves from TPAM the password required to establish the connection between the client and the target host without any further checks.
8. SPS authenticates the client to the target host, and establishes the connection.

Configuring SPS

This section provides detailed instructions as to what to configure on SPS:

- [upload the TPAM plugin to SPS](#)
- [configure a local Credential Store in the plugin to store the server_user_key securely](#)
- [set up gateway authentication on the connection that uses TPAM as the Credential Store](#)
- [set up DNS resolution](#)

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](#).

i 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**.

i 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 .
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 .

Storing sensitive plugin data securely

By default, the configuration of the plugin is stored on SPS in the configuration of SPS. Make sure that you store the sensitive parameters (`server_user_key`) of the plugin in an encrypted way.

To store sensitive plugin data securely

1. Obtain the `server_user_key`.
2. Log in to SPS and create a local Credential Store. For details, see "[Configuring local Credential Stores](#)" in the [Administration Guide](#).

Instead of usernames and passwords, you will store the configuration parameters of the plugin in this Credential Store.

3. Add the plugin parameters you want to store in an encrypted way to the Credential Store. You can store any configuration parameter of the plugin in the Credential Store, but note that if an option appears in the Credential Store, the plugin will use it. If the same parameter appears in the configuration of the plugin, it will be ignored.
 - Enter the name of the configuration section without the brackets in the **Host** field (tpam).
 - Enter the name of the plugin parameter in the **Username** field (server_user_key).
 - Enter the value of the plugin parameter in the **SSH Keys** field.
4. Commit your changes, and navigate to the configuration of the plugin on the **Policies > Credential Stores** page.
5. In the plugin configuration file, enter the name of the local Credential Store under the [plugin] section, in the `cred_store` parameter.

Configuring gateway authentication

To set up gateway authentication on the connection that uses TPAM as the Credential Store, follow the instructions in:

- For out-of-band gateway authentication: ["Configuring out-of-band gateway authentication" in the Administration Guide](#)
- For inband gateway authentication: ["Client-side authentication settings" in the Administration Guide](#)

Configuring DNS resolution

Since TPAM expects the address of the target host as a hostname rather than as an IP address, IP addresses must be transformed to hostnames. The hostname is then used as input when the TPAM plugin calculates the System name for TPAM. For this, you need to configure DNS resolution on SPS.

To resolve hostnames, SPS uses the Domain Name Servers set in **Basic Settings > Network > Naming > Primary DNS server** and **Secondary DNS server**. For details on these fields, see the section on naming in ["Network settings" in the Administration Guide](#).

Configuring TPAM

This section provides detailed instructions as to what to configure on TPAM:

- add an CLI user with Information Security Administrator (ISA) rights that SPS will use to communicate with TPAM
- assign ISA access policy for all relevant Accounts and Systems
- download the public key of TPAM
- enable custom attributes

Adding an ISA CLI user

Purpose:

When communicating with TPAM, SPS uses a CLI user with Information Security Administrator (ISA) rights to establish an SSH connection to TPAM. This user must be present in TPAM. In addition, in the TPAM plugin's configuration file, you need to provide the user name of this user (`server_user`).

To add an ISA CLI user

For details on how to add an ISA CLI user in TPAM, see section *Add a CLI user ID* in the [TPAM Administration Guide](#).

Ensure that the user has ISA permissions (configurable through access policies) for all Accounts and Systems whose credentials will be retrieved from TPAM. For details on how to assign ISA access policies, see [Assign ISA access policies to ISA CLI user](#) on page 14.

If this user is already present in TPAM, here is how you can obtain its user name:



1. Log in to TPAM using a TPAM Administrator account.
2. Navigate to **Users & Groups > User IDs > Manage User IDs**.
3. Click the **Listing** tab.
4. Look for the user name of the ICA CLI user in the **User Name** column.

Assign ISA access policies to ISA CLI user

Purpose:

In order for SPS to be able to retrieve information from TPAM, you must ensure that the ISA CLI user that is used for communicating with TPAM has ISA permission to all relevant Systems and Accounts whose credentials will be retrieved from TPAM.

To grant the ISA CLI user ISA permission

1. Log in to TPAM using a TPAM Administrator account.
2. Navigate to **Users & Groups > User IDs > Manage User IDs**.
3. Click the **Listing** tab.
4. Select the ISA CLI user.
5. Click the **Permissions > Results** tab.
6. Assign an ISA policy on the System level:
 - a. On the left, select the System you want to assign an ISA access policy to.
 - b. On the right, in the **Access Policy Details > Access policy** drop-down menu, select **ISA**.
 - c. Click .
7. Assign an ISA policy on the Account level:
 - a. On the left, select the Account you want to assign an ISA access policy to.
 - b. On the right, in the **Access Policy Details > Access policy** drop-down menu, select **ISA**.
 - c. Click .

Obtaining the private key of the ISA CLI user

Purpose:

In the TPAM plugin's configuration file, you need to provide the private key ([server_user_key](#)) of the CLI user with Information Security Administrator (ISA) access rights to TPAM ([server_user](#)) that SPS will use when communicating with TPAM. To obtain the key, download it from TPAM.

To download the private key of the ISA CLI user

1. Log in to TPAM using a TPAM Administrator account.
2. Navigate to **Users & Groups > User IDs > Manage User IDs**.
3. Click the **Listing** tab.
4. Select the ISA CLI user.
5. Click the **Details** tab.
6. Click the **Key Based** tab.
7. Select the **CLI** checkbox.
8. Click **Download Key**.

This key must be stored in a local Credential Store in SPS. For details on how to do that, see [Storing sensitive plugin data securely](#) on page 11.

Enabling custom attributes in TPAM

Purpose:

When mapping target user and target host names to their corresponding counterparts (Account and System names) in TPAM, an extra round of mapping may be necessary if the mapping option [system_maptoreal](#) is enabled.

The prerequisite of this extra mapping to happen is the enabling of custom attributes in TPAM.

To enable custom attributes in TPAM

1. Log in to TPAM with a TPAM System Administrator account.
2. Navigate to **System Status/Settings > Global Settings**.
3. Search for the **Custom Column Names** category.
4. For the **ManagedAccount.AccountCustom1** option, type **Real Account** in the field next to the option name.
5. For the **ManagedAccount.AccountCustom2** option, type **Real System** in the field next to the option name.
6. Click **Save Changes**.

Expected result:

You are now able to set these parameters per account on the **Custom Information** tab.

TPAM plugin parameter reference

This section describes the available options of the TPAM plugin.

The plugin uses an ini-style configuration file with sections and name=value pairs. This format consists of sections, led by a [section] header and followed by name=value entries. Note that the leading whitespace is removed from values. The values can contain format strings, which refer to other values in the same section. For example, the following section would resolve the %(dir)s value to the value of the dir entry (/var in this case).

```
[section name]
dirname=%(dir)s/mydirectory
dir=/var
```

All reference expansions are done on demand. Lines beginning with # or ; are ignored and may be used to provide comments.

You can edit the configuration file from the SPS web interface. The following code snippet is a sample configuration file.

```
[tpam]
authorization=policy
required_policy=<name-of-access-policy-required-to-be-present-for-authorization>

server=<hostname-or-IP-address-of-TPAM>
server_public_key=<public-key-of-TPAM>
server_port=<SSH-port-number-of-TPAM>
server_user=<TPAM-CLI-user-with-ISA-rights>
server_user_key=<private-key-of-server_user>
system_name_resolver=tpam
system_maptoreal=no
system_prefix=<your-preferred-prefix>
reuse_gateway_password=no
```

```
[plugin]
config_version=1
cred_store=<name-of-credential-store-hosting-sensitive-data>
log_level=info
```


[tpam]

This section contains the options related to the TPAM server.

```
[tpam]
authorization=policy
required_policy=<name-of-access-policy-required-to-be-present-for-authorization>

server=<hostname-or-IP-address-of-TPAM>
server_public_key=<public-key-of-TPAM>
server_port=<SSH-port-number-of-TPAM>
server_user=<TPAM-CLI-user-with-ISA-rights>
server_user_key=<private-key-of-server_user>
system_name_resolver=tpam
system_maptoREAL=no
system_prefix=<your-preferred-prefix>
reuse_gateway_password=no
```

authorization

Type:	approval gateway policy
Required:	no
Default:	gateway

Description: The authorization method used by TPAM to check whether the gateway user can be granted access to the target host. Possible values are:

- **approval:** Before providing the password required to access the target host, TPAM checks whether the gateway user is present in TPAM as a Requestor with approval to view the password.
- **gateway:** Access is granted without TPAM performing any checks.
- **policy:** Before providing the password required to access the target host, TPAM checks whether the gateway user has an access policy in TPAM for both the Account and System in question. If this value is configured, you must also configure [required_policy](#).

required_policy

Type:	string
Required:	no
Default:	Privileged Access

Description: Used only when authorization is set to `policy`. This parameter specifies the name of the access policy that is required to be present for access to be granted to the target host.

server

Type:	string
Required:	yes
Default:	N/A

Description: The address of the TPAM server, either a hostname or an IP address.

server_public_key

Type:	string
Required:	yes
Default:	N/A

Description: The public key corresponding to the hostname or IP address of the TPAM server, used for checking the TPAM server's identity.

Must be provided in the Open SSH [known_hosts](#) format, which includes:

- the address of the server
- the cipher suite used for encryption
- the hash of the key

Examples:

- current practice with hashed server address:
`|1|shAKuZdzJe1KykkXBo+14qpE+Fo=|J8oYavGEL2Rmo+u5R4r+Mdt7vuE= ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAy...`
- legacy practice with plain-text server address: `10.170.1.30 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAy...`

TIP:

To find out the public key of TPAM in the required format:

1. Log in to the core shell of SPS (through the console or SSH). For details on how to access the SPS console, see ["Accessing the One Identity Safeguard for Privileged Sessions \(SPS\) console" in the Administration Guide](#).
2. Set up an SSH connection to TPAM. When prompted about the key fingerprint, answer yes.
3. Search for the public key of TPAM in the known-hosts file using the following command:

```
ssh-keygen -F <hostname-or-IP-address-of-TPAM>
```

This command returns the key in the required format, which you can then copy and paste in the `server_public_key` field.

server_port

Type:	integer
Required:	no
Default:	22

Description: The port where TPAM is listening for SSH connections.

server_user

Type:	string
Required:	yes
Default:	N/A

Description: The user name of a CLI user with Information Security Administrator (ISA) access rights to TPAM. SPS sets up the SSH connection to TPAM using this ISA CLI user. This user must be present in TPAM.

For details on how to add this user in TPAM or how to obtain its user name if the user is already present, see [Adding an ISA CLI user](#) on page 13.

server_user_key

Type:	string
Required:	yes
Default:	N/A

Description: The SSH compatible, RSA-encrypted private key of `server_user`. This key must be stored in a Credential Store defined under `cred_store` in the `[plugin]` section. For details on how to obtain the key, see [Obtaining the private key of the ISA CLI user](#) on page 14.

For details on how to store the key in a local Credential Store policy on SPS, see [Storing sensitive plugin data securely](#) on page 11.

system_name_resolver

Type:	tpam dns
Required:	no
Default:	tpam

Description: TPAM expects the address of the target host as a hostname rather than as an IP address. You can specify where to take the target hostname from in case the address of the target host has been provided as an IP address:

- If set to `tpam`, the hostname is retrieved from TPAM. For this to happen, ensure that the ISA CLI user used for communicating with TPAM has ISA permissions at the System level. For details, see [Assign ISA access policies to ISA CLI user](#) on page 14.
- If set to `dns`, the hostname is retrieved from SPS following DNS resolution. For this to happen, you must configure DNS resolution in SPS. For details, see [Configuring DNS resolution](#) on page 12.

system_maptoreal

Type:	yes no
Required:	no
Default:	no

Description: If this parameter is set to `yes`, an additional lookup is performed on TPAM to map the Account-System pair to the custom attributes `ManagedAccount.AccountCustom1` and `ManagedAccount.AccountCustom2`. If the mapping is successful, the password corresponding to the custom pair is retrieved.

NOTE:

Custom attributes in TPAM must be enabled by a System Administrator. For details, see [Enabling custom attributes in TPAM](#) on page 15.

system_prefix

Type:	string
Required:	no
Default:	empty string

Description: Any prefix of your choice. The TPAM plugin appends this prefix followed by an underscore (_) to the target hostname when constructing the System name for TPAM.

reuse_gateway_password

Type:	yes no
Required:	no
Default:	no

Description: If this parameter is set to yes, then if the gateway user is the same as the target user accessing the protected server, the gateway password is reused as the password required to access the target host, effectively skipping password checkout from TPAM.

[plugin]

This section contains the options related to the plugin itself.

```
[plugin]
config_version=1
cred_store=<name-of-credential-store-hosting-sensitive-data>
log_level=info
```

config_version

Type:	integer
Required:	yes
Default:	1

Description: The version number of the configuration format. This is used to enable potentially incompatible changes in the future. If provided, the configuration will not be upgraded automatically. If not provided, the configuration will be upgraded automatically.

cred_store

Type:	string
Required:	yes
Default:	N/A

Description: The name of a local Credential Store policy configured on SPS. You can use this credential store to store sensitive information of the plugin in a secure way, such as the [server_user_key](#). For details, see [Storing sensitive plugin data securely](#) on page 11.

log_level

Type:	integer or string
Required:	no
Default:	info

Description: The logging verbosity of the plugin. The plugin sends the generated log messages to the SPS syslog system. You can check the log messages in the **Basic settings > Troubleshooting > View log files** section of the SPS web interface. Filter on the plugin: string to show only the messages generated by the plugins.

The possible values are:

- debug or 10
- info or 20
- warning or 30
- error or 40
- critical or 50

For details, see Python logging API's log levels: [Logging Levels](#).

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