One Identity Safeguard for Privileged Sessions 5.8

How to connect TPAM with One Identity Safeguard for Privileged Sessions

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

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

IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

SPS How to connect TPAM with One Identity Safeguard for Privileged Sessions
Updated - August 2018
Version - 5.8

Contents

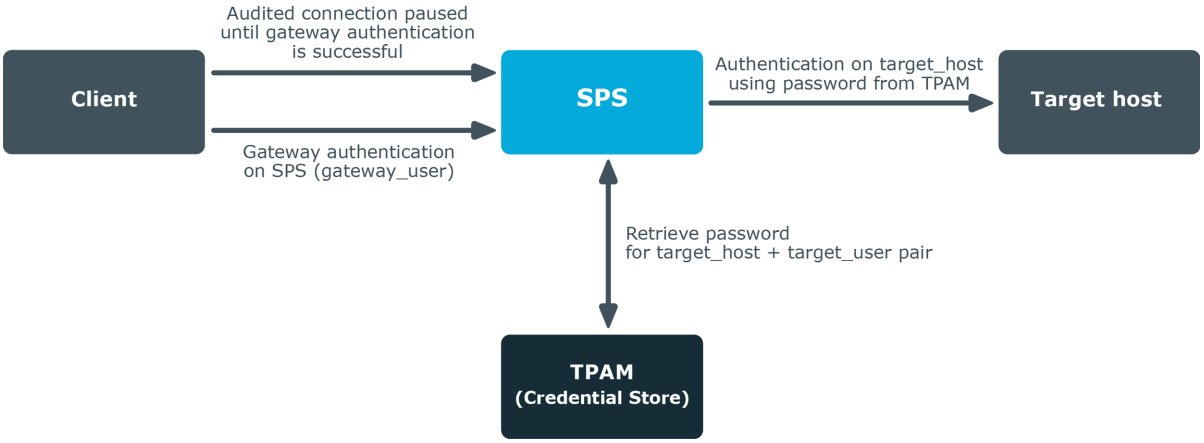
Introduction

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. SPS forwards the client's details to TPAM through the TPAM plugin.
7. The TPAM plugin maps the data received from SPS to corresponding data entries in TPAM so that TPAM receives data that it can process.
8. TPAM provides feedback to SPS about the result through the TPAM plugin.

   If the client is granted access, TPAM sends SPS the password required to establish the connection between the client and the target host.
9. 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 1.0.0 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 11.
- DNS resolution must be configured for the target host. For details, see Configuring DNS resolution on page 11.

In TPAM:

- The gateway user requesting to access the target host must be present in TPAM as a Requestor with approval to view the password.

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 11.

   For details on configuring a TPAM plugin, see Using a custom Credential Store plugin to authenticate on the target hosts on page 9.
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, see Adding an ISA CLI user on page 12.
5. SPS and TPAM mutually verify each other's identity. This requires the presence of the following keys:
   a. TPAM's `server_public_key`
   b. 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 10.
6. SPS forwards the client's details to TPAM through the TPAM plugin.
7. 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. 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, a DNS lookup is performed. For this to happen, you need to configure DNS resolution in SPS. For details, see Configuring DNS resolution on page 11.

   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 13.

8. TPAM provides feedback to SPS about the result through the TPAM plugin.

   If the client is granted access, TPAM sends SPS the password required to establish the connection between the client and the target host.

9. 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

Purpose:

To configure SPS to retrieve the credentials used to login to the target host using a custom plugin, complete the following steps.

Prerequisites:

To use a custom Credential Store plugin, you need to upload a working Credential Store plugin to SPS. This plugin is a script that uses the SPS API to access an external Credential Store or Password Manager. If you want to create such a plugin, contact our Support Team. For more information on creating a custom plugin, see "Troubleshooting plugins" in the Administration Guide.

> NOTE:
>
> Users accessing connections that use Credential Stores to authenticate on the target server must authenticate on SPS using gateway authentication or an AA plugin. Therefore gateway authentication or an AA plugin must be configured for these connections. For details, see "Performing inband gateway authentication in RDP connections" in the Administration Guide and "Integrating external authentication and authorization systems" in the Administration Guide.

To upload the custom Credential Store plugin you received, navigate to **Basic Settings > Plugins**, browse for the file and click **Upload**. Note that 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.

Steps:

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

Purpose:

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 do this, complete the following steps.

Steps:

1. Obtain the `server_user_key`.
2. Log in to SPS and create a local Credential Store. For details, see "Configuring password-protected 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 > AA Plugin Configurations** 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: "Local client-side authentication" 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 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
- 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`).

Steps:

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

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.

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.

Steps:

To download the private key of the ISA CLI user, complete the following steps:

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 10.

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.

Steps:

To enable custom attributes in TPAM, complete the following steps:

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]
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_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]
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_maptoreal=no
```

```
system_prefix=<your-preferred-prefix>
reuse_gateway_password=no
```

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|shAKuZdzJelKykkXBo+l4qpE+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 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:	yes
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 12.

server_user_key

Type:	string
Required:	yes
Default:	N/A

Description: The SSH compatible 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 12.

For details on how to store the key in a local Credential Store policy on SPS, see Storing sensitive plugin data securely on page 10.

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 13.

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 10.

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.

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 or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

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