One Identity Safeguard for Privileged Sessions 6.9.3

Evaluation Guide

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 Evaluation Guide
Updated - 30 April 2021, 09:25
Version - 6.9.3

Contents

Evaluating One Identity Safeguard for Privileged Sessions in a virtual environment

Before you start:

Before you start evaluating SPS, make sure you understand what SPS is and how it works. This information can greatly help you get SPS operational.

NOTE: 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. Please consult One Identity's Product Support Policies for more information on environment virtualization.

Supported virtual environments for evaluating One Identity Safeguard for Privileged Sessions

To evaluate One Identity Safeguard for Privileged Sessions as a virtual appliance, you can download and install the latest SPS ISO file into a virtual machine. The following virtual environments are supported for evaluation:

- Kernel-based Virtual Machine (KVM)
- Microsoft Hyper-V
- VMware
- vSphere (VMware ESX)

SPS may work in other virtual environments like VirtualBox as well, although these are officially not supported. You can obtain an evaluation license and the ISO file using your support portal account.

Download the evaluation version of SPS

1. Visit the Download Trials page, and navigate to **One Identity Safeguard for Privileged Sessions > Download Free trial**.
2. Complete the registration form, and click **Download Trial**.
3. You will receive the details on how to access your license key and the download the ISO files in email.

2

Setting up SPS and the virtual environment

1. To start using SPS, first install it in a virtual machine.
    - Setting up One Identity Safeguard for Privileged Sessions with vSphere
    - Setting up One Identity Safeguard for Privileged Sessions with VirtualBox
    - Setting up One Identity Safeguard for Privileged Sessions with Hyper-V
    - Setting up One Identity Safeguard for Privileged Sessions using Kernel-based Virtual Machine (KVM)
    - To deploy SPS as a Virtual Machine from the Azure Marketplace, see One Identity Safeguard for Privileged Sessions - Deployment from Azure Marketplace.
    - To set up and install SPS in an Amazon Web Services (AWS) virtual environment, see One Identity Safeguard for Privileged Sessions - Deployment on Amazon Web Services.
2. Then configure a simple scenario and start using SPS.

Setting up One Identity Safeguard for Privileged Sessions with vSphere

1. Download the vSphere application.

   Visit the vSphere webpage, and download latest version of the application for your operating system.
2. Install the vSphere application.

   Follow the instructions provided in the vSphere product documentation to install the application.
3. Install SPS. Follow the instructions provided in "One Identity Safeguard for Privileged Sessions VMware Installation Guide" in the Installation Guide.

4. Configure a simple scenario and evaluate SPS. For details, see Creating a simple scenario.

Setting up One Identity Safeguard for Privileged Sessions with VirtualBox

1. *Download the VirtualBox application*

   Download the latest version of VirtualBox for your operating system.
2. Install the VirtualBox application.
   - On Microsoft Windows, start the `VirtualBox.exe` file.
   - On Linux systems, follow the instructions provided in the VirtualBox manual.
3. Install SPS. Follow the instructions provided in "One Identity Safeguard for Privileged Sessions VMware Installation Guide" in the Installation Guide.
4. Configure a simple scenario and evaluate SPS. For details, see Creating a simple scenario.

Setting up One Identity Safeguard for Privileged Sessions with Hyper-V

1. Download the Hyper-V application.

   Visit the Hyper-V webpage, and download latest version of the application for your operating system.
2. Install the Hyper-V application.

   Follow the instructions provided in the Hyper-V product documentation to install the application.
3. Install SPS. Follow the instructions provided in "One Identity Safeguard for Privileged Sessions Hyper-V Installation Guide" in the Installation Guide.
4. Configure a simple scenario and evaluate SPS. For details, see Creating a simple scenario.

Setting up One Identity Safeguard for Privileged Sessions using Kernel-based Virtual Machine (KVM)

1. *Install a virtual machine manager application*

   Install a KVM manager application, for example, `virt-manager`.
2. Install SPS. Follow the instructions provided in "Installing One Identity Safeguard for Privileged Sessions as a Kernel-based Virtual Machine" in the Installation Guide.
3. Configure a simple scenario and evaluate SPS. For details, see Creating a simple scenario.

Creating a simple scenario

1. Connect to SPS.

   The SPS virtual machine acquires an IP address from your DHCP server accessible in the virtual environment. After SPS has booted up, the console displays the IP address of the SPS web interface at login prompt. To connect to SPS, use this IP address. For details, or tips if SPS cannot receive an IP address, see "The initial connection to One Identity Safeguard for Privileged Sessions (SPS)" in the Administration Guide.
2. Complete the Welcome Wizard as described in "Configuring One Identity Safeguard for Privileged Sessions (SPS) with the Welcome Wizard" in the Administration Guide. Upload the evaluation license file you have downloaded with your support portal account.
3. 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.
4. 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.
5. Replay your session in the browser. See "Replaying audit trails in your browser" in the Administration Guide for details.

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

General connection settings

SPS supports transparent and non-transparent proxy operation modes to make deployments in existing network infrastructures as easy as possible. SPS will automatically handle non-transparent and transparent connections simultaneously.

Modes of operation

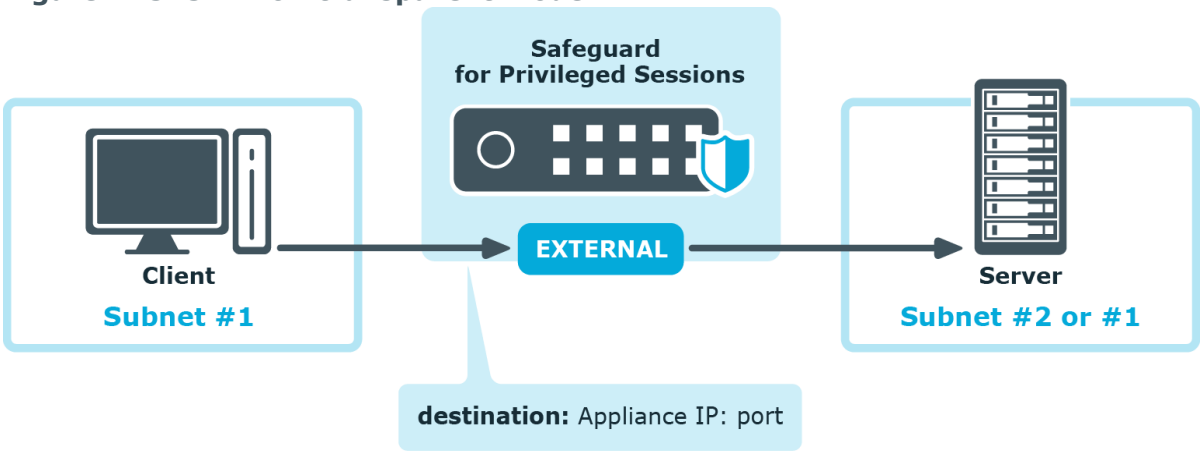
The following operation modes are possible:

- *Non-transparent proxy operation*: This guide will focus on this operation mode.
- *Transparent mode*: If you configure SPS proxies in transparent mode, the client usually addresses the target server directly. Therefore, you have to configure the connection policies in SPS accordingly.
- *Single-interface transparent mode*

Non-transparent proxy operation

This guide focuses on non-transparent proxy operation, which is the easiest to implement. In this configuration, clients connect to a server through SPS. That is, end-users address SPS explicitly, which then forwards connections to target systems based on various parameters depending on what destination selection method you select.

Figure 1: SPS in non-transparent mode



For an illustration of what happens when a client connects a server through SPS and how the different configuration options and policies of SPS affect this process, see:

- Connecting to a server through SPS using SSH
- Connecting to a server through SPS using RDP

Configuring the destination selection method

To configure the destination selection method, navigate to for example **SSH Control > Connections** (or the respective protocol control that you want to configure), and in the **Target** section, select the preferred method:

- *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.
- *NAT destination address*: Perform a network address translation on the target address.
- *Use fixed address*: The connection will connect always to this address, redirecting the clients to the server.
- *Inband destination selection*: Extract the address of the server from the username.

For details, see "Modifying the destination address" in the Administration Guide.

Network settings

Assigning logical interfaces to physical interfaces

To assign logical interfaces to the three physical interfaces of SPS, navigate to **Basic Settings > Network > Interfaces**.

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 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.

For details, see Network settings.

Routing uncontrolled traffic

To control how SPS routes uncontrolled traffic (that is, traffic that passes SPS but is not inspected or audited) between its network interfaces, navigate to **Basic Settings > Network > IP forwarding**.

You can connect interface pairs to each other, and SPS will route all uncontrolled traffic between these. 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.

For details, see Routing uncontrolled traffic between logical interfaces.

Configuring connections: SSH

The following procedures provide a skeleton to configure SSH connections in SPS. If you want to have a deeper understanding, read the in-depth detailed procedure.

Configure an SSH connection with fixed destination IP

The following describes how to configure a basic Secure Shell (SSH) connection in SPS. This Connection Policy uses a fixed destination IP, that is, it receives connections on an IP address of SPS, and forwards them to a server explicitly set in the policy.

The destination address is the address of the server where the clients finally connect to. To modify the destination address of a connection, complete the following steps.

Prerequisites:

- A SPS appliance where you have already completed the Welcome Wizard.
- An SSH server that is running on a host that you can access from SPS. That is, SPS must be able to access the network of the SSH server (adjust any routing and firewall settings in your network to permit this connection). If you only want to do a quick test, you can install an SSH server on the host you are configuring SPS from.

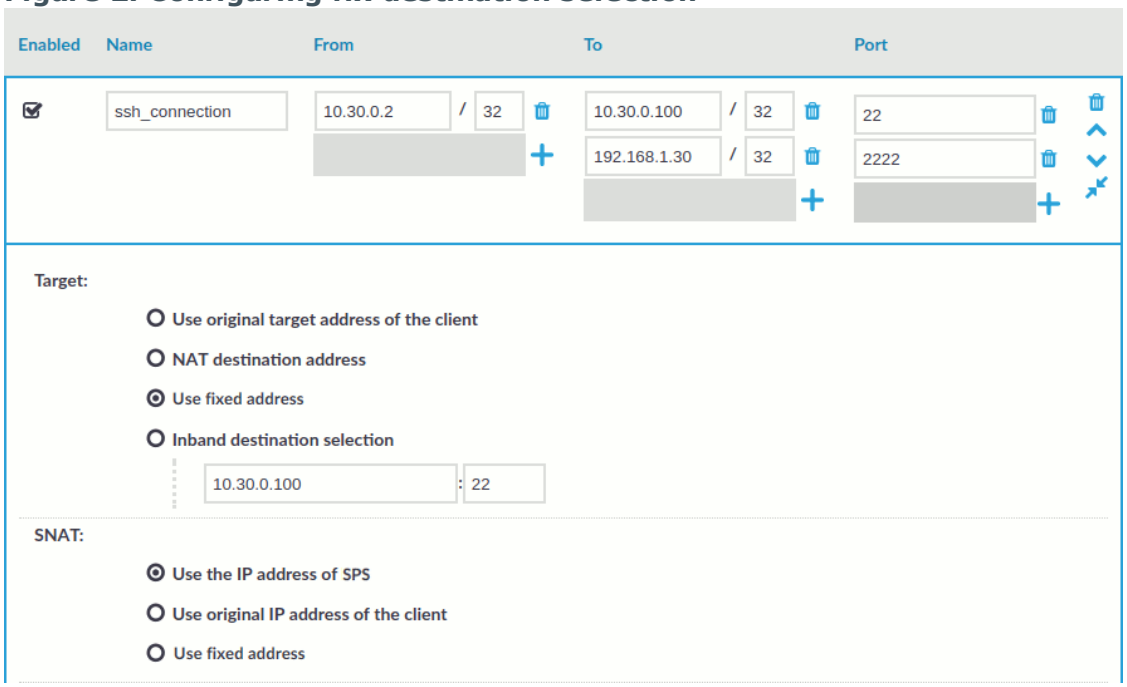
To configure a basic SSH connection in SPS

1. Navigate to **SSH 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.
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.

Enter the IP address that the clients will request into the **To** field. To test SPS the easiest is to use the IP address of SPS, meaning that the connection will be non-transparent. (To test transparent connections, you must place SPS into the network between the client and the server, or route the traffic that way.)

Figure 2: Configuring fix destination selection



4.

5. The **Target** section allows you to configure Network Address Translation (NAT) on the server side of SPS. Destination NAT determines the target IP address of the server-side connection. You can set the destination address as required for your environment. For this example non-transparent connection, select **Use fixed address**.
6. Enter the IP address and port number of the server. SPS will connect all incoming client-side connections to this server. For example, to redirect the connections to your computer (if it is running an SSH server), enter the IP address of your computer.

   You can also enter a hostname instead of the IP address, and 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.
   - Only IPv4 addresses are supported.
   - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.
7. 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. For details on organizing connections in non-transparent mode, see "Organizing connections in non-transparent mode" in the Administration Guide.

8. Click Commit to save the connection.

   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.

   TIP: To temporarily disable a connection, deselect the checkbox before the name of the connection.

9. Test the new configuration: try to initiate an SSH connection from the client (your computer) to the server.
10. After successfully connecting to the server, do something in the connection, for example, execute a simple command in SSH (for example, ls /tmp), then disconnect from the server.
11. Navigate to **Search** on the SPS web interface. Your sessions are displayed in the list of connections. Note that for the transparent connection, the client addresses the target server, while the non-transparent connection addresses SPS.
12. Click the icon. A summary will be displayed about the connection.

Server-side (only) password authentication

The default authentication method for SSH connection policies is to let the target system check credentials as it would happen when users access the server directly without SPS in place.

If you want to configure a different authentication method, create an authentication policy.

Figure 3: 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, navigate to **SSH Control > Authentication Policies**.

For details, see Authentication Policies.

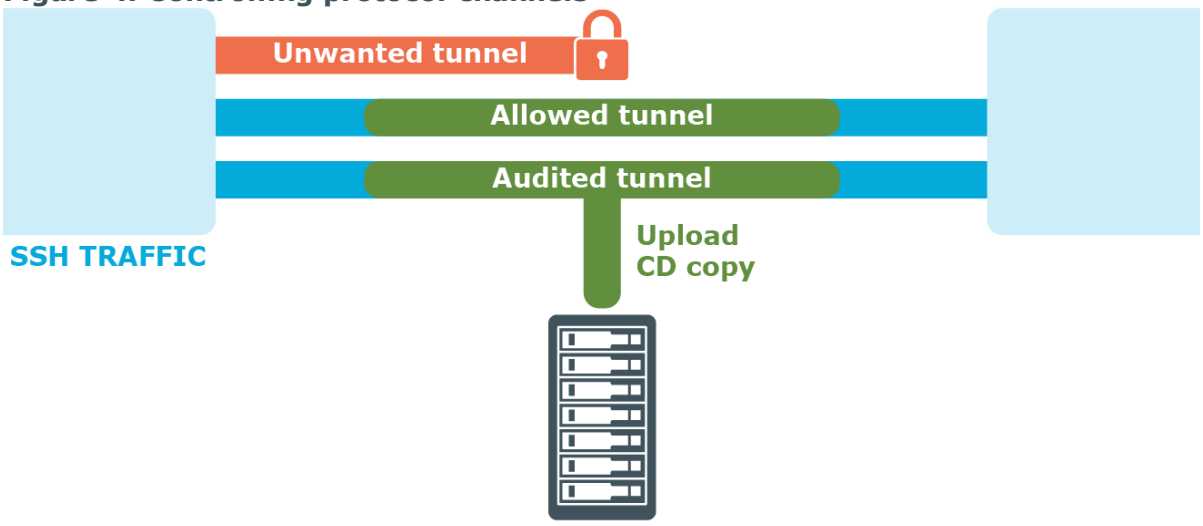
Permitting or denying access to SSH channels

For certain protocols, multiple channels are defined each of which is responsible for a specific functionality supported by the protocol. For example, the **Session Shell** channel is the traditional remote terminal session, while the **Session Exec** channel allows to execute a remote command (for example `rsync` without opening a session shell.

For details on the supported SSH channel types, see Supported SSH channel types.

SPS can permit/deny access to these functionalities based on various parameters of a connection (for example time of the day, username, and so on) to provide an additional level of access control and protection.

Figure 4: Controlling protocol channels



Access to sub-channels is controlled by channel policies. The default SSH channel policy allows session shell access only.

For details, see Creating and editing channel policies.

Configuring SCP and SFTP access in SSH

To configure SCP and SFTP access in SSH

1. Navigate to **SSH Control > Channel Policies** and click + to create a new channel policy. Enter a name for the policy into the **Channel Policy** field (for example, `shell_and_backup`).
2. Click + to add a new channel.
3. Select **Session Exec SCP** from the **Type** field.
4. Restrict the availability of the channel based on your preferences.

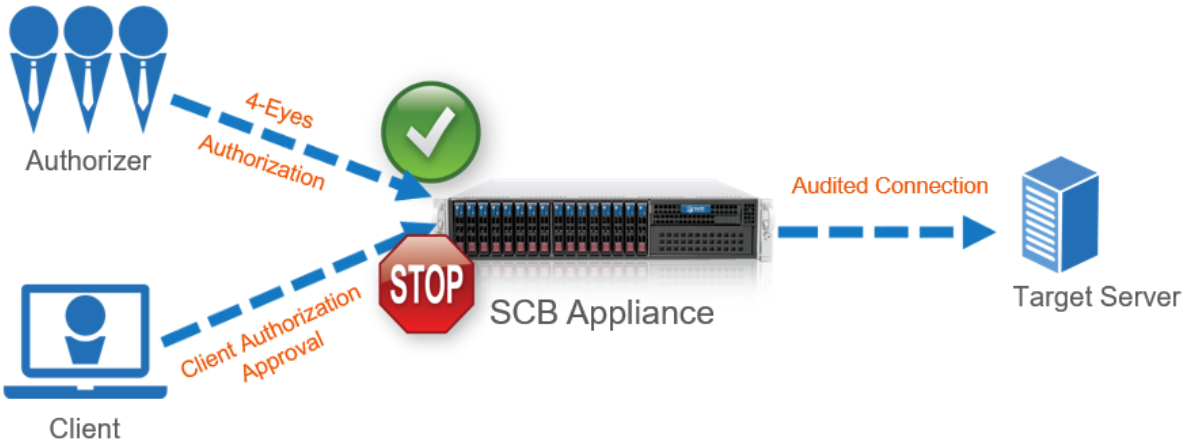
   For details, see Creating and editing channel policies.
5. To be able to extract the original file from the corresponding audit trail for further inspection, select the **Record audit trail** option to record the activities of the channel into audit trails.
6. (Optional) To also configure SFTP channel access, add a new channel and repeat the steps above, but this time, select **Session SFTP** from the **Type** field.

Authorizing and monitoring a connection personally in real-time

This is called four-eyes authorization in SPS terminology. When four-eyes authorization is required for a connection, a user (called authorizer) must authorize the connection on 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, outband 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.

Figure 5: Four-eyes authorization



For details on four-eyes authorization, see Four-eyes authorization.

Configuring four-eyes authorization

To configure four-eyes authorization

1. To enforce four-eyes authorization, navigate to **SSH Control > Connections**.
2. Select the connection policy to modify. 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** field.
4. Configure the parameters of four-eyes authorization. For details, see Configuring 4-eyes authorization.
5. Navigate to **SSH Control > Channel Policies**, and select the channel policy used in the connection.
6. Enable the **4 eyes** option for the channels which should be accessed only using four-eyes authorization.
7. Click Commit.

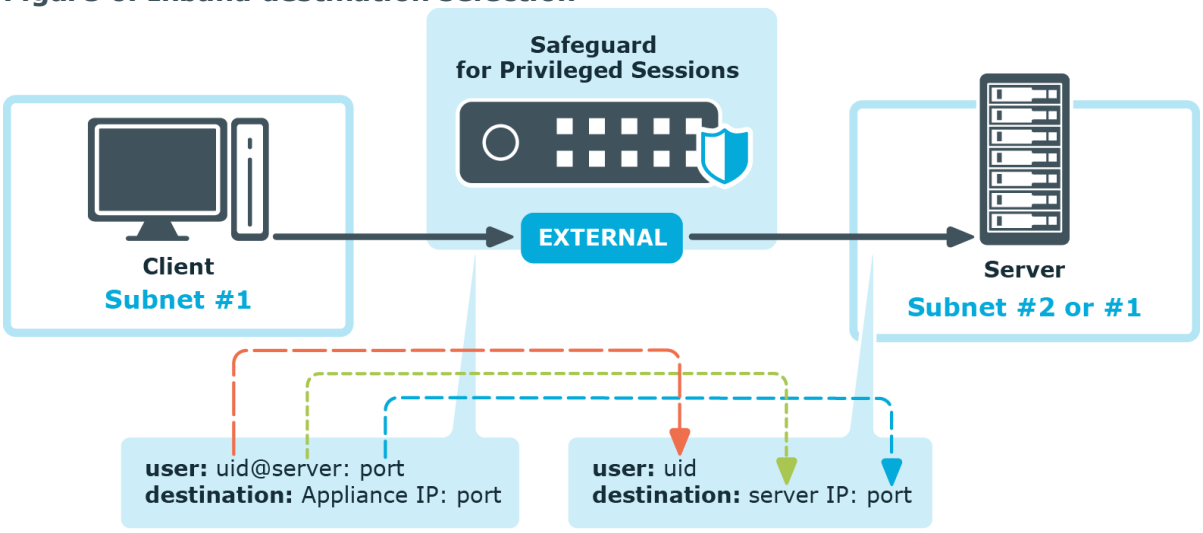
Inband destination selection

Using fix destination selection has the disadvantage of requiring one connection policy per protected server, because policies are mapped to servers based on IP addresses or port numbers.

Inband destination selection allows you to create a single connection policy and allow end-users to access any server. by including the name of the target server in their username

(for example `ssh username@targetserver:port@scb_address`). SPS can extract the address from the username and direct the connection to the target server.

Figure 6: Inband destination selection



The process looks like the following:

1. End-users specify the destination server as part of the username, for example in the format of `<username>@<server address>:<port>@<SPS address>`, where the server and SPS address can be either a hostname or an IP address.
2. SPS tokenizes the username and the server address to forward the connection to.
3. SPS forwards the connection to the server.

For details, see Using inband destination selection in SSH connections.

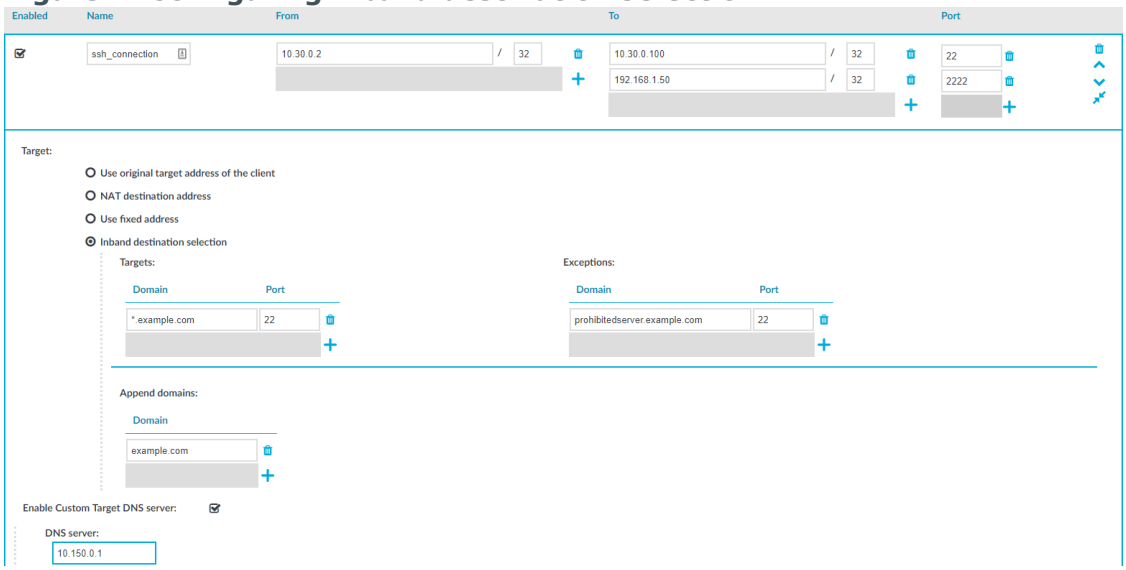
Configuring inband destination selection

The following describes how to configure a Connection Policy to extract the address of the server from the username.

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 7: Configuring inband destination selection



3. Enter the addresses of the servers that the users are permitted to access into the **Targets** field.
4. 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.
5. If there are any servers that the users cannot target using inband destination selection, add them to the **Exceptions** field.
6. Click Commit.

Example: Initiating a connection

Once the connection policy is configured correctly, a sample connection initiation would look like the following:

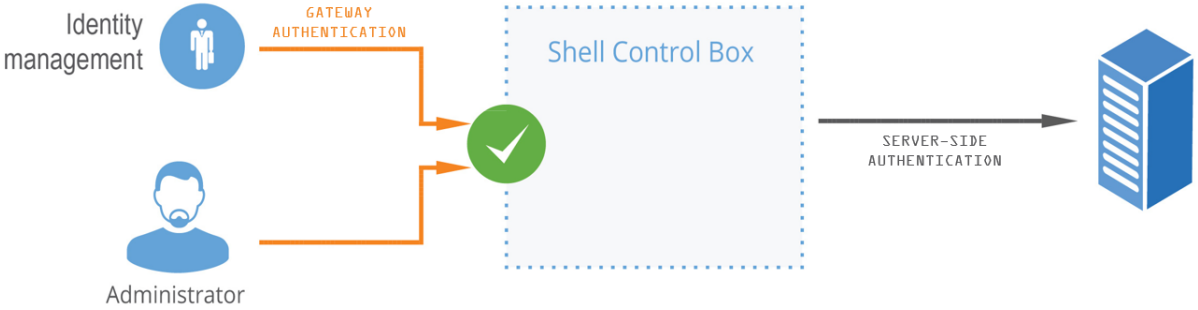
```
$ ssh root@192.168.56.10@192.168.56.200
```

- `root` = server user
- `192.168.56.10` = target server
- `192.168.56.200` = IP address of SPS

Gateway authentication

When gateway authentication is required for a connection, the user must authenticate on SPS as well. This additional authentication can be performed on the SPS web interface, so it provides a protocol-independent, outband authentication method. 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.

Figure 8: Gateway authentication



For details on gateway authentication, see The gateway authentication process.

Password-based gateway (local) + password-based server-side authentication from credential store

The goal of this scenario is to demonstrate an SSH connection in which end-users must authenticate themselves successfully with their own passwords against a local user database maintained on SPS and have a session opened to the requested destination with a different account without any further interaction (that is, have SPS complete the password-based login process).

To configure password-based gateway (local) + password-based server-side authentication from credential store

1. **Create a local user database:**

   Navigate to **Policies > Local User Databases** and create a local user database.

   For details, see Creating a Local User Database.

2. **Connect the local user database with a client-side gateway authentication policy:**

   Navigate to **SSH Control > Authentication Policies**. Create an authentication policy. Select **Authenticate the client to SPS using > Local**. Select **Password**. Configure the required settings.

   For details, see Local client-side authentication.

3. **Create a user list:**

   Navigate to **Policies > User lists** and create a user list.

   For details, see Creating and editing user lists.

4. **Create a usermapping policy:**

   Navigate to **Policies > Usermapping policies** and create a usermapping policy.

   For details, see Configuring usermapping policies.

5. Create a local or remote credential store with the server user and its password. SPS provides a plugin framework to integrate with other remote credential stores/password management systems.

   For details, see Using a custom Credential Store plugin to authenticate on the target hosts.

6. Expected outcome:

   If all prerequisites are met, SPS is ready to perform inband gateway authentication in an SSH session, which together with inband destination selection could be performed with the following connection string by an end-user:

   Example: Inband gateway authentication and destination selection

   ```
   $ ssh gu=myusername@root@192.168.56.10@192.168.56.200
   ```

   - `gu=myusername` = gateway user (myusername)
   - `root` = server user
   - `192.168.56.10` = target server
   - `192.168.56.200` = IP address of SPS

Public key-based gateway + password-based server-side authentication from credential store

This scenario differs from the previous one only in the client-side authentication method. In this case, the end-user is authenticated with the public key method, and if all permissions are granted by SPS (for example usermapping is allowed), they get logged in automatically to the requested server with the requested server account without having to enter a password.

To configure this, upload a public key for the user in the applied local user database, and make sure that the private key is accessible for the client application (openSSH, PuTTY, and so on).

To configure public key-based gateway + password-based server-side authentication from credential store

1. **Create a local user database:**

   Navigate to **Policies > Local User Databases** and create a local user database.

   For details, see Creating a Local User Database.

2. **Connect the local user database with a client-side gateway authentication policy:**

   Navigate to **SSH Control > Authentication Policies**. Create an authentication policy. Select **Authenticate the client to SPS using > Local**. Select **Public key**. Configure the required settings.

   For details, see Local client-side authentication.

3. **Create a user list:**

   Navigate to **Policies > User lists** and create a user list.

   For details, see Creating and editing user lists.

4. **Create a usermapping policy:**

   Navigate to **Policies > Usermapping policies** and create a usermapping policy.

   For details, see Configuring usermapping policies.

5. Create a local or remote credential store with the server user and its password. SPS provides a plugin framework to integrate with other remote credential stores/password management systems.

   For details, see Using a custom Credential Store plugin to authenticate on the target hosts.

6. Expected outcome:

   If all prerequisites are met, SPS is ready to perform inband gateway authentication in an SSH session, which together with inband destination selection could be performed with the following connection string by an end-user:

   Example: Inband gateway authentication and destination selection

   ```
   $ ssh gu=balabit@root@192.168.56.10@192.168.56.200
   ```

   - `gu=balabit` = gateway user (balabit)
   - `root` = server user
   - `192.168.56.10` = target server
   - `192.168.56.200` = IP address of SPS

Configuring an advanced channel policy for SSH

The channel policy lists the channels (for example, terminal session and SCP in SSH) 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 through SSH terminal, but the channel policy may restrict SCP access only to a single client. The rules defined in the channel policy are checked when the user attempts to open a particular channel type in the connection.

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. This can be helpful in the following cases:

- Four eyes authorization is required only for certain target user groups (Administrator/root)
- Group-based auditing: only certain user groups are audited
- Remote access to a certain server is only allowed during non-business hours for example to ensure that work is not interrupted because of system maintenance.

Based on the environment of the customer, such requirements could probably also be solved by creating multiple channel policies, hence having multiple-connection policies in place. But because connection policies are selected by only matching IP addresses and destination ports, especially group-based decisions cannot be solved at this stage. The third example can definitely be solved by a separate connection policy having a special channel policy assigned. But if this is the only difference to another connection policy,

solving the issue by an additional rule in the channel policy is much faster and keeps a clean ruleset.

For details, see Creating and editing channel policies.

To configure an advanced channel policy for SSH

1. Navigate to **SSH Control > Channel Policies** and click + to create a new channel policy. Enter a name for the policy.
2. Click + to add a new channel.
3. Select the channel to be enabled in the connection from the **Type** field.
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.
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.
6. 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.
7. 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.
8. Select a **Time** policy to narrow the availability of the channel.
9. **Decide which actions to trigger if the parameters configured above match:**

   Select the **4 eyes** option to require four-eyes authorization to access the channel.

   Select the **Record audit trail** option to record the activities of the channel into audit trails.
10. Click Commit.

Group-based auditing

It is sometimes a requirement to record connections of certain users only. If those uses cannot be identified by layer 3/4 connection parameters, this issue is not solved by creating separate connection policies with each having assigned a specific channel policy with auditing enabled/disabled.

To solve this issue, configure the group(s) that are affected by the channel settings:

Figure 9: Group-based auditing

In this scenario, only users of gateway groups `gr_supplier-A` and `gr_supplier-C` are audited when requesting a new SSH session shell channel. If these two groups are not matched, the second rule matches and no recording is triggered. Without the second rule, users that do not match the specified groups would be denied by this channel policy.

Integration with ticketing systems

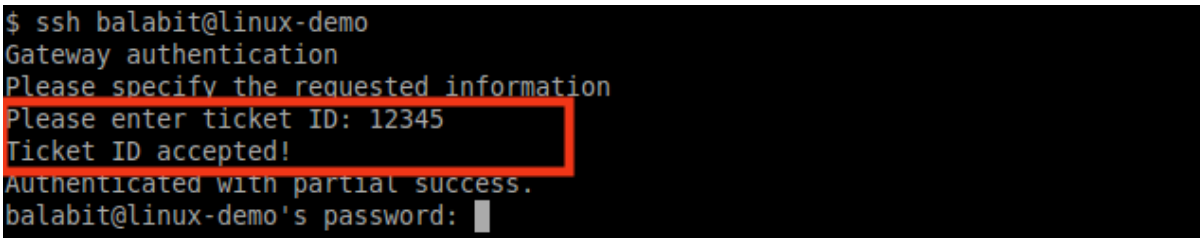
SPS provides a plugin framework to integrate SPS to external ticketing (or issue tracking) systems, allowing you to request a ticket ID from the user before authenticating on the target server. That way, SPS can verify that the user has a valid reason to access the server — and optionally terminate the connection if they do not.

Currently the following protocols are supported:

- Remote Desktop (RDP)
- Secure Shell (SSH)
- Telnet
- TN3270

The following is an example of a gateway authentication process with ticketing integration in PuTTY

Figure 10: Authentication with ticketing integration in PuTTY



For details, see Integrating ticketing systems.

Configuring connections: RDP

The following procedures will provide a skeleton of configuring RDP connections in SPS. If you want to have a deeper understanding, see the in-depth detailed procedure in Configuring connections.

Configure an RDP connection with fixed destination IP

The following describes how to configure a basic Remote Desktop (RDP) connection in SPS. This Connection Policy uses a fixed destination IP, that is, it receives connections on an IP address of SPS (on the default RDP port 3389), and forwards them to a server explicitly set in the policy.

The destination address is the address of the server where the clients finally connect to. To modify the destination address of a connection, complete the following steps.

Prerequisites:

- A SPS appliance where you have already completed the Welcome Wizard.
- A computer that accepts Remote Desktop connections (and RDP server). SPS must be able to access the network of the RDP server (adjust any routing and firewall settings in your network to permit this connection).

To configure a basic RDP connection in SPS

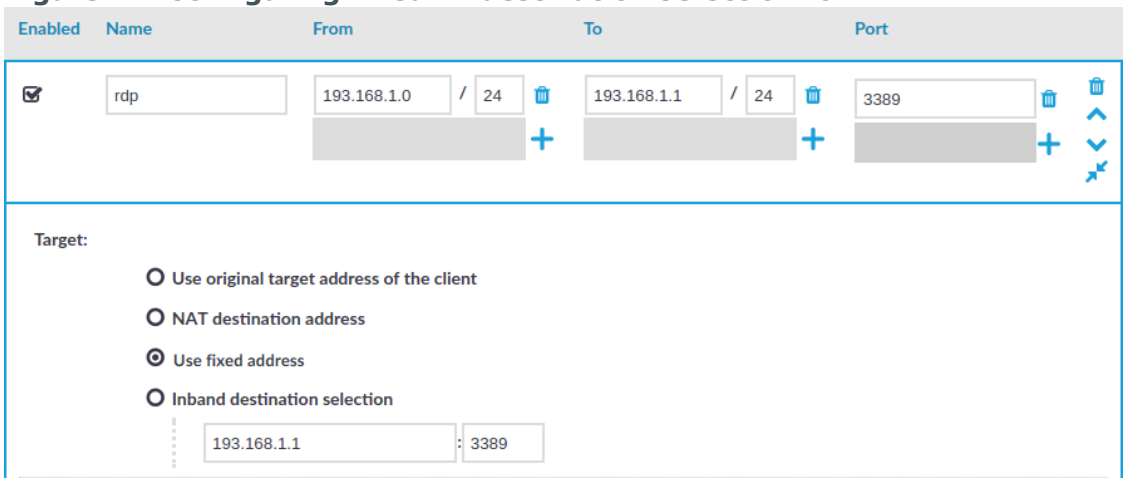
1. Navigate to **RDP Control > Connections**.
2. Click + to define a new connection and enter a name that will identify the connection (for example `admin_mainserver`).

   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.
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.

Enter the IP address that the clients will request into the **To** field. To test SPS the easiest is to use the IP address of SPS, meaning that the connection will be non-transparent. (To test transparent connections, you must place SPS into the network between the client and the server, or route the traffic that way.)

Figure 11: Configuring fixed IP destination selection for RDP



4.

5. The **Target** section allows you to configure Network Address Translation (NAT) on the server side of SPS. Destination NAT determines the target IP address of the server-side connection. You can set the destination address as required for your environment. For this example non-transparent connection, select **Use fixed address**.
6. Enter the IP address and port number of the server. SPS will connect all incoming client-side connections to this server.

   You can also enter a hostname instead of the IP address, and 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.
   - Only IPv4 addresses are supported.
   - If the Domain Name Server returns multiple IP addresses, SPS selects randomly from the list.
7. Click Commit to save the connection.

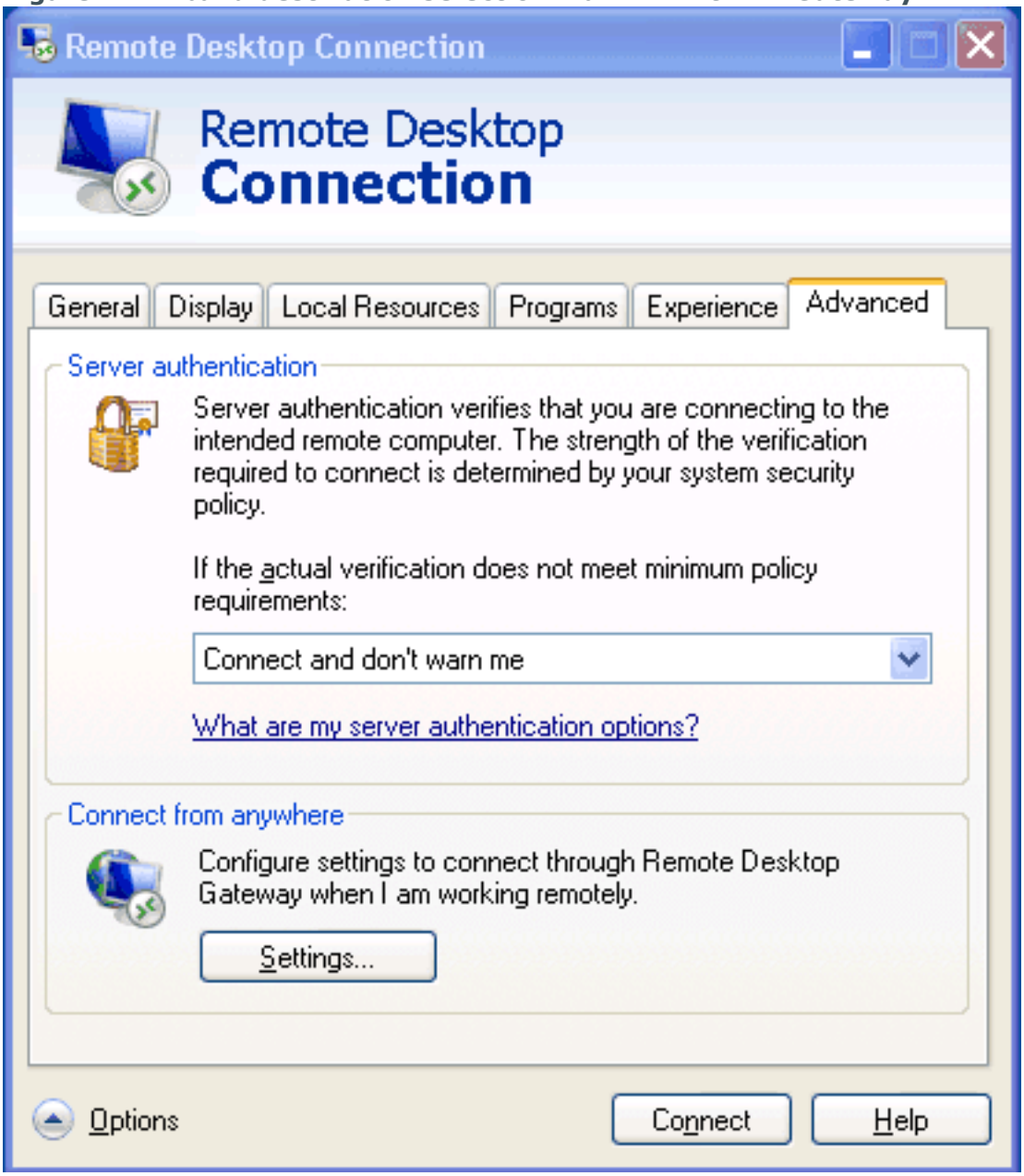
   This connection allows any user from the client machine to connect to the specified server, but permits only Desktop sessions — other RDP channels like disk redirection are disabled.
8. Test the new configuration: try to initiate an RDP connection from the client (your computer) to the server.

9. After successfully connecting to the server, do something in the connection, then disconnect from the server.
10. Navigate to **Search** on the SPS web interface. Your sessions are displayed in the list of connections. Note that for the transparent connection, the client addresses the target server, while the non-transparent connection addresses SPS.
11. Click the icon. A summary will be displayed about the connection.

Inband destination selection with Remote Desktop Gateway

Non-transparent operation with inband destination selection in RDP is supported with the implementation of the Remote Desktop Gateway protocol. When it is enabled, end-users configure their MSTSC client to use SPS as an RDP proxy/gateway and keep specifying target server addresses on the **General** tab the way they are used to.

Figure 12: Inband destination selection via RDP with RD Gateway



Configuring inband destination selection without RD Gateway

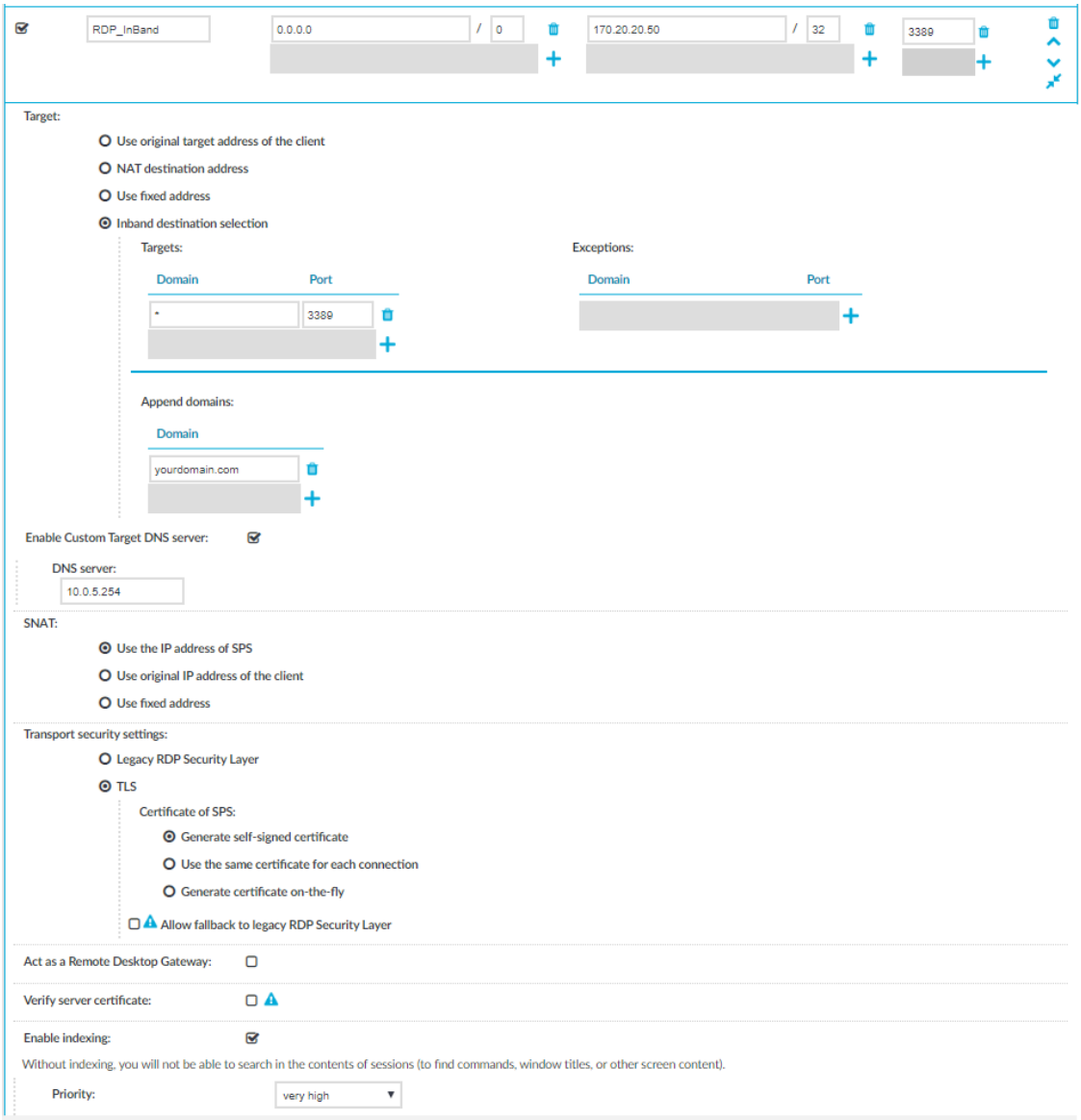
The following describes how to configure a Connection Policy to extract the address of the server from the username.

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 **RDP Control > Connections**.

   Select **Inband destination selection**.

   Figure 13: Configuring inband destination selection for Windows connections



2.
3. Enter the addresses of the servers that the users are permitted to access into the **Targets** field.
4. 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.

5. If there are any servers that the users cannot target using inband destination selection, add them to the **Exceptions** field.
6. To use inband destination selection with RDP connections without using SPS as a Remote Desktop Gateway, you must use SSL-encrypted RDP connections.

   For details, see Using TLS-encrypted RDP connections.
7. Click 

.
8. Start an RDP session from a Windows machine to SPS.

   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.

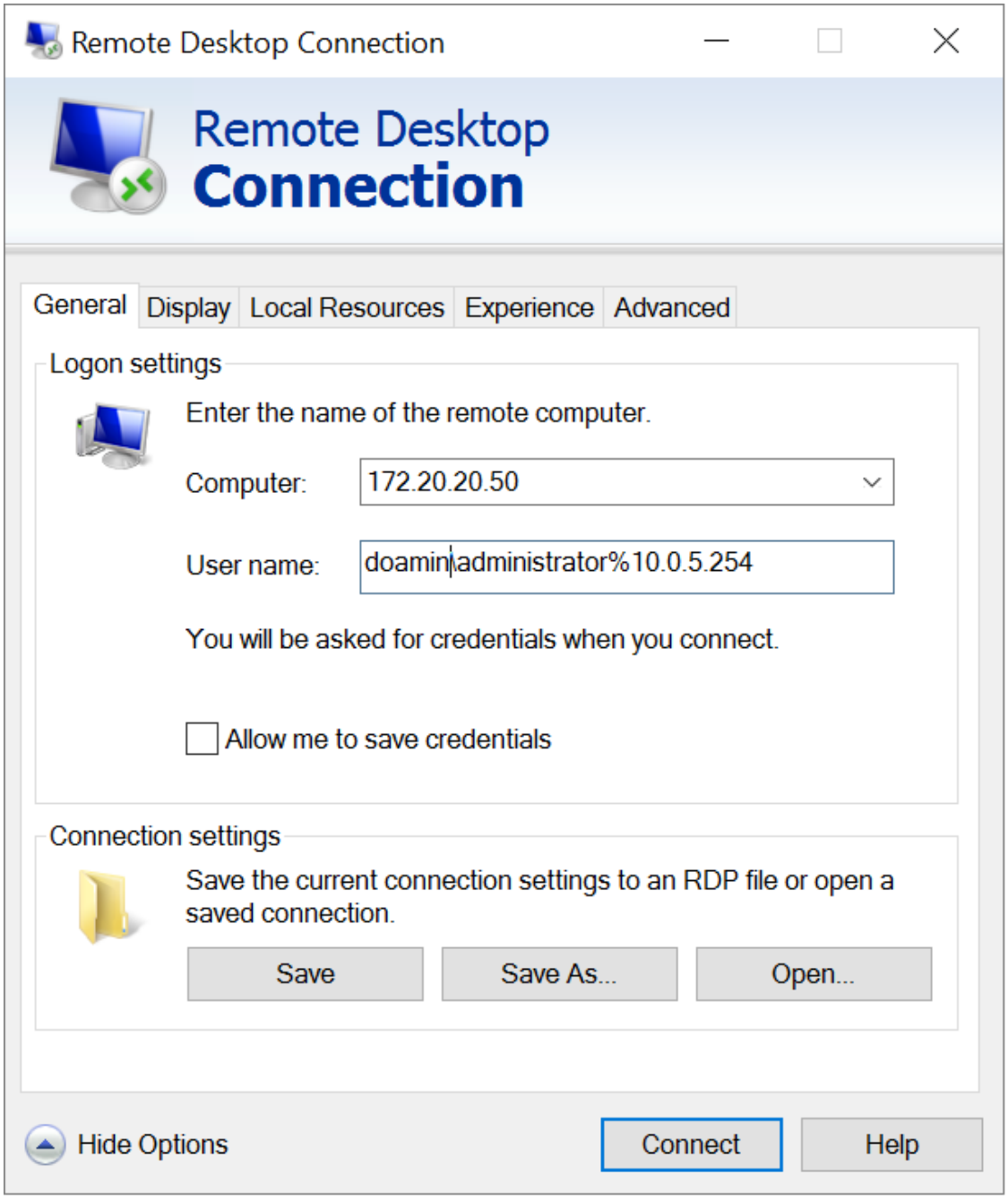
   When encoding the address of the destination server in the username, there are a few points to keep in mind. Since most RDP client applications limit which special characters can be used in usernames, this is not always intuitive.

   For the Microsoft Remote Desktop application (`mstsc`) and the login screen that SPS displays, note the following points:

   - Use the `%` character to separate the fields, for example: `username%my-targetserver`
   - Do not use the `@` character.
   - 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`.

   In the following example, a % symbol is passing the destination IP address to SPS, which redirects the connection to the proper client.

Figure 14: Configuring inband destination selection for Windows connections



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, 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 channels support content policies:

- SSH Session shell (event type: Commands/Screen Content/Credit card)
- Telnet (event type: Commands/Screen Content/Credit card)
- RDP Drawing (event type: Window title detection)
- VNC (event type: Window title detection)
- ICA Drawing (event type: Window title detection)

For details, see Real-time content monitoring with Content Policies.

NOTE: Using content policies significantly slows down connections (approximately 5 times slower), and can also cause performance problems when using the indexer service.

The following describes how to create a new content policy that performs an action if a predefined content appears in a connection.

For details, see Creating a new content policy.

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 **Event type** that you want to monitor.
3. Select **Match**, 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 **Ignore**

list.

4. To add an exception to the **Match** rule, select **Ignore**, click + and enter a string or regular expression.
5. Select the action to perform.
6. Click 

.
7. To use the content policy created in the previous steps, select the policy in the channel policy that is used to control the connections.

Indexing service

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.

For details, see Indexing audit trails.

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.

For details, see "Indexing audit trails" in the Administration Guide.

Configuring the internal indexer

Indexing audit trails allows you to search in the content of the audit trails, for example, to search for specific texts that the user has seen or typed in the session. The following describes how to configure SPS to index the audit trails. For details, see Configuring the internal indexer.

To configure SPS to index the audit trails

1. Navigate to **Basic Settings > Local Services > Indexer service**, and select **Indexer service**.
2. Define the **Maximum parallel audit trails to index on box**. The default value is set to the number of detected CPU cores.
3. (Optional) If you have encrypted audit trails and you want to index them, upload the necessary RSA keys (in PEM-encoded X.509 certificates).
4. Click 

.
5. Navigate to **Policies > Indexer Policies**.
6. To create a new Indexer Policy, click  .
7. To configure what languages to detect, select **Manual language selection**. Select the language(s) to detect.
8. Navigate to the Control page of the traffic type (for example **SSH Control**), and select the connection policy to index.
9. Select **Enable indexing**.
10. To determine the priority level of indexing this connection, select the appropriate **Priority** level.
11. Select the **Indexing Policy** to be used.
12. Click 

.
13. Check which channel policy is used in the connection, and navigate to the **Connection policies** page.
14. Select the channel policy used in the connection to index, and 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).
15. Click 

.
16. Test the new configuration: try to initiate a connection from the client (your computer) to the server.
17. 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.

18. Navigate to **Search** on the SPS web interface. Your sessions are displayed in the list of connections. Note that for the transparent connection, the client addresses the target server, while the non-transparent connection addresses SPS.
19. Click the icon. A summary will be displayed about the connection. Enter a text that was displayed in the connection into the search box, for example, the command you executed in SSH, or a menu item or other text you have seen in RDP (for example, `Start`). SPS will automatically generate a screenshot showing when the text was displayed in the connection.

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