# One Identity Safeguard for Privileged Sessions 5.11

# Starling Two-Factor Authentication-Tutorial

**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 Starling Two-Factor Authentication- Tutorial
Updated - April 2019
Version - 5.11

# Contents

# 1

# Introduction

This document describes how you can use the services of One Identity Starling 2FA to authenticate the sessions of your privileged users with One Identity Safeguard for Privileged Sessions (SPS).

### One Identity Safeguard for Privileged Sessions:

One Identity Safeguard for Privileged Sessions (SPS) controls privileged access to remote IT systems, records activities in searchable, movie-like audit trails, and prevents malicious actions. SPS is a quickly deployable enterprise device, completely independent from clients and servers — integrating seamlessly into existing networks. It captures the activity data necessary for user profiling and enables full user session drill down for forensic investigations.

SPS acts as a central authentication gateway, enforcing strong authentication before users access sensitive IT assets. SPS can integrate with remote user directories to resolve the group memberships of users who access nonpublic information. Credentials for accessing information systems can be retrieved transparently from SPS's local credential store or a third-party password management system. This method protects the confidentiality of passwords as users can never access them. When used together with Starling (or another multi-factor authentication provider), SPS directs all connections to the authentication tool, and upon successful authentication, it permits the user to access the information system.

### Integrating One Identity Starling 2FA with SPS:

SPS can interact with your Starling account and can automatically request strong multi-factor authentication for your privileged users who are accessing the servers and services protected by PSM. When used together with One Identity Starling 2FA, SPS prompts the user for a second factor authentication, and upon successful authentication, it permits the user to access the information system.

The integration adds an additional security layer to the gateway authentication performed on SPS. If the Starling 2FA App is installed on the user's device (smartphone, tablet, and so on), the user can generate a one-time password on the device. This will be used for the authentication to the One Identity platform. This way, the device turns into a two-factor authentication token for the user. The one-time password is changed after a few seconds.

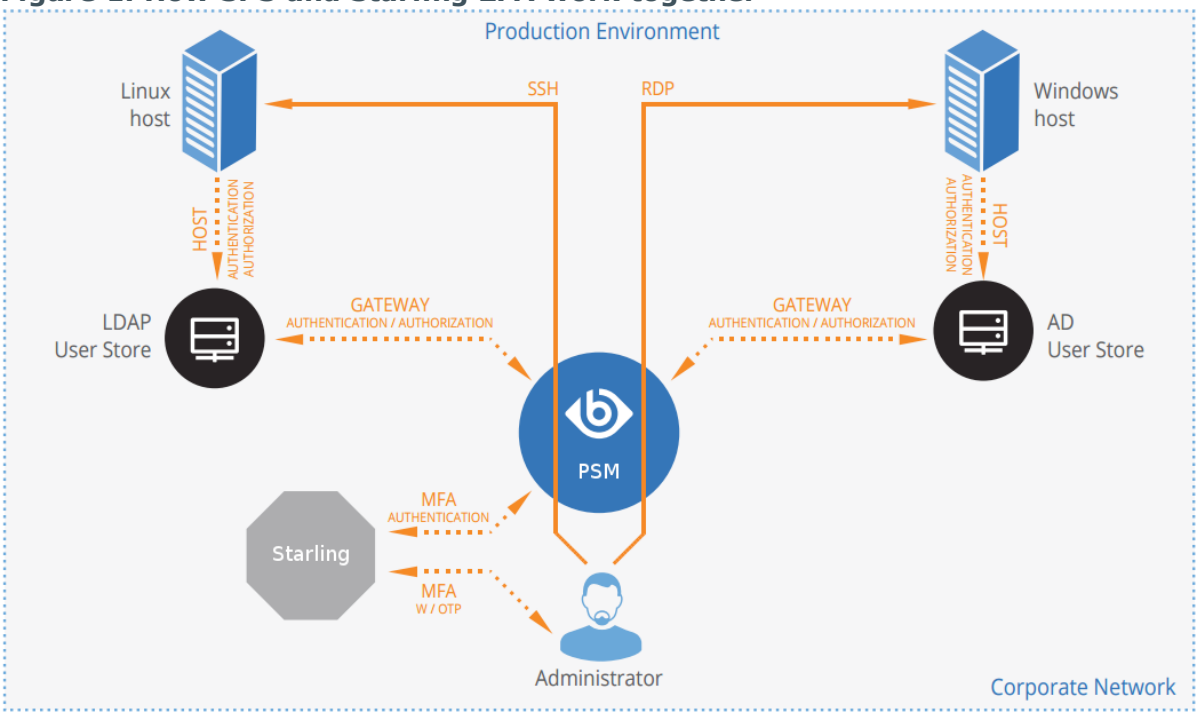
## Meet compliance requirements

ISO 27001, ISO 27018, SOC 2, and other regulations and industry standards include authentication-related requirements, for example, multi-factor authentication (MFA) for accessing production systems, and the logging of all administrative sessions. In addition to other requirements, using SPS and Starling helps you comply with the following requirements:

- PCI DSS 8.3: Secure all individual non-console administrative access and all remote access to the cardholder data environment (CDE) using multi-factor authentication.
- PART 500.12 Multi-Factor Authentication: Covered entities are required to apply multi-factor authentication for:
    - Each individual accessing the covered entity's internal systems.
    - Authorized access to database servers that allow access to nonpublic information.
    - Third parties accessing nonpublic information.
- NIST 800-53 IA-2, Identification and Authentication, network access to privileged accounts: The information system implements multi-factor authentication for network access to privileged accounts.

# How SPS and Starling 2FA work together

**Figure 1: How SPS and Starling 2FA work together**




1. A user attempts to log in to a protected server.

2. **Gateway authentication on SPS**

   SPS receives the connection request and authenticates the user. SPS can authenticate the user to a number of external user directories, for example, LDAP, Microsoft Active Directory, or RADIUS. This authentication is the first factor.

3. **Outband authentication on Starling**

   If gateway authentication is successful, SPS connects the Starling service to check which authentication methods are available for the user. Then SPS requests the second authentication factor from the user.

   - For OTP-like authentication methods, SPS requests the one-time password (OTP) from the user, and sends it to the Starling service for verification.
   - For the Starling push notification method, SPS asks the Starling service to check if the user successfully authenticated on the Starling service.

4. If multi-factor authentication is successful, the user can start working, while SPS records the user's activities. (Optionally, SPS can retrieve credentials from a local or external credential store or password vault, and perform authentication on the server with credentials that are not known to the user.)

# 3

# Technical requirements

In order to successfully connect SPS with Starling, you need the following components.

### In Starling:

- A valid Starling subscription that permits multi-factor authentication.
- Your users must be enrolled in Starling and their access must be activated. To create a new user account, log on to Starling, navigate to the **Users** tab and click **Add**.
- The users must install the Starling mobile app.
- To configure Starling 2FA Authentication in your product, you have to provide the Subscription Key that is available on the Starling 2FA Dashboard. To do this, log on to your Starling account. Navigate to **Dashboard**and click **Subscription Key**.

### In SPS:

- A One Identity Safeguard for Privileged Sessions appliance (virtual or physical), at least version 5 F1.
- A copy of the SPS Starling plugin. This plugin is an Authentication and Authorization (AA) plugin customized to work with the Starling multi-factor authentication service.
- SPS must be able to access the Internet (at least the API services). Since Starling is a cloud-based service provider, SPS must be able to access its web services to authorize the user.

  The connection also requires the API Key.
- Depending on the method you use to authenticate your users, your users might need Internet access on their cellphones.
- SPS supports AA plugins in the RDP, SSH, and Telnet protocols.
- In RDP, using an **AA plugin** together with Network Level Authentication in a Connection Policy has the same limitations as using Network Level Authentication without domain membership. For details, see "Network Level Authentication without domain membership" in the Administration Guide.
- In RDP, using an **AA plugin** requires TLS-encrypted RDP connections. For details, see "Enabling TLS-encryption for RDP connections" in the Administration Guide.

### Availability and support of the plugin

The SPS Starling plugin is available as-is, free of charge to every SPS customer from the Plugin Page. In case you need any customizations or additional features, contact professionalservices@balabit.com.

You can use the plugin on SPS 5 F5 and later. If you need to use the plugin on SPS 5 LTS, contact professionalservices@balabit.com.

# 4

# How SPS and Starling work together in detail

1. A user attempts to log in to a protected server.

2. **Gateway authentication on SPS**

   SPS receives the connection request and authenticates the user. SPS can authenticate the user to a number of external user directories, for example, LDAP, Microsoft Active Directory, or RADIUS. This authentication is the first factor.

3. **Check if the user is exempt from multi-factor authentication**

   You can configure SPS using whitelists and blacklists to selectively require multi-factor authentication for your users, for example, to create break-glass access for specific users.

   - If multi-factor authentication is not required, the user can start working, while SPS records the user's activities. The procedure ends here.
   - If multi-factor authentication is required, SPS continues the procedure with the next step.

   For details on creating exemption lists, see whitelist.

4. **Determining the Starling username**

   If the gateway usernames are different from the Starling usernames, you must configure the SPS Starling plugin to map the gateway usernames to the Starling usernames. The mapping can be as simple as appending a domain name to the gateway username, or you can query an LDAP or Microsoft Active Directory server. For details, see Mapping SPS usernames to Starling identities.

5. **Outband authentication on Starling**

   If gateway authentication is successful, SPS connects the Starling server to check which authentication factors are available for the user. Then SPS requests the second authentication factor from the user.

- For OTP-like authentication factors, SPS requests the OTP from the user, and sends it to the Starling server for verification.
- For the Starling push notification factor, SPS asks the Starling server to check if the user successfully authenticated on the Starling server.

6. If multi-factor authentication is successful, the user can start working, while SPS records the user's activities. (Optionally, SPS can retrieve credentials from a local or external credential store or password vault, and perform authentication on the server with credentials that are not known to the user.)
7. If the user opens a new session within a short period, they can do so without having to perform multi-factor authentication again. After this configurable grace period expires, the user must perform multi-factor authentication to open the next session. For details, see [cache].

# Mapping SPS usernames to Starling identities

By default, SPS assumes that the Starling username of the user is the same as the gateway username (that is, the username the user used to authenticate on SPS during the gateway authentication). To identify the users, SPS uses the username (login) field in Starling, which is an email address.

If the gateway usernames are different from the Starling usernames, you must configure the SPS Starling plugin to map the gateway usernames to the Starling usernames. You can use the following methods:

- To simply append a string to the gateway username, configure the `append_domain` parameter. In this case, SPS automatically appends the `@` character and the value of this option to the username from the session, and uses the resulting username on the Starling server to authenticate the user. For example, if the domain is set as `append_domain: example.com` and the username is `Example.User`, the SPS plugin will look for the user `Example.User@example.com` on the Starling server.
- To look up the Starling username of the user from an LDAP/Active Directory database, configure the `[ldap]` section of the SPS Starling plugin. Typically, the SPS plugin queries the email address corresponding to the username from your LDAP or Active Directory database. For details on LDAP parameters, see [ldap].
- If you configure both the `append_domain` parameter and the `[ldap]` section of the SPS Starling plugin, SPS appends the `@` character and the value of the `append_domain` parameter to the value retrieved from the LDAP database.
- If you have configured neither the `Domain` parameter nor the `[ldap]` section, SPS assumes that the Starling username of the user is the same as the gateway username.

# Bypassing Starling authentication

Having to perform multi-factor authentication to a remote server every time the user opens a session can be tedious and inconvenient for the users, and can impact their productivity. SPS offers the following methods to solve this problem:

- In SPS, the Connection policy determines the type of authentication required to access a server. If you do not need multi-factor authentication for accessing specific servers, configure your Connection policies accordingly.
- If the user opens a new session within a short period, they can do so without having to perform multi-factor authentication. After this configurable grace period expires, the user must perform multi-factor authentication to open the next session. For details, see [cache].
- You can configure SPS using whitelists and blacklists to selectively require multi-factor authentication for your users, for example, to create break-glass access for specific users. For details on creating exemption lists, see whitelist.

# Configure your Starling account for SPS

**Prerequisites:**

- Administrator access to your Starling account.
- Make sure that you have all the required components listed in Technical requirements.

1. **Add people to your Starling account.**

   The users you want to authenticate with SPS must have an activated account in Starling. For details on managing your user accounts, see Managing user accounts in the Starling documentation.

2. **Enable Multi-factor Authentication (MFA) for your organization.**

   For details on configuring the required methods for two-factor authentication, see Customizing user authentication in the Star documentation.

3. **Create an API token.**

   Navigate to *Admin* > *API* > *Tokens*, click *Create Token*, and save it.

# Configure SPS to use Starling multi-factor authentication

**Prerequisites:**

- Your Starling API token.

  > **CAUTION:**
  >
  > **According to the current Starling policies, your API token expires if it is not used for 30 days. Make sure that you use it regularly, because SPS will reject your sessions if the API token is expired.**

- Administrator access to SPS.
- Make sure that you have all the required components listed in Technical requirements.

***To configure SPS to use Starling multi-factor authentication***

1. **Download the SPS Starling plugin**

   SPS customers can download the plugin from Plugin Page.

2. **Upload the plugin to SPS**

   Upload the plugin to SPS. For details, see Administration Guide.

3. **Configure the plugin on SPS**

   The plugin includes a default configuration file, which is an ini-style configuration file with sections and name=value pairs. You can edit it on the **Policies > AA Plugin Configurations** page of the SPS web interface.

   a. Configure the usermapping settings if needed. SPS must find out which Starling user belongs to the username of the authenticated connection. For that, it can query your LDAP/Microsoft Active Directory server. For details, see Mapping SPS usernames to Starling identities.

b. Configure other parameters of your plugin as needed for your environment. For details, see SPS Starling plugin parameter reference.

4. **Configure a Connection policy and test it**

   Configure a Connection policy on SPS. In the **AA plugin** field of the Connection policy, select the SPS Starling plugin you configured in the previous step, then start a session to test it. For details on how a user can perform multi-factor authentication, see Perform multi-factor authentication with the SPS Starling plugin in terminal connections and Perform multi-factor authentication with the SPS Starling plugin in Remote Desktop connections.

   **CAUTION:**

   **According to the current Starling policies, your API token expires if it is not used for 30 days. Make sure that you use it regularly, because SPS will reject your sessions if the API token is expired.**

# 9

# SPS Starling plugin parameter reference

This section describes the available options of the SPS Starling 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.

```
[starling]
# Do NOT use api_key in production
; api_key=<Subscription-Key>
; api_url=https://api.2fa.cloud.oneidentity.com
timeout=60
rest_poll_interval=1

[users]
<exampleuser1>=123456789
<exampleuser2>=987654321

[plugin]
config_version=1
log_level=info
cred_store=<name-of-credstore-storing-sensitive-data>

[auth]
prompt=Hit Enter to send Starling push notification or provide the OTP:
whitelist=name-of-a-userlist
```

```
[username_transform]
append_domain=""

[ldap]
ldap_server_config=<SPS-LDAP-server-policy-name>
filter=(&(samAccountName={})(objectClass=user))
user_attribute=mail

[cache]
soft_timeout=15
hard_timeout=90
conn_limit=5

[question_1]
key=<name-of-name-value-pair>
prompt=<the-question-itself-in-text>
disable_echo=No

[question_2]...
```

# [starling]

This section contains the options related to your Starling account.

If you are using a Starling 2FA plugin, (that is, you have uploaded it to **Basic Settings > Plugins** and then configured it at **Policies > AA Plugin Configurations**) and the SPS node is joined to One Identity Starling, you do not have to specify `api_key` and `api_url` in the Starling 2FA plugin configuration. This configuration method is more secure.

```
[starling]
# Do NOT use api_key in production
; api_key=<Subscription-Key>
; api_url=https://api.2fa.cloud.oneidentity.com
timeout=60
rest_poll_interval=1
```

### api_key

| | |
|---|---|
| Type: | string |
| Required: | no \| yes for testing purposes if SPS is not joined to One Identity Starling |
| Default: | N/A |

**CAUTION:**

**This parameter contains sensitive data. Make sure to store this data in your local Credential Store. Type the $ value for this parameter in production.**

**For details, see "Store sensitive plugin data securely".**

**Only enter a value different than $ for this parameter in the configuration for testing purposes in a secure, non-production environment.**

*Description:* Your Subscription Key. Log on to your One Identity Starling account. Navigate to **Dashboard** and click **Subscription Key**. SPS uses this to communicate with the Starling server. For details on using a local Credential Store to host this data, read Store sensitive plugin data securely.

**CAUTION:**

**According to the current Starling policies, your API token expires if it is not used for 30 days. Make sure that you use it regularly, because SPS will reject your sessions if the API token is expired.**

### api_url

| | |
|---|---|
| Type: | string |
| Required: | yes |
| Default: | N/A |

*Description:* The URL where the One Identity Starling server can be accessed. Usually you can use the default value:

```
api_url=https://api.2fa.cloud.oneidentity.com
```

To override the access URL for the Starling API, change the value.

### timeout

| | |
|---|---|
| Type: | integer [seconds] |
| Required: | no |
| Default: | 60 |

*Description:* How long an HTTP request can take during communication with the Starling server.

### rest_poll_interval

| | |
|---|---|
| Type: | integer [seconds] |
| Required: | no |
| Default: | 1 |

*Description:* How often the plugin checks the Starling server to see if the push notification was successful.

# [users]

This section contains user-Starling 2FA application pairs.

```
[users]
<exampleuser1>=123456789
<exampleuser2>=987654321
```

### <exampleuser>

| | |
|---|---|
| Type: | integer [seconds] |
| Required: | no |
| Default: | 10 |

*Description:* To pair Starling 2FA applications with users, you have three options:

- Retrieve the name of the user an attribute of the user stored in LDAP/AD.
- Define a `[users]` section in the configuration file using the `user=uniqueid` format.

  When users install the app, they register with a mobile phone number that serves as their unique ID. Users can install the app on different devices and register with the same phone number in order to be able to have a backup device in case the primary device is inaccessible.
- Store the the user/device mapping in a credential store with the usual syntax: `host=users, user=exampleuser, password=deviceid`.

Use the second (`[users]` section) option only if there are not too many users, or for testing purposes. If there are too many users, it can cause performance issues.

# [plugin]

This section contains general plugin-related settings.

```
[plugin]
config_version=1
log_level=20
cred_store=<name-of-credstore-hosting-sensitive-data>
```

### 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: | no |
| 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, for example, the `ikey`/`skey` values in the `[starling]` section. For details, see Store sensitive plugin data securely.

### 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.

# [auth]

This section contains the options related to authentication.

```
[auth]
prompt=Hit Enter to send Starling push notification or provide the OTP:
whitelist=name-of-a-userlist
```

### prompt

| | |
|---|---|
| Type: | string |
| Required: | no |
| Default: | Hit Enter to send push notification or provide the OTP: |

*Description:* SPS displays this text to the user in a terminal connection to request an OTP interactively. The text is displayed only if the user uses an OTP-like factor, and does not send the OTP in the connection request.

```
prompt="Hit Enter to send Starling push notification or provide the OTP:"
```

### whitelist

| | |
|---|---|
| Type: | string |
| Required: | no |
| Default: | N/A |

*Description:* The name of a user list containing gateway users configured on SPS (**Policies > User Lists**). You can use this option to selectively require multi-factor authentication for your users, for example, to create break-glass access for specific users.

- If you set the **Default Policy** of the user list to `Reject`, then the list is a whitelist, so the plugin will not request Starling authentication from the users on the list.
- If you set the **Default Policy** of the user list to `Accept`, then the list is a blacklist, so the plugin will request Starling authentication only from the users on the list.

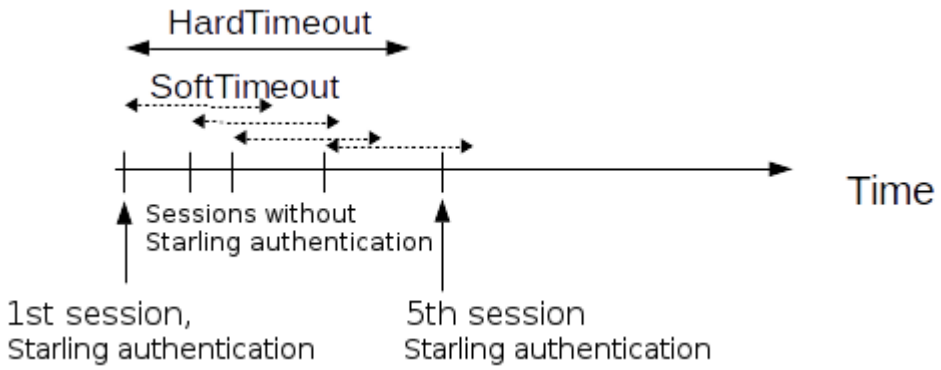
For details on creating user lists, see "Creating and editing user lists" in the Administration Guide.

# [cache]

This section contains the settings that determine how soon after performing a Starling authentication must the user repeat the authentication when opening a new session.

After the first Starling authentication of the user, SPS will not request a new Starling authentication from the user as long as the new authentications would happen within `soft_timeout` seconds from each other. After the `hard_timeout` expires (measured from the first Starling login of the user), SPS will request a new Starling authentication.

In other words, after opening the first session and authenticating on Starling, the user can keep opening other sessions without having to authenticate again on Starling as long as the time between opening any two sessions is less than `soft_timeout`, but must authenticate on Starling if `hard_timeout` expires.




```
[cache]
soft_timeout=15
hard_timeout=90
conn_limit=5
```

### soft_timeout

| | |
|---|---|
| Type: | integer [seconds] |
| Required: | yes, if you want caching |
| Default: | N/A |

*Description:* The time in seconds after which the SPS plugin requires a new Starling authentication for the next new session of the user, unless the user successfully authenticates another session within this period.

### hard_timeout

| | |
|---|---|
| Type: | integer [seconds] |

| | |
|---|---|
| Required: | yes, if you want caching |
| Default: | N/A |

*Description:* The time in seconds after which the SPS plugin requires a new Starling authentication for the next new session of the user. The time is measured from the last Starling authentication of the user.

### conn_limit

| | |
|---|---|
| Type: | integer [number of] |

*Description:* The cache can be used `conn_limit` times without multi-factor authentication. If the number of logins exceeds this number, the plugin will request multi-factor authentication again. If this parameter is not set, the number of logins from cache are unlimited.

# [ldap]

This section contains the settings you configure when you need to use an LDAP query to map the usernames from your audited sessions to the usernames in Starling.

To look up the Starling username of the user from an LDAP/Active Directory database, configure the `[ldap]` section of the SPS Starling plugin. Typically, the SPS plugin queries the email address corresponding to the username from your LDAP or Active Directory database. For details on LDAP parameters, see [ldap].

If you configure both the `append_domain` parameter and the `[ldap]` section of the SPS Starling plugin, SPS appends the `@` character and the value of the `append_domain` parameter to the value retrieved from the LDAP database.

For other methods of mapping gateway usernames to Starling usernames, see Mapping SPS usernames to Starling identities.

```
[ldap]
ldap_server_config=<SPS-LDAP-server-policy-name>
filter=(&(cn={})(objectClass=inetOrgPerson))
user_attribute=CN
```

### ldap_server_config

| | |
|---|---|
| Type: | string |
| Required: | no |
| Default: | N/A |

*Description:* The name of a configured LDAP server policy in SPS. For details on configuring LDAP policies, see "Authenticating users to an LDAP server" in the Administration Guide.

### filter

| | |
|---|---|
| Type: | string |
| Required: | no |
| Default: | (&(cn={})(objectClass=inetOrgPerson)) |

*Description:* The LDAP filter query that locates the user based on the gateway username. The plugin automatically replaces the {} characters with the gateway username from the session.

```
filter=(&(cn={})(objectClass=inetOrgPerson))
```

### user_attribute

| | |
|---|---|
| Type: | string |
| Required: | no |
| Default: | cn |

*Description:* The name of the LDAP attribute that contains the Starling username.

# [username_transform]

This section contains username transformation-related settings.

```
[username_transform]
append_domain=""
```

### append_domain

| | |
|---|---|
| Type: | string (nonrequired, no default) |
| Required: | no |
| Default: | N/A |

*Description:* If the gateway usernames are different from the Starling usernames, you must configure the SPS Starling plugin to map the gateway usernames to the Starling usernames.

To simply append a string to the gateway username, configure the `append_domain` parameter. In this case, SPS automatically appends the `@` character and the value of this option to the username from the session, and uses the resulting username on the Starling server to authenticate the user. For example, if the domain is set as `append_domain: example.com` and the username is `Example.User`, the SPS plugin will look for the user `Example.User@example.com` on the Starling server.

If you configure both the `append_domain` parameter and the `[ldap]` section of the SPS Starling plugin, SPS appends the `@` character and the value of the `append_domain` parameter to the value retrieved from the LDAP database.

For other methods of mapping gateway usernames to Starling usernames, see Mapping SPS usernames to Starling identities.

# [question_1]

| | |
|---|---|
| Type: | integer [seconds] |

*Description:* Used for communication between plugins. This is an interactive request/response right after authentication in order to supply data to credential store plugins. The question is transferred to the session cookie and all hooks of all plugins receive it.

For example, if you have an external authenticator app, you do not have to wait for the question to be prompted but can authenticate with a one-time password:

```
ssh otp=123456@root@scb
```

Name subsequent questions with the appropriate number, for example, [question_1], [question_2], and so on.

For details, see "Performing authentication with AA plugin in terminal connections" in the Administration Guide and "Performing authentication with AA plugin in Remote Desktop connections" in the Administration Guide.

### key

| | |
|---|---|
| Type: | string |
| Required: | yes |
| Default: | N/A |

*Description:* The name of the name-value pair.

### prompt

| | |
|---|---|
| Type: | string |
| Required: | yes |
| Default: | N/A |

*Description:* The question itself in text format.

### disable_echo

| | |
|---|---|
| Type: | boolean yes\|no |
| Required: | no |
| Default: | no |

*Description:* Whether the answer to the question is visible (yes), or replaced with asterisks (no).

# Store 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 (for example, api_key) of the plugin in an encrypted way.

***To store sensitive plugin data securely***

1. 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.
2. 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 (for example, `starling`).
   - Enter the name of the plugin parameter in the **USERNAME** field (for example, `api_key`).
   - Enter the value of the plugin parameter in the **PASSWORD** field.
3. Commit your changes, and navigate to the configuration of the plugin on the **Policies > AA Plugin Configurations** page.
4. In the plugin configuration file, enter the name of the local Credential Store under the `[plugin]` section, in the cred_store parameter.

# Perform multi-factor authentication with the SPS Starling plugin in terminal connections

The following describes how to establish a terminal connection (SSH, TELNET, or TN3270) to a server.

***To establish a terminal connection (SSH, TELNET, or TN3270) to a server***

1. Connect to the server.
    - If you can authenticate using an OTP or token, encode the OTP as part of the username. You can use the `@` as a field separator. For example:

      ```
      ssh otp=YOUR-ONE-TIME-PASSWORD@user@server
      ```

      Replace `YOUR-ONE-TIME-PASSWORD` with your actual OTP. If needed, you can specify the type of OTP as a prefix to the OTP. For example, to specify the OTP of a YubiKey token:

      ```
      ssh otp=y_YOUR-ONE-TIME-PASSWORD@user@server
      ```

        - Google Authenticator: `g`
        - inWebo Authenticator: `o`
        - Symantec token: `s`
        - YubiKey: `y`
        - RSA token: `r`
    - If you need to authenticate using the Starling Verify push notification, approve the connection in your mobile app.
2. If SPS prompts you for further information, enter the requested information. If you need to authenticate with an OTP, but you have not supplied the OTP in your username, you will be prompted to enter the OTP.
3. Authenticate on the server.
4. If authentication is successful, you can access the server.

# Perform multi-factor authentication with the SPS Starling plugin in Remote Desktop connections

The following describes how to establish a Remote Desktop (RDP) connection to a server when the **AA plugin** is configured.

***To establish an RDP connection to a server when the AA plugin is configured***

1. Open your Remote Desktop client application.
2. If you have to provide additional information to authenticate on the server, you must enter this information in your Remote Desktop client application in the *User name* field, before the regular content (for example, your username) of the field.

   If you can authenticate using an OTP or token, encode the OTP as part of the username. To encode additional data, you can use the following special characters:

   - `%` as a field separator
   - `~` as the equal sign
   - `^` as a colon (for example, to specify the port number or an IPv6 IP address)

   For example, use the following format:

   ```
   domain\otp~YOUR-ONE-TIME-PASSWORD%Administrator
   ```

   Replace `YOUR-ONE-TIME-PASSWORD` with your actual OTP. If needed, you can specify the type of OTP as a prefix to the OTP. For example, to specify the OTP of a YubiKey token: `domain\otp~y_YOUR-ONE-TIME-PASSWORD%Administrator`

   - Google Authenticator: `g`
   - inWebo Authenticator: `o`
   - Symantec token: `s`
   - YubiKey: `y`
   - RSA token: `r`
3. Connect to the server.

   If you need to authenticate using the Starling Authenticator push notification, approve the connection in your mobile app.

4. Authenticate on the server.
5. If authentication is successful, you can access the server.

# 13

# Learn more

To find out more about SPS, visit the One Identity page.

If you need help in connecting your Starling account with One Identity Safeguard for Privileged Sessions, contact our Sales Team or contact professionalservices@balabit.com.

# 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