Red Hat Enterprise Linux 8

Configuring authentication and authorization in RHEL

Using SSSD, authselect, and sssct1 to configure authentication and authorization
Red Hat Enterprise Linux 8 Configuring authentication and authorization in RHEL

Using SSSD, authselect, and sssctl to configure authentication and authorization
Abstract

This documentation collection provides instructions on how to configure authentication and authorization on a Red Hat Enterprise Linux 8 host.
Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE .................................................. 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION ............................ 4

CHAPTER 1. CONFIGURING USER AUTHENTICATION USING AUTHSELECT ........... 5
  1.1. WHAT IS AUTHSELECT USED FOR .............................................. 5
      1.1.1. Files and directories authselect modifies ........................... 5
      1.1.2. Data providers in /etc/nsswitch.conf ............................... 6
  1.2. CHOOSING AN AUTHSELECT PROFILE ...................................... 7
  1.3. MODIFYING A READY-MADE AUTHSELECT PROFILE ......................... 8
  1.4. CREATING AND DEPLOYING YOUR OWN AUTHSELECT PROFILE .......... 9
      Example ........................................................................... 10
  1.5. CONVERTING YOUR SCRIPTS FROM AUTHCONFIG TO AUTHSELECT .... 10

CHAPTER 2. UNDERSTANDING SSSD AND ITS BENEFITS ................................. 13
  2.1. HOW SSSD WORKS ................................................................ 13
  2.2. BENEFITS OF USING SSSD ..................................................... 13
  2.3. MULTIPLE SSSD CONFIGURATION FILES ON A PER-CLIENT BASIS .... 14
      How SSSD processes the configuration files .............................. 14
  2.4. IDENTITY AND AUTHENTICATION PROVIDERS FOR SSSD ............ 14
      Identity and Authentication Providers as SSSD domains ................ 14
      Proxy Providers .................................................................. 15
      Available Combinations of Identity and Authentication Providers ..... 15

CHAPTER 3. CONFIGURING SSSD TO USE LDAP AND REQUIRE TLS AUTHENTICATION .... 17
  3.1. AN OPENLDAP CLIENT USING SSSD TO RETRIEVE DATA FROM LDAP IN AN ENCRYPTED WAY 17
  3.2. CONFIGURING SSSD TO USE LDAP AND REQUIRE TLS AUTHENTICATION ..... 17

CHAPTER 4. SSSD CLIENT-SIDE VIEW ..................................................... 20
  4.1. OVERRIDING THE LDAP USERNAME ATTRIBUTE ............................. 20
  4.2. OVERRIDING THE LDAP UID ATTRIBUTE .................................... 21
  4.3. OVERRIDING THE LDAP GID ATTRIBUTE .................................... 23
  4.4. OVERRIDING THE LDAP HOME DIRECTORY ATTRIBUTE ................ 24
  4.5. OVERRIDING THE LDAP SHELL ATTRIBUTE ................................ 26
  4.6. LISTING OVERRIDES ON A HOST ............................................ 27
  4.7. REMOVING A LOCAL OVERRIDE ............................................. 28
  4.8. EXPORTING AND IMPORTING LOCAL VIEW .................................. 28

CHAPTER 5. CONFIGURING RHEL TO USE AD AS AN AUTHENTICATION PROVIDER ....... 29
  5.1. A STANDALONE RHEL HOST USING AD AS AN AUTHENTICATION PROVIDER 29
  5.2. CONFIGURING A RHEL HOST TO USE AD AS AN AUTHENTICATION PROVIDER 29

CHAPTER 6. REPORTING ON USER ACCESS ON HOSTS USING SSSD ................. 33
  6.1. THE SSSCTL COMMAND .......................................................... 33
  6.2. GENERATING ACCESS CONTROL REPORTS USING SSSCTL ............ 33
  6.3. DISPLAYING USER AUTHORIZATION DETAILS USING SSSCTL ........ 34

CHAPTER 7. QUERYING DOMAIN INFORMATION USING SSSD ......................... 35
  7.1. LISTING DOMAINS USING SSSCTL ........................................... 35
  7.2. VERIFYING THE DOMAIN STATUS USING SSSCTL ....................... 35

CHAPTER 8. ELIMINATING TYPOGRAPHICAL ERRORS IN LOCAL SSSD CONFIGURATION ...... 37
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.

- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the [Bugzilla](https://bugzilla.redhat.com) website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.
CHAPTER 1. CONFIGURING USER AUTHENTICATION USING AUTHSELECT

1.1. WHAT IS AUTHSELECT USED FOR

You can use the authselect utility to configure user authentication on a Red Hat Enterprise Linux 8 host.

You can configure identity information and authentication sources and providers by selecting one of the ready-made profiles:

- The default sssd profile enables the System Security Services Daemon (SSSD) for systems that use LDAP authentication.
- The winbind profile enables the Winbind utility for systems directly integrated with Microsoft Active Directory.
- The nis profile ensures compatibility with legacy Network Information Service (NIS) systems.
- The minimal profile serves only local users and groups directly from system files, which allows administrators to remove network authentication services that are no longer needed.

After selecting an authselect profile for a given host, the profile is applied to every user logging into the host.

Red Hat recommends using authselect in semi-centralized identity management environments, for example if your organization utilizes LDAP, Winbind, or NIS databases to authenticate users to use services in your domain.

**WARNING**

Do not use authselect if your host is part of Red Hat Enterprise Linux Identity Management (IdM). Joining your host to an IdM domain with the ipa-client-install command automatically configures SSSD authentication on your host.

Similarly, do not use authselect if your host is part of Active Directory via SSSD. Calling the realm join command to join your host to an Active Directory domain automatically configures SSSD authentication on your host.

1.1.1. Files and directories authselect modifies

The authconfig utility, used in previous Red Hat Enterprise Linux versions, created and modified many different configuration files, making troubleshooting more difficult. Authselect simplifies testing and troubleshooting because it only modifies the following files and directories:
The GNU C Library and other applications use this Name Service Switch (NSS) configuration file to determine the sources from which to obtain name-service information in a range of categories, and in what order. Each category of information is identified by a database name.

**/etc/nsswitch.conf**

<table>
<thead>
<tr>
<th>Database Name</th>
<th>NSS Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passwd</td>
<td>sss files</td>
<td>SSSD is the default provider for user information.</td>
</tr>
<tr>
<td>group</td>
<td>sss files</td>
<td>SSSD is the default provider for group information.</td>
</tr>
<tr>
<td>netgroup</td>
<td>sss files</td>
<td>SSSD is the default provider for NIS netgroup information.</td>
</tr>
<tr>
<td>automount</td>
<td>sss files</td>
<td>SSSD is the default provider for NFS automount information.</td>
</tr>
<tr>
<td>services</td>
<td>sss files</td>
<td>SSSD is the default provider for service information.</td>
</tr>
</tbody>
</table>

This means that the system first looks to SSSD if information concerning one of those items is requested:

- **passwd** for user information
- **group** for user group information
- **netgroup** for NIS netgroup information
- **automount** for NFS automount information
- **services** for information regarding services

Linux-PAM (Pluggable Authentication Modules) is a system of modules that handle the authentication tasks of applications (services) on the system. The nature of the authentication is dynamically configurable: the system administrator can choose how individual service-providing applications will authenticate users.

The configuration files in the `/etc/pam.d/` directory list the PAMs that will perform authentication tasks required by a service, and the appropriate behavior of the PAM-API in the event that individual PAMs fail.

Among other things, these files contain information about:

- user password lockout conditions
- the ability to authenticate with a smart card
- the ability to authenticate with a fingerprint reader

This directory holds configuration profiles for the `dconf` utility, which you can use to manage settings for the GNOME Desktop Graphical User Interface (GUI).

### 1.1.2. Data providers in `/etc/nsswitch.conf`

The default **sssd** profile establishes SSSD as a source of information by creating **sss** entries in `/etc/nsswitch.conf`:

- **passwd**: sss files
- **group**: sss files
- **netgroup**: sss files
- **automount**: sss files
- **services**: sss files

This means that the system first looks to SSSD if information concerning one of those items is requested:
Only if the requested information is not found in the \texttt{sssd} cache and on the server providing authentication, or if \texttt{sssd} is not running, the system looks at the local files, that is \texttt{/etc/*}.

For example, if information is requested about a user ID, the user ID is first searched in the \texttt{sssd} cache. If it is not found there, the \texttt{/etc/passwd} file is consulted. Analogically, if a user’s group affiliation is requested, it is first searched in the \texttt{sssd} cache and only if not found there, the \texttt{/etc/group} file is consulted.

In practice, the local files database is not normally consulted. The most important exception is the case of the root user, which is never handled by \texttt{sssd} but by files.

\section*{1.2. CHOOSING AN AUTHSELECT PROFILE}

As a system administrator, you can select a profile for the \texttt{authselect} utility for a specific host. The profile will be applied to every user logging into the host.

\subsection*{Prerequisites}
- You need root credentials to run \texttt{authselect} commands

\subsection*{Procedure}
- Select the \texttt{authselect} profile that is appropriate for your authentication provider. For example, for logging into the network of a company that uses LDAP, choose \texttt{sssd}.

\begin{verbatim}
  # authselect select sssd
\end{verbatim}

- (Optional) You can modify the default profile settings by adding the following options to the \texttt{authselect select sssd} or \texttt{authselect select winbind} command, for example:

\begin{itemize}
  \item with-faillock
  \item with-smartcard
  \item with-fingerprint
\end{itemize}

To see the full list of available options, see Section 1.5, “Converting your scripts from \texttt{authconfig} to \texttt{authselect}” or the \texttt{authselect-migration(7)} man page.

\begin{note}
Make sure that the configuration files that are relevant for your profile are configured properly before finishing the \texttt{authselect select} procedure. For example, if the \texttt{sssd} daemon is not configured correctly and active, running \texttt{authselect select} results in only local users being able to authenticate, using \texttt{pam_unix}.
\end{note}

\subsection*{Verification Steps}
1. Verify sss entries for SSSD are present in \texttt{/etc/nsswitch.conf}:

\begin{verbatim}
  passwd:  sss files
  group:   sss files
  netgroup: sss files
\end{verbatim}
automount: sss files
services: sss files
...

2. Review the contents of the /etc/pam.d/system-auth file for pam_sss.so entries:

```bash
# Generated by authselect on Tue Sep 11 22:59:06 2018
# Do not modify this file manually.
auth        required        pam_env.so
auth        required        pam_faildelay.so delay=2000000
auth        [default=1 ignore=ignore success=ok]    pam_succeed_if.so uid >= 1000 quiet
auth        [default=1 ignore=ignore success=ok]    pam_localuser.so
auth        sufficient      pam_unix.so nullok try_first_pass
auth        requisite      pam_succeed_if.so uid >= 1000 quiet_success
auth        sufficient      pam_sss.so forward_pass
auth        required        pam_deny.so

account     required        pam_unix.so
account     sufficient      pam_localuser.so
...
```

Additional Resources

- For a list of ready-made authselect profiles, see Section 1.1, “What is authselect used for”.

- If adjusting a ready-made profile by adding one of the authselect select command-line options described above is not enough for your use case, you can:
  - modify a ready-made profile by changing the /etc/authselect/user-nsswitch.conf file. For details, see Section 1.3, “Modifying a ready-made authselect profile”.
  - create your own custom profile. For details, see Section 1.4, “Creating and deploying your own authselect profile”.

1.3. MODIFYING A READY-MADE AUTHSELECT PROFILE

As a system administrator, you can modify one of the default profiles to suit your needs.

You can modify any of the items in the /etc/authselect/user-nsswitch.conf file with the exception of:

- passwd
- group
- netgroup
- automount
- services

Running authselect select profile_name afterwards will result in transferring permissible changes from /etc/authselect/user-nsswitch.conf to the /etc/nsswitch.conf file. Unacceptable changes are overwritten by the default profile configuration.
IMPORTANT
Do not modify the `/etc/nsswitch.conf` file directly.

Procedure

1. Select an `authselect` profile, for example:
   ```
   # authselect select sssd
   ```

2. Edit the `/etc/authselect/user-nsswitch.conf` file with your desired changes.

3. Apply the changes from the `/etc/authselect/user-nsswitch.conf` file:
   ```
   # authselect apply-changes
   ```

Verification steps

- Review the `/etc/nsswitch.conf` file to verify that the changes from `/etc/authselect/user-nsswitch.conf` have been propagated there.

Additional Resources

- For a list of ready-made `authselect` profiles, see Section 1.1, “What is authselect used for”.

1.4. CREATING AND DEPLOYING YOUR OWN AUTHSELECT PROFILE

As a system administrator, you can create and deploy a custom profile by making a customized copy of one of the default profiles.

This is particularly useful if Section 1.3, “Modifying a ready-made authselect profile” is not enough for your needs. When you deploy a custom profile, the profile is applied to every user logging into the given host.

Procedure

1. Create your custom profile by using the `authselect create-profile` command. For example, to create a custom profile called `user-profile` based on the ready-made `sssd` profile but one in which you can configure the items in the `/etc/nsswitch.conf` file yourself:
   ```
   # authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
   New profile was created at /etc/authselect/custom/user-profile
   ```

   Including the `--symlink-pam` option in the command means that PAM templates will be symbolic links to the origin profile files instead of their copy; including the `--symlink-meta` option means that meta files, such as README and REQUIREMENTS will be symbolic links to the origin profile files instead of their copy. This ensures that all future updates to the PAM templates and meta files in the original profile will be reflected in your custom profile, too.

   The command creates a copy of the `/etc/nsswitch.conf` file in the `/etc/authselect/custom/user-profile/` directory.

2. Configure the `/etc/authselect/custom/user-profile/nsswitch.conf` file.
3. Select the custom profile by running the `authselect select` command, and adding
   `custom/name_of_the_profile` as a parameter. For example, to select the `user-profile` profile:
   ```
   # authselect select custom/user-profile
   ```
   Selecting the `user-profile` profile for your machine means that if the `sssd` profile is
   subsequently updated by Red Hat, you will benefit from all the updates with the exception of
   updates made to the `/etc/nsswitch.conf` file.

Example
The following procedure shows how to create a profile based on the `sssd` profile which only consults the
local static table lookup for hostnames in the `/etc/hosts` file, not in the `dns` or `myhostname` databases.

1. Edit the `/etc/nsswitch.conf` file by editing the following line:
   ```
   hosts:      files
   ```
2. Create a custom profile based on `sssd` that excludes changes to `/etc/nsswitch.conf`:
   ```
   # authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
   ```
3. Select the profile:
   ```
   # authselect select custom/user-profile
   ```
4. Optionally, check that selecting the custom profile has
   - created the `/etc/pam.d/system-auth` file according to the chosen `sssd` profile
   - left the configuration in the `/etc/nsswitch.conf` unchanged:
     ```
     hosts:      files
     ```
   **NOTE**
   Running `authselect select sssd` would, in contrast, result in
   ```
   hosts:      files dns myhostname
   ```

Additional Resources
- For a list of ready-made `authselect` profiles, see Section 1.1, “What is authselect used for”.

1.5. CONVERTING YOUR SCRIPTS FROM AUTHCONFIG TO AUTHSELECT

If you use `ipa-client-install` or `realm join` to join a domain, you can safely remove any `authconfig` call in
your scripts. If this is not possible, replace each `authconfig` call with its equivalent `authselect` call. In
doing that, select the correct profile and the appropriate options. In addition, edit the necessary
configuration files:
- `/etc/krb5.conf`
- `/etc/sssd/sssd.conf` (for the `sssd` profile) or `/etc/samba/smb.conf` (for the `winbind` profile)
Relation of authconfig options to authselect profiles and Authselect profile option equivalents of authconfig options show the authselect equivalents of authconfig options.

Table 1.1. Relation of authconfig options to authselect profiles

<table>
<thead>
<tr>
<th>Authconfig options</th>
<th>Authselect profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>--enableldap --enableldapauth</td>
<td>sssd</td>
</tr>
<tr>
<td>--enablesssd --enablesssdauth</td>
<td>sssd</td>
</tr>
<tr>
<td>--enablekrb5</td>
<td>sssd</td>
</tr>
<tr>
<td>--enablewinbind --enablewinbindauth</td>
<td>winbind</td>
</tr>
<tr>
<td>--enablenis</td>
<td>nis</td>
</tr>
</tbody>
</table>

Table 1.2. Authselect profile option equivalents of authconfig options

<table>
<thead>
<tr>
<th>Authconfig option</th>
<th>Authselect profile feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>--enablesmartcard</td>
<td>with-smartcard</td>
</tr>
<tr>
<td>--enablefingerprint</td>
<td>with-fingerprint</td>
</tr>
<tr>
<td>--enableecryptfs</td>
<td>with-ecryptfs</td>
</tr>
<tr>
<td>--enablemkhomedir</td>
<td>with-mkhomedir</td>
</tr>
<tr>
<td>--enablefaillock</td>
<td>with-faillock</td>
</tr>
<tr>
<td>--enablepamaccess</td>
<td>with-pamaccess</td>
</tr>
<tr>
<td>--enablewinbindkrb5</td>
<td>with-krb5</td>
</tr>
</tbody>
</table>

Table 1.3, “Examples of authselect command equivalents to authconfig commands” shows example transformations of Kickstart calls to authconfig into Kickstart calls to authselect.

Table 1.3. Examples of authselect command equivalents to authconfig commands

<table>
<thead>
<tr>
<th>authconfig command</th>
<th>authselect equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>authconfig --enableldap --enableldapauth -- enablefaillock -- updateall</td>
<td>authselect select sssd with-faillock</td>
</tr>
<tr>
<td>authconfig --enablesssd --enablesssdauth -- enablesmartcard --smartcardmodule=sssd -- updateall</td>
<td>authselect select sssd with-smartcard</td>
</tr>
<tr>
<td>authconfig command</td>
<td>authselect equivalent</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>authconfig --enableecryptfs -- enablepamaccess --updateall</td>
<td>authselect select sssd with-ecryptfs with-pamaccess</td>
</tr>
<tr>
<td>authconfig --enablewinbind -- enablewinbindauth -- winbindjoin=Administrator --updateall</td>
<td>realm join -U Administrator --client-software=winbind WINBINDDOMAIN</td>
</tr>
</tbody>
</table>
CHAPTER 2. UNDERSTANDING SSSD AND ITS BENEFITS

2.1. HOW SSSD WORKS

The System Security Services Daemon (SSSD) is a system service that allows you to access remote directories and authentication mechanisms. You can connect a local system, an SSSD client, to an external back-end system, a provider, for example:

- An LDAP directory
- An Identity Management (IdM) domain
- An Active Directory (AD) domain
- A Kerberos realm

SSSD works in two stages:

1. It connects the client to a remote provider to retrieve identity and authentication information.
2. It uses the obtained authentication information to create a local cache of users and credentials on the client.

Users on the local system are then able to authenticate using the user accounts stored in the remote provider.

SSSD does not create user accounts on the local system. However, SSSD can be configured to create home directories for IdM users. Once created, an IdM user home directory and its contents on the client are not deleted when the user logs out.

Figure 2.1. How SSSD works

SSSD can also provide caches for several system services, such as Name Service Switch (NSS) or Pluggable Authentication Modules (PAM).

2.2. BENEFITS OF USING SSSD

Using the System Security Services Daemon (SSSD) provides multiple benefits regarding user identity retrieval and user authentication.

Offline authentication

SSSD optionally keeps a cache of user identities and credentials retrieved from remote providers. In this setup, a user – provided they have already authenticated once against the remote provider at the start of the session – can successfully authenticate to resources even if the remote provider or
the client are offline.

**A single user account: improved consistency of the authentication process**

With SSSD, it is not necessary to maintain both a central account and a local user account for offline authentication. The conditions are:

- In a particular session, the user must have logged in at least once: the client must be connected to the remote provider when the user logs in for the first time.

- Caching must be enabled in SSSD.

Without SSSD, remote users often have multiple user accounts. For example, to connect to a virtual private network (VPN), remote users have one account for the local system and another account for the VPN system. In this scenario, you must first authenticate on the private network to fetch the user from the remote server and cache the user credentials locally.

With SSSD, thanks to caching and offline authentication, remote users can connect to network resources simply by authenticating to their local machine. SSSD then maintains their network credentials.

**Reduced load on identity and authentication providers**

When requesting information, the clients first check the local SSSD cache. SSSD contacts the remote providers only if the information is not available in the cache.

### 2.3. MULTIPLE SSSD CONFIGURATION FILES ON A PER-CLIENT BASIS

The default configuration file for SSSD is `/etc/sssd/sssd.conf`. Apart from this file, SSSD can read its configuration from all `*.conf` files in the `/etc/sssd/conf.d/` directory.

This combination allows you to use the default `/etc/sssd/sssd.conf` file on all clients and add additional settings in further configuration files to extend the functionality individually on a per-client basis.

**How SSSD processes the configuration files**

SSSD reads the configuration files in this order:

1. The primary `/etc/sssd/sssd.conf` file

2. Other `*.conf` files in `/etc/sssd/conf.d/`, in alphabetical order

If the same parameter appears in multiple configuration files, SSSD uses the last read parameter.

**NOTE**

SSSD does not read hidden files (files starting with .) in the `conf.d` directory.

### 2.4. IDENTITY AND AUTHENTICATION PROVIDERS FOR SSSD

**Identity and Authentication Providers as SSSD domains**

Identity and authentication providers are configured as `domains` in the SSSD configuration file, `/etc/sssd/sssd.conf`. The providers are listed in the `[domain/name of the domain]` or `[domain/default]` section of the file.

A single domain can be configured as one of the following providers:
- An identity provider, which supplies user information such as UID and GID.
  - Specify a domain as the identity provider by using the id_provider option in the [domain/name of the domain] section of /etc/sssd/sssd.conf file.

- An authentication provider, which handles authentication requests.
  - Specify a domain as the authentication provider by using the auth_provider option in the [domain/name of the domain] section of /etc/sssd/sssd.conf.

- An access control provider, which handles authorization requests.
  - Specify a domain as the access control provider using the access_provider option in the [domain/name of the domain] section of /etc/sssd/sssd.conf. By default, the option is set to permit, which always allows all access. See the sssd.conf(5) man page for details.

- A combination of these providers, for example if all the corresponding operations are performed within a single server.
  - In this case, the id_provider, auth_provider, and access_provider options are all listed in the same [domain/name of the domain] or [domain/default] section of /etc/sssd/sssd.conf.

**NOTE**

You can configure multiple domains for SSSD. You must configure at least one domain, otherwise SSSD will not start.

**Proxy Providers**

A proxy provider works as an intermediary relay between SSSD and resources that SSSD would otherwise not be able to use. When using a proxy provider, SSSD connects to the proxy service, and the proxy loads the specified libraries.

You can configure SSSD to use a proxy provider in order to enable:

- Alternative authentication methods, such as a fingerprint scanner
- Legacy systems, such as NIS
- A local system account defined in the /etc/passwd file as an identity provider and a remote authentication provider, for example Kerberos

**Available Combinations of Identity and Authentication Providers**

You can configure SSSD to use the following combinations of identity and authentication providers.

**Table 2.1. Available Combinations of Identity and Authentication Providers**

<table>
<thead>
<tr>
<th>Identity Provider</th>
<th>Authentication Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity Management [a]</td>
<td>Identity Management</td>
</tr>
<tr>
<td>Active Directory</td>
<td>Active Directory</td>
</tr>
<tr>
<td>LDAP</td>
<td>LDAP</td>
</tr>
</tbody>
</table>
### Additional resources

- You can configure SSSD using the **authselect** utility. For more details about using **authselect**, see Chapter 1, *Configuring user authentication using authselect*.

- If your host is enrolled in Identity Management (IdM) that is in a trust agreement with an Active Directory (AD) forest, you can list and verify the status of the domains using the **sssctl** utility. For more details, see Chapter 7, *Querying domain information using SSSD*.

- You can use the **sssctl** utility to create access control reports and display user data. For more details, see Chapter 6, *Reporting on user access on hosts using SSSD*.

---

**Identity Provider | Authentication Provider**

| LDAP | Kerberos |
| Proxy | Proxy |
| Proxy | LDAP |
| Proxy | Kerberos |

[a] An extension of the LDAP provider type.
CHAPTER 3. CONFIGURING SSSD TO USE LDAP AND REQUIRE TLS AUTHENTICATION

3.1. AN OPENLDAP CLIENT USING SSSD TO RETRIEVE DATA FROM LDAP IN AN ENCRYPTED WAY

The System Security Services Daemon (SSSD) is a daemon that manages identity data retrieval and authentication on a RHEL 8 host. A system administrator can configure the SSSD on the host to use a standalone LDAP server database as the user account database. Examples of an LDAP server include the OpenLDAP server and the Red Hat Directory Server. In this chapter, the scenario also includes the requirement that the connection with the LDAP server must be encrypted with a TLS certificate.

The authentication method of the LDAP objects can be either a Kerberos password or an LDAP password. Note that the questions of authentication and authorization of the LDAP objects are not addressed in this chapter.

IMPORTANT

Configuring SSSD with LDAP is a complex procedure requiring a high level of expertise in SSSD and LDAP. Consider using an integrated and automated solution such as Active Directory or Red Hat Identity Management (IdM) instead. For details about IdM, see Planning Identity Management.

3.2. CONFIGURING SSSD TO USE LDAP AND REQUIRE TLS AUTHENTICATION

Complete this procedure to configure your Red Hat Enterprise Linux (RHEL) system as an OpenLDAP client with the following client configuration:

- The RHEL system authenticates users stored in an OpenLDAP user account database.
- The RHEL system uses the System Security Services Daemon (SSSD) service to retrieve user data.
- The RHEL system communicates with the OpenLDAP server over a TLS-encrypted connection.

NOTE

You can alternatively use this procedure to configure your RHEL system as a client of a Red Hat Directory Server.

Prerequisites

- The OpenLDAP server is installed and configured with user information.
- You have root permissions on the host you are configuring as the LDAP client.
- On the host you are configuring as the LDAP client, the /etc/sssd/sssd.conf file has been created and configured to specify ldap as the autofs_provider and the id_provider.
- You have a PEM-formatted copy of the root CA signing certificate chain from the Certificate Authority that issued the OpenLDAP server certificate, stored in a local file named core-dirsrv.ca.pem.
Procedure

1. Install the requisite packages:

   ```bash
   # dnf -y install openldap-clients sssd sssd-ldap oddjob-mkhomedir
   ```

2. Switch the authentication provider to `sssd`:

   ```bash
   # authselect select sssd with-mkhomedir
   ```

3. Copy the `core-dirsrv.ca.pem` file containing the root CA signing certificate chain from the Certificate Authority that issued the OpenLDAP server’s SSL/TLS certificate into the `/etc/openldap/certs` folder.

   ```bash
   # cp core-dirsrv.ca.pem /etc/openldap/certs
   ```

4. Add the URL and suffix of your LDAP server to the `/etc/openldap/ldap.conf` file:

   ```ini
   URI ldap://ldap-server.example.com/
   BASE dc=example,dc=com
   ```

5. In the `/etc/openldap/ldap.conf` file, add a line pointing the `TLS_CACERT` parameter to `/etc/openldap/certs/core-dirsrv.ca.pem`:

   ```ini
   # When no CA certificates are specified the Shared System Certificates
   # are in use. In order to have these available along with the ones specified
   # by TLS_CACERTDIR one has to include them explicitly:
   TLS_CACERT /etc/openldap/certs/core-dirsrv.ca.pem
   ```

6. In the `/etc/sssd/sssd.conf` file, add your environment values to the `ldap_uri` and `ldap_search_base` parameters:

   ```ini
   [domain/default]
   id_provider = ldap
   autofs_provider = ldap
   auth_provider = ldap
   chpassword_provider = ldap
   ldap_uri = ldap://ldap-server.example.com/
   ldap_search_base = dc=example,dc=com
   ldap_id_use_start_tls = True
   cache_credentials = True
   ldap_tls_cacertdir = /etc/openldap/certs
   ldap_tls_reqcert = allow
   ```

   ```ini
   [sssd]
   services = nss, pam, autofs
   domains = default
   ```

   ```ini
   [nss]
   homedir_substring = /home
   ```

7. In `/etc/sssd/sssd.conf`, specify the TLS authentication requirement by modifying the `ldap_tls_cacert` and `ldap_tls_reqcert` values in the `[domain]` section:
...  
cache_credentials = True  
ldap_tls_cacert = /etc/openldap/certs/core-dirsrv.ca.pem  
ldap_tls_reqcert = hard  
...

8. Change the permissions on the /etc/sssd/sssd.conf file:

```bash
# chmod 600 /etc/sssd/sssd.conf
```

9. Restart and enable the SSSD service and the oddjobd daemon:

```bash
# systemctl restart sssd oddjobd
# systemctl enable sssd oddjobd
```

10. (Optional) If your LDAP server uses the deprecated TLS 1.0 or TLS 1.1 protocols, switch the system-wide cryptographic policy on the client system to the LEGACY level to allow RHEL 8 to communicate using these protocols:

```bash
# update-crypto-policies --set LEGACY
```

For more details, see the Deprecated Functionality section in the RHEL 8.0 Release Notes.

Verification steps

- Verify you can retrieve user data from your LDAP server by using the id command and specifying an LDAP user:

```bash
# id ldap_user
uid=17388(ldap_user) gid=45367(sysadmins)
groups=45367(sysadmins),25395(engineers),10(wheel),1202200000(admins)
```

The system administrator can now query users from LDAP using the id command. The command returns a correct user ID and group membership.
CHAPTER 4. SSSD CLIENT-SIDE VIEW

SSSD provides the `sss_override` utility, which allows you to create a local view that displays values for POSIX user or group attributes that are specific to your local machine. You can configure overrides for all `id_provider` values, except `ipa`.

If you are using the `ipa` provider, define ID views centrally in IPA. For more information, see the **ID Views** section.

For information about a potential negative impact on the SSSD performance, see the **Potential Negative Impact on SSSD Performance** section.

4.1. OVERRIDING THE LDAP USERNAME ATTRIBUTE

As an administrator, you can configure an existing host to use accounts from LDAP. However, the values for a user (name, UID, GID, home directory, shell) in LDAP are different from the values on the local system. You can override the LDAP `username` attribute by defining a secondary `username` with the following procedure.

**Prerequisites**

- **root** access
- Installed `sssd-tools`

**Procedure**

1. Display the current information for the user:

   ```
   # id username
   
   Replace `username` with the name of the user.
   ```

2. Add the secondary `username`:

   ```
   # sss_override user-add username -n secondary-username
   
   Replace `username` with the name of the user and replace `secondary-username` with the new `username`.
   ```

3. After creating the first override using the `sss_override user-add` command, restart SSSD for the changes to take effect:

   ```
   # systemctl restart sssd
   ```

**Verification steps**

- Verify that the new `username` is added:

  ```
  # id secondary-username
  ```

- Optional. Display the overrides for the user:
Example 4.1. Defining a secondary username

To add a secondary **username** *sarah* for the user *sjones*:

1. Display the current information for the user *sjones*:

   ```
   # id sjones
   uid=1001(sjones) gid=6003 groups=6003,10(wheel)
   ```

2. Add the secondary **username**:

   ```
   # sss_override user-add sjones -n sarah
   ```

3. Verify that the new **username** has been added and overrides for the user display correctly:

   ```
   # id sarah
   uid=1001(sjones) gid=6003(sjones) groups=6003(sjones),10(wheel)
   
   # sss_override user-show sjones
   user@ldap.example.com:sarah::::::
   ```

Additional resources

- **sss_override** man page

### 4.2. OVERRIDING THE LDAP UID ATTRIBUTE

As an administrator, you can configure an existing host to use accounts from LDAP. However, the values for a user (name, UID, GID, home directory, shell) in LDAP are different from the values on the local system. You can override the LDAP UID attribute by defining a different UID with the following procedure.

**Prerequisites**

- **root** access
- Installed **sssd-tools**

**Procedure**

1. Display the current UID of the user:

   ```
   # id -u user-name
   ```

   Replace *user-name* with the name of the user.

2. Override the UID of the user’s account:
# sss_override user-add user-name -u new-UID

Replace user-name with the name of the user and replace new-UID with the new UID number.

3. Expire the in-memory cache:

```
# sss_cache --users
```

4. After creating the first override using the sss_override user-add command, restart SSSD for the changes to take effect:

```
# systemctl restart sssd
```

### Verification steps

- Verify that the new UID has been applied:

```
# id -u user-name
```

- **Optional.** Display the overrides for the user:

```
# sss_override user-show user-name
user@ldap.example.com::new-UID::::::
```

#### Example 4.2. Overriding the UID of the user

To override the UID of the user sarah with UID 6666:

1. Display the current UID of the user sarah:

```
# id -u sarah
1001
```

2. Override the UID of the user sarah's account with UID 6666:

```
# sss_override user-add sarah -u 6666
```

3. Manually expire the in-memory cache:

```
# sss_cache --users
```

4. Restart SSSD for the changes to take effect:

```
# systemctl restart sssd
```

5. Verify that the new UID is applied and overrides for the user display correctly:

```
# id sarah
6666

# sss_override user-show sarah
user@ldap.example.com::6666::::::
```
4.3. OVERRIDING THE LDAP GID ATTRIBUTE

As an administrator, you can configure an existing host to use accounts from LDAP. However, the values for a user (name, UID, GID, home directory, shell) in LDAP are different from the values on the local system. You can override the LDAP GID attribute by defining a different GID with the following procedure.

Prerequisites

- root access
- Installed sssd-tools

Procedure

1. Display the current GID of the user:

   ```
   # id -g user-name
   ```

   Replace user-name with the name of the user.

2. Override the GID of the user’s account:

   ```
   # sss_override user-add user-name -u new-GID
   ```

   Replace user-name with the name of the user and replace new-GID with the new GID number.

3. Expire the in-memory cache:

   ```
   # sss_cache --users
   ```

4. After creating the first override using the sss_override user-add command, restart SSSD for the changes to take effect:

   ```
   # systemctl restart sssd
   ```

Verification steps

- Verify that the new GID is applied:

  ```
  # id -g user-name
  ```

- Optional. Display the overrides for the user:

  ```
  # sss_override user-show user-name
  ```

  user@ldap.example.com:::6666:::

Additional resources

- sss_override man page
Example 4.3. Overriding the GID of the user

To override the GID of the user sarah with GID 6666:

1. Display the current GID of the user sarah:
   ```
   # id -g sarah
   6003
   ```

2. Override the GID of the user sarah's account with GID 6666:
   ```
   # sss_override user-add sarah -g 6666
   ```

3. Manually expire the in-memory cache:
   ```
   # sss_cache --users
   ```

4. If this is your first override, restart SSSD for the changes to take effect:
   ```
   # systemctl restart sssd
   ```

5. Verify that the new GID is applied and overrides for the user display correctly:
   ```
   # id -g sarah
   6666
   # sss_override user-show sarah
   user@ldap.example.com::6666:::::
   ```

Additional resources

- **sss_override** man page

### 4.4. OVERRIDING THE LDAP HOME DIRECTORY ATTRIBUTE

As an administrator, you can configure an existing host to use accounts from LDAP. However, the values for a user (name, UID, GID, home directory, shell) in LDAP are different from the values on the local system. You can override the LDAP home directory attribute by defining a different home directory with the following procedure.

**Prerequisites**

- **root** access
- Installed **sssd-tools**

**Procedure**

1. Display the current home directory of the user:
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/home-directory:/bin/bash

Replace user-name with the name of the user.

2. Override the home directory of the user:

# sss_override user-add user-name -h new-home-directory

Replace user-name with the name of the user and replace new-home-directory with the new home directory.

3. Restart SSSD for the changes to take effect:

# systemctl restart sssd

Verification steps

- Verify that the new home directory is defined:

# getent passwd user-name
user-name:x:XXXX:XXXX::/new-home-directory:/bin/bash

- Optional. Display the overrides for the user:

# sss_override user-show user-name
user@ldap.example.com:::::::new-home-directory::

Example 4.4. Overriding the home directory of the user

To override the home directory of the user sarah with admin:

1. Display the current home directory of the user sarah:

# getent passwd sarah
sarah:x:1001:6003::sarah:/bin/bash

2. Override the home directory of the user sarah with new home directory admin:

# sss_override user-add sarah -h admin

3. Restart SSSD for the changes to take effect:

# systemctl restart sssd

4. Verify that the new home directory is defined and overrides for the user display correctly:

# getent passwd sarah
sarah:x:1001:6003::admin:/bin/bash

# sss_override user-show user-name
user@ldap.example.com:::::::admin::
4.5. OVERRIDING THE LDAP SHELL ATTRIBUTE

As an administrator, you can configure an existing host to use accounts from LDAP. However, the values for a user (name, UID, GID, home directory, shell) in LDAP are different from the values on the local system. You can override the LDAP shell attribute by defining a different shell with the following procedure.

Prerequisites

- root access
- Installed sssd-tools

Procedure

1. Display the current shell of the user:

   ```
   # getent passwd user-name
   user-name:x:XXXX:XXXX::/home/home-directory:/bin/bash
   ```

   Replace user-name with the name of the user.

2. Override the shell of the user:

   ```
   # sss_override user-add user-name -s new-shell
   ```

   Replace user-name with the name of the user and replace new-shell with the new shell.

3. Restart SSSD for the changes to take effect:

   ```
   # systemctl restart sssd
   ```

Verification steps

- Verify that the new shell is defined:

  ```
  # getent passwd user-name
  user-name:x:XXXX:XXXX::/home/home-directory:new-shell
  ```

- Optional. Display the overrides for the user:

  ```
  # sss_override user-show user-name
  user@ldap.example.com::::::new-shell:
  ```

Example 4.5. Overriding the shell of the user

To change the shell of the user sarah from /bin/bash to sbin/nologin:
1. Display the current shell of the user sarah:

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/bin/bash
```

2. Override the shell of the user sarah with new `/sbin/nologin` shell:

```
# sss_override user-add sarah -s /sbin/nologin
```

3. Restart SSSD for the changes to take effect:

```
# systemctl restart sssd
```

4. Verify that the new shell is defined and overrides for the user display correctly:

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/sbin/nologin

# sss_override user-show user-name
user@ldap.example.com::::::/sbin/nologin:
```

Additional resources

- `sss_override` man page

### 4.6. LISTING OVERRIDES ON A HOST

As an administrator, you can list all user and group overrides on a host to verify that the correct attributes have been overridden.

**Prerequisites**

- **root** access
- Installed `sssd-tools`

**Procedure**

- List all user overrides:

```
# sss_override user-find
user1@ldap.example.com::8000::::::/bin/zsh:
user2@ldap.example.com::8001::::::/bin/bash:
...
```

- List all group overrides:

```
# sss_override group-find
group1@ldap.example.com::7000
group2@ldap.example.com::7001
...
```
### 4.7. REMOVING A LOCAL OVERRIDE

If you want to remove local override that is defined in the global LDAP directory, use the following procedure.

**Prerequisites**
- **root** access
- Installed `sssd-tools`

**Procedure**
- To remove the override for a user account, use:
  
  ```
  # sss_override user-del user-name
  ```
  
  Replace `user-name` with the name of the user. The changes take effect immediately.

- To remove an override for a group, use:
  
  ```
  # sss_override group-del group-name
  ```

- After removing the first override using the `ssss_override user-del` or `ssss_override group-del` command, restart SSSD for the changes to take effect:

  ```
  # systemctl restart sssd
  ```

When you remove overrides for a user or group, all overrides for this object are removed.

### 4.8. EXPORTING AND IMPORTING LOCAL VIEW

Your local overrides are stored in the local SSSD cache. You can export user and group overrides from this cache to a file to create a backup. This ensures that even if the cache is cleared, you can restore the configurations later.

**Prerequisites**
- **root** access
- Installed `sssd-tools`

**Procedure**
- To back up user and group view, use:

  ```
  # sss_override user-export /var/lib/sss/backup/sssd_user_overrides.bak
  # sss_override group-export /var/lib/sss/backup/sssd_group_overrides.bak
  ```

- To restore user and group view, use:

  ```
  # sss_override user-import /var/lib/sss/backup/sssd_user_overrides.bak
  # sss_override group-import /var/lib/sss/backup/sssd_group_overrides.bak
  ```
CHAPTER 5. CONFIGURING RHEL TO USE AD AS AN AUTHENTICATION PROVIDER

5.1. A STANDALONE RHEL HOST USING AD AS AN AUTHENTICATION PROVIDER

As a system administrator, you can use Active Directory (AD) as the authentication provider for a Red Hat Enterprise Linux (RHEL) host without joining the host to AD if, for example:

- You do not want to grant AD administrators the control over enabling and disabling the host.
- The host, which can be a corporate PC, is only meant to be used by one user in your company.

**IMPORTANT**

Implement this procedure only in the rare cases where this approach is preferred.

Consider fully joining the system to AD or Red Hat Identity Management (IdM) instead. Joining the RHEL host to a domain makes the setup easier to manage. If you are concerned about client access licences related to joining clients into AD directly, consider leveraging an IdM server that is in a trust agreement with AD. For more information on an IdM-AD trust, see [Planning a cross-forest trust between IdM and AD](#) and [Installing a trust between IdM and AD](#).

5.2. CONFIGURING A RHEL HOST TO USE AD AS AN AUTHENTICATION PROVIDER

Complete this procedure to enable the user named `AD_user` to log in to the `rhel8_host` system using the password set in the Active Directory AD user database in the `example.com` domain. In this example, the `EXAMPLE.COM` Kerberos realm corresponds to the `example.com` domain.

**Prerequisites**

- You have root access to `rhel8_host`.
- The `AD_user` user account exists in the `example.com` domain.
- The Kerberos realm is `EXAMPLE.COM`.
- `rhel8_host` has not been joined to AD using the `realm join` command.

**Procedure**

1. Create the `AD_user` user account locally without assigning a password to it:
   ```bash
   # useradd AD_user
   ```

2. Open the `/etc/nsswitch.conf` file for editing, and make sure that it contains the following lines:
   ```conf
   passwd:     sss files systemd
   group:      sss files systemd
   shadow:     files sss
   ```
3. Open the `/etc/krb5.conf` file for editing, and make sure that it contains the following sections and items:

```plaintext
# To opt out of the system crypto-policies configuration of krb5, remove the
# symlink at /etc/krb5.conf.d/crypto-policies which will not be recreated.
includedir /etc/krb5.conf.d/

[logging]
default = FILE:/var/log/krb5libs.log
dc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
spake_preauth_groups = edwards25519
default_realm = EXAMPLE.COM
default_ccache_name = KEYRING:persistent:%{uid}

[realms]
EXAMPLE.COM = {
    kdc = ad.example.com
    admin_server = ad.example.com
}

[domain_realm]
.example.com = EXAMPLE.COM
test.com = EXAMPLE.COM
```

4. Create the `/etc/sssd/sssd.conf` file and insert the following sections and lines into it:

```plaintext
[sssd]
services = nss, pam
domains = EXAMPLE.COM

[domain/EXAMPLE.COM]
id_provider = files
auth_provider = krb5
krb5_realm = EXAMPLE.COM
krb5_server = ad.example.com
```

5. Change the permissions on the `/etc/sssd/sssd.conf` file:

```plaintext
# chmod 600 /etc/sssd/sssd.conf
```

6. Start the Security System Services Daemon (SSSD):

```plaintext
# systemctl start sssd
```

7. Enable SSSD:

```plaintext
- 
```
8. Open the `/etc/pam.d/system-auth` file, and modify it so that it contains the following sections and lines:

```
# Generated by authselect on Wed May  8 08:55:04 2019
# Do not modify this file manually.
auth  required             pam_env.so
auth  required             pam_faildelay.so delay=2000000
auth  [default=1 ignore=ignore success=ok]  pam_succeed_if.so uid >= 1000 quiet
auth  [default=1 ignore=ignore success=ok]  pam_localuser.so
auth  sufficient           pam_unix.so nullok try_first_pass
auth  sufficient           pam_succeed_if.so uid >= 1000 quiet_success
auth  sufficient           pam_sss.so forward_pass
auth  required             pam_deny.so
account required           pam_unix.so
account sufficient         pam_localuser.so
account sufficient         pam_succeed_if.so uid < 1000 quiet
account [default=bad success=ok user_unknown=ignore] pam_sss.so
account required           pam_permit.so
password requisite         pam_pwquality.so try_first_pass local_users_only
password sufficient        pam_unix.so sha512 shadow nullok try_first_pass
use_authtok
password sufficient        pam_sss.so use_authtok
password required          pam_deny.so
session optional           pam_keyinit.so revoke
session required           pam_limits.so
-ses-sion optional         pam_systemd.so
session [success=1 default=ignore]  pam_succeed_if.so service in crond quiet
use_uid
session required           pam_unix.so
session optional           pam_sss.so
```

9. Copy the contents of the `/etc/pam.d/system-auth` file into the `/etc/pam.d/password-auth` file. Enter `yes` to confirm the overwriting of the current contents of the file:

```
# cp /etc/pam.d/system-auth /etc/pam.d/password-auth
cp: overwrite `/etc/pam.d/password-auth'? yes
```

**Verification steps**

1. Request a Kerberos ticket-granting ticket (TGT) for **AD_user**. Enter the password of **AD_user** as requested:

```
# kinit AD_user
Password for AD_user@EXAMPLE.COM:
```

2. Display the obtained TGT:

```
# klist
```
Ticket cache: KEYRING:permanent:0:0
Default principal: AD_user@EXAMPLE.COM

Valid starting     Expires            Service principal
11/02/20 04:16:38  11/02/20 14:16:38  krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 18/02/20 04:16:34

AD_user has successfully logged in to rhel8_host using the credentials from the EXAMPLE.COM Kerberos domain.
CHAPTER 6. REPORTING ON USER ACCESS ON HOSTS USING SSSD

The Security System Services Daemon (SSSD) tracks which users can or cannot access clients. This chapter describes creating access control reports and displaying user data using the sssctl tool.

Prerequisites

- SSSD packages are installed in your network environment.

6.1. THE SSSCTL COMMAND

sssctl is a command-line tool using Security System Services Daemon (SSSD) to gather information about:

- domain state
- client user authentication
- user access on clients of a particular domain
- information about cached content

With the sssctl tool, you can:

- manage the SSSD cache
- manage logs
- check configuration files

**NOTE**

The sssctl tool replaces sss_cache and sss_debuglevel tools.

Additional resources

- For details about sssctl, enter:

  ```
  # sssctl --help
  ```

6.2. GENERATING ACCESS CONTROL REPORTS USING SSSCTL

You can list the access control rules applied to the machine on which you are running the report because SSSD controls which users can log in to the client.

**NOTE**

The access report is not accurate because the tool does not track users locked out by the Key Distribution Center (KDC).
6.3. DISPLAYING USER AUTHORIZATION DETAILS USING SSSCTL

The `sssctl user-checks` command helps debug problems in applications that use the System Security Services Daemon (SSSD) for user lookup, authentication, and authorization.

The `sssctl user-checks [USER_NAME]` command displays user data available through Name Service Switch (NSS) and the InfoPipe responder for the D-Bus interface. The displayed data shows whether the user is authorized to log in using the `system-auth` Pluggable Authentication Module (PAM) service.

The command has two options:

- `-a` for a PAM action
- `-s` for a PAM service

If you do not define `-a` and `-s` options, the `sssctl` tool uses default options: `-a acct -s system-auth`.

Prerequisites

- You must be logged in with administrator privileges
- The `sssctl` tool is available on RHEL 7 and RHEL 8 systems

Procedure

To display user data for a particular user, enter:

```
[root@client1 ~]# sssctl user-checks -a acct -s sshd example.user
user: example.user
action: acct
service: sshd
....
```

Additional resources

- For details on `sssctl user-checks`, use the following command:

```
sssctl user-checks --help
```
CHAPTER 7. QUERYING DOMAIN INFORMATION USING SSSD

Security System Services Daemon (SSSD) can list domains in Identity Management (IdM), including Active Directory domains in the cross-forest trust. You can also verify the status of each of the listed domains:

- Listing domains using the sssctl command
- Verifying the domain status using the sssctl command

7.1. LISTING DOMAINS USING SSSCTL

The sssctl domain-list command helps debug problems with the domain topology.

**NOTE**

The status might not be available immediately. If the domain is not visible, repeat the command.

**Prerequisites**

- You must be logged in with administrator privileges
- The sssctl is available on RHEL 7 and RHEL 8 systems

**Procedure**

1. To display help for the sssctl command, enter:

   ```bash
   [root@client1 ~]# sssctl --help
   ....
   ```

2. To display a list of available domains, enter:

   ```bash
   [root@client1 ~]# sssctl domain-list
   implicit_files
   idm.example.com
   ad.example.com
   sub1.ad.example.com
   ```

   The list includes domains in the cross-forest trust between Active Directory and Identity Management.

7.2. VERIFYING THE DOMAIN STATUS USING SSSCTL

The sssctl domain-status command helps debug problems with the domain topology.

**NOTE**

The status might not be available immediately. If the domain is not visible, repeat the command.

**Prerequisites**
You must be logged in with administrator privileges

The `sssctl` is available on RHEL 7 and RHEL 8 systems.

Procedure

1. To display help for the `sssctl` command, enter:

   ```bash
   [root@client1 ~]# sssctl --help
   ```

2. To display user data for a particular domain, enter:

   ```bash
   [root@client1 ~]# sssctl domain-status idm.example.com
   Online status: Online
   Active servers:
   IPA: server.idm.example.com
   Discovered IPA servers:
   - server.idm.example.com
   ```

   The domain `idm.example.com` is online and visible from the client where you applied the command.

   If the domain is not available, the result is:

   ```bash
   [root@client1 ~]# sssctl domain-status ad.example.com
   Unable to get online status
   ```
CHAPTER 8. ELIMINATING TYPOGRAPHICAL ERRORS IN LOCAL SSSD CONFIGURATION

You can test if the /etc/sssd/sssd.conf file on your host contains any typographical errors using the sssctl config-check command.

Prerequisites

- You are logged in as root.

Procedure

1. Enter the sssctl config-check command:

```
# sssctl config-check
```

Issues identified by validators: 1
[rule/allowed_domain_options]: Attribute 'ldap_search' is not allowed in section 'domain/example1'. Check for typos.

Messages generated during configuration merging: 0

Used configuration snippet files: 0

2. Open the /etc/sssd/sssd.conf file and correct the typo. If you, for example, received the error message in the previous step, replace ldap_search with ldap_search_base:

```
[...]
domain/example1
ldap_search_base = dc=example,dc=com
[...]
```

3. Save the file.

4. Restart SSSD:

```
# systemctl restart sssd
```

Verification steps

- Enter the sssctl config-check command:

```
# sssctl config-check
```

Issues identified by validators: 0

Messages generated during configuration merging: 0

Used configuration snippet files: 0

The /etc/sssd/sssd.conf file now has no typographical errors.