Oracle® Linux 8

Setting Up System Users and Authentication
Abstract

This manual provides security guidelines for the Oracle Linux 8 operating system.

Document generated on: 2020-02-12 (revision: 9256)
# Table of Contents

Preface ................................................................................................. v  
1 Configuring and Managing Local Accounts ........................................ 1  
  1.1 About User and Group Configuration ........................................... 1  
  1.2 About Files Storing User and Group Information ......................... 1  
  1.3 Changing Default Settings for User Accounts .............................. 2  
  1.4 Creating User Accounts ............................................................. 2  
    1.4.1 About umask and the setgid and Restricted Deletion Bits .......... 3  
  1.5 Locking an Account .................................................................. 3  
  1.6 Modifying or Deleting User Accounts ......................................... 4  
  1.7 Creating Groups ...................................................................... 4  
  1.8 Modifying or Deleting Groups .................................................. 4  
  1.9 Configuring Password Ageing ................................................... 4  
  1.10 Granting sudo Access to Users ............................................... 5  
2 Configuring System Authentication .................................................. 7  
  2.1 About Authentication in Oracle Linux 8 ....................................... 7  
    2.1.1 About the authselect Utility ................................................. 7  
    2.1.2 About Profiles and Supported Features ................................. 8  
    2.1.3 About the Default Authentication Profile .............................. 9  
    2.1.4 Enabling and Disabling Profile Features ............................... 9  
    2.1.5 Selecting the Winbind Profile ............................................ 11  
    2.1.6 Modifying Ready-Made Profiles ......................................... 12  
    2.1.7 Creating Custom Profiles ................................................. 12  
    2.1.8 Migrating From authconfig to authselect ............................ 13  
  2.2 More About the System Security Services Daemon ....................... 14  
    2.2.1 Customizing SSSD ............................................................ 14  
    2.2.2 About Pluggable Authentication Modules ............................ 17
Preface

*Oracle® Linux 8: Setting Up System Users and Authentication* provides information about how to set up user accounts and authentication mechanisms on Oracle Linux 8 the Oracle Linux 8 release.

**Audience**

This document is intended for administrators who need to install and configure Oracle Linux 8. It is assumed that readers are familiar with web technologies and have a general understanding of administering the Linux operating system.

**Document Organization**

The document is organized as follows:

- **Chapter 1, Configuring and Managing Local Accounts** describes how to configure and manage local user and group accounts.

- **Chapter 2, Configuring System Authentication** describes how to configure various authentication methods that Oracle Linux can use and also how to configure the System Security Services Daemon feature to provide centralized identity and authentication management.

If you are interested in centralized implementation of authentication and authorization, see [https://www.oracle.com/middleware/technologies/identity-management/](https://www.oracle.com/middleware/technologies/identity-management/).

**Related Documents**

The documentation for this product is available at:


**Conventions**

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><strong>italic</strong></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>

**Documentation Accessibility**

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at [https://www.oracle.com/corporate/accessibility/](https://www.oracle.com/corporate/accessibility/).

**Access to Oracle Support**

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
Access to Oracle Support

https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.
Chapter 1 Configuring and Managing Local Accounts

Table of Contents

1.1 About User and Group Configuration ................................................................. 1
1.2 About Files Storing User and Group Information .................................................. 1
1.3 Changing Default Settings for User Accounts ....................................................... 2
1.4 Creating User Accounts ....................................................................................... 2
  1.4.1 About umask and the setgid and Restricted Deletion Bits ............................. 3
1.5 Locking an Account ............................................................................................... 3
1.6 Modifying or Deleting User Accounts ................................................................. 4
1.7 Creating Groups .................................................................................................. 4
1.8 Modifying or Deleting Groups .............................................................................. 4
1.9 Configuring Password Ageing ............................................................................... 4
1.10 Granting sudo Access to Users ............................................................................ 5

This chapter describes how to configure and manage local user and group accounts in Oracle Linux 8.

1.1 About User and Group Configuration

You use the `useradd` and `groupadd` commands to add and delete users and groups, as well as to modify settings such as passwords, home directories, login shells, and group membership.

In an enterprise environment that might have hundreds of servers and thousands of users, user and group account information is more likely to be held in a central repository rather than in files on individual servers. You can configure user and group information on a central server and then retrieve this information by using services such as the Lightweight Directory Access Protocol (LDAP) or the Network Information Service (NIS). You can also create home directories on a central server and then automatically mount or access these remote file systems when the user logs in to a system.

1.2 About Files Storing User and Group Information

Unless you select a different authentication mechanism during installation or use the `authselect` command to create an authentication profile, Oracle Linux verifies a user's identity by using the information that is stored in the `/etc/passwd` and `/etc/shadow` files.

The `/etc/passwd` file stores account information for each user such as his or her unique user ID (or `UID`, which is an integer), user name, home directory, and login shell. A user logs in using his or her user name, but the operating system uses the associated UID. When the user logs in, he or she is placed in his or her home directory and his or her login shell runs.

The `/etc/group` file stores information about groups of users. A user also belongs to one or more groups, and each group can contain one or more users. If you can grant access privileges to a group, all members of the group receive the same access privileges. Each group account has a unique group ID (`GID`, again an integer) and an associated group name.

By default, Oracle Linux implements the `user private group` (UPG) scheme where adding a user account also creates a corresponding UPG with the same name as the user, and of which the user is the only member.

Only the `root` user can add, modify, or delete user and group accounts. By default, both users and groups use shadow passwords, which are cryptographically hashed and stored in `/etc/shadow` and `/etc/`
Changing Default Settings for User Accounts

gshadow respectively. These shadow password files are readable only by the root user. The root user can set a group password that a user must enter to become a member of the group. If a group does not have a password, a user can only join the group if the root user adds that user as a member.

A user can use the newgrp command to log into a new group or to change the current group ID during a login section. If the user has a password, he or she can add group membership on a permanent basis. See the newgrp(1) manual page.

The /etc/login.defs file defines parameters for password aging and related security policies.

For more information about the content of these files, see the group(5), gshadow(5), login.defs(5), passwd(5), and shadow(5) manual pages.

1.3 Changing Default Settings for User Accounts

To display the default settings for a user account, use the following command:

```bash
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

**INACTIVE**: Specifies after how many days the system locks an account if a user's password expires. If set to 0, the system locks the account immediately. If set to -1, the system does not lock the account.

**SKEL**: Defines a template directory, whose contents are copied to a newly created user's home directory. The contents of this directory should match the default shell defined by SHELL.

You can specify options to useradd -D to change the default settings for user accounts. For example, to change the defaults for INACTIVE, HOME and SHELL:

```bash
# useradd -D -f 3 -b /home2 -s /bin/sh
```

**Note**

If you change the default login shell, you would most likely also create a new SKEL template directory that contains contents that are appropriate to the new shell.

If you specify /sbin/nologin for a user's SHELL, that user cannot log into the system directly but processes can run with that user's ID. This setting is typically used for services that run as users other than root.

The default settings are stored in the /etc/default/useradd file.

For more information, see Section 1.9, “Configuring Password Ageing” and the useradd(8) manual page.

1.4 Creating User Accounts

To create a user account by using the useradd command:

1. Create a user account by using the useradd command:

```bash
# useradd [options] username
```

You can specify options to change the account's settings from the default ones.
By default, if you specify a user name argument but do not specify any options, `useradd` creates a locked user account using the next available UID and assigns a user private group (UPG) rather than the value defined for `GROUP` as the user’s group.

2. Assign a password to the account to unlock it as follows:

```
# passwd username
```

The command prompts you to enter a password for the account.

If you want to change the password non-interactively (for example, from a script), use the `chpasswd` command instead:

```
echo "USERNAME:PASSWORD" | chpasswd
```

Alternatively, you can use the `newusers` command to create a number of user accounts at the same time.

For more information, see the `chpasswd(8), newusers(8), passwd(1), and useradd(8)` manual pages.

To create users by using the web-based GUI, see https://docs.oracle.com/en/operating-systems/oracle-linux/8/obe-cockpit-usermanage/index.html.

### 1.4.1 About umask and the setgid and Restricted Deletion Bits

Users whose primary group is not a UPG have a `umask` of 0022 set by `/etc/profile` or `/etc/bashrc`, which prevents other users, including other members of the primary group, from modifying any file that the user owns.

A user whose primary group is a UPG has a `umask` of 0002. It is assumed that no other user has the same group.

To grant users in the same group write access to files within the same directory, change the group ownership on the directory to the group, and set the `setgid` bit on the directory:

```
# chgrp groupname directory
# chmod g+s directory
```

Files that are created in such a directory have their group set to that of the directory rather than the primary group of the user who creates the file.

The restricted deletion bit prevents unprivileged users from removing or renaming a file in the directory unless they own either the file or the directory.

To set the restricted deletion bit on a directory:

```
# chmod a+t directory
```

For more information, see the `chmod(1)` manual page.

### 1.5 Locking an Account

To lock a user's account, use the `passwd` command with the `-l` option:

```
# passwd -l username
```

To unlock the account, specify the `-u` option:
Modifying or Deleting User Accounts

1.6 Modifying or Deleting User Accounts

To modify a user account, use the `usermod` command:

```
# usermod [options] username
```

For example, to add a user to a supplementary group (other than his or her login group):

```
# usermod -aG groupname username
```

You can use the `groups` command to display the groups to which a user belongs, for example:

```
# groups root
root : root
```

To delete a user's account, use the `userdel` command:

```
# userdel username
```

For more information, see the `passwd(1)`, `userdel(8)` and `usermod(8)` manual pages.

1.7 Creating Groups

To create a group by using the `groupadd` command:

```
# groupadd [options] groupname
```

Typically, you might want to use the `-g` option to specify the group ID (GID). For example:

```
# groupadd -g 1000 devgrp
```

For more information, see the `groupadd(8)` manual page.

1.8 Modifying or Deleting Groups

To modify a group, use the `groupmod` command:

```
# groupmod [options] username
```

To delete a user's account, use the `groupdel` command:

```
# groupdel username
```

For more information, see the `groupdel(8)` and `groupmod(8)` manual pages.

1.9 Configuring Password Ageing

To specify how users' passwords are aged, edit the following settings in the `/etc/login.defs` file:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASS_MAX_DAYS</td>
<td>Maximum number of days for which a password can be used before it must be changed. The default value is 99,999 days.</td>
</tr>
</tbody>
</table>
### Granting sudo Access to Users

**PASS_MIN_DAYS**
- Minimum number of days that is allowed between password changes. The default value is 0 days.

**PASS_WARN_AGE**
- Number of days warning that is given before a password expires. The default value is 7 days.

For more information, see the `login.defs(5)` manual page.

To change how long a user's account can be inactive before it is locked, use the `usermod` command. For example, to set the inactivity period to 30 days:

```bash
# usermod -f 30 username
```

To change the default inactivity period for new user accounts, use the `useradd` command:

```bash
# useradd -D -f 30
```

A value of -1 specifies that user accounts are not locked due to inactivity.

For more information, see the `useradd(8)` and `usermod(8)` manual pages.

#### 1.10 Granting sudo Access to Users

By default, an Oracle Linux system is configured so that you cannot log in directly as the root user. You must log in as a named user before using either `su` or `sudo` to perform tasks as root. This configuration allows system accounting to trace the original login name of any user who performs a privileged administrative action. If you want to grant certain users authority to be able to perform specific administrative tasks via `sudo`, use the `visudo` command to modify the `/etc/sudoers` file.

For example, the following entry grants the user `erin` the same privileges as root when using `sudo`, but defines a limited set of privileges to `frank` so that he can run commands such as `systemctl`, `rpm`, and `dnf`:

<table>
<thead>
<tr>
<th>User</th>
<th>Command Group</th>
<th>Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>erin</td>
<td>ALL=(ALL)</td>
<td>ALL</td>
</tr>
<tr>
<td>frank</td>
<td>ALL=(SERVICES, SOFTWARE)</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see the `su(1), sudo(8), sudoers(5), and visudo(8)` manual pages.
Chapter 2 Configuring System Authentication

Table of Contents

2.1 About Authentication in Oracle Linux 8 .............................. 7
  2.1.1 About the authselect Utility .................................... 7
  2.1.2 About Profiles and Supported Features .......................... 8
  2.1.3 About the Default Authentication Profile ......................... 9
  2.1.4 Enabling and Disabling Profile Features ......................... 9
  2.1.5 Selecting the Winbind Profile .................................. 11
  2.1.6 Modifying Ready-Made Profiles ................................ 12
  2.1.7 Creating Custom Profiles ....................................... 12
  2.1.8 Migrating From authconfig to authselect ....................... 13

2.2 More About the System Security Services Daemon .................. 14
  2.2.1 Customizing SSSD ............................................... 14
  2.2.2 About Pluggable Authentication Modules ....................... 17

This chapter describes how to configure various authentication methods that Oracle Linux can use, including NIS, LDAP, Kerberos, and Winbind, and how you can configure the System Security Services Daemon feature to provide centralized identity and authentication management.

2.1 About Authentication in Oracle Linux 8

Authentication is the verification of the identity of an entity, such as a user, to a system. A user logs in by providing a user name and a password, and the operating system authenticates the user's identity by comparing this information to data stored on the system. If the login credentials match and the user account is active, the user is authenticated and can successfully access the system.

The information that verifies a user's identity is located on the local system in the `/etc/passwd` and `/etc/shadow` files, or on remote systems by using Identity Policy Audit (IPA), the Lightweight Directory Access Protocol (LDAP), or Winbind. In addition, IPAv2, and LDAP data files can use the Kerberos authentication protocol, which allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner.

2.1.1 About the authselect Utility

In previous releases, the `authconfig` command was the tool for controlling authentication of user logins on local and remote systems. However, over time, authentication methods have grown in number, and a variety of authentication configurations using each of these methods have also multiplied. Consequently, the `authconfig` command has become too complex to function as a central management tool as well as for troubleshooting problems with authentication setup.

The introduction of the `authselect` utility reflects a different approach to authentication, which is profile-based. It is automatically included in an Oracle Linux 8 installation.

The `authselect` utility consists of the following components:

- `authselect` command to manage system authentication. This command can only be run by `root`.
- Three ready-made profiles that implement specific types of authentication. The following profiles are automatically installed in the `usr/share/authselect/default` directory:
  - `sssd` profile. This profile uses the `sssd` service to perform system authentication. This profile is automatically selected by default after an Oracle Linux 8 installation.
• **winbind** profile. This profile uses the **winbind** service to perform system authentication.

• The **nis** profile is also included in the directory but only for purposes of maintaining compatibility with legacy configurations. NIS is deprecated in Oracle Linux 8. Thus, converting to using the **sssd** profile instead is highly recommended.

The **authselect** utility also enables you to create customized profiles if the ready-made profiles are not sufficient for your environment. Because of this flexibility offered by the utility, authentication profiles are stored depending on the type of profile:

• **/usr/share/authselect/default** contains the ready-made profiles provided by Oracle Linux 8.

• **/usr/share/authselect/vendor** contains the profiles that are provided by vendors. These profiles can override those that are in the default directory.

• **/etc/authselect/custom** contains any profiles you create for your specific environment.

**Important**

The **authselect** utility applies the specifications in your selected profile. However, the utility does not modify the configuration files of the service on which the profile is based. If, for example, you use the **sssd** profile, you must configure SSSD for the service to function properly. Consult the proper documentation to configure the profile’s service. You must also ensure that the service is started and enabled.

For more details about the utility, refer to the **authselect(8)** manual page.

### 2.1.2 About Profiles and Supported Features

Each profile has associated features you can enable to make the profile's service perform a particular method of authentication, such as smart card authentication, fingerprint authentication, kerberos, and so on. After you select a profile and enable preferred features, **authselect** automatically reads the appropriate configuration files of those features to run the relevant authentication processes. Every user who logs in to the host is authenticated based on that configured profile.

The following table shows features for the **sssd** profile:

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>with-faillock</strong></td>
<td>Lock the account after too many authentication failures.</td>
</tr>
<tr>
<td><strong>with-mkhome<code>dir</code></strong></td>
<td>Create home directory on user's first log in.</td>
</tr>
<tr>
<td><strong>with-ecryptfs</strong></td>
<td>Enable automatic per-user ecryptfs.</td>
</tr>
<tr>
<td><strong>with-smartcard</strong></td>
<td>Authenticate smart cards through SSSD.</td>
</tr>
<tr>
<td><strong>with-smartcard-lock-on-removal</strong></td>
<td>Lock the screen when the smart card is removed. Requires that <strong>with-smartcard</strong> is also enabled.</td>
</tr>
<tr>
<td><strong>with-smartcard-required</strong></td>
<td>Only smart card authentication is operative; others, including password, are disabled. Requires that <strong>with-smartcard</strong> is also enabled.</td>
</tr>
<tr>
<td><strong>with-fingerprint</strong></td>
<td>Authenticate through fingerprint reader.</td>
</tr>
</tbody>
</table>
### About the Default Authentication Profile

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>with-silent-lastlog</code></td>
<td>Disable generation of <code>pam_lostlog</code> messages during login</td>
</tr>
<tr>
<td><code>with-sudo</code></td>
<td>Enable <code>sudo</code> to use SSSD for rules besides <code>/etc/sudoers</code>.</td>
</tr>
<tr>
<td><code>with-pamaccess</code></td>
<td>Refer to <code>/etc/access.conf</code> for account authorization.</td>
</tr>
<tr>
<td><code>without-nullock</code></td>
<td>Do not add the <code>nullock</code> parameter to <code>pam_unix</code>.</td>
</tr>
</tbody>
</table>

The following table shows features for the `winbind` profile:

**Table 2.2 Features Supported by winbind Profile**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>with-faillock</code></td>
<td>Lock the account after too many authentication failures.</td>
</tr>
<tr>
<td><code>with-mkhomedir</code></td>
<td>Create home directory on user's first log in.</td>
</tr>
<tr>
<td><code>with-ecryptfs</code></td>
<td>Enable automatic per-user ecryptfs.</td>
</tr>
<tr>
<td><code>with-fingerprint</code></td>
<td>Authenticate through fingerprint reader.</td>
</tr>
<tr>
<td><code>with-krb5</code></td>
<td>Use Kerberos authentication.</td>
</tr>
<tr>
<td><code>with-silent-lastlog</code></td>
<td>Disable generation of <code>pam_lostlog</code> messages during login</td>
</tr>
<tr>
<td><code>with-pamaccess</code></td>
<td>Refer to <code>/etc/access.conf</code> for account authorization.</td>
</tr>
<tr>
<td><code>without-nullock</code></td>
<td>Do not add the <code>nullock</code> parameter to <code>pam_unix</code></td>
</tr>
</tbody>
</table>

For details about each profile, refer to the profile’s corresponding `/usr/share/authselect/default/profile/README` file. See also the `authselect-profiles(5)` manual page.

#### 2.1.3 About the Default Authentication Profile

On an Oracle Linux 8 system, the `sssd` profile is selected by default to manage authentication on the system. This profile covers most authentication cases including PAM authentication, Kerberos, and so on. The `sssd` is recommended as the profile of choice.

As a default profile, some `sssd` features are automatically enabled. You can verify the information with the following command:

```bash
# authselect current
Profile ID: sssd
Enabled features:
- with-fingerprint
- with-silent-lastlog
```

The output indicates that the SSSD service manages the system's authentication. At a minimum, authentication with the fingerprint reader is enforced through `pam_fprintd`. Additionally, no `pam_lastlog` message is displayed on the screen when users log in.

#### 2.1.4 Enabling and Disabling Profile Features

Enabled features of a profile determine the manner of authentication on the system. You can enable profile features in one of two ways:
Enabling and Disabling Profile Features

- Specify additional features to be enabled in the current profile.
- Replace current features of a selected profile. This method is discussed in Section 2.1.5, “Selecting the Winbind Profile”.

To enable more features of a profile, do the following:

1. (Optional): Identify the current profile.
   
   Enabling additional features works only on the current profile. The procedure does not work on unselected profiles.
   
   ```
   # authselect current
   ```

2. If necessary, identify the feature requirements for the feature to work properly.
   
   ```
   # authselect requirements profile feature
   ```

3. Complete the indicated listed feature requirements as needed.

4. Enable the feature.
   
   Note that you can enable features only one at a time.
   
   ```
   # authselect enable-feature feature
   ```

To disable a feature:

```
# authselect disable-feature feature
```

**Example 2.1 Examples of Enabling Profile Features**

This example shows how to add the following additional functionalities to the default sssd profile:

- Automatically lock an account after too many authentication failures (with-faillock).
- Automatically create a user home directory at the user’s first time log in (with-mkhomedir).

```
# authselect requirements sssd with-faillock
Make sure that SSSD service is configured and enabled. See SSSD documentation for more information.
```

```
# authselect requirements sssd with-mkhomedir
Make sure that SSSD service is configured and enabled. See SSSD documentation for more information.
```

- with-mkhomedir is selected, make sure pam_oddjob_mkhomedir module is present and oddjobd service is enabled
  - systemctl enable oddjobd.service
  - systemctl start oddjobd.service

```
# authselect enable-feature with-faillock
# authselect enable-feature with-mkhomedir
```

```
# authselect current
Profile ID: sssd
Enabled features:
- with-fingerprint
- with-silent-lastlog
- with-faillock
- with-mkhomedir
```

This second example shows how to cause the system to check `/etc/security/access.conf` to authenticate and authorize users. In this case, the PAM access feature needs to be added as an enabled feature for sssd.
Selecting the Winbind Profile

Winbind is a client-side service that resolves user and group information on a Windows server, and allows Oracle Linux to understand Windows users and groups.

If you do not want to use the current default profile (`sssd`) and use the `winbind` profile instead, do the following:

1. Install the `samba-winbind` package.
   
   ```
   # dnf install samba-winbind
   ```

2. Select the `winbind` profile.
   
   When selecting a profile, you can enable multiple features in the same command, for example:
   
   ```
   # authselect select winbind with-faillock with-mkhomedir [options]
   Profile "winbind" was selected.
   The following nsswitch maps are overwritten by the profile:
   - passwd
   - group
   - with-mkhomedir is selected, make sure pam_oddjob_mkhomedir module
     is present and oddjobd service is enabled
   - systemctl enable oddjobd.service
   - systemctl start oddjobd.service
   ```

   For other options you can use with the `authselect select` command, see the `authselect(8)` manual page.

   Note that the output includes other information and instructions about additional actions that are required for the authentication to be properly implemented. The information is important especially for the different features you want to enable for the profile. Ensure that these requirements are also fulfilled.

3. Start the `winbind` service.
   
   ```
   # systemctl start winbind
   # systemctl enable winbind
   ```
2.1.6 Modifying Ready-Made Profiles

To modify and customize a ready-made profile, revisions must be applied to the system’s /etc/nsswitch.conf file. However, you must not edit the /etc/nsswitch.conf directly. Instead, follow these steps:

1. If necessary, select the profile to make it current, for example:

   ```
   # authselect select sssd
   ```

2. Edit the /etc/authselect/user-nsswitch.conf file as required.

   Do not modify any of the following configuration in the file. If you do, those modifications will be ignored:
   - passwd
   - group
   - netgroup
   - automount
   - services

3. Apply the changes.

   ```
   # authselect apply-changes
   ```

   The changes in /etc/authselect/user-nsswitch.conf are applied to /etc/nsswitch.conf and will be used by the current profile.

Note

If the system is part of an environment that uses either Identity Management or Active Directory, do not use authselect to manage authentication. When the host is made to join either Identity Management or Active Directory, their respective tools take care of managing authentication of the environment.

2.1.7 Creating Custom Profiles

If you do not want to use the ready-made profiles from Oracle Linux 8 or vendor-provided profiles, you can create your own specific profile. Follow these steps:

1. Create the profile.

   ```
   # authselect create-profile newprofile -b template
   --symlink-meta --symlink-pam
   ```

   `newprofile` Name of your custom profile.

   `template` Base to be used for the custom profile, which is either `sssd` or `winbind`.

Note

If you use the authselect select to select a profile that is already current and active while enabling features at the same time, those features will replace whatever features were previously enabled.
Migrating From authconfig to authselect

--symlink-meta Creates symbolic links to the meta files in the original directory of the template profile you are using as base.

--symlink-pam Creates symbolic links to the PAM templates in the original directory of the template profile you are using as base.

This command creates an /etc/authselect/custom/newprofile directory that contains the symbolic links to the files in the base’s original directory. The only file that is not a symbolic link in this directory is nsswitch.conf.

2. Edit the /etc/authselect/custom/newprofile/nsswitch.conf file according to your preference.

3. Select your custom profile.

   # authselect select custom/newprofile

   This command also creates a backup of the original /etc/nsswitch.conf file and replaces it with a symbolic link to the corresponding file in your custom profile’s directory.

   You can test this result by comparing the symbolic link /etc/nsswitch.conf with the original /etc/nsswitch.conf.bak and verify that the original file’s contents remain intact.

4. Enable features for your new profile as needed.

   See Section 2.1.4, “Enabling and Disabling Profile Features” for reference.

5. (Optional) Verify the configuration of the custom profile.

   # authselect current

2.1.8 Migrating From authconfig to authselect

In Oracle Linux 8, authselect support for backward compatibility with authconfig is minimal. Thus, migrating to authselect is highly recommended. Migrating requires you to take several actions, including the following:

• Convert scripts.

   If you use the ipa-client-install command or the realm join command to make the host join a domain, you can just remove any authconfig call in your scripts. Otherwise, you will need to replace each authconfig call with its equivalent authselect call.

• Update configuration files.

   Files for the various services need to be reconfigured, including those that apply to the following: Kerberos, LDAP, NIS, SSSD, and Winbind.

• Enforce password quality restrictions for authselect.

   The pam_pwquality module enforces password quality restrictions for local users. You configure this module in the /etc/security/pwquality.conf file, according to the information that is provided in the pam_pwquality(8) man page.

• Switch from the authconfig’s cacertdir_rehash tool to the native openssl rehash directory command.

• Start the appropriate services.
Depending on the profile you select for the authselect implementation, start the service for that profile. If you select the sssd profile, for example, then you would enable and start the SSSD service.

```
# systemctl start sssd; systemctl enable sssd
```

For complete migration instructions and examples, see the authselect-migration(7) manual page. See also the authselect(8) manual page.

## 2.2 More About the System Security Services Daemon

The System Security Services Daemon (SSSD) feature provides access on a client system to remote identity and authentication providers. The SSSD acts as an intermediary between local clients and any back-end provider that you configure.

The benefits of configuring SSSD include the following:

- **Reduced system load**
  
  Clients do not have to contact the identification or authentication servers directly.

- **Offline authentication**
  
  You can configure SSSD to maintain a cache of user identities and credentials.

- **Single sign-on access**
  
  If you configure SSSD to store network credentials, users need only authenticate once per session with the local system to access network resources.

Because the Oracle Linux 8 authselect sssd profile is used by default, the SSSD service is automatically installed and enabled on a newly installed system. The default configuration makes use of the Pluggable Authentication Modules (PAM) and the Name Service Switch (NSS) for managing access and authentication on a system. No further configuration is required, unless you wish to use different authentication services or wish to customize the configuration to use alternative values to the default settings.

### 2.2.1 Customizing SSSD

By default the SSSD service used by the sssd profile uses Pluggable Authentication Modules (PAM) and the Name Service Switch (NSS) for managing access and authentication on a system. As you enable additional features for the profile to customize SSSD authentication, you must also configure SSSD for the enabled feature.

Customized configuration is performed by creating configuration files within the /etc/sssd/conf.d directory. Each configuration file must have the .conf suffix to enable it when SSSD is started. Configuration files are formatted using ini-style syntax and consist of sections, indicated using square brackets, and parameters, listed as `key = value` entries. The man pages provided for SSSD are comprehensive and provide detailed information on the options that are available.

The following example demonstrates how one might configure SSSD to authenticate against an LDAP provider with Kerberos configured:

1. Create a configuration file for the feature and store it in /etc/sssd/conf.d, for example /etc/sssd/conf.d/00-ldap.conf

2. Configure /etc/sssd/conf.d/00-ldap.conf with parameter definitions, such as the following:
```bash
config_file_version = 2
domains = LDAP
services = nss, pam

[domain/LDAP]
id_provider = ldap
ldap_uri = ldap://ldap.mydom.com
ldap_search_base = dc=mydom,dc=com

auth_provider = krb5
krb5_server = krbsvr.mydom.com
krb5.realm = MYDOM.COM
cache_credentials = true

min_id = 5000
max_id = 25000
enumerate = false

[nss]
filter_groups = root
filter_users = root
reconnection_retries = 3
entry_cache_timeout = 300

[pam]
reconnection_retries = 3
offline_credentials_expiration = 2
offline_failed_login_attempts = 3
offline_failed_login_delay = 5
```

Contains configuration settings for SSSD monitor options, domains, and services. The SSSD monitor service manages the services that SSSD provides.

- **services** defines the supported services, which should include **nss** for the Name Service Switch and **pam** for Pluggable Authentication Modules.

- The **domains** entry specifies the name of the sections that define authentication domains.

 Defines a domain for an LDAP identity provider that uses Kerberos authentication. Each domain defines where user information is stored, the authentication method, and any configuration options. SSSD can work with LDAP identity providers such as OpenLDAP, Red Hat Directory Server, IPA, and Microsoft Active Directory, and it can use either native LDAP or Kerberos authentication.

- **id_provider** specifies the type of provider (in this example, LDAP).

- **ldap_uri** specifies a comma-separated list of the Universal Resource Identifiers (URIs) of the LDAP servers, in order of preference, to which SSSD can connect.

- **ldap_search_base** specifies the base distinguished name (dn) that SSSD should use when performing LDAP user operations on a relative distinguished name (RDN) such as a common name (cn).
Customizing SSSD

- **auth_provider** entry specifies the authentication provider (in this example, Kerberos).

- **krb5_server** specifies a comma-separated list of Kerberos servers, in order of preference, to which SSSD can connect.

- **krb5_realm** specifies the Kerberos realm.

- **cache_credentials** specifies if SSSD caches user credentials such as tickets, session keys, and other identifying information to support offline authentication and single sign-on.

  **Note**

  To allow SSSD to use Kerberos authentication with an LDAP server, you must configure the LDAP server to use both Simple Authentication and Security Layer (SASL) and the Generic Security Services API (GSSAPI). For more information about configuring SASL and GSSAPI for OpenLDAP, see [http://www.openldap.org/doc/admin24/sasl.html](http://www.openldap.org/doc/admin24/sasl.html).

- **min_id** and **max_id** specify upper and lower limits on the values of user and group IDs.

- **enumerate** specifies whether SSSD caches the complete list of users and groups that are available on the provider. The recommended setting is `False` unless a domain contains relatively few users or groups.

[nss]

Configures the Name Service Switch (NSS) module that integrates the SSS database with NSS.

- **filter_users** and **filter_groups** prevent NSS from extracting information about the specified users and groups being retrieved from SSS.

- **reconnection_retries** specifies the number of times that SSSD should attempt to reconnect if a data provider crashes.

- **enum_cache_timeout** specifies the number of seconds for which SSSD caches user information requests.
### About Pluggable Authentication Modules

**[pam]**

Configures the PAM module that integrates SSSD with PAM.

- **offline_credentials_expiration** specifies the number of days for which to allow cached logins if the authentication provider is offline.

- **offline_failed_login_attempts** specifies how many failed login attempts are allowed if the authentication provider is offline.

- **offline_failed_login_delay** specifies how many minutes after the limit of allowed failed login attempts have been exceeded before a new login attempt is permitted.

3. Change the mode of `/etc/sssd/conf.d/00-ldap.conf` to 0600:

   ```bash
   # chmod 0600 /etc/sssd/conf.d/00-ldap.conf
   ```

4. If it is not started yet, enable the SSSD service:

   ```bash
   # authselect select sssd
   ```

5. Select the `sssd` profile.

For more information about the SSSD service, see the `sssd(8)` man page as well as [https://pagure.io/SSSD/sssd/](https://pagure.io/SSSD/sssd/). Additionally, you can consult `sssd.conf(5)`, `sssd-ldap(5)`, `sssd-krb5(5)`, `sssd-IPA(5)`, and other man pages.

#### 2.2.2 About Pluggable Authentication Modules

The Pluggable Authentication Modules (PAM) feature is an authentication mechanism used by the `sssd` profile that allows you to configure how applications use authentication to verify the identity of a user.

The PAM configuration files, which are located in the `/etc/pam.d` directory, describe the authentication procedure for an application. The name of each configuration file is the same as, or is similar to, the name of the application for which the module provides authentication. For example, the configuration files for `passwd` and `sudo` are named `passwd` and `sudo`.

Each PAM configuration file contains a list or stack of calls to authentication modules. For example, the following listing shows the default content of the `login` configuration file:

```bash
#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth include system-auth
auth include postlogin
account required pam_nologin.so
account include system-auth
password include system-auth
# pam_selinux.so close should be the first session rule
session required pam_selinux.so close
session required pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session required pam_selinux.so open
session required pam_namespace.so
session optional pam_keyinit.so force revoke
session include system-auth
session include postlogin
~session optional pam_ck_connector.so
```

Comments in the file start with a `#` character. The remaining lines each define an operation type, a control flag, the name of a module such as `pam_rootok.so` or the name of an included configuration file such
as `system-auth`, and any arguments to the module. PAM provides authentication modules as shared libraries in `/usr/lib64/security`.

For a particular operation type, PAM reads the stack from top to bottom and calls the modules listed in the configuration file. Each module generates a success or failure result when called.

The following operation types are defined for use:

- **auth**
  The module tests whether a user is authenticated or authorized to use a service or application. For example, the module might request and verify a password. Such modules can also set credentials, such as a group membership or a Kerberos ticket.

- **account**
  The module tests whether an authenticated user is allowed access to a service or application. For example, the module might check if a user account has expired or if a user is allowed to use a service at a given time.

- **password**
  The module handles updates to an authentication token.

- **session**
  The module configures and manages user sessions, performing tasks such as mounting or unmounting a user’s home directory.

If the operation type is preceded with a dash (`-`), PAM does not add an create a system log entry if the module is missing.

With the exception of `include`, the control flags tell PAM what to do with the result of running a module. The following control flags are defined for use:

- **optional**
  The module is required for authentication if it is the only module listed for a service.

- **required**
  The module must succeed for access to be granted. PAM continues to execute the remaining modules in the stack whether the module succeeds or fails. PAM does not immediately inform the user of the failure.

- **requisite**
  The module must succeed for access to be granted. If the module succeeds, PAM continues to execute the remaining modules in the stack. However, if the module fails, PAM notifies the user immediately and does not continue to execute the remaining modules in the stack.

- **sufficient**
  If the module succeeds, PAM does not process any remaining modules of the same operation type. If the module fails, PAM processes the remaining modules of the same operation type to determine overall success or failure.

The control flag field can also define one or more rules that specify the action that PAM should take depending on the value that a module returns. Each rule takes the form `value=action`, and the rules are enclosed in square brackets, for example:

```
[user_unknown=ignore success=ok ignore=ignore default=bad]
```

If the result that is returned by a module matches a value, PAM uses the corresponding action, or, if there is no match, it uses the default action.

The `include` flag specifies that PAM must also consult the PAM configuration file specified as the argument.
Most authentication modules and PAM configuration files have their own manual pages. Relevant files are stored in the `/usr/share/doc/pam` directory.

For more information, see the `pam(8)` manual page. In addition, each PAM module has its own manual page, for example `pam_unix(8), postlogin(5), and system-auth(5).`