# Oracle Linux 8.4 Common Criteria Guidance Document

April 12, 2023

0.8

Prepared By:
Acumen Security
2400 Research Blvd Suite 395
Rockville, MD, 20850
www.acumensecurity.net

Prepared for:
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA
Tel.: +1.650.506.7000
www.oracle.com

**Trademarks**

Oracle Linux and the Oracle logo are trademarks or registered trademarks of Oracle Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Intel, Xeon, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

**Legal Notice**

## Table of Contents

# Revision History

| Version | Date | Description |
|---|---|---|
| 0.1 | September 9, 2021 | Initial draft. |
| 0.2 | September 14, 2021 | Minor updates. |
| 0.3 | August 31, 2022 | Updated to address validator comments. |
| 0.4 | September 13, 2022 | Updates to match ST claims. |
| 0.5 | November 14, 2022 | Updates to match ST claims. |
| 0.6 | January 25, 2023 | Updates to address certifier comments. |
| 0.7 | March 17, 2023 | Updates to address certifier comments. |
| 0.8 | April 12, 2023 | Addition of TOE identifier. |

# 1 Purpose of this document

This document is intended to be a supplement to the Oracle public user documentation. This Common Criteria guidance document contains configuration information needed to configure and administer Oracle Linux 8.4. Oracle Linux conforms to the Protection Profile for General Purpose Operating Systems Version 4.2.1 [PP_OS_V4.2.1] and the Functional Package for Secure Shell (SSH) Version 1.0 [PKG_SSH_V1.0]. The information contained in this document is intended for Administrators who would be responsible for the configuration and management of Oracle Linux 8.4.

## 1.1 TOE Overview

Oracle Linux 8.4 (herein referred to as the TOE) is a Linux-based operating system. Oracle Linux is a general purpose, multi-user, multi-tasking Linux based operating system. The TOE provides a platform for a variety of applications and satisfies all the criterion to meet the Protection Profile for General Purpose Operating Systems Version 4.2.1 and Functional Package for Secure Shell (SSH) Version 1.0.

## 1.2 TOE Description

The TOE in the evaluated configuration consists of the following platforms:

- Oracle Linux 8.4 running on AMD EPYC 7551
- Oracle Linux 8.4 running on Intel Xeon Platinum 8167M

The TOE is identified as follows:

| Oracle Linux 8.4 with the following package updates:<br><br>Note: Dependencies for additional packages will be installed automatically. | |
|---|---|
| kernel-uek | 5.4.17-2136.312.3.4.el8uek |
| openssl | 1:1.1.1k-7.el8_6 |
| platform-python | 3.6.8-47.0.1.el8_6 |
| expat | 2.2.5-8.0.1.el8_6.3 |
| file | 5.33-20.el8 |
| glibc | 2.28-189.5.0.1.el8_6 |
| gnutls | 3.6.16-5.el8_6 |
| grub2-common | 2.02-123.0.10.el8_6.8 |
| nettle | 3.4.1-7.el8 |
| libsolv | 07.20-1.el8 |
| libtirpc | 1.1.4-6.el8 |
| libxml2 | 2.9.7-13.el8_6.1 |
| lz4-libs | 1.8.3-3.el8_4 |

| | |
|---|---|
| nss | 3.79.0-10.el8_6 |
| polkit | 0.115-13.0.1.el8_5.2 |
| sssd-common | 2.6.2-4.0.2.el8_6.1 |
| vim-minimal | 2:8.0.1763-19.0.1.el8_6.4 |
| bind-export-libs | 32:9.11.36-3.el8_6.1 |
| c-ares | 1.13.0-6.el8 |
| cpio | 2.12-11.el8 |
| cryptsetup-libs | 2.3.7-2.el8 |
| curl | 7.61.1-22.el8_6.4 |
| cyrus-sasl-lib | 2.1.27-6.el8_5 |
| dnf | 4.7.0-8.0.1.el8 |
| libgcc | 8.5.0-10.1.0.1.el8_6 |
| libgomp | 8.5.0-10.1.0.1.el8_6 |
| libssh | 0.9.6.3.el8 |
| libstdc++ | 8.5.0-10.1.0.1.el8_6 |
| glib2 | 2.56.4-158.el8_6.1 |
| gnupg2 | 2.2.20-3.el8_6 |
| gzip | 1.9-13.el8_5 |
| json-c | 0.13.1-3.el8 |
| kpartx | 0.8.4-22.el8_6.2 |
| libarchive | 3.3.3-3.el8_5 |
| libgcrypt | 1.8.5-7.el8_6 |
| libksba | 1.3.5-8.el8_6 |
| lua-libs | 5.3.4-12.el8 |
| microcode_ctl | 4:20220207-1.20220510.1.0.1.el8_6 |
| ncurses | 6.1-9.20180224.el8 |
| openssh | 8.0p1-13.el8 |
| pcre | 8.42-6.el8 |
| pcre2 | 10.32-3.el8_6 |
| rpm | 4.14.3-24.el8_6 |
| rsyslog | 8.2102.0-7.el8_6.1 |
| shim | 15.6-1.0.3.el8 |
| systemd | 239-58.0.1.el8.6.8 |
| sqlite-libs | 3.26.0-16.el8_6 |
| udisks2 | 2.9.0-9.el8 |
| xz | 5.2.4-4.el8_6 |
| zlib | 1.2.11-19.el8_6 |
| NetworkManager | 1:1.36.0-9.0.1.el8_6 |

| | |
|---|---|
| kexec-tools | 2.0.20-68.0.3.el8 |
| libsepol | 2.9-3.el8 |
| platform-python-pip | 9.0.3-22.el8 |
| libsss_autofs | 2.6.2-4.0.2.el8_6.1 |
| libsss_sudo | 2.6.2-4.0.2.el8_6.1 |
| sssd-nfs-idmap | 2.6.2-4.0.2.el8_6.1 |
| python3-pip-wheel | 9.0.3-22.el8 |

## 1.3 Assumptions

The following Assumptions are for the Operational Environment:

| Assumptions | Operational Environment |
|---|---|
| A.PLATFORM | The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP. |
| A.PROPER_USER | The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act as the user, so requirements which confine malicious subjects are still in scope. |
| A.PROPER_ADMIN | The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy. |

**Table 1 Assumptions on the Operational Environment**

## 1.4 TOE Delivery

The TOE software can be download from the Oracle website:
https://yum.oracle.com/ISOS/OracleLinux/OL8/u4/x86_64/OracleLinux-R8-U4-x86_64-dvd.iso

When software updates are available via the website, the users can obtain, verify the integrity, and install the updates.

# 2 Prerequisites for Installation

Before beginning this procedure, you must ensure that:

- Oracle Linux 8.4 ISO image is required for Common Criteria certification as applicable for the list of platforms listed in Section 1.2 of this guidance document.
- Ensure that the ISO image's integrity verification has been performed for all files downloaded from the Oracle website. The integrity is verified using SHA-256 hash sum provided by the Oracle web site.

  The minimum system requirements are the following:

  - Minimum of 2 logical CPUs up to 2048 logical CPUs
  - 1.5 GB of memory per logical CPU, up to a maximum of 64 TB
  - At least 10 GB of disk space (20 GB is the recommended minimum)
  - On UEFI systems, ensure that the target disk uses GPT (GUID Partition Table), as some UEFI firmware does not support UEFI/MBR boot.

NOTE: A graphical user interface (GUI) is not included in the evaluated configuration.

# 3 Installation of Oracle Linux 8.4

The steps below are applicable whether Oracle Linux 8.4 is installed on a server, or virtual device.

The ISO can be downloaded from the following link:
https://yum.oracle.com/ISOS/OracleLinux/OL8/u4/x86_64/OracleLinux-R8-U4-x86_64-dvd.iso

After gaining access to Oracle Linux ISO, follow the instructions in Section 3 - Installing Oracle Linux Manually in the "Installing Oracle Linux" guidance below.

https://docs.oracle.com/en/operating-systems/oracle-linux/8/install/

During the installation procedure select "Minimal Installation" in the software selection options.

# 4 Enabling FIPS Mode of Operation

To be complaint for Common Criteria, the user must ensure FIPS mode is enabled.

Follow the steps below to enable FIPS mode on the system:

To enable FIPS mode on the system, run the following command:

# sudo fips-mode-setup --enable

The following output would be displayed:

Setting system policy to FIPS.  FIPS mode will be enabled.

Please reboot the system for the setting to take effect.

You must reboot the system for the setting to take effect.

Running the previous command configures FIPS mode implicitly by setting the system-wide cryptographic policy to FIPS.

Note that using the **update-crypto-policies** command to set FIPS mode is not sufficient, as shown in the following output:

# update-crypto-policies --set FIPS

The following output is displayed:

Warning: Using 'update-crypto-policies --set FIPS' is not sufficient for FIPS compliance.

Use # 'fips-mode-setup --enable' command instead.

Verify that FIPS is enabled by running any of the following commands:

# fips-mode-setup --check

# update-crypto-policies --show

# cat /etc/system-fips

# sysctl crypto.fips_enabled

# crypto.fips_enabled = 1

For the command output in the last example, a response of 1 indicates that FIPS is enabled. One must reboot the system for the settings to take effect.

# 5 Configuring SSH

The OS implements SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, 4344, 5656, 6668, and 8332 as a client and server. The TOE supports password-based authentication and public key based authentication.

## 5.1 Configuring SSH Server

The SSH server is allowed to use only approved ciphers. This can be configured by using the 'update-crypto-policies' tool.

*Note: The information about the update-crypto-policies tool and instructions on how to use the tool can be found in Oracle's Enhancing System Security document, section 4 Implementing Additional Security Features and Best Practices, under 'Configuring System Cryptographic Policies' sub-section.*

A default Oracle Linux installation includes the openssh and openssh-server packages, and the sshd service is enabled by default.

You can set sshd configuration options using the 'update-crypto-policies' tool. You can determine the current system-wide cryptographic policy by running '*update-crypto-policies --show*'. This policy should be set to FIPS.

Next, update the '/etc/crypto-policies/back-ends/opensshserver.config' file to include these the following claimed parameters:

- The following public key algorithms: rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, and ecdsa-sha2-nistp384.
- The SSH client shall ensure that, as described in RFC 4253, packets greater than 262144 bytes in an SSH transport connection are dropped.
- The TOE supports the following encryption algorithms: aes128-ctr, aes256-ctr, aes128-cbc, and aes256-cbc.
- The TOE supports the following data integrity MAC algorithms: hmac-sha2-256 and hmac-sha2-512.
- The TOE supports the following key exchange algorithms: ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.

Modify the following lines in '/etc/sshd_config'

GSSAPIAuthentication no
ChallengeResponseAuthentication no

Note: After making changes to the configuration file, you must restart the sshd service for the changes to take effect.

## 5.2 Configuring SSH Client

To configure OpenSSH, you need the openssh and openssh-clients packages. A default Oracle Linux installation includes both packages. However, if necessary, install or update the openssh and openssh-clients packages on your system as follows:

# sudo dnf install openssh openssh-clients

Next, verify that '/etc/crypto-policies/back-ends/openssh.config' includes the following claimed parameters. If the parameters mismatch, update the openssh.config file accordingly:

- The following public key algorithms: rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, and ecdsa-sha2-nistp384.
- The SSH client shall ensure that, as described in RFC 4253, packets greater than 262144 bytes in an SSH transport connection are dropped.
- The TOE supports the following encryption algorithms: aes128-ctr, aes256-ctr, aes128-cbc, and aes256-cbc.
- The TOE supports the following data integrity MAC algorithms: hmac-sha2-256, and hmac-sha2-512.
- The TOE supports the following key exchange algorithms: ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.

## 5.3 Using Public Key Authentication

The OS supports public key authentication. To generate a public and public key pair, the following command should be used:

# ssh-keygen

Press Enter each time that the command prompts you to enter a passphrase. Use the ssh-copy-id script to append the public key in the local ~/.ssh/id_rsa.pub file to the ~/.ssh/authorized_keys file on the remote system. You can now use the OpenSSH utilities to access the remote system without supplying a

password. As the script suggests, you should use ssh to log into the remote system to verify that the ~/.ssh/ authorized_keys file contains only the keys for the systems from which you expect to connect.

### 5.4 Configuring SSH Connection Rekey Limits

Oracle Linux 8.4 supports configuration of the threshold for rekeying of symmetric session keys in an SSH session by modification of the “RekeyLimit” parameter in the following configuration files:

**/etc/ssh/sshd_conf** on the SSH server.
**/etc/ssh/ssh_conf** on the SSH client.

To configure the TOE, uncomment and modify the **RekeyLimit** parameter in each of the relevant files as follows:

**RekeyLimit 1G 1h**

# 6 Configuring TLS

TOE supports RSA key sizes of 2048 bits, 3072, and 4096 bits for key generation and ECDSA. The RSA keys and ECDSA keys are used in support of digital signature for TLS sessions.
The TOE supports FFC schemes using cryptographic key sizes of 2048-bits or greater. The FFC scheme is used as part of key generation.

Any Certificate Authority can be used to generate or sign the certificate. TLS v1.2 is supported.

The following ciphersuites are supported by the OS for TLS session establishments:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

The following command is used to generate RSA private key:
# openssl genrsa

The following command is used to generate a certificate signing request:

# openssl req

The following command is used to sign certificate signing request:

# openssl ca

# 7 Cryptographic Key Destruction

The TOE is capable of performing key destruction.

The TOE does not support delayed key destruction.

When using SSDs, the wear levelling mechanism prevents software to overwrite the exact physical location where the keys are stored.

Key data may still reside on the physical data store albeit it cannot be retrieved by the operating system anymore. Yet, forensic tools may recover that data. Thus, an SSD must be physically destroyed at the end of life to guarantee that no cryptographic keys remain.

The system uses many more keys than outlined in the preceding sections. Those keys are always ephemeral and maintained in RAM. These keys will be securely erased by the system without user intervention.

# 8 Configuring User Authentication

The Pluggable Authentication Modules (PAM) feature allows you to enforce strong user authentication and password policies, including rules for password complexity, length, age, expiration and the reuse of previous passwords. You can configure PAM to block user access after too many failed login attempts, after normal working hours, or if too many concurrent sessions are opened.

The PAM configuration file (/etc/pam.d/system-auth) contains the following default entries for testing a password's strength:

The line for pam_pwquality.so defines that a user gets three attempts to choose a good password. From the module's default settings, the password length must a minimum of six characters, of which three characters must be different from the previous password. The module only tests the quality of passwords for users who are defined in /etc/passwd.

```
password  requisite   pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
password  sufficient  pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password  required    pam_deny.so
```

The line for pam_unix.so specifies that the module tests the password previously specified in the stack before prompting for a password, if necessary (pam_pwquality will already have performed such checks

for users defined in /etc/passwd), uses SHA-512 password hashing and the /etc/shadow file, and allows access if the existing password is null.

# 9 Mounting Filesystems

To access a file system's contents, you must attach its block device to a mount point in the directory hierarchy. Any directory can be used to function as a mount point.

Generally, you create a new directory for a mount point. If you use an existing directory, the contents remain hidden until you unmount the overlying file system.

You use the mount command to attach the device containing the file system to the mount point as follows:

# sudo mount [options] device mount_point.

The device can be mounted by referencing its name, UUID, or label. For example, to mount the file system that was created in the previous section to /var/projects, any of the following commands can be used after you create the directory by running the following commands:

# sudo mkdir /var/projects
# sudo mount /dev/sdb1 /var/projects
# sudo mount UUID="ad8113d7-b279-4da8-b6e4-cfba045f66ff" /var/projects
# sudo mount LABEL="Projects" /var/projects

Issuing the mount command by itself displays all of the currently mounted file systems. In the following example, an extract of the command's output indicates the following:

- /dev/sdb1 with an ext4 file system is mounted on /var/projects for both reading and writing

- /dev/mapper/vg_host01-lv_root, an LVM logical volume also with an ext4 file system, is mounted on / for both reading and writing:

# sudo mount

# 10 Creating User Accounts

To create a user account by using the useradd command:

Enter the following command to create a user account:

# 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.

Assign a password to the account to unlock it:

# 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.

## 10.1 Locking an Account

To lock a user's account, enter:
# passwd -l username
To unlock the account:
# passwd -u username

## 10.2 Modifying or Deleting User Accounts

To modify a user account, use the usermod command:
Creating Groups
# 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 bin daemon sys adm disk wheel
To delete a user's account, use the userdel command:
# userdel username

### 10.3 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

### 10.4 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

### 10.5 Changing User Passwords

One can change the user password by using the following command:

# passwd <username>

### 10.6 Password Policy

It is recommended that users choose password that are strong. The initial password set by the administrator be changed when you first login to the system. Password length should be a minimum of 8 characters. Passwords can be comprised of special characters, upper case, lower case, and numeric characters.

## 11 Management Functions

The TOE maintains the following roles: Administrator and User.

The management functions are listed below:

| **Management Function** | **Administrator** | **User** |
|---|---|---|
| Enable/disable [*session timeout*] | X | |
| Configure [*session]* inactivity timeout | X | |
| Configure local audit storage capacity | X | |
| Configure minimum password Length | X | |
| Configure minimum number of special characters in password | X | |
| Configure minimum number of numeric characters in password | X | |

| Management Function | Administrator | User |
|---|---|---|
| Configure minimum number of uppercase characters in password | X | |
| Configure minimum number of lowercase characters in password | X | |
| Configure lockout policy for unsuccessful authentication attempts through [*limiting number of attempts during a time period]* | X | |
| Configure host-based firewall | X | |
| Configure audit rules | X | |
| Configure name/address of network time server | X | |
| Enable/disable automatic software update | X | |

# 12 System Firewall

You can configure the firewall by using the firewall-cmd command and its multiple options.

In Oracle Linux 8.4, the firewall service, firewalld, is enabled by default. The service is controlled by the systemctl command.

To start the service:

# systemctl unmask firewalld

# systemctl start firewalld

To ensure that the service starts automatically when the system starts, run the following command after starting the firewall:

# sudo systemctl enable firewalld

To stop the firewall service and prevent it from automatically starting when the system starts, run the following command:

# sudo systemctl stop firewalld

# sudo systemctl disable firewalld

To prevent the firewall service from being started by other services or through the firewalld D-Bus interface, run the following command after disabling the firewall:

# sudo systemctl mask firewalld

To display the current status of the firewall service:

3 sudo systemctl status firewalld

# 13 Network Time Service

The chrony is a feature that implements NTP to maintain accurate timekeeping on the network. In Oracle Linux 8, the chrony daemon service replaces ntpd for the management of NTP. Chrony has two components, which are provided in the chrony package:

- chronyd service daemon
- chornyc service utility

The chronyd service daemon enables mobile systems and virtual machines to update their system clock after a period of suspension or disconnection from a network. The service can also be used to implement a simple NTP client or NTP server. As an NTP server, chronyd can synchronize with higher stratum NTP servers or act as a stratum 1 server using time signals that are received from the Global Positioning System (GPS) or radio broadcasts such as DCF77, MSF, or WWVB. In an Oracle Linux 8 system, this service daemon is enabled by default.

To configure the chronyd service on a system:

1. Install the chrony package.

# sudo dnf install chrony

2. If remote access to the local NTP service is required, configure the system firewall to allow access to the NTP service in the appropriate zones, for example:

# firewall-cmd --zone=zone --add-service=ntp

# firewall-cmd --zone=zone --permanent --add-service=ntp

3. If necessary, start the chronyd service and configure it to start following a system reboot.

Note that by default, chrony is enabled after installation.

# sudo systemctl start chronyd

# sudo systemctl enable chronyd

4. In the /etc/chrony.conf file, the default configuration assumes that the system has network access to public NTP servers with which it can synchronise. The following example configuration for a system enables it to access three NTP servers:

pool NTP_server_1

pool NTP_server_2

pool NTP_server_3

driftfile /var/lib/chrony/drift

keyfile /etc/chrony.keys

...

5. To configure chronyd to act as an NTP server for a specified client or subnet, use the allow directive, as shown in bold in the following example:

pool NTP_server_1

pool NTP_server_2

pool NTP_server_3

allow 192.168.2/24

driftfile /var/lib/chrony/drift

keyfile /etc/chrony.keys

...

# 14 Session Timeout

The OS supports CLI and SSH session timeouts.

The session inactivity timeout on the terminal is defined by a time-out in /etc/profile file by adjusting the following parameter:

```
# TMOUT=<time in milliseconds for delay>
```

For remote SSH, the session timeout can be set with the ClientAliveInterval. As an administrator user, open the sshd_config file:

```
# vi /etc/ssh/sshd_config
```

Locate and set the ClientAliveInterval option to 60 (in seconds) or add the value if it is not there.

Locate and set the ClientAliveCountMax option to 0 or add the value if it is not there.

```
ClientAliveInterval 60

ClientAliveCountMax 0
```

Note : ClientAliveInterval: number of seconds that the server will wait before sending a null packet to the client (to keep the connection alive).

Restart sshd daemon :

```
# sudo systemctl restart sshd.service
```

# 15 Storage of Sensitive Data

Keys and configuration files are stored in /etc directory. Privileges are controlled by permissions to invoke applications and to access data. Due to privileges being controlled by permissions, this prevents users from performing management functions that they do not have access to.

# 16 Application Developers

Application developers should use the following compiler options as best practice when developing applications invoking the GCC compiler and linker.

The stack-protector-strong flag has been developed to broaden the scope of the stack protection without extending it to every function in the program. The following compiler flags are used to enable stack protection with the GCC compiler:

**-fstack-protector-strong --param=ssp-buffer-size=4**

ASLR improves executable security in terms of memory randomization and access protection. The following compiler flags are used to enable ASLR with the GCC compiler:

**-fpie -Wl, -pie**

# 17 Setting System Time and Date

Date and time representation on a system can be set to match a specific timezone. To list all the available timezones, run:

# timedatectl list-timezones

To set the system timezone to match a value returned from the available timezones, you can run:

# timedatectl set-timezone America/Los_Angeles

One can check your system's current date and time configuration by running the **timedatectl** command on its own

To set system time manually, you can use the timedatectl set-time command. For example. you can run:

# timedatectl set-time "2018-10-28 01:59:59"

This command sets the current system time based on the time specified assuming the currently set system timezone. The command also updates the system Real Time Clock (RTC).

# 18 Applying Updates

Oracle provides regular updates to the Oracle Linux operating system. After initial installation, the update mechanism is fully configured to obtain updates.

To update the system, administrators can use the DNF package manager. Administrators can use the 'dnf | less' command to view the usage of the dnf command one page at a time. This displays the list of commands, a list of plugin commands, and general dnf options.

To list all of the packages installed on the system, run the following command:

# dnf list installed

To display any upgrades that are available for the software packages that are already installed:

# dnf check-update

To upgrade a software package:

# dnf upgrade <package name>

# 19 Auditing

Auditing collects data at the kernel level that you can analyze to identify unauthorized activity. Auditing collects more data in greater detail than system logging. The process of examining audit trails to locate events of interest can be a significant challenge that you will probably need to automate. The audit configuration file, /etc/audit/auditd.conf, defines the data retention policy, the maximum size of the audit volume, the action to take if the capacity of the audit volume is exceeded, and the locations of local and remote audit trail volumes. The default audit trail volume is /var/log/audit/audit.log.

The TOE generates audit events for all start-up and shut-down function. The TOE leverages the Lightweight Audit Framework (LAF) audit system. Audit events are generated for the following audit functions:

- Start-up and shut-down of the audit functions
- Authentication events (Success/Failure)
- Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes)
- Privilege or role escalation events (Success/Failure)

Sample audit records for each of the above audit functions are shown below:

**Start-up of the audit function**

type=SERVICE_START msg=audit(1668998947.188:31): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=kdump comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'

**Shut-down of the audit function**

type=SERVICE_STOP msg=audit(1668998673.598:7626): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-update-utmp comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'

**Authentication Events**

type=USER_AUTH msg=audit(1668577495.686:7305): pid=15023 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=PAM:authentication grantors=pam_unix acct="root" exe="/usr/sbin/sshd" hostname=192.168.2.6 addr=192.168.2.6 terminal=ssh res=success'

type=USER_AUTH msg=audit(1666310642.902:43): pid=3430 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 msg='op=PAM:authentication grantors=? acct="root" exe="/usr/sbin/sshd" hostname=192.168.2.16 addr=192.168.2.16 terminal=ssh res=failed'

**Privilege Escalation**

type=USER_AUTH msg=audit(1668999350.246:68): pid=3735 uid=0 auid=0 ses=1 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:authentication grantors=pam_rootok acct="user1" exe="/usr/bin/su" hostname=oracle-toe addr=? terminal=pts/0 res=success'

type=USER_AUTH msg=audit(1669000203.002:153): pid=4002 uid=1000 auid=0 ses=1 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:authentication grantors=? acct="root" exe="/usr/bin/su" hostname=oracle-toe addr=? terminal=pts/0 res=failed'

**Use of privileged/special rights**

type=USER_CHAUTHTOK msg=audit(1669000184.333:148): pid=3974 uid=0 auid=0 ses=1 subj=unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023 msg='op=PAM:chauthtok grantors=pam_pwquality,pam_unix acct="tester" exe="/usr/bin/passwd" hostname=oracle-toe addr=? terminal=pts/0 res=success'

type=USER_CHAUTHTOK msg=audit(1669002803.249:243): pid=4146 uid=1000 auid=1000 ses=7 subj=unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023 msg='op=attempted-to-change-password id=1000 exe="/usr/bin/passwd" hostname=oracle-toe addr=? terminal=pts/0 res=failed'

Each audit record contains the following information:

- Date and time are marked with “time”
- Type of event is referenced with “type”
- Subject identity is specified by “uid” containing the numeric user ID of the respective user
- Outcome of the event identified with the field “success”
- User identity is given with the “auid” field

The audit configuration file, /etc/audit/auditd.conf, defines the data retention policy, the maximum size of the audit volume, the action to take if the capacity of the audit volume is exceeded, and the locations of local and remote audit trail volumes. The default audit trail volume is /var/log/audit/audit.log.

By default, auditing captures specific events such as system logins, modifications to accounts, and **sudo** actions. You can also configure auditing to capture detailed system call activity or modifications to certain files. The kernel audit daemon (auditd) records the events that you configure, including the event type, a time stamp, the associated user ID, and success or failure of the system call.

The entries in the audit rules file, /etc/audit/audit.rules, determine which events are audited. Each rule is a command-line option that is passed to the **auditctl** command. You should typically configure this file to match your site's security policy.

The following are examples of rules that you might set in the /etc/audit/audit.rules file.

Record all unsuccessful exits from open and truncate system calls for files in the /etc directory hierarchy.

-a exit,always -S open -S truncate -F /etc -F success=0

Record all files opened by a user with UID 10.

-a exit,always -S open -F uid=10

Record all files that have been written to or that have their attributes changed by any user who originally logged in with a UID of 500 or greater.

-a exit,always -S open -F auid>=500 -F perm=wa

Record requests for write or file attribute change access to /etc/sudoers, and tag such record with the string sudoers-change.

-w /etc/sudoers -p wa -k sudoers-change

Record requests for write and file attribute change access to the /etc directory hierarchy.

-w /etc/ -p wa

Require a reboot after changing the audit configuration. If specified, this rule should appear at the end of the /etc/audit/audit.rules file.

-e 2

## 19.1 Starting and Stopping Audit

If the audit daemon is stopped, audit events are not saved until the system is restarted.

To configure auditd to start at boot time:

# systemctl reload auditd

Once auditd is configured, start the service to collect Audit information and store it in the log files. Use the following command as the root user to start auditd:

# service auditd start

To stop audit:

# service auditd stop

The kernel parameter audit=1 to your boot loader configuration file to ensure that all processes, including those launched before the auditd service, are properly connected to the audit subsystem.

## 19.2 Storage of Audit Records

The audit configuration stores audit records in the /var/log/audit/ directory by default. This is configured in the /etc/audit/auditd.conf file. The auditd.conf file can be altered based on the users' local requirements.

You can configure the audit daemon settings appropriately for your local requirements, for example by changing the log file retention policy to never delete old audit logs with the following setting in the /etc/audit/auditd.conf file:

max_log_file_action = KEEP_LOGS

The following settings are found in the /etc/audit/auditd.conf file if a fail-secure audit system is required:

admin_space_left_action = SINGLE

disk_full_action = HALT

disk_error_action = HALT

### 19.3 Retrieving Audit Records

The following command can be used to retrieve information from the audit events:

Searching for events by process ID:

# ausearch -p 4690

# 20 Access Control Lists

POSIX Access Control Lists (ACLs) provide a richer access control model than traditional UNIX Discretionary Access Control (DAC) that sets read, write, and execute permissions for the owner, group, and all other system users. You can configure ACLs that define access rights for more than just a single user or group, and specify rights for programs, processes, files, and directories. If you set a default ACL on a directory, its descendants inherit the same rights automatically. An ACL consists of a set of rules that specify how a specific user or group can access the file or directory with which the ACL is associated. A regular ACL entry specifies access information for a single file or directory. A default ACL entry is set on directories only, and specifies default access information for any file within the directory that does not have an access ACL.

### 20.1 Configuring Access Control Lists

Ensure the acl package is installed. The following command can be used to install it:

# sudo dnf install acl

Edit /etc/fstab and change the entries for the file systems with which you want to use ACLs so that they include the appropriate option that supports ACLs, for example:

LABEL=/work  /work  ext4  acl  0 0

### 20.2 Setting and Displaying Access Control Lists

To add or modify the ACL rules for file, use the setfacl command:

# setfacl -m rules file ...

The rules take the following forms:

[d:]u:user[:permissions]

Sets the access ACL for the user specified by name or user ID. The permissions apply to the owner if a user is not specified.
[d:]g:group[:permissions]

Sets the access ACL for a group specified by name or group ID. The permissions apply to the owning group if a group is not specified.
[d:]m[:][:permissions]

Sets the effective rights mask, which is the union of all permissions of the owning group and all of the user and group entries.

[d:]o[:][:permissions]

Sets the access ACL for other (everyone else to whom no other rule applies).

The permissions are r, w, and x for read, write, and execute as used with chmod. The d: prefix is used to apply the rule to the default ACL for a directory.

To display a file's ACL, use the getfacl command, for example:

# getfacl foofile

If extended ACLs are active on a file, the -l option to ls displays a plus sign (+) after the permissions, for example:

# ls -l foofile
-rw-r--r--+ 1 bob bob  105322 Apr 11 11:02 foofile

The following are examples of how to set and display ACLs for directories and files.

Grant read access to a file or directory by a user.

# setfacl -m u:*user*:r *file*

Display the name, owner, group, and ACL for a file or directory.

# getfacl *file*

Remove write access to a file for all groups and users by modifying the effective rights mask rather than the ACL.

# setfacl -m m::rx *file*

The -x option removes rules for a user or group.

Remove the rules for a user from the ACL of a file.

# setfacl -x u:*user file*

Remove the rules for a group from the ACL of a file.

# setfacl -x g:*group file*

The -b option removes all extended ACL entries from a file or directory.

# setfacl -b *file*

Copy the ACL of file *f1* to file *f2*.

# getfacl *f1* | setfacl --set-file=- *f2*

Set a default ACL of read and execute access for other on a directory:

# setfacl -m d:o:rx *directory*

Promote the ACL settings of a directory to default ACL settings that can be inherited.

# getfacl --access *directory* | setfacl -d -M- *directory*

The -k option removes the default ACL from a directory.

# setfacl -k *directory*

# 21 Self-tests

When an Oracle Linux system boots, it performs the following operations:

The computer's BIOS performs a power-on self-test (POST), and then locates and initializes any peripheral devices including the hard disk.

The BIOS reads the Master Boot Record (MBR) into memory from the boot device. (For GUID Partition Table (GPT) disks, this MBR is the protective MBR on the first sector of the disk.) The MBR stores information about the organization of partitions on that device. On a computer with x86 architecture, the MBR occupies the first 512 bytes of the boot device. The first 446 bytes contain boot code that points to the boot loader program, which can be on the same device or on another device. The next 64 bytes contain the partition table. The final two bytes are the boot signature, which is used for error detection.

The default boot loader program used on Oracle Linux is GRUB 2, which stands for Grand Unified Bootloader version 2.

The boot loader loads the vmlinuz kernel image file into memory and extracts the contents of the initramfs image file into a temporary, memory-based file system (tmpfs).

The kernel loads the driver modules from the initramfs file system that are needed to access the root file system.

The kernel starts the systemd process with a process ID of 1 (PID 1). systemd is the ancestor of all processes on a system. systemd reads its configuration from files in the /etc/systemd directory. The /etc/systemd/system.conf file controls how systemd handles system initialization.

systemd reads the file linked by /etc/systemd/system/default.target, for example /usr/lib/systemd/system/multi-user.target, to determine the default system target.

GRUB 2 can load many operating systems in addition to Oracle Linux and it can chain-load proprietary operating systems. GRUB 2 understands the formats of file systems and kernel executables, which allows it to load an arbitrary operating system without needing to know the exact location of the kernel on the boot device. GRUB 2 behavior is based on configuration files. On BIOS-based systems, the configuration file is /boot/grub2/grub.cfg. On UEFI-based systems, the configuration file is /boot/efi/EFI/redhat/grub.cfg. Each kernel version's boot parameters are stored in independent configuration files in /boot/loader/entries. GRUB 2 requires only the file name and drive partitions to load a kernel. One can configure this information by using the GRUB 2 menu or by entering it on the command line.

The default menu entry is determined by the value of the GRUB_DEFAULT parameter in /etc/default/grub. The value saved allows you to use the grub2-set-default and grub2-reboot commands to specify the default entry. grub2-set-default sets the default entry for all subsequent reboots and grub2-reboot sets the default entry for the next reboot only.

If you specify a numeric value as the value of GRUB_DEFAULT or as an argument to either grub2-reboot or grub2-set-default, GRUB 2 counts the menu entries in the configuration file starting at 0 for the first entry.# grub2-mkconfig

The default menu entry is determined by the value of the GRUB_DEFAULT parameter in /etc/default/grub.

Use the following command to display all of the kernels that are installed and configured on your system:

# sudo grubby --info=ALL

# 22 Reference Identifiers

The TOE verifies that the presented identifier matches the reference identifier according to RFC 6125.

The reference identifiers that are supported are the following:

- DNS host name or IP address can be set in the Common Name field
- DNS host name can be set in the SAN field

The TOE will verify the above identifiers with the presented certificates to ensure that it matches.

Wild cards are supported, and certificate pinning is unsupported.

# 23 References

| Document Name | Version | Date |
|---|---|---|
| Oracle Linux 8.4 Security Target | Version 1.12 | 12 April, 2023 |
| Oracle Linux 8 Installing Oracle Linux | F13930-24 | August 2022 |
| Oracle Linux 8 Enhancing System Security | F22907-21 | May 2022 |
| Oracle Linux Connecting to Remote Systems with OpenSSH | F22963-09 | June 2022 |
| Oracle Linux 8 Setting Up System Users and Authentication | F21455-09 | November 2022 |