Oracle Linux 7

Security Guide

E54670-52
May 2023

Oracle Linux 7 Security Guide,

E54670-52

Contents

Preface

1 About System Security

2 Security Guidelines

3 Secure Installation and Configuration

4 Implementing Oracle Linux Security

5 Using OpenSCAP to Scan for Vulnerabilities

6 FIPS 140-2 Compliance in Oracle Linux 7

Preface

Oracle® Linux 7: Security Guide provides security guidelines for the Oracle Linux 7 operating system. The guide presents steps that you can take to harden an Oracle Linux system and the features that you can use to protect your data and applications. You can tailor the recommendations in the guide to suit your site security policy.

Conventions

The following text conventions are used in this document:

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

Documentation Accessibility

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

For information about the accessibility of the Oracle Help Center, see the Oracle Accessibility Conformance Report at https://www.oracle.com/corporate/accessibility/templates/t2-11535.html.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1 About System Security

This chapter discusses topics related to system security and its implementation in Oracle Linux.

Overview of System Security in Oracle Linux

Oracle Linux provides a complete security stack, from network firewall control to access control security policies, and is designed to be secure by default.

Traditional Linux security is based on a Discretionary Access Control (DAC) policy, which provides minimal protection from broken software or from malware that is running as a normal user or as `root`. The SELinux enhancement to the Linux kernel implements the Mandatory Access Control (MAC) policy, which allows you to define a security policy that provides granular permissions for all users, programs, processes, files, and devices. The kernel's access control decisions are based on all the security relevant information available, and not solely on the authenticated user identity. By default, SELinux is enabled when you install an Oracle Linux system.

Oracle Linux has evolved into a secure enterprise-class operating system that can provide the performance, data integrity, and application uptime necessary for business-critical production environments.

Thousands of production systems at Oracle run Oracle Linux and numerous internal developers use it as their development platform. Oracle Linux is also at the heart of several Oracle engineered systems, including the Oracle Exadata Database Machine, Oracle Exalytics In-Memory Machine, Oracle Exalogic Elastic Cloud, and Oracle Database Appliance.

Oracle On Demand services, which deliver software as a service (SaaS) at a customer's site, via an Oracle data center, or at a partner site, use Oracle Linux at the foundation of their solution architectures. Backed by Oracle support, these mission-critical systems and deployments depend fundamentally on the built-in security and reliability features of the Oracle Linux operating system.

Released under an open-source license, Oracle Linux includes the Unbreakable Enterprise Kernel that provides the latest Linux innovations while offering tested performance and stability. Oracle has been a key participant in the Linux community, contributing code enhancements such as Oracle Cluster File System and the Btrfs file system. From a security perspective, having roots in open source is a significant advantage. The Linux community, which includes many experienced developers and security experts, reviews posted Linux code extensively prior to its testing and release. The open-source Linux community has supplied many security improvements over time, including access control lists (ACLs), cryptographic libraries, and trusted utilities.

Understanding the Oracle Linux Environment

To better understand your security needs, ask yourself the following questions:

Which resources am I protecting?
Many resources in the production environment can be protected, including information in databases accessed by WebLogic Server and the availability, performance, applications, and the integrity of the Web site. Consider the resources you want to protect when deciding the level of security you must provide.

From whom am I protecting the resources?
For most Web sites, resources must be protected from everyone on the Internet. But should the Web site be protected from the employees on the intranet in your enterprise? Should your employees have access to all resources within the WebLogic Server environment? Should the system administrators have access to all WebLogic resources? Should the system administrators be able to access all data? You might consider giving access to highly confidential data or strategic resources to only a few well trusted system administrators. Perhaps it would be best to allow no system administrators access to the data or resources.

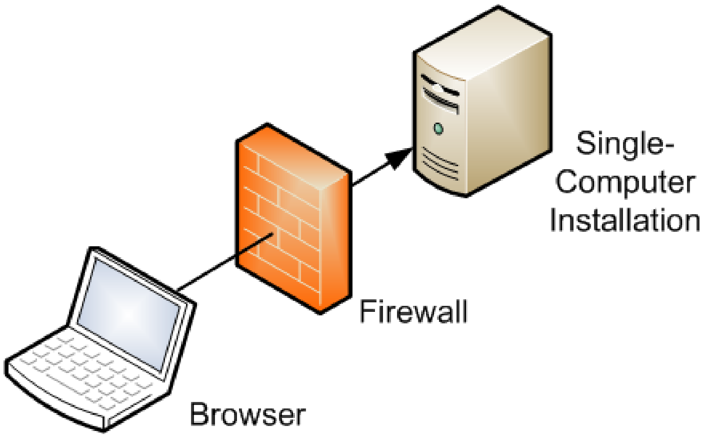
What will happen if the protections on strategic resources fail?
In some cases, a fault in your security scheme is easily detected and considered nothing more than an inconvenience. In other cases, a fault might cause great damage to companies or individual clients that use the Web site. Understanding the security ramifications of each resource will help you protect it properly.

Recommended Deployment Configurations

This section describes recommended architectures for deploying Oracle products with secure Internet access.

Figure 1-1 shows a simple deployment architecture.

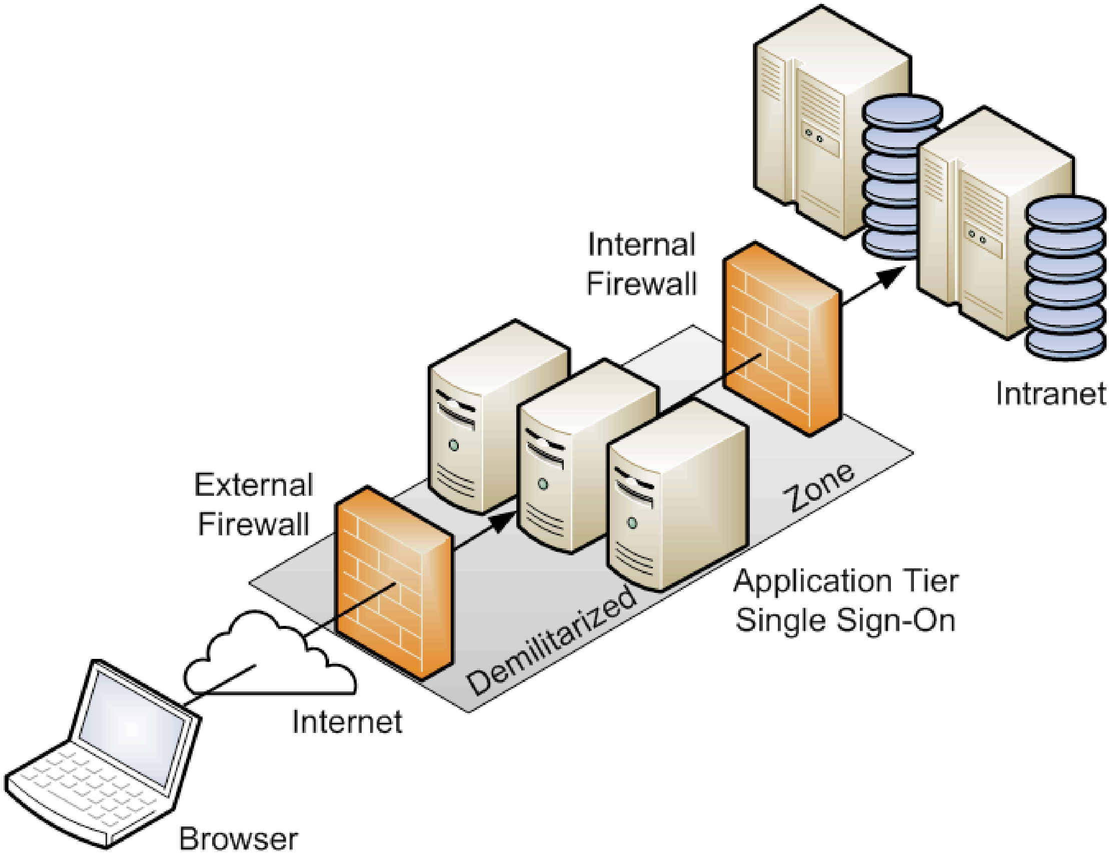
Figure 1-1 Simple Firewall Deployment Configuration



This single-computer deployment may be cost effective for small organizations. However, it cannot provide high availability because all components are stored on the same computer.

Figure 1-2 shows the recommended configuration, which uses the well-known and generally accepted Internet-Firewall-DMZ-Firewall-Intranet architecture.

Figure 1-2 DMZ Deployment Configuration



A *demilitarized zone* (DMZ) refers to a server that is isolated by firewalls from both the Internet and the intranet, and which acts a buffer between them. The firewalls that separate DMZ zones provide two essential functions:

- Block any traffic types that are not permitted.
- Provide intrusion containment in the event that successful intrusions take over processes or processors.

Component Security

Each application software component usually has its own security considerations that you should take into account independently of those that apply to the operating system. Refer to the security guidelines for each component to determine how best to configure it for the requirements of security at your site.

Basic Security Considerations

The following are fundamental principles for using Oracle Linux securely.

Keep Software Up to Date

One of the principles of good security practice is to keep all software versions and patches up to date. Throughout this document, we assume a maintenance level of Oracle Linux Release 7 or later.

For more information, see Configuring and Using Software Management

Restrict Network Access to Critical Services

Keep both middle-tier applications and databases behind a firewall. In addition, place a firewall between middle-tier applications and databases if these are hosted on separate servers. The firewalls provide assurance that access to these systems is restricted to a known network route, which can be monitored and restricted, if necessary. As an alternative, a firewall router substitutes for multiple, independent firewalls.

If firewalls cannot be used, restrict access based upon IP address. Restricting database access by IP address often causes application client/server programs to fail for DHCP clients. To resolve this, consider using static IP addresses, a software/hardware VPN or Windows Terminal Services or its equivalent.

For more information, see Configuring Access to Network Services.

Follow the Principle of Least Privilege

The principle of least privilege states that users should be given the least amount of privilege to perform their jobs. Over ambitious granting of responsibilities, roles, grants, and so on, especially early on in an organization’s life cycle when people are few and work needs to be done quickly, often leaves a system wide open for abuse. User privileges should be reviewed periodically to determine relevance to current job responsibilities.

For more information, see Checking User Accounts and Privileges.

Monitor System Activity

System security stands on three legs: good security protocols, proper system configuration, and system monitoring. Auditing and reviewing audit records address the third requirement. Each component within a system has some degree of monitoring capability. Follow audit advice in this document and regularly monitor audit records.

For more information, see Configuring and Using Auditing.

Keep Up to Date on the Latest Security Information

Oracle continually improves its software and documentation. Check regularly on the Oracle Technology Network at https://www.oracle.com/technetwork/server-storage/linux for revisions. For information about common vulnerabilities and exposures (CVE) and errata that are available on the Unbreakable Linux Network, see https://linux.oracle.com/cve and https://linux.oracle.com/errata.

Security Considerations for Developers

This chapter discusses principles and guidelines that developers need to follow towards writing secure code.

Design Principles for Secure Coding

The following well-established design principles are recommended for secure coding:

Least privilege
A process or user should be given only those privileges that are necessary to complete a task. User privileges should be assigned according to their role, but not otherwise. To create a minimal protection domain, assign rights when a process or thread requires them and remove them afterwards. This principle limits the potential damage that can result from attacks and user errors.

Economy of mechanism
Keep the design simple. There is less to go wrong, fewer inconsistencies are possible, and the code is easier to understand and debug.

Complete mediation
Check every attempt to access to a resource, not just the first. For example, Linux checks access permissions when a process opens a file but not thereafter. If a file's permissions change while a process has the file open, unauthorized access can result. Ideally, one could argue that the permissions should be checked whenever an open file is accessed. In practise, such checking is considered to be an unnecessary overhead given the circumstances under which access was first obtained.

Open design
Security should not depend on the secrecy of the code's design or implementation, sometimes referred to as *security through obscurity*. For example, an open back door to a system is only as secure as the knowledge of its existence. Of course, this principle does not apply to information such as passwords or cryptographic keys, knowledge of which should also be shared among as few people as possible. For this reason, many secure authentication schemes also rely on biometric identification or the possession of a physical artifact such a hardware token or smart card, in addition to knowledge of a PIN code or password.

Separation of privilege
Divide the code into modules, where each module requires a specific, limited set of privileges to perform a specific task. Typically, multiple privileges should be required to grant access to a sensitive operation. This principle ensures separation of duty and provides defense in depth. For example, a main thread that has no privileges can generate a privileged thread to perform a task. A successful attack against the main thread thus gains minimal access to the system.

Least common mechanism
A system should isolate users and their activities from each other. Users should not share processes or threads and information channels should not be shared between users.

Fail-safe defaults
The default action should be to deny access to an operation. Should an attempt to perform an operation be denied, the system is as secure as it was before the operation started.

Accountability
Log the user and their privileges for each action that he or she attempts to perform. Any logs should be capable of being rotated and archived to avoid filling up a file system.

Psychological acceptability
Security mechanisms should be easy to install, configure, and use so that a user is less tempted to try to bypass them.

General Guidelines for Secure Coding

The following coding practices are recommended:

- Check that input data is what the program expects by performing type, length, and bound checking. Inputs include command-line arguments and environment variables in addition to data that a user enters.
- Check input data for the inclusion of constructs such as shell commands, SQL statements, and XML and HTML code that might be used in an injection attack.
- Check the type, length, and bounds of arguments to system calls and library routines. If possible, use library routines that guard against buffer overflows.
- Use full pathnames for file-name arguments, do not use files in world-writable directories, verify that a file being written to is not actually a symbolic link, and protect against the unintended overwriting of existing files.
- Check the type, length, and bounds of values returned by system calls and library routines. Make the code respond appropriately to any error codes that system calls and library functions set or return.
- Do not assume the state of the shell environment. Check any settings that a program inherits from the shell, such as the user file-creation mask, signal handling, file descriptors, current working directory, and environment variables, especially `PATH` and `IFS` . Reset the settings if necessary.
- Perform assert checking on variables that can take a finite set of values.
- Log information about privileged actions and error conditions. Do not allow the program to dump a core file on an end-user system.
- Do not echo passwords to the screen, or transmit or store them as clear text. Before transmitting or storing a password, combine it with a salt value and use a secure one-way algorithm such as SHA-512 to create a hash.
- If your program uses a pseudo-random number generating routine, verify that the numbers that it generates are sufficiently random for your requirements. You should also use a good random seed that a potential attacker should not be able to predict. See RFC 4086, Randomness Requirements for Security, for more information.
- It is recommended that Address Space Layout Randomization (ASLR) is enabled on the host system as this feature can help defeat certain types of buffer overflow attacks. See Address Space Layout Randomization.
- When compiling and linking your program, use the Position Independent Executables (PIE) feature to generate a position-independent binary. See Position Independent Executables.
- Consider using `chroot()` to confine the operating boundary of your program to a specified location within a file system.

- Do not execute a shell command by calling `popen()` or `syscall()` from within a program, especially from a `setuid` or `setgid` program.

The following guidelines apply if your program has its `setuid` or `setgid` bit set so that it can perform privileged actions on behalf of a user who does not possess those privileges:

- Do not set the `setuid` or `setgid` bit on shell scripts. However, if you use Perl scripts that are `setuid` or `setgid`, `perl` runs in *taint mode*, which is claimed to be more secure than using the equivalent C code. See the `perlsec(1)` manual page for details.
- Restrict the use of the privilege that `setuid` or `setgid` grants to the functionality that requires it, and then return the effective UID or GID to that of the user. If possible, perform the privileged functionality in a separate, closely-monitored thread or process.
- Do not allow a `setuid` or `setgid` program to execute child processes using `execlp()` or `execvp()`, which use the `PATH` environment variable.

General Guidelines for Network Programs

The following coding practices are recommended for network programs:

- Perform a reverse lookup on an IP address to obtain the fully qualified domain name, and then use that domain name look up the IP address. The two IP addresses should be identical.
- Protect a service against Denial of Service (DoS) attacks by allowing it to stop processing requests if it becomes overloaded.
- Set timeouts on read and write requests over the network.
- Check the content, bounds, value, and type of data received over the network, and reject any data that does not conform to what the program expects.
- Use certificates or preshared keys to authenticate the local and remote ends of the network connection.
- Use a well-established technology such as TLS or SSL to encrypt data sent over the network connection.
- Wherever possible, use existing networking protocols and technologies whose security characteristics are well known.
- Log information about successful and unsuccessful connection attempts, data reception and transmission errors, and changes to the service state.

2 Security Guidelines

This chapter provides guidelines that help secure your Oracle Linux system.

For information about how to use OpenSCAP to scan a system for vulnerabilities, see Using OpenSCAP to Scan for Vulnerabilities.

Minimizing the Software Footprint

On systems on which Oracle Linux has been installed, remove unneeded RPMs to minimize the software footprint. For example, you could uninstall the X Windows package (`xorg-x11-server-Xorg`) if it is not required on a server system.

To discover which package provides a given command or file, use the `yum provides` command, as shown in the following example:

```
yum provides /usr/sbin/sestatus

...
policycoreutils-2.0.83-19.24.0.1.el6.x86_64 : SELinux policy core utilities
Repo        : installed
Matched from:
Other       : Provides-match: /usr/sbin/sestatus
```

To display the files that a package provides, use the `repoquery` utility, which is included in the `yum-utils` package. For example, the following command lists the files that the `btrfs-progs` package provides.

```
repoquery -l btrfs-progs

/sbin/btrfs
/sbin/btrfs-convert
/sbin/btrfs-debug-tree
.
.
.
```

To uninstall a package, use the `yum remove` command, as shown in this example:

```
sudo yum remove xinetd

Loaded plugins: refresh-packagekit, security
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package xinetd.x86_64 2:2.3.14-35.el6_3 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package       Arch          Version                   Repository          Size
================================================================================
Removing:
 xinetd        x86_64        2:2.3.14-35.el6_3         @ol6_latest        259 k

Transaction Summary
================================================================================
Remove        1 Package(s)

Installed size: 259 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing    : 2:xinetd-2.3.14-35.el6_3.x86_64                              1/1
  Verifying  : 2:xinetd-2.3.14-35.el6_3.x86_64                              1/1

Removed:
  xinetd.x86_64 2:2.3.14-35.el6_3

Complete!
```

The following list contains packages that you should not install or that you should remove using the `yum remove` command if they are already installed.

- **`krb5-appl-clients`**

  Kerberos versions of `ftp`, `rcp`, `rlogin`, `rsh` and `telnet`. If possible, use SSH instead.

- **`rsh`, `rsh-server`**

  `rcp`, `rlogin`, and `rsh` use unencrypted communication that can be snooped. Use SSH instead.

- **`samba`**

  Network services used by Samba. Remove this package if the system is not acting as an Active Directory server, a domain controller, or as a domain member, and it does not provide Microsoft Windows file and print sharing functionality.

- **`talk`, `talk-server`**

  `talk` is considered obsolete.

- **`telnet`, `telnet-server`**

  `telnet` uses unencrypted communication that can be snooped. Use SSH instead.

- **`tftp`, `tftp-server`**

  TFTP uses unencrypted communication that can be snooped. Use only if required to support legacy hardware. If possible, use SSH or other secure protocol instead.

- **`xinetd`**

  The security model used by the Internet listener daemon is deprecated.

- **`ypbind`, `ypserv`**

  The security model used by NIS is inherently flawed. Use an alternative such as LDAP or Kerberos instead.

Configuring System Logging

Verify that the rsyslog logging service is running:

```
sudo systemctl is-active rsyslog
```

```
active
```

If the rsyslogd service is not running, start it and enable it to start when the system is rebooted:

```
sudo systemctl start rsyslog
sudo systemctl enable rsyslog
```

Ensure that each log file referenced in /etc/rsyslog.conf exists and is owned and only readable by root:

```
touch logfile
sudo chown root:root logfile
sudo chmod 0600 logfile
```

It is also recommended that you use a central log server and that you configure Logwatch on that server. See Configuring and Using System Logging.

Disabling Core Dumps

Core dumps can contain information that an attacker might be able to exploit and they take up a large amount of disk space. To prevent the system creating core dumps when the operating system terminates a program due to a segment violation or other unexpected error, add the following line to /etc/security/limits.conf:

```
*  hard  core  0
```

You can restrict access to core dumps to certain users or groups, as described in the limits.conf(5) manual page.

By default, the system prevents setuid and setgid programs, programs that have changed credentials, and programs whose binaries do not have read permission from dumping core. To ensure that the setting is permanently recorded, add the following lines to /etc/sysctl.conf:

```
# Disallow core dumping by setuid and setgid programs
fs.suid_dumpable = 0
```

Then, run the sysctl -p command.

> **Note:**
>
> A value of 1 permits core dumps that are readable by the owner of the dumping process. A value of 2 permits core dumps that are readable only by root for debugging purposes.

Minimizing Active Services

Restrict services to only those that a server requires. The default installation for an Oracle Linux server configures a minimal set of services:

- `cupsd` and `lpd` (print services)
- `sendmail` (email delivery service)
- `sshd` (openSSH services)

If possible, configure one type of service per physical machine, virtual machine, or Linux Container. This technique limits exposure if a system is compromised.

If a service is not used, remove the software packages that are associated with the service. If it is not possible to remove a service because of software dependencies, use the `chkconfig` and `service` commands to disable the service.

For services that are in use, apply the latest Oracle support patches and security updates to keep software packages up to date. To protect against unauthorized changes, ensure that the `/etc/services` file is owned by `root` and writable only by `root`.

```
ls -Z /etc/services

-rw-r--r--. root root system_u:object_r:etc_t:SystemLow /etc/services
```

Unless specifically stated otherwise, consider disabling the services that are described in the following list, if they are not used on your system.

`anacron`
Executes commands periodically. Primarily intended for use on laptop and user desktop machines that do not run continuously.

`automount`
Manages mount points for the automatic file-system mounter. Disable this service on servers that do not require automounter functionality.

`bluetooth`
Supports the connections of Bluetooth devices. Primarily intended for use on laptop and user desktop machines. Bluetooth provides an additional potential attack surface. Disable this service on servers that do not require Bluetooth functionality.

`gpm`
(General Purpose Mouse) Provides support for the mouse pointer in a text console.

`hidd`
(Bluetooth Human Interface Device daemon) Provides support for Bluetooth input devices such as a keyboard or mouse. Primarily intended for use on laptop and user desktop machines. Bluetooth provides an additional potential attack surface. Disable this service on servers that do not require Bluetooth functionality.

`irqbalance`
Distributes hardware interrupts across processors on a multiprocessor system. Disable this service on servers that do not require this functionality.

`iscsi`
Controls logging in to iSCSI targets and scanning of iSCSI devices. Disable this service on servers that do not access iSCSI devices.

`iscsid`
Implements control and management for the iSCSI protocol. Disable this service on servers that do not access iSCSI devices.

`kdump`
Enables a `kdump` kernel to be loaded into memory at boot time or a kernel dump to be saved if the system panics. Disable this service on servers that you do not use for debugging or testing.

`mcstrans`
Controls the SELinux Context Translation System service.

`mdmonitor`
Checks the status of all software RAID arrays on the system. Disable this service on servers that do not use software RAID.

`pcscd`
(PC/SC Smart Card Daemon) Supports communication with smart-card readers. Primarily intended for use on laptop and user desktop machines to support smart-card authentication. Disable this service on servers that do not use smart-card authentication.

`sandbox`
Sets up `/tmp`, `/var/tmp`, and home directories to be used with the `pam_namespace`, `sandbox`, and `xguest` application confinement utilities. Disable this service if you do not use these programs.

`setroubleshoot`
Controls the SELinux Troubleshooting service, which provides information about SELinux Access Vector Cache (AVC) denials to the `sealert` tool.

`smartd`
Communicates with the Self-Monitoring, Analysis and Reporting Technology (SMART) systems that are integrated into many ATA-3 and later, and SCSI-3 disk drives. SMART systems monitor disk drives to measure reliability, predict disk degradation and failure, and perform drive testing.

`xfs`
Caches fonts in memory to improve the performance of X Window System applications.

Consider disabling the network services that are described in the following table, if they are not used on your system.

`avahi-daemon`
Implements Apple's Zero configuration networking (also known as Rendezvous or Bonjour). Primarily intended for use on laptop and user desktop machines to support music and file sharing. Disable this service on servers that do not require this functionality.

`cups`
Implements the Common UNIX Printing System. Disable this service on servers that do not need to provide this functionality.

hplip
Implements HP Linux Imaging and Printing to support faxing, printing, and scanning operations on HP inkjet and laser printers. Disable this service on servers that do not require this functionality.

isdn
(Integrated Services Digital Network) Provides support for network connections over ISDN devices. Disable this service on servers that do not directly control ISDN devices.

netfs
Mounts and unmounts network file systems, including NCP, NFS, and SMB. Disable this service on servers that do not require this functionality.

network
Activates all network interfaces that are configured to start at boot time.

NetworkManager
Switches network connections automatically to use the best connection that is available.

nfslock
Implements the Network Status Monitor (NSM) used by NFS. Disable this service on servers that do not require this functionality.

nmb
Provides NetBIOS name services used by Samba. Disable this service and remove the `samba` package if the system is not acting as an Active Directory server, a domain controller, or as a domain member, and it does not provide Microsoft Windows file and print sharing functionality.

portmap
Implements Remote Procedure Call (RPC) support for NFS. Disable this service on servers that do not require this functionality.

rhnsd
Queries the Unbreakable Linux Network (ULN) for updates and information.

rpcgssd
Used by NFS. Disable this service on servers that do not require this functionality.

rpcidmapd
Used by NFS. Disable this service on servers that do not require this functionality.

smb
Provides SMB network services used by Samba. Disable this service and remove the `samba` package if the system is not acting as an Active Directory server, a domain controller, or as a domain member, and it does not provide Microsoft Windows file and print sharing functionality.

To stop a service and prevent it from starting when you reboot the system, used the following commands:

```
sudo systemctl stop service_name
sudo systemctl disable service_name
```

Locking Down Network Services

> **Note:**
>
> It is recommended that you do not install the `xinetd` Internet listener daemon. If you do not need this service, remove the package altogether by using the `yum remove xinetd` command.

If you must enable `xinetd` on your system, minimize the network services that `xinetd` can launch by disabling those services that are defined in the configuration files in `/etc/xinetd.d` and which are not needed.

To counter potential Denial of Service (DoS) attacks, you can configure the resource limits for such services by editing `/etc/xinetd.conf` and related configuration files. For example, you can set limits for the connection rate, the number of connection instances to a service, and the number of connections from an IP address:

```
# Maximum number of connections per second and
# number of seconds for which a service is disabled
# if the maximum number of connections is exceeded
cps             = 50 10

# Maximum number of connections to a service
instances       = 50

# Maximum number of connections from an IP address
per_source      = 10
```

For more information, see the `xinetd(8)` and `xinetd.conf(5)` manual pages.

Configuring a Packet-Filtering Firewall

You can configure the Netfilter feature to act as a packet-filtering firewall that uses rules to determine whether network packets are received, dropped, or forwarded.

The primary interfaces for configuring the packet-filter rules are the `firewall-cmd` command and the Firewall Configuration GUI (`firewall-config`) or the `iptables` and `ip6tables` utilities. By default, the rules should drop any packets that are not destined for a service that the server hosts or that originate from networks other than those to which you want to allow access.

In addition, you can use Network Address Translation (NAT) to hide IP addresses behind a public IP address, and IP masquerading to alter IP header information for routed packets. You can also set rule-based packet logging and define a dedicated log file in `/etc/syslog.conf`.

For more information, see Configuring Packet-filtering Firewalls.

Configuring TCP Wrappers

The TCP wrappers feature mediates requests from clients to services, and control access based on rules that you define in the /etc/hosts.deny and /etc/hosts.allow files. You can restrict and permit service access for specific hosts or whole networks. A common way of using TCP wrappers is to detect intrusion attempts. For example, if a known malicious host or network attempts to access a service, you can deny access and send a warning message about the event to a log file or to the system console.

For more information, see Configuring TCP Wrappers.

Configuring Kernel Parameters

You can use several kernel parameters to counteract various kinds of attack.

- kernel.randomize_va_space: Controls Address Space Layout Randomization (ASLR), which can help defeat certain types of buffer overflow attacks. A value of 0 disables ASLR, 1 randomizes the positions of the stack, virtual dynamic shared object (VDSO) page, and shared memory regions, and 2 randomizes the positions of the stack, VDSO page, shared memory regions, and the data segment. The default and recommended setting is 2.
- net.ipv4.conf.all.accept_source_route: Controls the handling of source-routed packets, which might have been generated outside the local network. A value of 0 rejects such packets, and 1 accepts them. The default and recommended setting is 0.
- net.ipv4.conf.all.rp_filter: Controls reversed-path filtering of received packets to counter IP address spoofing. A value of 0 disables source validation, 1 causes packets to be dropped if the routing table entry for their source address does not match the network interface on which they arrive, and 2 causes packets to be dropped if source validation by reversed path fails (see RFC 1812). The default setting is 0. A value of 2 can cause otherwise valid packets to be dropped if the local network topology is complex and RIP or static routes are used.
- net.ipv4.icmp_echo_ignore_broadcasts: Controls whether ICMP broadcasts are ignored to protect against Smurf DoS attacks. A value of 1 ignores such broadcasts, and 0 accepts them. The default and recommended setting is 1.
- net.ipv4.icmp_ignore_bogus_error_message: Controls whether ICMP bogus error message responses are ignored. A value of 1 ignores such messages, and 0 accepts them. The default and recommended setting is 1.

To change the value of a kernel parameter, add the setting to /etc/sysctl.conf, for example:

```
kernel.randomize_va_space = 1
```

Then, run the sysctl -p command.

For additional security configurations on the kernel, see Configuring and Using Kernel Security Mechanisms.

Restricting Access to SSH Connections

The Secure Shell (SSH) allows protected, encrypted communication with other systems. As SSH is an entry point into the system, disable it if it is not required, or alternatively, edit the /etc/ssh/sshd_config file to restrict its use.

For example, the following setting does not allow root to log in using SSH:

```
PermitRootLogin no
```

You can restrict remote access to certain users and groups by specifying the AllowUsers, AllowGroups, DenyUsers, and DenyGroups settings, for example:

```
DenyUsers carol dan
AllowUsers alice bob
```

The ClientAliveInterval and ClientAliveCountMax settings cause the SSH client to time out automatically after a period of inactivity, for example:

```
# Disconnect client after 300 seconds of inactivity
ClientAliveCountMax 0
ClientAliveInterval 300
```

After changing the configuration file, restart the sshd service for the changes to take effect.

For more information, see the sshd_config(5) manual page.

Configuring File System Mounts, File Permissions, and File Ownership

Use separate disk partitions for operating system and user data to prevent a *file system full* issue from impacting the operation of a server. For example, you might create separate partitions for /home, /tmp, p, /oracle, and so on.

Establish disk quotas to prevent a user from accidentally or intentionally filling up a file system and denying access to other users.

To prevent the operating system files and utilities from being altered during an attack, mount the /usr file system read-only. If you need to update any RPMs on the file system, use the -o remount,rw option with the mount command to remount /usr for both read and write access. After performing the update, use the -o remount,ro option to return the /usr file system to read-only mode.

To limit user access to non-root local file systems such as /tmp or removable storage partitions, specify the -o noexec, nosuid, nodev options to mount. These option prevent the execution of binaries (but not scripts), prevent the setuid bit from having any effect, and prevent the use of device files.

Use the find command to check for unowned files and directories on each file system, for example:

```
find mount_point -mount -type f -nouser -o -nogroup -exec ls -l {} \;
find mount_point -mount -type d -nouser -o -nogroup -exec ls -l {} \;
```

Unowned files and directories might be associated with a deleted user account, they might indicate an error with software installation or deleting, or they might a sign of an intrusion on the system. Correct the permissions and ownership of the files and directories that you find, or remove them. If possible, investigate and correct the problem that led to their creation.

Use the `find` command to check for world-writable directories on each file system, for example:

```
find mount_point -mount -type d -perm /o+w -exec ls -l {} \;
```

Investigate any world-writable directory that is owned by a user other than a system user. The user can remove or change any file that other users write to the directory. Correct the permissions and ownership of the directories that you find, or remove them.

You can also use `find` to check for `setuid` and `setgid` executables.

```
find path -type f \( -perm -4000 -o -perm -2000 \) -exec ls -l {} \;
```

If the `setuid` and `setgid` bits are set, an executable can perform a task that requires other rights, such as `root` privileges. However, buffer overrun attacks can exploit such executables to run unauthorized code with the rights of the exploited process.

If you want to stop a `setuid` and `setgid` executable from being used by non-`root` users, you can use the following commands to unset the `setuid` or `setgid` bit:

```
sudo chmod u-s file
sudo chmod g-s file
```

The following table lists programs for which you might want to consider unsetting the `setuid` and `setgid`.

> **Note:**
>
> The list is not exhaustive, as many optional packages contain `setuid` and `setgid` programs.

Program File	Bit Set	Description of Usage
`/usr/bin/chage`	`setuid`	Determines password aging information (via the `-l` option).
`/usr/bin/chfn`	`setuid`	Changes `finger` information.
`/usr/bin/chsh`	`setuid`	Changes the login shell.
`/usr/bin/crontab`	`setuid`	Edits, lists, or removes a `crontab` file.
`/usr/bin/wall`	`setgid`	Sends a system-wide message.
`/usr/bin/write`	`setgid`	Sends a message to another user.

Program File	Bit Set	Description of Usage
`/usr/bin/Xorg`	`setuid`	Invokes the X Windows server.
`/usr/libexec/openssh/ssh-keysign`	`setuid`	Runs the SSH helper program for host-based authentication.

Program File	Bit Set	Description of Usage
`/usr/sbin/mount.nfs`	`setuid`	Mounts an NFS file system.

Note:

`/sbin/mount.nfs4`, `/sbin/umount.nfs`, and `/`

Program File	Bit Set	Description of Usage
		`sbin/umount.nfs4` are symbolic links to this file.
`/usr/sbin/netreport`	`setgid`	Requests notification of changes to network interfaces.

Program File	Bit Set	Description of Usage
/usr/sbin/usernetctl	setuid	Controls network interfaces. Permission for a user to alter the state of a network interface also requires USERCTL=yes to be set in the interface file. You can also grant users and groups the privilege to run the ip command by creating a suitable entry in the /etc/sudoers file.

Checking User Accounts and Privileges

Check the system for unlocked user accounts on a regular basis by using a command similar to the following:

```
for u in `cat /etc/passwd | cut -d: -f1 | sort`; do passwd -S $u; done

abrt LK 2012-06-28 0 99999 7 -1 (Password locked.)
adm LK 2011-10-13 0 99999 7 -1 (Alternate authentication scheme in use.)
apache LK 2012-06-28 0 99999 7 -1 (Password locked.)
avahi LK 2012-06-28 0 99999 7 -1 (Password locked.)
avahi-autoipd LK 2012-06-28 0 99999 7 -1 (Password locked.)
bin LK 2011-10-13 0 99999 7 -1 (Alternate authentication scheme in use.)
...
```

In the output that is shown in this example, the second field indicates whether a user account is locked (LK), does not have a password (NP), or has a valid password (PS). The third field shows the date on which the user last changed their password. The remaining fields show the minimum age, maximum age, warning period, and inactivity period for the password and additional information about the password's status. The unit of time is days.

Use the passwd command to set passwords on any accounts that are not protected.

Use the passwd -l command to lock unused accounts. Alternatively, use userdel to remove the accounts entirely.

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

To specify how user passwords are aged, edit the settings in the /etc/login.defs file. These settings are described in the following list.

- PASS_MAX_DAYS: Maximum number of days for which a password can be used before it must be changed. The default value is 99,999 days.
- 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:

```
sudo usermod -f 30 username
```

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

```
sudo 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.

Verify that no user accounts other than root have a user ID of 0.

```
awk -F":" '$3 == 0 { print $1 }' /etc/passwd
```

```
root
```

If you install software that creates a default user account and password, change the vendor's default password immediately. Centralized user authentication using an LDAP implementation such as OpenLDAP can help to simplify user authentication and management tasks, and also reduces the risk arising from unused accounts or accounts without a password.

By default, an Oracle Linux system is configured so that you cannot log in directly as root. 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 rpm and yum:

```
erin            ALL=(ALL)       ALL
frank           ALL=SOFTWARE
```

Oracle Linux supports the pluggable authentication modules (PAM) feature, which makes it easier 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.

PAM is highly customizable by its use of different modules with customisable parameters. For example, the default password integrity checking module pam_pwquality.so tests password strength. The PAM configuration file (/etc/pam.d/system-auth) contains the following default entries for testing a password's strength:

```
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_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.

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.

You can modify the control flags and module parameters to change the checking that is performed when a user changes his or her password, for example:

```
password  required  pam_pwquality.so retry=3 minlen=8 difok=5 minclass=-1
password  required  pam_unix.so use_authtok sha512 shadow remember=5
password  required  pam_deny.so
```

The line for pam_pwquality.so defines that a user gets three attempts to choose a good password with a minimum of eight characters, of which five characters must be different from the previous password, and which must contain at least one upper case letter, one lower case letter, one numeric digit, and one non-alphanumeric character.

The line for pam_unix.so specifies that the module does not perform password checking, uses SHA-512 password hashing and the /etc/shadow file, and saves information about the previous five passwords for each user in the /etc/security/opasswd file. As nullok is not specified, a user cannot change his or her password if the existing password is null.

The omission of the try_first_pass keyword means that the user is always asked for their existing password, even if he or she entered it for the same module or for a previous module in the stack.

For more information, see Configuring and Using Pluggable Authentication Modules and the pam_deny(8), pam_pwquality(8), and pam_unix(8) manual pages.

3 Secure Installation and Configuration

This chapter outlines the planning process for a secure installation and describes how the choices that you make during installation affect system security.

Pre-Installation Tasks

An important consideration is the security of the physical system on which you will install Oracle Linux. If possible, keep server systems in a locked data center and limit access to authorized personnel. Such personnel should also receive appropriate administrative training as human error is often the cause of a security breach. For more information about the available Oracle Linux coursework and certification options, see https://education.oracle.com.

Aside from the risks of theft and data compromise, physical security is critical because it prevents an unauthorized user from possibly modifying the system BIOS, altering the boot device, and booting from an alternate medium. If a system is not kept in a locked data center, consider password-protecting the BIOS. Consult the system manufacturer's documentation for information about setting a BIOS password. Edit the BIOS settings to disable booting from the CD-ROM drive, floppy disk drive, USB ports, and other external devices. In addition, you can configure disk encryption during installation, or password-protect the GRUB boot loader after installation.

> **Note:**
>
> Setting a BIOS, encrypted disk, or boot-loader password requires you to enter the password whenever you reboot the system. Only disk encryption can prevent access to the data on disk when an attacker uses techniques such as resetting the BIOS, accessing the disk by booting an operating system from a memory stick, or simply removing the hard drive to read its contents on another system.

Installing Oracle Linux

When you install Oracle Linux, you can reduce the attack surface by installing only the software packages that are required for operation. Software packages are a potential source of `setuid` programs, network services, and libraries that an attacker can potentially use to gain access illegitimately and compromise a system.

You can use a pretested kickstart profile to provide consistent and precise control over what is installed. Automated installation using a kickstart profile reduces both security risk and administrative effort.

Alternatively, you can use Oracle Enterprise Manager Ops Center, which supports the import of OS images and explicit provisioning profiles. For more information, refer to the Oracle Enterprise Manager Ops Center documentation.

Shadow Passwords and Hashing Algorithms

By default, an Oracle Linux system is configured to use password hashes that are stored in the /etc/shadow file rather than in the world-readable /etc/passwd file. If shadow passwords were not used, an attacker is much more likely to be able to discover a password by applying cracking software to the hashes. Similarly, using a password-hashing algorithm that is weaker than SHA-512 would make it much easier to find likely candidates that match a hash value.

Strong Passwords

During installation, you are prompted to enter passwords for root and one additional user, if you choose the user to be authenticated locally rather than over the network. The passwords that you enter should be strong in that they should be extremely difficult to deduce by guesswork or by other means, such as automated FTP or SSH logins. By default, the installation process rejects null passwords and warns about weak passwords, but it does not enforce strong passwords. It is your responsibility to ensure that passwords are sufficiently strong.

Some general guidelines for creating a strong password are:

- Make the password at least eight characters long.
- Use a mixture of lower and upper case letters, numbers, and other characters.
- Do not include whole words from English, LEET speak, or any other language or technology, even if you spell the words in reverse order.
- Do not include personal information such as names, dates, addresses, email addresses, or telephone numbers.
- Do not use well-known acronyms, abbreviations, or character sequences such as QWERTY.
- Do not use a password that is the same as or very similar to a password that you used previously on the system.
- Use a password for root that is different from the password for any other user.

Separate Disk Partitions

The National Security Agency (NSA) recommendations state that you should set up user-writable file systems such as /home, /tmp, and /var/tmp on partitions that are separate from /. In addition, /boot must be a dedicated file system if you encrypt the root file system.

Encrypted Disk Partitions

When choosing a disk layout, you have the option of encrypting disk partitions with the Linux Unified Key Setup (LUKS) format. As for any other password, ensure that you enter a strong passphrase if you choose to encrypt any partitions.

> **Note:**
>
> The `/boot` file system cannot be encrypted.

Software Selection

If you choose to customize the software to be installed on a system, you can select or deselect packages from the default set. For example, the basic server configuration does not install the Gnome and KDE desktop software and the X Windows System packages from the **Desktops** section. Additional packages that you might want to install on a server system are available under the **Servers**, **Web Services**, **Databases**, and other section headings.

Network Time Service

If you choose to synchronize the data and time over the network, the system is configured as an NTP client that uses the `[012].rhel.pool.ntp.org` public servers by default. If your systems rely on Kerberos authentication, which requires close synchronization of the clocks on each participating system, you might prefer to configure your systems to use a local NTP server instead.

Post-Installation Tasks

For information about the way that you can configure the security of an Oracle Linux system, see Implementing Oracle Linux Security.

For guidelines about hardening an Oracle Linux system, see Security Guidelines.

4 Implementing Oracle Linux Security

This chapter describes the various ways in which you can configure the security of an Oracle Linux system.

Configuring Access to Network Services

As networks are usually the primary point of entry point into IT systems, you can use network intrusion prevention and detection tools to help avert or uncover a security breach. You can then take steps such as disabling unused network services and configure a packet-filtering firewall and TCP wrappers.

There are several open-source tools for performing packet logging and analysis. For example, tcpdump and Snort capture TCP traffic and analyze it for suspicious usage patterns, such as those that typically occur with port scans or network DoS attacks. Sguil incorporates tcpdump, Snort, and the Wireshark protocol analyzer to provide a network intrusion and detection system that simplifies log analysis and reporting.

You can check what services are running on a system by using port scanning utilities. The following examples show the information that the `netstat`, `lsof`, and `nmap` commands return about open TCP ports and the associated services:

```
netstat -tulp

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/
Program name
tcp        0      0 localhost:9003          0.0.0.0:*               LISTEN      1776/
osms-agent
tcp        0      0 0.0.0.0:sunrpc          0.0.0.0:*               LISTEN      1042/
rpcbind
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      2051/
sshd
tcp6       0      0 [::]:sunrpc             [::]:*                  LISTEN      1042/
rpcbind
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      2051/
sshd
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*                           1465/
dhclient
udp        0      0 0.0.0.0:sunrpc          0.0.0.0:*                           1042/
rpcbind
udp        0      0 localhost:323           0.0.0.0:*                           1062/
chronyd
udp        0      0 0.0.0.0:789             0.0.0.0:*                           1042/
rpcbind
udp6       0      0 [::]:sunrpc             [::]:*                              1042/
rpcbind
udp6       0      0 localhost:323           [::]:*                              1062/
chronyd
udp6       0      0 [::]:789                [::]:*                              1042/
rpcbind

lsof -iTCP -sTCP:LISTEN
```

```
COMMAND     PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
rpcbind   1042  rpc    8u  IPv4  19998      0t0  TCP *:sunrpc (LISTEN)
rpcbind   1042  rpc   11u  IPv6  20001      0t0  TCP *:sunrpc (LISTEN)
osms-agen 1776 root   10u  IPv4  26707      0t0  TCP localhost:9003 (LISTEN)
sshd      2051 root    3u  IPv4  25784      0t0  TCP *:ssh (LISTEN)
sshd      2051 root    4u  IPv6  25786      0t0  TCP *:ssh (LISTEN)

nmap -sTU 10.0.2.15

Starting Nmap 5.51 ( http://nmap.org ) at 2012-12-10 09:37 GMT
Nmap scan report for 10.0.2.15
Host is up (0.0017s latency).
Not shown: 1993 closed ports
PORT     STATE         SERVICE
22/tcp   open          ssh
111/tcp  open          rpcbind
68/udp   open|filtered dhcpc
111/udp  open          rpcbind
123/udp  open          ntp
631/udp  open|filtered ipp
5353/udp open|filtered zeroconf

Nmap done: 1 IP address (1 host up) scanned in 12.66 seconds
```

For more information, see the `lsof(8)`, `netstat(8)`, and `nmap(1)` manual pages.

> **Caution:**
>
> Before installing or using the `nmap` command, check the local legislation relating to port scanning software. In some jurisdictions, the possession or use of port scanning software is considered as unlawful criminal activity. Some ISPs might also have acceptable use policies that forbid using such software outside of your private networks.

The two sections in this chapter, Configuring Packet-filtering Firewalls and Configuring TCP Wrappers, are specific methods to restrict access to network services.

Configuring Packet-filtering Firewalls

A packet filtering firewall filters incoming and outgoing network packets based on the packet header information. You can create packet filter rules that determine whether packets are accepted or rejected. For example, if you create a rule to block a port, any request is made to that port that is blocked by the firewall, and the request is ignored. Any service that is listening on a blocked port is effectively disabled.

The Oracle Linux kernel uses the Netfilter feature to provide packet filtering functionality for IPv4 and IPv6 packets.

Netfilter consists of two components:

- A `netfilter` kernel component consisting of a set of tables in memory for the rules that the kernel uses to control network packet filtering.

- Utilities to create, maintain, and display the rules that netfilter stores. In Oracle Linux 7, the default firewall utility is firewall-cmd, which is provided by the firewalld package.

  If you prefer, you can enable the iptables and iptables6 services and use the iptables and ip6tables utilities, provided by the iptables package. These were the default utilities for firewall configuration in Oracle Linux 6.

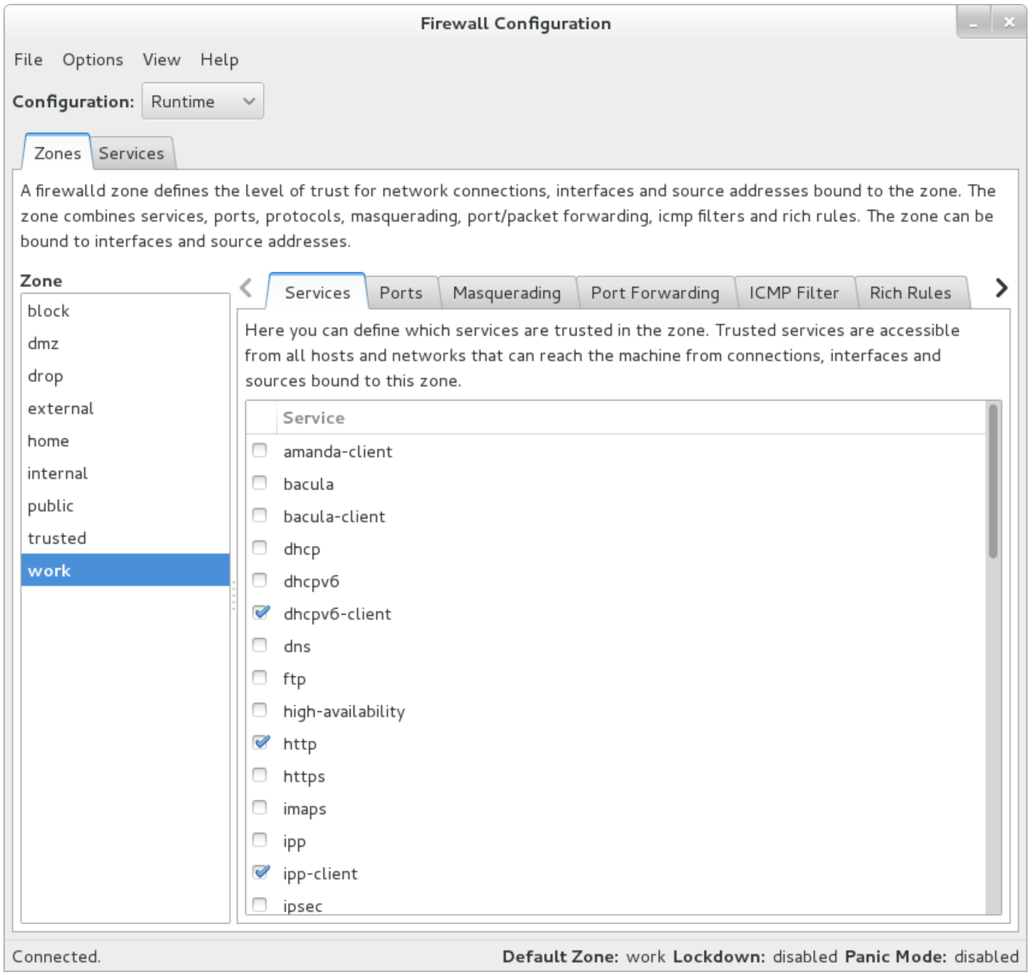
The firewalld-based firewall has the following advantages over an iptables-based firewall:

- Unlike the iptables and ip6tables commands, using firewalld-cmd does not restart the firewall and disrupt established TCP connections.
- firewalld supports dynamic zones, which allow you to implement different sets of firewall rules for systems such as laptops that can connect to networks with different levels of trust. You are unlikely to use this feature with server systems.
- firewalld supports D-Bus for better integration with services that depend on firewall configuration.

To implement a general-purpose firewall, you can use the Firewall Configuration GUI (firewall-config), provided by the firewall-config package.

Figure 4-1 shows the Firewall Configuration GUI.

Figure 4-1 Firewall Configuration



Configuration: Runtime

A firewalld zone defines the level of trust for network connections, interfaces and source addresses bound to the zone. The zone combines services, ports, protocols, masquerading, port/packet forwarding, icmp filters and rich rules. The zone can be bound to interfaces and source addresses.

block
dmz
drop
external
home
internal
public
trusted
work

Here you can define which services are trusted in the zone. Trusted services are accessible from all hosts and networks that can reach the machine from connections, interfaces and sources bound to this zone.

☐ amanda-client
☐ bacula
☐ bacula-client
☐ dhcp
☐ dhcpv6
☑ dhcpv6-client
☐ dns
☐ ftp
☐ high-availability
☑ http
☐ https
☐ imaps
☐ ipp
☑ ipp-client
☐ ipsec

Connected. **Default Zone:** work **Lockdown:** disabled **Panic Mode:** disabled

To create or modify a firewall configuration from the command line, use the `firewall-cmd` utility (or, if you prefer, the `iptables`, or `ip6tables` utilities) to configure the packet filtering rules.

The packet filtering rules are recorded in the `/etc/firewalld` hierarchy for `firewalld` and in the `/etc/sysconfig/iptables` and `/etc/sysconfig/ip6tables` files for `iptables` and `ip6tables`.

Controlling the firewalld Firewall Service

The `firewalld` service is enabled by default in Oracle Linux 7. You can use the `systemctl` command to start, stop, or restart the service, and to query its status.

Configuring the firewalld Zone

To check the zone for which your system's firewall is configured:

```
sudo firewall-cmd --get-active-zone
```

The command does not display any results if the system has not been assigned to a zone.

Use the following command to display all available zones:

```
sudo firewall-cmd --get-zones

block dmz drop external home internal public trusted work
```

To configure your system for the `work` zone on a local network connected via the `em1` interface:

```
sudo firewall-cmd --zone=work --change-interface=em1

success
```

Querying the current zone now shows that the firewall is configured on the interface `em1` for the `work` zone:

```
sudo firewall-cmd --get-active-zone

work
  interfaces: em1
```

To make the change permanent, you can change the default zone for the system, for example:

```
sudo firewall-cmd --get-default-zone

public

sudo firewall-cmd --set-default-zone=work

success

sudo firewall-cmd --get-default-zone

work
```

Controlling Access to Services

You can permit or deny access to a service by specifying its name. The following command lists the services to which access is allowed on the local system for the `work` zone:

```
sudo firewall-cmd --zone=work --list-services

ssh samba
```

In this example, the system allows access by SSH and Samba clients.

To permit access by NFS and HTTP clients when the `work` zone is active, use the `--add-service` option:

```
sudo firewall-cmd --zone=work --add-service=http --add-service=nfs

success

sudo firewall-cmd --zone=work --list-services

http nfs ssh samba
```

> **Note:**
>
> If you do not specify the zone, the change is applied to the default zone, not the currently active zone.

To make rule changes persist across reboots, run the command again, additionally specifying the `--permanent` option:

```
sudo firewall-cmd --permanent --zone=work --add-service=http --add-service=nfs

success
```

To remove access to a service, use the `--remove-service` option, for example:

```
sudo firewall-cmd --zone=work --remove-service=samba

success

sudo firewall-cmd --zone=work --list-services

http nfs ssh
```

Controlling Access to Ports

You can permit or deny access to a port by specifying the port number and the associated protocol. The `--list-port` option lists the ports and associated protocols to which you have explicitly allowed access, for example:

```
sudo firewall-cmd --zone=work --list-ports

3689/tcp
```

You can use the `--add-port` option to permit access:

```
sudo firewall-cmd --zone=work --add-port=5353/udp

success

sudo firewall-cmd --zone=work --list-ports

5353/udp 3689/tcp
```

Similarly, the `--remove-port` option removes access to a port. Remember to re-run the command with the `--permanant` option if you want to make the change persist.

To display all the firewall rules that are defined for a zone, use the `--list-all` option:

```
sudo firewall-cmd --zone=work --list-all

work (default,active)
  interfaces: em1
  sources:
  services: http nfs ssh
  ports: 5353/udp 3689/tcp
  masquerade: no
  forward-ports:
```

```
  icmp-blocks:
  rich rules:
```

For more information, see the `firewall-cmd(1)` manual page.

Controlling the iptables Firewall Service

If you want to use `iptables` instead of `firewalld`, first stop and disable the `firewalld` service before starting the `iptables` firewall service and enabling it to start when the system boots:

```
sudo systemctl stop firewalld
sudo systemctl disable firewalld
sudo systemctl start iptables
sudo systemctl enable iptables
```

To save any changes that you have made to the firewall rules to `/etc/sysconfig/iptables`, so that the service loads them when it next starts:

```
sudo /sbin/iptables-save > /etc/sysconfig/iptables
```

To restart the service so that it re-reads its rules from `/etc/sysconfig/iptables`:

```
sudo systemctl restart iptables
```

To stop the service:

```
sudo systemctl stop iptables
```

To control IPv6 filtering, use `ip6tables` instead of `iptables`.

For more information, see the `iptables(8)`, and `ip6tables(8)` manual pages.

About netfilter Tables Used by iptables and ip6tables

The `netfilter` tables used by `iptables` and `ip6tables` include the following:

`Filter`
The default table, which is mainly used to drop or accept packets based on their content.

`Mangle`
This table is used to alter certain fields in a packet.

`NAT`
The Network Address Translation table is used to route packets that create new connections.

The kernel uses the rules stored in these tables to make decisions about network packet filtering. Each rule consists of one or more criteria and a single action. If a criterion in a rule matches the information in a network packet header, the kernel applies the action to the packet. Examples of actions include:

`ACCEPT`
Continue processing the packet.

`DROP`
End the packet’s life without notice.

REJECT
As DROP, and additionally notify the sending system that the packet was blocked.

Rules are stored in chains, where each chain is composed of a default policy plus zero or more rules. The kernel applies each rule in a chain to a packet until a match is found. If there is no matching rule, the kernel applies the chain's default action (policy) to the packet.

Each netfilter table has several predefined chains. The filter table contains the following chains:

FORWARD
Packets that are not addressed to the local system pass through this chain.

INPUT
Inbound packets to the local system pass through this chain.

OUTPUT
Locally created packets pass through this chain.

The chains are permanent and you cannot delete them. However, you can create additional chains in the filter table.

Listing Firewall Rules

Use the iptables -L command to list firewall rules for the chains of the filter table. The following example shows the default rules for a newly installed system:

```
iptables -L

Chain INPUT (policy ACCEPT)
target     prot opt source       destination
ACCEPT     all  --  anywhere     anywhere        state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere     anywhere
ACCEPT     all  --  anywhere     anywhere
ACCEPT     tcp  --  anywhere     anywhere        state NEW tcp dpt:ssh
ACCEPT     udp  --  anywhere     anywhere        state NEW udp dpt:ipp
ACCEPT     udp  --  anywhere     224.0.0.251     state NEW udp dpt:mdns
ACCEPT     tcp  --  anywhere     anywhere        state NEW tcp dpt:ipp
ACCEPT     udp  --  anywhere     anywhere        state NEW udp dpt:ipp
REJECT     all  --  anywhere     anywhere        reject-with icmp-host-
prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source       destination
REJECT     all  --  anywhere     anywhere        reject-with icmp-host-
prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source       destination
```

In this example, the default policy for each chain is ACCEPT. A more secure system could have a default policy of DROP, and the additional rules would only allow specific packets on a case-by-case basis.

If you want to modify the chains, specify the --line-numbers option to see how the rules are numbered.

```
iptables -L --line-numbers
```

```
Chain INPUT (policy ACCEPT)
num  target     prot opt source           destination
1    ACCEPT     all  --  anywhere         anywhere         state RELATED,ESTABLISHED
2    ACCEPT     icmp --  anywhere         anywhere
3    ACCEPT     all  --  anywhere         anywhere
4    ACCEPT     tcp  --  anywhere         anywhere         state NEW tcp dpt:ssh
5    ACCEPT     udp  --  anywhere         anywhere         state NEW udp dpt:ipp
6    ACCEPT     udp  --  anywhere         224.0.0.251      state NEW udp dpt:mdns
7    ACCEPT     tcp  --  anywhere         anywhere         state NEW tcp dpt:ipp
8    ACCEPT     udp  --  anywhere         anywhere         state NEW udp dpt:ipp
9    REJECT     all  --  anywhere         anywhere         reject-with icmp-host-
prohibited

Chain FORWARD (policy ACCEPT)
num  target     prot opt source           destination
1    REJECT     all  --  anywhere         anywhere         reject-with icmp-host-
prohibited

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source           destination
```

Inserting and Replacing Rules in a Chain

Use the `iptables -I` command to insert a rule in a chain. For example, the following command inserts a rule in the `INPUT` chain to allow access by TCP on port 80:

```
sudo iptables -I INPUT 4 -p tcp -m tcp --dport 80 -j ACCEPT
sudo iptables -L --line-numbers

Chain INPUT (policy ACCEPT)
num  target     prot opt source           destination
1    ACCEPT     all  --  anywhere         anywhere         state RELATED,ESTABLISHED
2    ACCEPT     icmp --  anywhere         anywhere
3    ACCEPT     all  --  anywhere         anywhere
4    ACCEPT     tcp  --  anywhere         anywhere         tcp dpt:http
5    ACCEPT     tcp  --  anywhere         anywhere         state NEW tcp dpt:ssh
6    ACCEPT     udp  --  anywhere         anywhere         state NEW udp dpt:ipp
7    ACCEPT     udp  --  anywhere         224.0.0.251      state NEW udp dpt:mdns
8    ACCEPT     tcp  --  anywhere         anywhere         state NEW tcp dpt:ipp
9    ACCEPT     udp  --  anywhere         anywhere         state NEW udp dpt:ipp
10   REJECT     all  --  anywhere         anywhere         reject-with icmp-host-
prohibited

Chain FORWARD (policy ACCEPT)
num  target     prot opt source           destination
1    REJECT     all  --  anywhere         anywhere         reject-with icmp-host-
prohibited

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source           destination
```

The output from `iptables -L` shows that the new entry has been inserted as rule 4, and the old rules 4 through 9 are pushed down to positions 5 through 10. The TCP destination port of 80 is represented as `http`, which corresponds to the following definition in the `/etc/services` file (the HTTP daemon listens for client requests on port 80):

```
http            80/tcp          www www-http    # WorldWideWeb HTTP
```

To replace the rule in a chain, use the `iptables -R` command. For example, the following command replaces rule 4 in the `INPUT` chain to allow access by TCP on port 443:

```
sudo iptables -I INPUT 4 -p tcp -m tcp --dport 443 -j ACCEPT
sudo iptables -L --line-numbers

Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state 
RELATED,ESTABLISHED
2    ACCEPT     icmp --  anywhere             anywhere
3    ACCEPT     all  --  anywhere             anywhere
4    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:https
...
```

The TCP destination port of 443 is represented as `https`, which corresponds to the following definition in the `/etc/services` file for secure HTTP on port 443:

```
https           443/tcp                         # http protocol over TLS/SSL
```

Deleting Rules in a Chain

Use the `iptables -D` command to delete a rule in a chain. For example, the following command deletes rule 4 from the `INPUT` chain:

```
sudo iptables -D INPUT 4
```

To delete all rules in a chain, enter:

```
sudo iptables -F chain
```

To delete all rules in all chains, enter:

```
sudo iptables -F
```

Saving Rules

To save your changes to the firewall rules so that they are loaded when the `iptables` service next starts, use the following command:

```
sudo /sbin/iptables-save /etc/sysconfig/iptables
```

The command saves the rules to `/etc/sysconfig/iptables`. For IPv6, you can use `/sbin/ip6tables-save > /etc/sysconfig/ip6tables` to save the rules to `/etc/sysconfig/ip6tables`.

Configuring OpenSSH

OpenSSH is suite of network connectivity tools that provides secure communications between systems. OpenSSH provides another layer of protection to your organization by ensuring that network traffic is safe from external threats. For more information, see Oracle® Linux: Connecting to Remote Systems With OpenSSH.

Configuring TCP Wrappers

TCP wrappers provide basic filtering of incoming network traffic. You can allow or deny access from other systems to certain *wrapped* network services running on a Linux server. A wrapped network service is one that has been compiled against the libwrap.a library. You can use the ldd command to determine if a network service has been wrapped as shown in the following example for the sshd daemon:

```
sudo ldd /usr/sbin/sshd | grep libwrap

libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f877de07000)
```

When a remote client attempts to connect to a network service on the system, the wrapper consults the rules in the configuration files /etc/hosts.allow and /etc/hosts.deny files to determine if access is permitted.

The wrapper for a service first reads /etc/hosts.allow from top to bottom. If the daemon and client combination matches an entry in the file, access is allowed. If the wrapper does not find a match in /etc/hosts.allow, it reads /etc/hosts.deny from top to bottom. If the daemon and client combination matches and entry in the file, access is denied. If no rules for the daemon and client combination are found in either file, or if neither file exists, access to the service is allowed.

The wrapper first applies the rules specified in /etc/hosts.allow, so these rules take precedence over the rules specified in /etc/hosts.deny. If a rule defined in /etc/hosts.allow permits access to a service, any rule in /etc/hosts.deny that forbids access to the same service is ignored.

The rules take the following form:

```
daemon_list : client_list [: command] [: deny]
```

In the previous example, *daemon_list* and *client_list* are comma-separated lists of daemons and clients, and the optional *command* is run when a client tries to access a daemon. You can use the keyword ALL to represent all daemons or all clients. Subnets can be represented by using the * wildcard, for example 192.168.2.*. Domains can be represented by prefixing the domain name with a period (.), for example .example.com. The optional deny keyword causes a connection to be denied even for rules specified in the /etc/hosts.allow file.

The following are some sample rules.

Match all clients for scp, sftp, and ssh access (sshd).

```
sshd : ALL
```

Match all clients on the 192.168.2 subnet for FTP access (vsftpd).

```
vsftpd : 192.168.2.*
```

Match all of the clients in the example.com domain to gain access to all wrapped services.

```
ALL : .example.com
```

Match all clients for FTP access, and displays the contents of the banner file /etc/banners/vsftpd. The banner file must have the same name as the daemon.

```
vsftpd : ALL : banners /etc/banners/
```

Match all of the clients on the 200.182.68 subnet for all wrapped services, and logs all such events. The %c and %d tokens are expanded to the names of the client and the daemon.

```
ALL : 200.182.68.* : spawn /usr/bin/echo `date` “Attempt by %c to connect to %d"
>> /var/log/tcpwr.log
```

Match all of the clients for scp, sftp, and ssh access, and log the event as an emerg message, which is displayed on the console.

```
sshd : ALL : severity emerg
```

Match all of the clients in the forbid.com domain for scp, sftp, and ssh access, log the event, and deny access (even if the rule appears in /etc/hosts.allow).

```
sshd : .forbid.com : spawn /usr/bin/echo `date` "sshd access denied for %c"
>>/var/log/sshd.log : deny
```

For more information, see the hosts_access(5) manual page.

Using chroot Jails to Protect the Root (/) Directory

A chroot operation changes the apparent root directory for a running process and its children. It allows you to run a program with a root directory other than /. The program cannot see or access files outside the designated directory tree. Such an artificial root directory is called a *chroot jail*, and its purpose is to limit the directory access of a potential attacker. The chroot jail locks down a given process and any user ID that it is using so that all they see is the directory in which the process is running. To the process, it appears that the directory in which it is running is the root directory.

> **Note:**
>
> The chroot mechanism cannot defend against intentional tampering or low-level access to system devices by privileged users. For example, a chroot root user could create device nodes and mount file systems on them. A program can also break out of a chroot jail if it can gain root privilege and use chroot() to change its current working directory to the real root directory. For this reason, you should ensure that a chroot jail does not contain any setuid or setgid executables that are owned by root.

For a chroot process to be able to start successfully, you must populate the chroot directory with all required program files, configuration files, device nodes, and shared libraries at their expected locations relative to the level of the chroot directory.

Running DNS and FTP Services in a Chroot Jail

If the DNS name service daemon (named) runs in a chroot jail, any hacker that enters your system via a BIND exploit is isolated to the files under the chroot jail directory. Installing the bind-chroot package creates the /var/named/chroot directory, which becomes the chroot jail for all BIND files.

You can configure the vsftpd FTP server to automatically start chroot jails for clients. By default, anonymous users are placed in a chroot jail. However, local users that access an vsftpd FTP server are placed in their home directory. Specify the chroot_local_user=YES option in the /etc/vsftpd/vsftpd.conf file to place local users in a chroot jail based on their home directory.

Creating a Chroot Jail

To create a chroot jail:

1. Create the directory that will become the root directory of the chroot jail, for example:

   ```
   mkdir /home/oracle/jail
   ```

2. Use the ldd command to find out which libraries are required by the command that you intend to run in the chroot jail, for example /usr/bin/bash:

   ```
   ldd /usr/bin/bash

   linux-vdso.so.1 =>  (0x00007fffdedfe000)
   libtinfo.so.5 => /lib64/libtinfo.so.5 (0x0000003877000000)
   libdl.so.2 => /lib64/libdl.so.2 (0x0000003861c00000)
   libc.so.6 => /lib64/libc.so.6 (0x0000003861800000)
   /lib64/ld-linux-x86-64.so.2 (0x0000003861000000)
   ```

   > **Note:**
   >
   > Although the path is displayed as /lib64, the actual path is /usr/lib64 because /lib64 is a symbolic link to /usr/lib64. Similarly, /bin is a symbolic link to /usr/bin. You need to recreate such symbolic links within the chroot jail.

3. Create subdirectories of the chroot jail's root directory that have the same relative paths as the command binary and its required libraries have to the real root directory, for example:

   ```
   mkdir -p /home/oracle/jail/usr/bin
   mkdir -p /home/oracle/jail/usr/lib64
   ```

4. Create the symbolic links that link to the binary and library directories in the same manner as the symbolic links that exists in the real root directory.

   ```
   ln -s /home/oracle/jail/usr/bin /home/oracle/jail/bin
   ln -s /home/oracle/jail/usr/lib64 /home/oracle/jail/lib64
   ```

5. Copy the binary and the shared libraries to the directories under the chroot jail's root directory, for example:

   ```
   cp /usr/bin/bash /home/oracle/jail/usr/bin
   cp /usr/lib64/{libtinfo.so.5,libdl.so.2,libc.so.6,ld-linux-x86-64.so.2} /home/
   oracle/jail/usr/lib64
   ```

Using a Chroot Jail

To run a command in a chroot jail in an existing directory (*chroot_jail*), use the following command:

```
chroot chroot_jail command
```

If you do not specify a command argument, chroot runs the value of the SHELL environment variable or /usr/bin/sh if SHELL is not set.

For example, to run /usr/bin/bash in a chroot jail (having previously set it up as described in Creating a Chroot Jail):

```
chroot /home/oracle/jail
bash-4.2# pwd

/

bash-4.2# ls

bash: ls: command not found

bash-4.2# exit

#
```

You can run built-in shell commands such as `pwd` in this shell, but not other commands unless you have copied their binaries and any required shared libraries to the chroot jail.

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

Configuring and Using Software Management

Oracle Linux provides the `yum` command that you can use to install or upgrade RPM packages. The main benefit of using yum is that it also installs or upgrades any package dependencies. The `yum` command downloads packages from repositories such as those that are available on the Oracle Linux yum server and the Unbreakable Linux Network (ULN), but you can also set up your own repositories on systems that do not have Internet access.

For more information about managing software with the `yum` utility, see Oracle® Linux 7: Managing Software.

The Oracle Linux yum server is a convenient way to install Oracle Linux packages rather than installing them from installation media. You can also subscribe to the Oracle Linux errata mailing list, and obtain bug fixes, security fixes and enhancements. You can access the server at https://yum.oracle.com/.

If you have registered your system with ULN, you can use `yum` with the ULN channels to maintain the software on your system

You can use the RPM package manager to verify the integrity of installed system files. The `rpm -V` *`package`* and `rpm -Vf` *`filename`* commands verify packages and files respectively by comparing them with package metadata in the RPM database. The verify operation compares file size, MD5 sum, permissions, type, owner, and group and displays any discrepancies. To see more verbose information, specify the `–v` option. You can use the `rpm -qa` command to verify the integrity of all the packages that are installed on a system, for example:

```
for i in `rpm -qa`
> do
> rpm -V $i > .tmp || echo -e "\nDiscepancies for package $i" && cat .tmp
> rm -f .tmp
> done

Discepancies for package gdm-2.30.4-33.0.1.el6_2.x86_64
.M....G..    /var/log/gdm
.M.......    /var/run/gdm
missing     /var/run/gdm/greeter
```

```
Discepancies for package libgcj-4.4.6-4.el6.x86_64
..5....T.  c /usr/lib64/security/classpath.security

Discepancies for package sudo-1.7.4p5-12.el6_3.x86_64
S.5....T.  c /etc/sudoers

Discepancies for package libcgroup-0.37-4.el6.x86_64
S.5....T.  c /etc/cgconfig.conf

Discepancies for package yum-3.2.29-30.0.1.el6.noarch
.......T.  c /etc/yum.conf

Discepancies for package kernel-2.6.32-279.el6.x86_64
.......T.    /etc/ld.so.conf.d/kernel-2.6.32-279.el6.x86_64.conf
...
```

A string of character codes indicates the discrepancies between an installed file and the metadata for that file. The following list describes the meanings of the character codes in the output of the `rpm -V` command.

- `5`: MD5 sum
- `D`: Device major or minor number.
- `G`: Group ownership.
- `L`: Symbolic link path.
- `M`: Mode including permissions or file type.
- `P`: Capabilities.
- `S`: File size.
- `T`: Modification time.
- `U`: User ownership.
- `.`: None (test passed).
- `?`: Unknown (test could not be performed).

If displayed, a single character code preceding the affected file denotes the file type, and can take the values that are shown in the following list.

- `c`: Configuration file.
- `d`: Documentation file.
- `g`: Ghost file, whose file contents are not included in the package payload.
- `l`: License file.
- `r`: Readme file

Most discrepancies are caused by editing the configuration files of subsystems. To see which files change over time, create a baseline file of discrepancies immediately after installation, and `diff` this file against the results found by `rpm -V` at a later date.

You can also use a file integrity checker to test whether a system has been compromised. There are several available open source and commercial file integrity checking tools, including AIDE (Advanced Intrusion Detection Environment) and Tripwire. AIDE and Tripwire are intrusion detection systems that scan file systems and record cryptographic hashes of each file in a database. After creating the database, you should then move it to a read-only medium to avoid tampering. On subsequent file system checks, the tool alerts you if the stored checksums do not match those for the current files. For more information, see the AIDE or Tripwire websites.

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

Configuring Update and Patch Management

Effective security practice relies on keeping system software up to date. It is therefore essential to apply system security updates as soon as they are published. It is strongly recommended that you register every IT system with an update management infrastructure. For Oracle Linux systems, the Unbreakable Linux Network (ULN) tracks system software release levels, and advises you as soon as critical updates become available. Updates and errata are also available at no charge from the Oracle Linux yum server.

Updating the kernel or core system libraries typically requires a system reboot. In mission-critical enterprise and cloud environments, crucial updates might not get installed until you reboot the systems during a scheduled maintenance window. As a result, systems that support critical business applications could be running while they are not protected from known vulnerabilities. To tackle this problem, Oracle Linux Premier Support includes access to Oracle Ksplice, an innovative technology that enables you to apply security updates, patches, and critical bug fixes to the running kernel without requiring a reboot. Ksplice improves the security, reliability, and availability of Oracle Linux systems by enabling zero downtime updates, helping to keep systems up to date without downtime or service disruption.

For more information about Ksplice, see https://oss.oracle.com/ksplice/docs/ksplice-quickstart.pdf.

Installing and Using the Yum Security Plugin

The `yum-plugin-security` package enables you to use the `yum` command to obtain a list of all of the errata that are available for your system, including security updates. You can also use Oracle Enterprise Manager 12c Cloud Control or management tools such as Katello, Pulp, Red Hat Satellite, Spacewalk, and SUSE Manager to extract and display information about errata.

To install the `yum-plugin-security` package, enter the following command:

```
sudo yum install yum-plugin-security
```

To list the errata that are available for your system, enter:

```
sudo yum updateinfo list

Loaded plugins: refresh-packagekit, rhnplugin, security
ELBA-2012-1518 bugfix         NetworkManager-1:0.8.1-34.el6_3.x86_64
ELBA-2012-1518 bugfix         NetworkManager-glib-1:0.8.1-34.el6_3.x86_64
ELBA-2012-1518 bugfix         NetworkManager-gnome-1:0.8.1-34.el6_3.x86_64
ELBA-2012-1457 bugfix         ORBit2-2.14.17-3.2.el6_3.x86_64
ELBA-2012-1457 bugfix         ORBit2-devel-2.14.17-3.2.el6_3.x86_64
ELSA-2013-0215 Important/Sec. abrt-2.0.8-6.0.1.el6_3.2.x86_64
ELSA-2013-0215 Important/Sec. abrt-addon-ccpp-2.0.8-6.0.1.el6_3.2.x86_64
ELSA-2013-0215 Important/Sec. abrt-addon-kerneloops-2.0.8-6.0.1.el6_3.2.x86_64
ELSA-2013-0215 Important/Sec. abrt-addon-python-2.0.8-6.0.1.el6_3.2.x86_64
ELSA-2013-0215 Important/Sec. abrt-cli-2.0.8-6.0.1.el6_3.2.x86_64
```

```
ELSA-2013-0215 Important/Sec. abrt-desktop-2.0.8-6.0.1.el6_3.2.x86_64
...
```

The output from the command sorts the available errata in order of their IDs, and it also specifies whether each erratum is a security patch (*severity* /Sec.), a bug fix (bugfix), or a feature enhancement (enhancement). Security patches are listed by their severity: Important, Moderate, or Low.

You can use the --sec-severity option to filter the security errata by severity, for example:

```
sudo yum updateinfo list --sec-severity=Moderate

Loaded plugins: refresh-packagekit, rhnplugin, security
ELSA-2013-0269 Moderate/Sec. axis-1.2.1-7.3.el6_3.noarch
ELSA-2013-0668 Moderate/Sec. boost-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-date-time-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-devel-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-filesystem-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-graph-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-iostreams-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-program-options-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-python-1.41.0-15.el6_4.x86_64
...
```

To list the security errata by their Common Vulnerabilities and Exposures (CVE) IDs instead of their errata IDs, specify the keyword cves as an argument:

```
sudo yum updateinfo list cves

Loaded plugins: refresh-packagekit, rhnplugin, security
 CVE-2012-5659 Important/Sec. abrt-2.0.8-6.0.1.el6_3.2.x86_64
 CVE-2012-5660 Important/Sec. abrt-2.0.8-6.0.1.el6_3.2.x86_64
 CVE-2012-5659 Important/Sec. abrt-addon-ccpp-2.0.8-6.0.1.el6_3.2.x86_64
 CVE-2012-5660 Important/Sec. abrt-addon-ccpp-2.0.8-6.0.1.el6_3.2.x86_64
 CVE-2012-5659 Important/Sec. abrt-addon-kerneloops-2.0.8-6.0.1.el6_3.2.x86_64
 CVE-2012-5660 Important/Sec. abrt-addon-kerneloops-2.0.8-6.0.1.el6_3.2.x86_64
 CVE-2012-5659 Important/Sec. abrt-addon-python-2.0.8-6.0.1.el6_3.2.x86_64
 CVE-2012-5660 Important/Sec. abrt-addon-python-2.0.8-6.0.1.el6_3.2.x86_64
...
```

Similarly, the keywords bugfix, enhancement, and security filter the list for all bug fixes, enhancements, and security errata.

You can use the --cve option to display the errata that correspond to a specified CVE, for example:

```
sudo yum updateinfo list --cve CVE-2012-2677

Loaded plugins: refresh-packagekit, rhnplugin, security
ELSA-2013-0668 Moderate/Sec. boost-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-date-time-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-devel-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-filesystem-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-graph-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-iostreams-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-program-options-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-python-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-regex-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-serialization-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-signals-1.41.0-15.el6_4.x86_64
```

```
ELSA-2013-0668 Moderate/Sec. boost-system-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-test-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-thread-1.41.0-15.el6_4.x86_64
ELSA-2013-0668 Moderate/Sec. boost-wave-1.41.0-15.el6_4.x86_64
updateinfo list done
```

To display more information, specify `info` instead of `list`, for example:

```
sudo yum updateinfo info --cve CVE-2012-2677

Loaded plugins: refresh-packagekit, rhnplugin, security
===============================================================================
   boost security update
===============================================================================
  Update ID : ELSA-2013-0668
    Release : Oracle Linux 6
       Type : security
     Status : final
     Issued : 2013-03-21
       CVEs : CVE-2012-2677
Description : [1.41.0-15]
            : - Add in explicit dependences between some boost
            :   subpackages
            :
            : [1.41.0-14]
            : - Build with -fno-strict-aliasing
            :
            : [1.41.0-13]
            : - In Boost.Pool, be careful not to overflow
            :   allocated chunk size (boost-1.41.0-pool.patch)
            :
            : [1.41.0-12]
            : - Add an upstream patch that fixes computation of
            :   CRC in zlib streams.
            : - Resolves: #707624
   Severity : Moderate
updateinfo info done
```

To update all packages for which security-related errata are available to the latest versions of the packages, even if those packages include bug fixes or new features but not security errata, enter:

```
sudo yum --security update
```

To update all packages to the latest versions that contain security errata, ignoring any newer packages that do not contain security errata, enter:

```
sudo yum --security update-minimal
```

To update all kernel packages to the latest versions that contain security errata, enter:

```
sudo yum --security update-minimal kernel*
```

You can also update only those packages that correspond to a CVE or erratum, for example:

```
yum update --cve CVE-2012-3954
sudo yum update --advisory ELSA-2012-1141
```

Note:

Some updates might require you to reboot the system. By default, the boot manager will automatically enable the most recent kernel version.

For more information, see the `yum-security(8)` manual page.

Configuring and Using Data Encryption

You can use data encryption to protect data that is stored or that is being transmitted. Data on storage devices and media can be at risk of theft or device loss. Data being transmitted over local area networks and the Internet can be intercepted or altered. In addition, data encryption to protect privacy and personal data is increasingly being made a mandatory requirement of corporate security policy and by governmental regulations (for example, HIPAA, GLBA, SOX, and PCI DSS).

Oracle Linux systems provide the following strategies for protecting data:

- When installing systems and application software, only accept RPM packages that have been digitally signed. To ensure that downloaded software packages are signed, set `gpgcheck=1` in the repository configuration file and import the GPG key provided by the software supplier. You can also install RPMs using the Secure Sockets Layer (SSL) protocol, which uses encryption to protect the communications channel.
- To protect against data theft, consider using full-disk encryption, especially on laptops, external hard drives, or removable devices such as USB memory sticks. Oracle Linux supports block device encryption using the `dm-crypt` kernel module and the Linux Unified Key Setup (LUKS) format. The `cryptsetup` administration command is available in the `cryptsetup` package. These technologies encrypt device partitions so that the data is inaccessible when a system is turned off. When the system boots and you supply the appropriate passphrase, the device is decrypted and its data is accessible. For more infomation, see the `cryptsetup(8)` manual page.
- Oracle Linux uses encryption to support Virtual Private Networks (VPN), Secure Shell (`ssh`), and password protection. By default, Oracle Linux uses a strong password hashing algorithm (SHA-512) and stores hashed passwords in the `/etc/shadow` file.
- Oracle Linux takes advantage of hardware-accelerated encryption on Intel CPUs that support the Advanced Encryption Standard New Instructions (AES-NI) instruction set, which speeds up the execution of AES algorithms as well as SHA-1 and RC4 algorithms on x86 and x86_64 architectures.

Configuring and Using Certificate Management

Public Key Infrastructure (PKI) provides the tools and framework to encrypt and validate network connections. It also provides an authentication mechanism in the form of signed certificates. Managing your certificates and implementing strong public key infrastructure is an important part of maintaining good security within your organization. For more information, see Oracle® Linux: Managing Certificates and Public Key Infrastructure

Configuring and Using Authentication

Authentication is a method of verifying the identity of users. The operating system authenticates user names and passwords by comparing this information to data stored on the system. If the login credentials match the data, then access to the system is opened. For more information, see Oracle® Linux 7: Setting Up System Accounts and Authentication.

Configuring and Using Pluggable Authentication Modules

The Pluggable Authentication Modules (PAM) feature is an authentication mechanism for applications to verify user credentials. For more information, see Oracle® Linux 7: Setting Up System Accounts and Authentication.

Configuring and Using 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 descendents inherit the same rights automatically. The kernel provides ACL support for `ext3`, `ext4`, and NFS-exported file systems.

The following are examples of setting and displaying ACLs for directories and files.

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

```
sudo setfacl -m u:user:r file
```

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

```
sudo getfacl file
```

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

```
sudo setfacl -m m::rx file
```

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

```
sudo setfacl -x g:group file
```

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

```
sudo getfacl f1 | setfacl --set-file=- f2
```

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

```
sudo getfacl --access dir | setfacl -d -M- dir
```

For more information about how to manage ACLs, see the `setfacl(1)` and `getfacl(1)` manual pages.

Configuring and Using SELinux

SELinux is a kernel module that enforces and implements access control policies on Oracle Linux systems to protect services and files from malicious or unauthorized access. Use the SELinux user space tools to manage policies and to resolve access issues. For more information, see Oracle® Linux: Administering SELinux for more info

Configuring and Using 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, but most audited events are uninteresting and insignificant. 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`. For more information, see the `auditd.conf(5)` manual page.

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
```

You can find more examples of audit rules in `/usr/share/doc/audit-`*`version`*`/stig.rules`, and in the `auditctl(8)` and `audit.rules(7)` manual pages.

Stringent auditing requirements can impose a significant performance overhead and generate large amounts of audit data. Some site security policies stipulate that a system must shut down if events cannot be recorded because the audit volumes have exceeded their capacity. As a general rule, you should direct audit data to separate file systems in rotation to prevent overspill and to facilitate backups.

You can use the `-k` option to tag audit records so that you can locate them more easily in an audit volume with the `ausearch` command. For example, to examine records tagged with the string `sudoers-change`, you would enter:

```
sudo ausearch -k sudoers-change
```

The `aureport` command generates summaries of audit data. You can set up `cron` jobs that run `aureport` periodically to generate reports of interest. For example, the following command generates a reports that shows every login event from 1 second after midnight on the previous day until the current time:

```
sudo aureport -l -i -ts yesterday -te now
```

For more information, see the `ausearch(8)` and `aureport(8)` manual pages.

Configuring and Using System Logging

The log files contain messages about the system, kernel, services, and applications. The `journald` logging daemon, which is part of `systemd`, records system messages in non-persistent journal files in memory and in the `/run/log/journal` directory. `journald` forwards messages to the system logging daemon, `rsyslog`. As files in `/run` are volatile, the log data is lost after a reboot unless you create the directory `/var/log/journal`. You can use the `journalctl` command to query the journal logs.

For more information, see the `journalctl(1)` and `systemd-journald.service(8)` manual pages.

The configuration file for `rsyslogd` is `/etc/rsyslog.conf`, which contains global directives, module directives, and rules. By default, `rsyslog` processes and archives only `syslog` messages. If required, you can configure `rsyslog` to archive any other messages that `journald` forwards, including kernel, boot, `initrd`, `stdout`, and `stderr` messages.

Global directives specify configuration options that apply to the `rsyslogd` daemon. All configuration directives must start with a dollar sign (`$`) and only one directive can be specified on each line. The following example specifies the maximum size of the `rsyslog` message queue:

```
$MainMsgQueueSize 50000
```

The available configuration directives are described in the file `/usr/share/doc/rsyslog-`*`version-number`*`/rsyslog_conf_global.html`.

The design of rsyslog allows its functionality to be dynamically loaded from modules, which provide configuration directives. To load a module, specify the following directive:

```
$ModLoad MODULE_name
```

Modules have the following main categories:

- Input modules gather messages from various sources. Input module names always start with the im prefix (examples include imfile and imrelp).
- Filter modules allow rsyslogd to filter messages according to specified rules. The name of a filter module always starts with the fm prefix.
- Library modules provide functionality for other loadable modules. rsyslogd loads library modules automatically when required. You cannot configure the loading of library modules.
- Output modules provide the facility to store messages in a database or on other servers in a network, or to encrypt them. Output module names always starts with the om prefix (examples include omsnmp and omrelp).
- Message modification modules change the content of an rsyslog message.
- Parser modules allow rsyslogd to parse the message content of messages that it receives. The name of a parser module always starts with the pm prefix.
- String generator modules generate strings based on the content of messages in cooperation with rsyslog's template feature. The name of a string generator module always starts with the sm prefix.

Input modules receive messages, which pass them to one or more parser modules. A parser module creates a representation of a message in memory, possibly modifying the message, and passes the internal representation to output modules, which can also modify the content before outputting the message.

A description of the available modules can be found in RSyslog documentation at https://www.rsyslog.com/doc/.

An rsyslog rule consists of a filter part, which selects a subset of messages, and an action part, which specifies what to do with the selected messages. To define a rule in the /etc/rsyslog.conf configuration file, specify a filter and an action on a single line, separated by one or more tabs or spaces.

You can configure rsyslog to filter messages according to various properties. The most commonly used filters are:

- Expression-based filters, written in the rsyslog scripting language, select messages according to arithmetic, boolean, or string values.
- Facility/priority-based filters filter messages based on facility and priority values that take the form *facility.priority*.
- Property-based filters filter messages by properties such as timegenerated or syslogtag.

The following list identifies the available facility keywords for facility/priority-based filters:

- auth, authpriv: Security, authentication, or authorization messages.
- cron: crond messages.
- daemon: Messages from system daemons other than crond and rsyslogd.

- kern: Kernel messages.
- lpr: Line printer subsystem.
- mail: Mail system.
- news: Network news subsystem.
- syslog: Messages generated internally by rsyslogd.
- user: User-level messages.
- UUCP: UUCP subsystem.
- local0 - local7: Local use.

The following list identifies the available priority keywords for facility/priority-based filters, in ascending order of importance:

- debug: Debug-level messages.
- info: Informational messages.
- notice: Normal but significant condition.
- warning: Warning conditions.
- err: Error conditions.
- crit: Critical conditions.
- alert: Immediate action required.
- emerg: System is unstable.

All messages of the specified priority and higher are logged according to the specified action. An asterisk (*) wildcard specifies all facilities or priorities. Separate the names of multiple facilities and priorities on a line with commas (,). Separate multiple filters on one line with semicolons (;). Precede a priority with an exclamation mark (!) to select all messages except those with that priority.

The following are examples of facility/priority-based filters.

Select all kernel messages with any priority.

```
kern.*
```

Select all mail messages with crit or higher priority.

```
mail.crit
```

Select all daemon and kern messages with warning or err priority.

```
daemon,kern.warning,err
```

Select all cron messages except those with info or debug priority.

```
cron.!info,!debug
```

By default, /etc/rsyslog.conf includes the following rules:

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog

# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

You can send the logs to a central log server over TCP by adding the following entry to the `forwarding rules` section of `/etc/rsyslog.conf` on each log client:

```
*.*        @@logsvr:port
```

In the previous example, *logsvr* is the domain name or IP address of the log server and port is the port number (usually, 514).

On the log server, add the following entry to the `MODULES` section of `/etc/rsyslog.conf`:

```
$ModLoad imtcp
$InputTCPServerRun port
```

In the previous example, *port* corresponds to the port number that you set on the log clients.

To manage the rotation and archival of the correct logs, edit `/etc/logrotate.d/syslog` so that it references each of the log files that are defined in the `RULES` section of `/etc/rsyslog.conf`. You can configure how often the logs are rotated and how many past copies of the logs are archived by editing `/etc/logrotate.conf`.

It is recommended that you configure Logwatch on your log server to monitor the logs for suspicious messages, and disable Logwatch on log clients. However, if you do use Logwatch, disable high precision timestamps by adding the following entry to the GLOBAL DIRECTIVES section of `/etc/rsyslog.conf` on each system:

```
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
```

For more information, see the `logrotate(8)`, `logwatch(8)`, `rsyslogd(8)` and `rsyslog.conf(5)` manual pages, the HTML documentation in the `/usr/share/doc/rsyslog-5.8.10` directory, and the documentation at https://www.rsyslog.com/doc/.

Configuring Logwatch

Logwatch is a monitoring system that you can configure to report on areas of interest in the system logs. After you install the `logwatch` package, the `/etc/cron.daily/0logwatch` script runs every night and sends an email report to `root`. You can set local configuration options in `/etc/logwatch/conf/logwatch.conf` that override the main configuration file `/usr/share/logwatch/default.conf/logwatch.conf`, including:

- Log files to monitor, including log files that are stored for other hosts.
- Names of services to monitor, or to be excluded from monitoring.
- Level of detail to report.
- User to be sent an emailed report.

You can also run `logwatch` directly from the command line.

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

Configuring and Using Process Accounting

The `psacct` package implements the process accounting service in addition to the following utilities that you can use to monitor process activities:

`ac`
Displays connection times in hours for a user as recorded in the `wtmp` file (by default, `/var/log/wtmp`).

`accton`
Turns on process accounting to the specified file. If you do not specify a file name argument, process accounting is stopped. The default system accounting file is `/var/account/pacct`.

`lastcomm`
Displays information about previously executed commands as recorded in the system accounting file.

`sa`
Summarizes information about previously executed commands as recorded in the system accounting file.

> **Note:**
>
> As for any logging activity, ensure that the file system has enough space to store the system accounting and `wtmp` files. Monitor the size of the files and, if necessary, truncate them.

For more information, see the `ac(1)`, `accton(8)`, `lastcomm(1)`, and `sa(8)` manual pages.

Configuring and Using Linux Containers

The Linux Containers (LXC) feature provides a way to isolate a group of processes from other processes that are running on an Oracle Linux system. LXC is a lightweight operating system virtualization technology that uses the control group (cgroup) feature to provide resource management and namespace isolation in a similar manner to `chroot`. Within a container, processes can have their own private view of the operating system with its own process ID space, file system structure, and network interfaces.

See the following documentation:

- See Oracle® Linux 7: Working With LXC for more information about how to configure and use Linux Containers.
- Oracle Cloud Native Environment documentation at https://docs.oracle.com/en/operating-systems/olcne/.

Configuring and Using Kernel Security Mechanisms

The Linux kernel features some additional security mechanisms that you can use to enhance the security of a system. These mechanisms randomize the layout of a process's address space or prevent code from being executed in non-executable memory.

Address Space Layout Randomization

Address Space Layout Randomization (ASLR) can help defeat certain types of buffer overflow attacks. ASLR can locate the base, libraries, heap, and stack at random positions in a process's address space, which makes it difficult for an attacking program to predict the memory address of the next instruction. ASLR is built into the Linux kernel and is controlled by the parameter `/proc/sys/kernel/randomize_va_space`. The `randomize_va_space` parameter can take the following values:

- **0**: Disable ASLR. This setting is applied if the kernel is booted with the `norandmaps` boot parameter.
- **1**: Randomize the positions of the stack, virtual dynamic shared object (VDSO) page, and shared memory regions. The base address of the data segment is located immediately after the end of the executable code segment.
- **2**: Randomize the positions of the stack, VDSO page, shared memory regions, and the data segment. This is the default setting.

You can change the setting temporarily by writing a new value to `/proc/sys/kernel/randomize_va_space`, for example:

```
echo value > /proc/sys/kernel/randomize_va_space
```

To change the value permanently, add the setting to `/etc/sysctl.conf`, for example:

```
kernel.randomize_va_space = value
```

Then, run the `sysctl -p` command.

If you change the value of `randomize_va_space`, you should test your application stack to ensure that it is compatible with the new setting.

If necessary, you can disable ASLR for a specific program and its child processes by using the following command:

```
setarch `uname -m` -R program [args ...]
```

Data Execution Prevention

The Data Execution Prevention (DEP) feature prevents an application or service from executing code in a non-executable memory region. Hardware-enforced DEP works in conjunction with the NX (Never eXecute) bit on compatible CPUs. Oracle Linux does not emulate the NX bit in software for CPUs that do not implement the NX bit in hardware.

> **Note:**
>
> You cannot disable the DEP feature.

Position Independent Executables

The Position Independent Executables (PIE) feature loads executable binaries at random memory addresses so that the kernel can disallow text relocation. To generate a position-independent binary:

- Specify the `–fpie` option to `gcc` when compiling.
- Specify the `–pie` option to `ld` when linking.

To test whether a binary or library is relocatable, use the following command:

```
sudo readelf -d elfname | grep TEXTREL
```

5

Using OpenSCAP to Scan for Vulnerabilities

This chapter describes how to use OpenSCAP to scan your Oracle Linux system for security vulnerabilities.

About SCAP

The Security Content Automation Protocol (SCAP) provides an automated, standardized methodology for managing system security, including measuring and managing system vulnerability, and evaluating policy compliance against security standards such as the Federal Information Security Management Act (FISMA). The U.S. government content repository for SCAP standards is the National Vulnerability Database (NVD), which is managed by the National Institute of Standards and Technology (NIST).

Oracle Linux provides the following SCAP packages for Oracle Linux 7:

`openscap-utils`
The `openscap-utils` package contains command-line tools that use the OpenSCAP library. This package previously included the `oscap` command-line configuration and vulnerability scanner, but this is now made available separately in the `openscap-scanner` package. The `openscap-scanner` package is installed as a dependency when you install the `openscap-utils` package.

`openscap-scanner`
Provides the `oscap` command-line configuration and vulnerability scanner, which can perform compliance checking against SCAP content including the SCAP Security Guide. This is a dependency of the `openscap-utils` package. The package also includes the `oscap-chroot` utility that allows you to scan an offline file system within a chroot environment.

`openscap-containers`
The `openscap-containers` package provides the `oscap-docker` utility that can be used to scan containers and container images. Note that some dependencies for this package are included in the `ol7_addons` yum repository.

`openscap`
Provides the OpenSCAP open-source libraries for generating SCAP-compliance documentation. OpenSCAP received SCAP 1.2 certification from NIST in April 2014.

`scap-security-guide`
Provides system-hardening guidance in SCAP format, including links to government requirements. The guide provides security profiles that you can modify to comply with the security policies that you have established for your site. Starting from version v0.1.46-11.0.2.el7, a stig profile is included in this package to align with the DISA STIG for Oracle Linux 7 V1R1 published at https://public.cyber.mil/stigs/. See Displaying Available Profiles.

Installing the SCAP Packages

Use the `yum` command to install the SCAP packages from the `ol7_<arch>_latest` channel on ULN or the `ol7_latest` repository on the Oracle Linux yum server:

```
sudo yum install scap-security-guide
```

Note that if you intend to install the `openscap-containers` package to scan containers and container images, you must also enable the `ol7_<arch>_addons` channel on ULN or the `ol7_addons` repository on the Oracle Linux yum server.

About the oscap Command

The `oscap` command has the following general syntax:

```
oscap [options] module operation [operation_options_and_arguments]
```

For *module*, `oscap` supports the following module types:

`cpe`
Performs operations using a Common Platform Enumeration (CPE) file.

`cve`
Performs operations using a Common Vulnerabilities and Exposures (CVE) file.

`cvss`
Performs operations using a Common Vulnerability Scoring System (CVSS) file.

`ds`
Performs operations using a SCAP Data Stream (DS).

`info`
Determines a file's type and prints information about the file.

`oval`
Performs operations using an Open Vulnerability and Assessment Language (OVAL) file.

`xccdf`
Performs operations using a file in eXtensible Configuration Checklist Description Format (XCCDF).

The `info`, `oval`, and `xccdf` modules are the most generally useful for scanning Oracle Linux systems.

For *operation*, the value you can specify depends on the module type. The following operations are the most generally useful with the `oval` and `xccdf` modules on Oracle Linux systems:

- **`eval`**

  For an OVAL file, `oscap` probes the system, evaluates each definition in the file, and prints the results to the standard output.

  For a specified profile in an XCCDF file, `oscap` tests the system against each rule in the file and prints the results to the standard output.

- **`generate`**

  For an OVAL XML results file, `generate report` converts the specified file to an HTML report.

  For an XCCDF file, `generate guide` outputs a full security guide for a specified profile.

- **`validate`**

  Validates an OVAL or XCCDF file against an XML schema to check for errors.

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

Displaying the Available SCAP Information

To display the supported SCAP specifications, any loaded plug-in capabilities, the locations of schema, Common Platform Enumeration (CPE), and probe files, inbuilt CPE names, and supported Open Vulnerability and Assessment Language (OVAL) objects and associated SCAP probes, use the `oscap -V` command, for example:

```
oscap -V

OpenSCAP command line tool (oscap) 1.2.10
Copyright 2009--2016 Red Hat Inc., Durham, North Carolina.

==== Supported specifications ====
XCCDF Version: 1.2
OVAL Version: 5.11.1
CPE Version: 2.3
CVSS Version: 2.0
CVE Version: 2.0
Asset Identification Version: 1.1
Asset Reporting Format Version: 1.1

==== Capabilities added by auto-loaded plugins ====
No plugins have been auto-loaded...

==== Paths ====
Schema files: /usr/share/openscap/schemas
Default CPE files: /usr/share/openscap/cpe
Probes: /usr/libexec/openscap

==== Inbuilt CPE names ====
Red Hat Enterprise Linux - cpe:/o:redhat:enterprise_linux
Red Hat Enterprise Linux 5 - cpe:/o:redhat:enterprise_linux:5
Red Hat Enterprise Linux 6 - cpe:/o:redhat:enterprise_linux:6
Red Hat Enterprise Linux 7 - cpe:/o:redhat:enterprise_linux:7
Community Enterprise Operating System 5 - cpe:/o:centos:centos:5
Community Enterprise Operating System 6 - cpe:/o:centos:centos:6
Community Enterprise Operating System 7 - cpe:/o:centos:centos:7
Scientific Linux 5 - cpe:/o:scientificlinux:scientificlinux:5
Scientific Linux 6 - cpe:/o:scientificlinux:scientificlinux:6
Scientific Linux 7 - cpe:/o:scientificlinux:scientificlinux:7
Fedora 16 - cpe:/o:fedoraproject:fedora:16
Fedora 17 - cpe:/o:fedoraproject:fedora:17
Fedora 18 - cpe:/o:fedoraproject:fedora:18
Fedora 19 - cpe:/o:fedoraproject:fedora:19
Fedora 20 - cpe:/o:fedoraproject:fedora:20
Fedora 21 - cpe:/o:fedoraproject:fedora:21
Fedora 22 - cpe:/o:fedoraproject:fedora:22
Fedora 23 - cpe:/o:fedoraproject:fedora:23
Fedora 24 - cpe:/o:fedoraproject:fedora:24
Fedora 25 - cpe:/o:fedoraproject:fedora:25
SUSE Linux Enterprise all versions - cpe:/o:suse:sle
SUSE Linux Enterprise Server 10 - cpe:/o:suse:sles:10
SUSE Linux Enterprise Desktop 10 - cpe:/o:suse:sled:10
SUSE Linux Enterprise Server 11 - cpe:/o:suse:sles:11
SUSE Linux Enterprise Desktop 11 - cpe:/o:suse:sled:11
SUSE Linux Enterprise Server 12 - cpe:/o:suse:sles:12
SUSE Linux Enterprise Desktop 12 - cpe:/o:suse:sled:12
openSUSE 11.4 - cpe:/o:opensuse:opensuse:11.4
openSUSE 13.1 - cpe:/o:opensuse:opensuse:13.1
openSUSE 13.2 - cpe:/o:opensuse:opensuse:13.2
openSUSE 42.1 - cpe:/o:novell:leap:42.1
openSUSE All Versions - cpe:/o:opensuse:opensuse
Red Hat Enterprise Linux Optional Productivity Applications - cpe:/
a:redhat:rhel_productivity
Red Hat Enterprise Linux Optional Productivity Applications 5 - cpe:/
a:redhat:rhel_productivity:5
Oracle Linux 5 - cpe:/o:oracle:linux:5
Oracle Linux 6 - cpe:/o:oracle:linux:6
Oracle Linux 7 - cpe:/o:oracle:linux:7

==== Supported OVAL objects and associated OpenSCAP probes ====
system_info                  probe_system_info
family                       probe_family
filehash                     probe_filehash
environmentvariable          probe_environmentvariable
textfilecontent54            probe_textfilecontent54
textfilecontent              probe_textfilecontent
variable                     probe_variable
xmlfilecontent               probe_xmlfilecontent
environmentvariable58        probe_environmentvariable58
filehash58                   probe_filehash58
inetlisteningservers         probe_inetlisteningservers
rpminfo                      probe_rpminfo
partition                    probe_partition
iflisteners                  probe_iflisteners
rpmverify                    probe_rpmverify
rpmverifyfile                probe_rpmverifyfile
rpmverifypackage             probe_rpmverifypackage
selinuxboolean               probe_selinuxboolean
selinuxsecuritycontext       probe_selinuxsecuritycontext
systemdunitproperty          probe_systemdunitproperty
systemdunitdependency        probe_systemdunitdependency
file                         probe_file
interface                    probe_interface
password                     probe_password
process                      probe_process
runlevel                     probe_runlevel
shadow                       probe_shadow
uname                        probe_uname
xinetd                       probe_xinetd
sysctl                       probe_sysctl
process58                    probe_process58
fileextendedattribute        probe_fileextendedattribute
routingtable                 probe_routingtable
symlink                      probe_symlink
```

Displaying Information About a SCAP File

To display information about a SCAP file, use the `oscap info` command, for example:

```
oscap info com.oracle.elsa-2017.xml

Document type: OVAL Definitions
OVAL version: 5.3
Generated: 2017-06-01T00:00:00
Imported: 2017-06-13T23:12:06
```

This output shows that the file `com.oracle.elsa-2017.xml` is an OVAL definitions file.

Displaying Available Profiles

You can use the `oscap info` command to display the profiles that are supported by a checklist file such as the SCAP Security Guide, for example:

```
oscap info "/usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml"

Document type: XCCDF Checklist
Checklist version: 1.1
Imported: 2020-04-21T19:46:55
Status: draft
Generated: 2020-04-21
Resolved: true
Profiles:
    Title: DISA STIG for Oracle Linux 7
        Id: stig
Referenced check files:
    ssg-ol7-oval.xml
        system: http://oval.mitre.org/XMLSchema/oval-definitions-5
    ssg-ol7-ocil.xml
        system: http://scap.nist.gov/schema/ocil/2
    https://linux.oracle.com/security/oval/com.oracle.elsa-all.xml.bz2
        system: http://oval.mitre.org/XMLSchema/oval-definitions-5
[vagrant@localhost ~]$
```

> **Note:**
>
> Other profiles are available and located in a different set of files, such as `ssg-rhel7-*` files. For example, to view other profiles, replace `ssg-ol7-xccdf.xml` in the command with `ssg-rhel7-xccdf.xml`.

This output shows that `ssg-ol7-xccdf.xml` provides the DISA STIG for Oracle Linux 7 (`stig`). A profile contains generic security recommendations that apply to all Oracle Linux installations and additional security recommendations that are specific to the intended usage of a system.

To obtain information about a specific profile, specify the `--profile` option.

```
oscap info --profile stig /usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml
```

```
Document type: XCCDF Checklist
Profile
        Title: DISA STIG for Oracle Linux 7
        Id: stig

        Description: This profile contains configuration checks that align to
the DISA STIG for Oracle Linux V1R1.
```

This DISA STIG profile can be used to check compliance with the published Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG) for Oracle Linux. For more information, see https://public.cyber.mil/stigs/.

> **Note:**
>
> Starting from version v0.1.46-11.0.2.el7 the DISA STIG profile and it's checklist definition is deprecated in the `ssg-rhel7-xccdf.xml` and `ssg-rhel7-ds.xml` files . The `scap-security-guide` package now provides a profile aligned to DISA STIG for Oracle Linux V1R1 and the profile is available in the `ssg-ol7-xccdf.xml` and `ssg-ol7-ds.xml` files.

Note that the provided profiles in Oracle Linux 7 might not all be appropriate to your system. However, you can use them to create new profiles that test compliance with your site's security policies.

Validating OVAL and XCCDF Files

To validate an OVAL or XCCDF file against its schema, use the `oscap validate` command and examine the exit code, for example:

```
oscap oval validate com.oracle.elsa-2017.xml && echo "ok" || echo "exit code
= $? not ok"

ok

oscap xccdf validate /usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml && echo
"ok" || echo "exit code = $? not ok"

ok
```

An exit code of 0 indicates that the file is valid, 1 indicates an error prevented validation, and 2 indicates that the file is invalid. Error messages are written to the standard error output.

Running a Scan Against a Profile

To scan a system against an XCCDF profile, use the `oscap xccdf eval` command, for example:

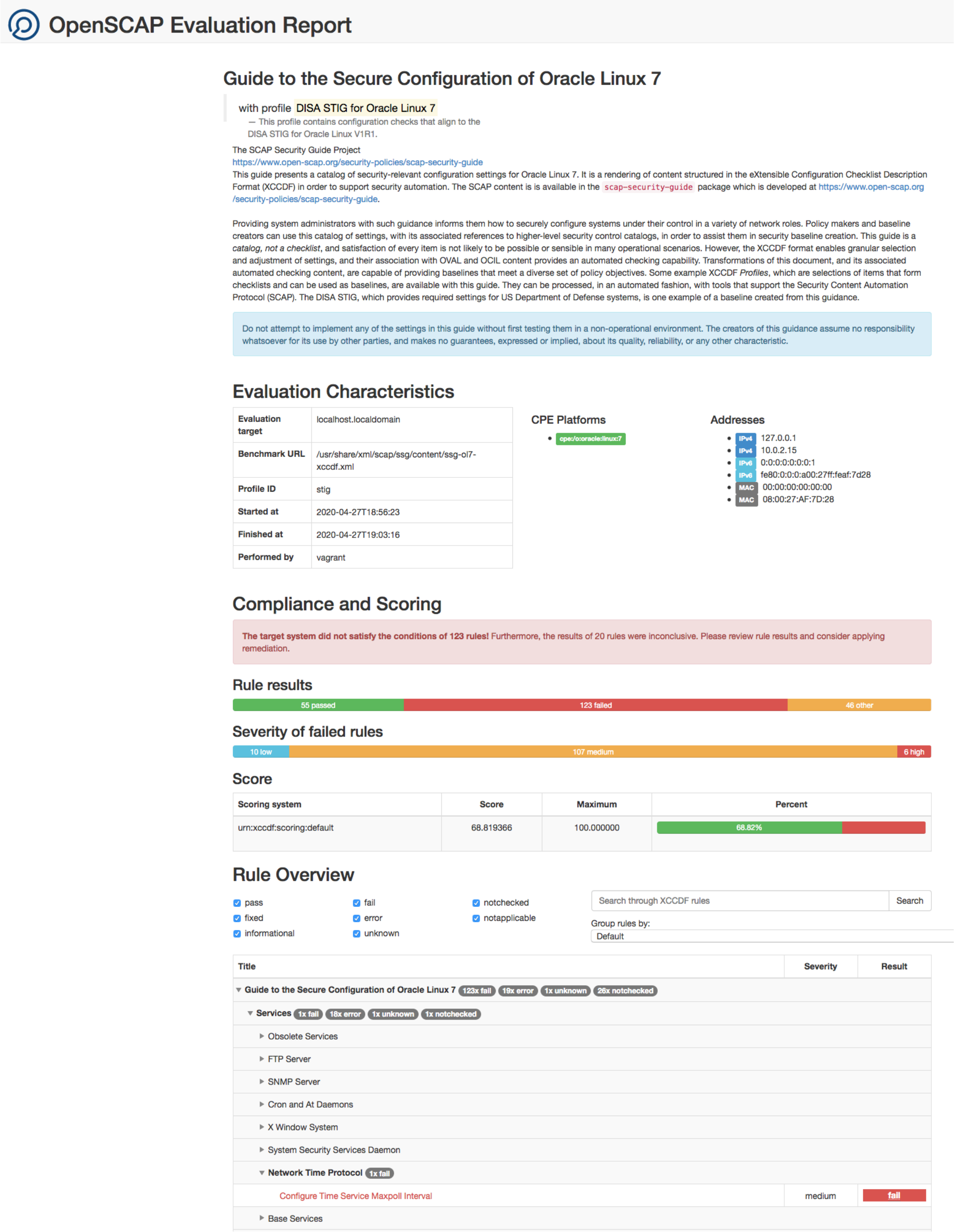
```
oscap xccdf eval --profile stig  \
--results /tmp/`hostname`-ssg-results.xml \
--report /var/www/html/`hostname`-ssg-results.html \
--cpe /usr/share/xml/scap/ssg/content/ssg-ol7-cpe-dictionary.xml \
/usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml
```

```
WARNING: This content points out to the remote resources.
   Use `--fetch-remote-resources' option to download them.
WARNING: Skipping https://linux.oracle.com/security/oval/com.oracle.elsa-all.xml.bz2
file
   which is referenced from XCCDF content
Title   Remove User Host-Based Authentication Files
Rule    no_user_host_based_files
Result  pass

Title   Remove Host-Based Authentication Files
Rule    no_host_based_files
Result  pass

Title   Uninstall rsh-server Package
Rule    package_rsh-server_removed
Result  pass

Title   Uninstall telnet-server Package
Rule    package_telnet-server_removed
Result  pass

...
```

This example scan performs the scan against the stig profile of the ssg-ol7-xccdf.xml checklist using the ssg-ol7-cpe-dictionary.xml CPE dictionary, and outputs the XML results and HTML report files to /tmp and /var/www/html respectively. Any rule in a profile that results in a fail potentially requires the system to be reconfigured.

You can view the HTML report in a browser as shown in Figure 5-1.

Figure 5-1 Sample Scan Report



OpenSCAP Evaluation Report

Guide to the Secure Configuration of Oracle Linux 7

> with profile DISA STIG for Oracle Linux 7
> — This profile contains configuration checks that align to the DISA STIG for Oracle Linux V1R1.

The SCAP Security Guide Project
https://www.open-scap.org/security-policies/scap-security-guide
This guide presents a catalog of security-relevant configuration settings for Oracle Linux 7. It is a rendering of content structured in the eXtensible Configuration Checklist Description Format (XCCDF) in order to support security automation. The SCAP content is is available in the `scap-security-guide` package which is developed at https://www.open-scap.org/security-policies/scap-security-guide.

Providing system administrators with such guidance informs them how to securely configure systems under their control in a variety of network roles. Policy makers and baseline creators can use this catalog of settings, with its associated references to higher-level security control catalogs, in order to assist them in security baseline creation. This guide is a *catalog, not a checklist*, and satisfaction of every item is not likely to be possible or sensible in many operational scenarios. However, the XCCDF format enables granular selection and adjustment of settings, and their association with OVAL and OCIL content provides an automated checking capability. Transformations of this document, and its associated automated checking content, are capable of providing baselines that meet a diverse set of policy objectives. Some example XCCDF *Profiles*, which are selections of items that form checklists and can be used as baselines, are available with this guide. They can be processed, in an automated fashion, with tools that support the Security Content Automation Protocol (SCAP). The DISA STIG, which provides required settings for US Department of Defense systems, is one example of a baseline created from this guidance.

Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. The creators of this guidance assume no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

Evaluation Characteristics

Evaluation target	localhost.localdomain
Benchmark URL	/usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml
Profile ID	stig
Started at	2020-04-27T18:56:23
Finished at	2020-04-27T19:03:16
Performed by	vagrant

CPE Platforms

- cpe:/o:oracle:linux:7

Addresses

- IPv4 127.0.0.1
- IPv4 10.0.2.15
- IPv6 0:0:0:0:0:0:0:1
- IPv6 fe80:0:0:0:a00:27ff:feaf:7d28
- MAC 00:00:00:00:00:00
- MAC 08:00:27:AF:7D:28

Compliance and Scoring

The target system did not satisfy the conditions of 123 rules! Furthermore, the results of 20 rules were inconclusive. Please review rule results and consider applying remediation.

Rule results

55 passed | 123 failed | 46 other

Severity of failed rules

10 low | 107 medium | 6 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	68.819366	100.000000	68.82%

Rule Overview

☑ pass
☑ fixed
☑ informational
☑ fail
☑ error
☑ unknown
☑ notchecked
☑ notapplicable

Title	Severity	Result
▼ **Guide to the Secure Configuration of Oracle Linux 7** 123x fail 19x error 1x unknown 26x notchecked		
▼ **Services** 1x fail 18x error 1x unknown 1x notchecked		
▶ Obsolete Services		
▶ FTP Server		
▶ SNMP Server		
▶ Cron and At Daemons		
▶ X Window System		
▶ System Security Services Daemon		
▼ **Network Time Protocol** 1x fail		
Configure Time Service Maxpoll Interval	medium	fail
▶ Base Services		

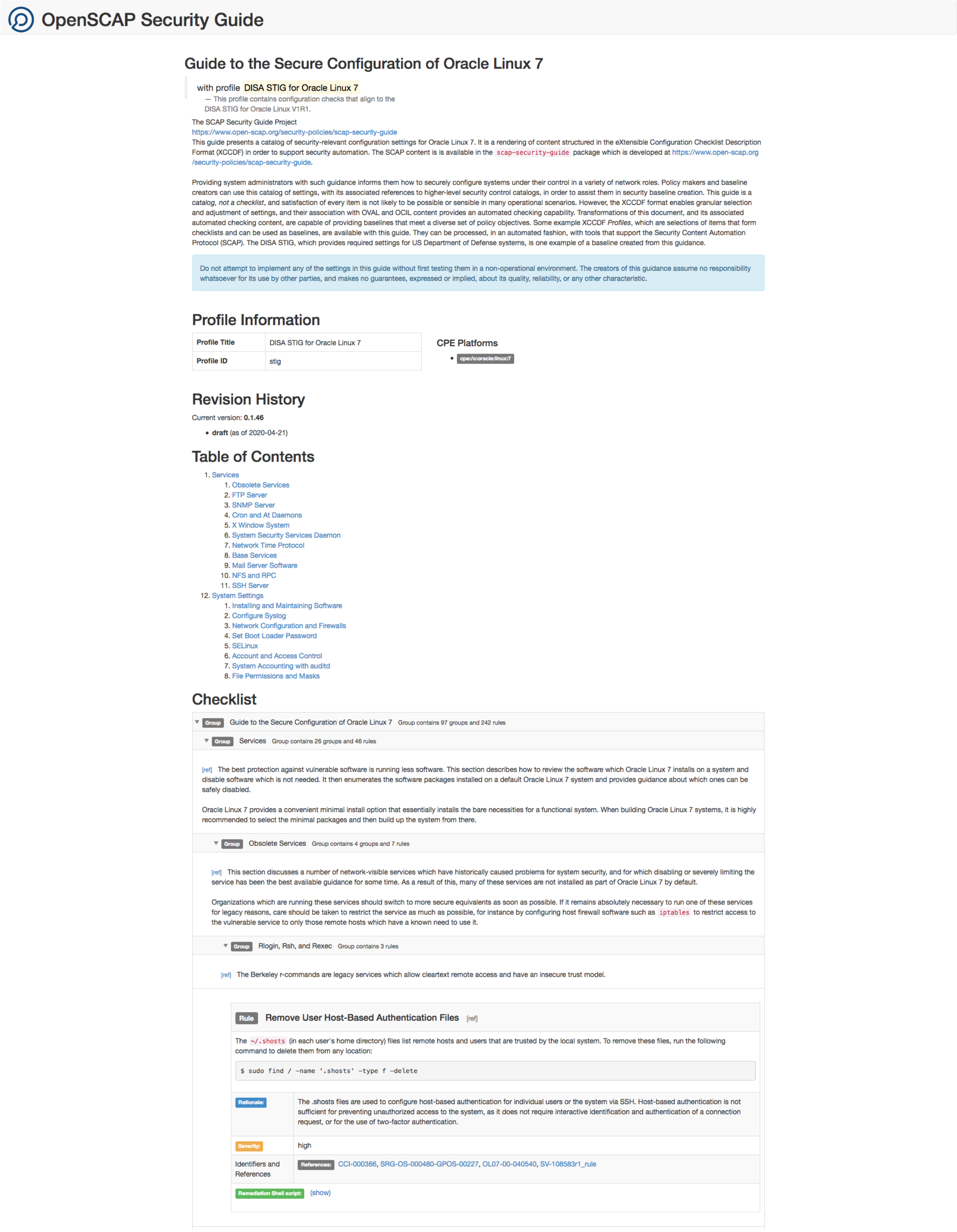
Generating a Full Security Guide

To create a full security guide for a system based on an XCCDF profile, use the `oscap xccdf generate guide` command, for example:

```
oscap xccdf generate guide --profile stig \
--cpe /usr/share/xml/scap/ssg/content/ssg-ol7-cpe-dictionary.xml \
/usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml > /var/www/html/
security_guide.html
```

You can view the security guide in a browser as shown in Figure 5-2.

Figure 5-2 Sample Security Guide



OpenSCAP Security Guide

Guide to the Secure Configuration of Oracle Linux 7

with profile DISA STIG for Oracle Linux 7
— This profile contains configuration checks that align to the DISA STIG for Oracle Linux V1R1.

The SCAP Security Guide Project
https://www.open-scap.org/security-policies/scap-security-guide
This guide presents a catalog of security-relevant configuration settings for Oracle Linux 7. It is a rendering of content structured in the eXtensible Configuration Checklist Description Format (XCCDF) in order to support security automation. The SCAP content is is available in the `scap-security-guide` package which is developed at https://www.open-scap.org/security-policies/scap-security-guide.

Providing system administrators with such guidance informs them how to securely configure systems under their control in a variety of network roles. Policy makers and baseline creators can use this catalog of settings, with its associated references to higher-level security control catalogs, in order to assist them in security baseline creation. This guide is a *catalog, not a checklist*, and satisfaction of every item is not likely to be possible or sensible in many operational scenarios. However, the XCCDF format enables granular selection and adjustment of settings, and their association with OVAL and OCIL content provides an automated checking capability. Transformations of this document, and its associated automated checking content, are capable of providing baselines that meet a diverse set of policy objectives. Some example XCCDF *Profiles*, which are selections of items that form checklists and can be used as baselines, are available with this guide. They can be processed, in an automated fashion, with tools that support the Security Content Automation Protocol (SCAP). The DISA STIG, which provides required settings for US Department of Defense systems, is one example of a baseline created from this guidance.

Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. The creators of this guidance assume no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

Profile Information

Profile Title	DISA STIG for Oracle Linux 7
Profile ID	stig

CPE Platforms

- cpe:/o:oracle:linux:7

Revision History

Current version: **0.1.46**

- **draft** (as of 2020-04-21)

Table of Contents

Checklist

Group Guide to the Secure Configuration of Oracle Linux 7 Group contains 97 groups and 242 rules

Group Services Group contains 26 groups and 46 rules

[ref] The best protection against vulnerable software is running less software. This section describes how to review the software which Oracle Linux 7 installs on a system and disable software which is not needed. It then enumerates the software packages installed on a default Oracle Linux 7 system and provides guidance about which ones can be safely disabled.

Oracle Linux 7 provides a convenient minimal install option that essentially installs the bare necessities for a functional system. When building Oracle Linux 7 systems, it is highly recommended to select the minimal packages and then build up the system from there.

Group Obsolete Services Group contains 4 groups and 7 rules

[ref] This section discusses a number of network-visible services which have historically caused problems for system security, and for which disabling or severely limiting the service has been the best available guidance for some time. As a result of this, many of these services are not installed as part of Oracle Linux 7 by default.

Organizations which are running these services should switch to more secure equivalents as soon as possible. If it remains absolutely necessary to run one of these services for legacy reasons, care should be taken to restrict the service as much as possible, for instance by configuring host firewall software such as `iptables` to restrict access to the vulnerable service to only those remote hosts which have a known need to use it.

Group Rlogin, Rsh, and Rexec Group contains 3 rules

[ref] The Berkeley r-commands are legacy services which allow cleartext remote access and have an insecure trust model.

Rule **Remove User Host-Based Authentication Files** [ref]

The `~/.shosts` (in each user's home directory) files list remote hosts and users that are trusted by the local system. To remove these files, run the following command to delete them from any location:

```
$ sudo find / -name '.shosts' -type f -delete
```

Rationale:	The .shosts files are used to configure host-based authentication for individual users or the system via SSH. Host-based authentication is not sufficient for preventing unauthorized access to the system, as it does not require interactive identification and authentication of a connection request, or for the use of two-factor authentication.
Severity:	high
Identifiers and References	References: CCI-000366, SRG-OS-000480-GPOS-00227, OL07-00-040540, SV-108583r1_rule

Remediation Shell script: (show)

Running an OVAL Auditing Scan

Oracle provides OVAL definitions for all errata on ULN. You can use these definitions to ensure that all applicable errata are installed on an Oracle Linux system. For example, Spacewalk allows you to schedule regular auditing scans.

The following OVAL definition files are available:

- `com.oracle.elsa-`*`cve`*`.xml`

  OVAL definition file for a single ELSA security patch. For example, `com.oracle.elsa-20150377.xml` relates to ELSA-2015-0377.

- `com.oracle.elsa-`*`year`*`.xml.bz2`

  Compressed archive of OVAL definition files for all ELSA patches released in a given *year*.

- `com.oracle.else-all.xml.bz2`

  Compressed archive of all applicable OVAL definition files for all available ELSA patches.

To download an OVAL definitions file and perform an audit on a system:

1. Use `wget` or a similar command to download a definitions file from https://linux.oracle.com/security, for example:

   ```
   wget https://linux.oracle.com/security/oval/com.oracle.elsa-2017.xml.bz2
   ```

2. In the definitions file is a compressed `bz2` archive, use `bzip2` to extract the OVAL definitions file:

   ```
   bzip2 -d com.oracle.elsa-2017.xml.bz2
   ```

3. Use `oscap oval eval` to audit a system using an OVAL definitions file, for example:

   ```
   oscap oval eval --results /tmp/elsa-results-oval.xml \
   --report /var/www/html/elsa-report-oval.html \
   /tmp/com.oracle.elsa-2017.xml

   Definition oval:com.oracle.elsa:def:20173580: false
   Definition oval:com.oracle.elsa:def:20173579: true
   Definition oval:com.oracle.elsa:def:20173576: false
   Definition oval:com.oracle.elsa:def:20173575: false
   Definition oval:com.oracle.elsa:def:20173574: true
   Definition oval:com.oracle.elsa:def:20173567: false
   Definition oval:com.oracle.elsa:def:20173566: false
   Definition oval:com.oracle.elsa:def:20173565: true
   Definition oval:com.oracle.elsa:def:20173539: true
   Definition oval:com.oracle.elsa:def:20173538: false
   Definition oval:com.oracle.elsa:def:20173537: false
   ...
   Evaluation done.
   ```

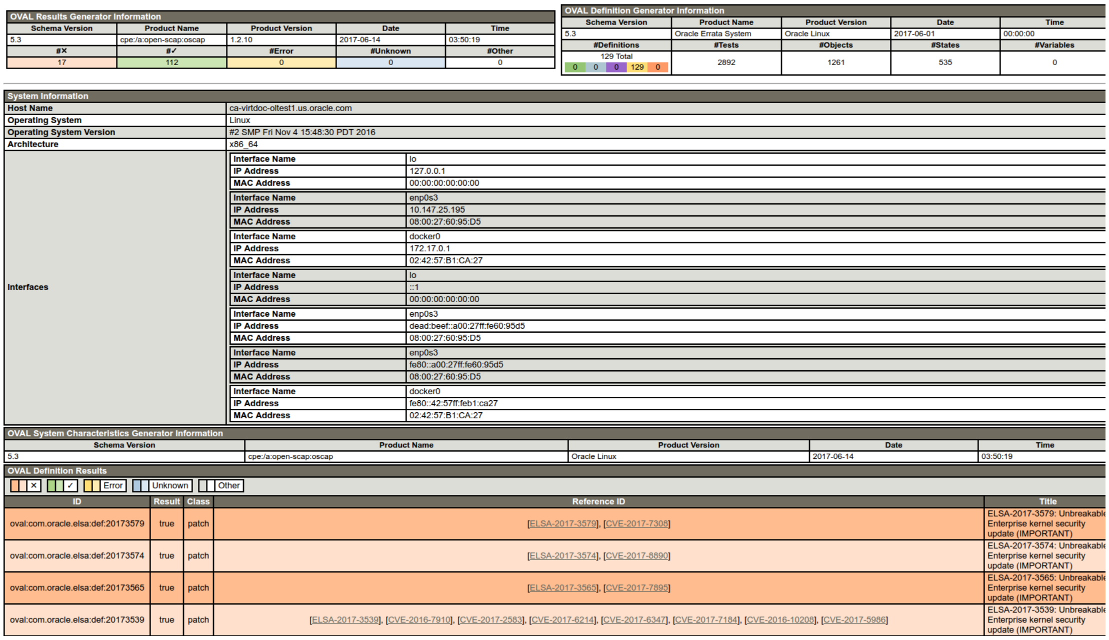
   This example scan uses the OVAL definitions in `com.oracle.elsa-2017.xml` and outputs the XML results and HTML report files to `/tmp` and `/var/www/html` respectively. A result of `true` for a patch means that it has not been applied to a system; a result of `false` means that it has been applied.

If you generate an XML results file but not the HTML report, you can use `oscap oval generate report` to convert the results file to an HTML report, for example:

```
oscap oval generate report /tmp/elsa-results-oval.xml > /var/www/html/elsa-report-
oval.html
```

You can view the HTML report in a browser as shown in Figure 5-3.

Figure 5-3 Sample OVAL Report



OVAL Results Generator Information				
Schema Version	Product Name	Product Version	Date	Time
5.3	cpe:/a:open-scap:oscap	1.2.10	2017-06-14	03:50:19
#✕	#✓	#Error	#Unknown	#Other
17	112	0	0	0

OVAL Definition Generator Information				
Schema Version	Product Name	Product Version	Date	Time
5.3	Oracle Errata System	Oracle Linux	2017-06-01	00:00:00
#Definitions	#Tests	#Objects	#States	#Variables
129 Total 0 0 0 129 0	2892	1261	535	0

System Information		
Host Name	ca-virtdoc-oltest1.us.oracle.com	
Operating System	Linux	
Operating System Version	#2 SMP Fri Nov 4 15:48:30 PDT 2016	
Architecture	x86_64	
Interfaces	Interface Name	lo
	IP Address	127.0.0.1
	MAC Address	00:00:00:00:00:00
	Interface Name	enp0s3
	IP Address	10.147.25.195
	MAC Address	08:00:27:60:95:D5
	Interface Name	docker0
	IP Address	172.17.0.1
	MAC Address	02:42:57:B1:CA:27
	Interface Name	lo
	IP Address	::1
	MAC Address	00:00:00:00:00:00
	Interface Name	enp0s3
	IP Address	dead:beef::a00:27ff:fe60:95d5
	MAC Address	08:00:27:60:95:D5
	Interface Name	enp0s3
	IP Address	fe80::a00:27ff:fe60:95d5
	MAC Address	08:00:27:60:95:D5
	Interface Name	docker0
	IP Address	fe80::42:57ff:feb1:ca27
	MAC Address	02:42:57:B1:CA:27

OVAL System Characteristics Generator Information				
Schema Version	Product Name	Product Version	Date	Time
5.3	cpe:/a:open-scap:oscap	Oracle Linux	2017-06-14	03:50:19

OVAL Definition Results

✕ ✓ Error Unknown Other

ID	Result	Class	Reference ID	Title
oval:com.oracle.elsa:def:20173579	true	patch	[ELSA-2017-3579], [CVE-2017-7308]	ELSA-2017-3579: Unbreakabl Enterprise kernel security update (IMPORTANT)
oval:com.oracle.elsa:def:20173574	true	patch	[ELSA-2017-3574], [CVE-2017-8890]	ELSA-2017-3574: Unbreakabl Enterprise kernel security update (IMPORTANT)
oval:com.oracle.elsa:def:20173565	true	patch	[ELSA-2017-3565], [CVE-2017-7895]	ELSA-2017-3565: Unbreakabl Enterprise kernel security update (IMPORTANT)
oval:com.oracle.elsa:def:20173539	true	patch	[ELSA-2017-3539], [CVE-2016-7910], [CVE-2017-2583], [CVE-2017-6214], [CVE-2017-6347], [CVE-2017-7184], [CVE-2016-10208], [CVE-2017-5986]	ELSA-2017-3539: Unbreakabl Enterprise kernel security update (IMPORTANT)

Scanning Containers, Container Images and Offline File Systems

OpenSCAP includes utilities that allow you to scan Docker containers or container images using the `oscap-docker` command; or to scan offline file systems hosting an operating system using the `oscap-chroot` command. Note that due to the mechanisms that are used to perform these scans, these tools are only supported in the context of a scan against an Oracle Linux 7 based system.

Scan Container Images and Containers

Scan Docker containers or container images using the `oscap-docker` command. This tool assesses vulnerabilities in the container or image and checks compliance with security policies similarly to the `oscap` command. The tool uses offline scanning to perform all assessments and checks by performing a temporary read-only mount of the container or image file system. No changes are made to the container or image and no additional tools are required within the container or image.

To scan an image for vulnerabilities using the appropriate CVE stream for the image variant and to output this information in HTML format, run:

```
oscap-docker --disable-atomic image-cve ol7-image --report report.html
```

Note that you must use the --disable-atomic option when running the command. Atomic containers are not supported on Oracle Linux.

To scan an image for compliance with a security policy specified in an XCCDF checklist and to output the result in HTML format, run:

```
oscap-docker --disable-atomic image ol7-image xccdf eval  \
--fetch-remote-resources \
--profile <profile> \
--results results.xml \
--report report.html \
--cpe /usr/share/xml/scap/ssg/content/ssg-ol7-cpe-dictionary.xml \
/usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml
```

You can change the subcommand from image or image-cve to container or container-cve if you would prefer to scan a specific container. Note that you must use the --disable-atomic option when running the command. Atomic containers are not supported on Oracle Linux.

See the oscap-docker(8) manual page for more information.

Scan Offline File Systems

Use the oscap-chroot command to perform an offline scan of a file system that is mounted at a specified path. This tool can be used for scanning of custom objects that are not supported by oscap-docker, like containers that use an alternate format to Docker or virtual machine disk files. The options for this tool are similar to the oscap command.

For example, to audit a file system mounted at /mnt audit using an OVAL definitions file, run:

```
oscap-chroot /mnt oval eval --results /tmp/elsa-results-oval.xml \
--report elsa-report-oval.html \
/tmp/com.oracle.elsa-2021.xml
```

See the oscap-chroot(8) manual page for more information.

6

FIPS 140-2 Compliance in Oracle Linux 7

Oracle Linux provides a set of cryptographic libraries, services, and user-level cryptographic applications that are validated at the Federal Information Processing Standard (FIPS) Publication 140-2.

FIPS Publication 140-2, Security Requirements for Cryptographic Modules, specifies the security requirements that must be satisfied by a cryptographic module that is used within a security system to protect sensitive, but unclassified information. The NIST/CSE Cryptographic Module Validation Program (CMVP) validates cryptographic modules to FIPS 140-2. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

For security requirements and guidance that apply to FIPS validated cryptographic modules for Oracle Linux 7.8, see Changes to Requirements and Guidance for FIPS Validated Cryptographic Modules for Oracle Linux 7.8.

FIPS Validated Cryptographic Modules for Oracle Linux 7.8

The following table describes Oracle's FIPS 140-2 Level 1 certifications for cryptographic components that reside within Oracle Linux 7.8 on the x86_64 and aarch64 platforms. The package versions that are listed reflect information that is found in the logical cryptographic boundary for the specific module.

> **Important:**
>
> Several changes that impact requirements and guidance for FIPS validated cryptographic modules are introduced in this release. See Changes to Requirements and Guidance for FIPS Validated Cryptographic Modules for Oracle Linux 7.8 for more details about these changes.

Cryptographic Module Name	Package Version	Certificate Number
Oracle Linux 7 OpenSSL Cryptographic Module	`openssl-libs-1.0.2k-21.0.1.el7_9.x86_64` `openssl-libs-1.0.2k-21.0.1.el7_9.aarch64`	4170 (x86_64 and aarch64)
Oracle Linux 7 OpenSSH Client Cryptographic Module	`openssh-clients-7.4p1-21.0.3.el7.x86_64` `openssh-clients-7.4p1-21.0.3.el7.aarch64`	4206 (x86_64 and aarch64)

Cryptographic Module Name	Package Version	Certificate Number
Oracle Linux 7 OpenSSH Server Cryptographic Module	openssh-server-7.4p1-21.0.3.el7.x86_64 openssh-server-7.4p1-21.0.3.el7.aarch64	4176 (x86_64 and aarch64)
Oracle Linux 7 libgcrypt Cryptographic Module	libgcrypt-1.5.3-14.el7.x86_64 libgcrypt-1.5.3-14.el7.aarch64	3604 (x86_64 and aarch64)
Oracle Linux 7 NSS Cryptographic Module	nss-softokn-3.53.1-6.0.1.el7_9.x86_64 nss-softokn-3.53.1-6.0.1.el7_9.aarch64	4171 (x86_64 and aarch64)
Oracle Linux 7 Libreswan Cryptographic Module	libreswan-3.25-9.1.0.3.el7_8.x86_64 libreswan-3.25-9.1.0.3.el7_8.aarch64	4266 (x86_64 and aarch64)
Oracle Linux 7 GnuTLS Cryptographic Module	gnutls-3.3.29-9.el7_6.x86_64 gnutls-3.3.29-9.el7_6.aarch64	3757 (x86_64 and aarch64)
Oracle Linux Unbreakable Enterprise Kernel (UEK 6) Cryptographic Module	kernel-uek-5.4.17-2102.200.13.el7uek.x86_64 kernel-uek-5.4.17-2102.200.13.el7uek.aarch64	4169 (x86_64 and aarch64)

Locations of Packages for FIPS Validated Cryptographic Modules for Oracle Linux 7.8

The following are the dedicated Unbreakable Linux Network (ULN) channels and yum repository containing FIPS validated cryptographic modules for Oracle Linux 7.8:

x86_64 Platform:

- `ol7_x86_64_u8_security_validation` ULN channel
- `ol7_u8_security_validation` yum repository

aarch64 Platform:

- `ol7_aarch64_u8_security_validation` ULN channel
- `ol7_u8_security_validation` yum repository

Note that the `ol7_u8_security_validation` yum repository is a common repository name for the x86_64 and aarch64 platforms and contains FIPS validated packages for both architectures.

> **Important:**
>
> With Oracle Linux 7.8, FIPS validated versions of cryptographic modules for `OpenSSL`, `OpenSSH`, and `libreswan` receive security errata updates through the `ol7_x86_64_u8_security_validation` and `ol7_aarch64_u8_security_validation` ULN channels, as well as the corresponding `ol7_u8_security_validation` yum repository, which is a common repository name that is used for the x86_64 and aarch64 platforms. This repository contains the FIPS validated packages for both architectures. For these packages, on top of the package version, Epoch was increased to a substantially higher number to ensure that these packages are never updated by package versions that do not contain the required FIPS patches.

Note that packages for FIPS validated cryptographic modules for Oracle Linux 7.3, Oracle Linux 7.5, and Oracle Linux 7.6 continue to be made available in the `ol7_x86_64_security_validation` ULN channel and the corresponding `ol7_security_validation` and `ol7_latest` yum repositories.

For instructions on installing FIPS validated cryptographic modules on Oracle Linux 7, see Installing FIPS Validated Cryptographic Modules for Oracle Linux.

Changes to Requirements and Guidance for FIPS Validated Cryptographic Modules for Oracle Linux 7.8

The Cryptographic Module Validation Program (CMVP) has made changes to the FIPS requirements and guidance for the FIPS cryptographic modules that are described in FIPS Validated Cryptographic Modules for Oracle Linux 7.8.

With Oracle Linux 7.8, new requirements that are specifically related to the Special Publication (SP 800-56Ar3) document issued by NIST have been implemented. Also note that the OpenSSH package has been updated to disable the `diffie-hellman-group-exchange-sha256` KEX algorithm for FIPS mode, per SP 800-56Ar3 compliance, as this algorithm is not a FIPS approved safe-prime group.

These new FIPS 140-2 requirements that pertain to Oracle Linux 7.8, which are based on the implementation of NIST SP 800-56Ar3 compliance, stipulate that server connections only allow the use of safe primes as part of the Diffie-Hellman key agreement. As such, attempts by clients to connect to a server that is connected in FIPS mode that does not support NIST SP 800-56Ar3 safe primes fail. For more information, see https://nvlpubs.nist.gov/nistpubs/SpecialPublications/nist.sp.800-56Ar3.pdf.

For instructions on how to configure the module for FIPS mode, refer to Section 10.1 of the Crypto Officer Guidance document when you install the module to verify that the package was FIPS 140-2 validated and ensure that you correctly enable the module for FIPS mode.

These changes pertain to the following packages for Oracle Linux 7.8 on the x86_64 and aarch64 platforms:

- Oracle Linux 7 OpenSSL Cryptographic Module

  - `openssl-libs-1.0.2k-21.0.1.el7_9.x86_64`
  - `openssl-libs-1.0.2k-21.0.1.el7_9.aarch64`
- Oracle Linux 7 NSS Cryptographic Module
  - `nss-softokn-3.53.1-6.0.1.el7_9.x86_64`
  - `nss-softokn-3.53.1-6.0.1.el7_9.aarch64`
- Oracle Linux 7 OpenSSH Server Cryptographic Module
  - `openssh-server-7.4p1-21.0.3.el7.x86_64`
  - `openssh-server-7.4p1-21.0.3.el7.aarch64`
- Oracle Linux 7 OpenSSH Client Cryptographic Module
  - `openssh-clients-7.4p1-21.0.3.el7.x86_64`
  - `openssh-clients-7.4p1-21.0.3.el7.aarch64`
- Oracle Linux 7 Libreswan Cryptographic Module
  - `libreswan-3.25-9.1.0.3.el7_8.x86_64`
  - `libreswan-3.25-9.1.0.3.el7_8.aarch64`

FIPS Validated Cryptographic Modules for Oracle Linux 7.5 and Oracle Linux 7.6

Oracle has completed FIPS 140-2 Level 1 certifications for cryptographic components that reside within Oracle Linux 7.5 and Oracle Linux 7.6 on the x86_64 platform. The completed certification is described in the following table.

Cryptographic Module Name	Oracle Linux Release and Package Version	Certificate Number
Oracle Linux 7 OpenSSL Cryptographic Module	Oracle Linux 7.5: `openssl-libs-1.0.2k-12.0.3.el7.x86_64`	3474
Oracle Linux 7 OpenSSL Cryptographic Module	Oracle Linux 7.6: `openssl-libs-1.0.2k-16.0.1.el7.x86_64`	3474
Oracle Linux 7 OpenSSH Client Cryptographic Module	Oracle Linux 7.6: `openssh-clients-7.4p1-16.el7.x86_64`	3582
Oracle Linux 7 OpenSSH Server Cryptographic Module	Oracle Linux 7.6: `openssh-server-7.4p1-16.el7.x86_64`	3590
Oracle Linux 7 libgcrypt Cryptographic Module	Oracle Linux 7.6: `libgcrypt-1.5.3-14.el7.x86_64`	3604

Cryptographic Module Name	Oracle Linux Release and Package Version	Certificate Number
Oracle Linux 7 NSS Cryptographic Module	Oracle Linux 7.6: `nss-softokn-3.36.0-5.0.1.el7_5.x86_64`	3616
Oracle Linux 7 Libreswan Cryptographic Module	Oracle Linux 7.6: `libreswan-3.25-4.1.0.1.el7_6.x86_64`	3699
Oracle Linux 7 GnuTLS Cryptographic Module	Oracle Linux 7.6: `gnutls-3.3.29-9.el7_6.x86_64`	3757
Oracle Linux Unbreakable Enterprise Kernel (UEK 5) Cryptographic Module	Oracle Linux 7.6: `kernel-uek-4.14.35-1902.300.11.el7uek.x86_64`	3893
Oracle Linux Unbreakable Enterprise Kernel (UEK 4) Cryptographic Module	Oracle Linux 7.6: `kernel-uek-4.1.12-124.36.1.el7uek.x86_64`	3921

FIPS Validated Cryptographic Modules for Oracle Linux 7.3

Oracle has completed FIPS 140-2 Level 1 certifications for cryptographic components that reside within Oracle Linux 7.3 on the x86_64 platform. The completed certifications include those that are described in the following table.

Cryptographic Module Name	Package Version	Certificate Number
Oracle Linux OpenSSL Cryptographic Module	`openssl-1.0.1e60.0.1.el7_3.1.x86_64`	3017
Oracle Linux 7 OpenSSH Server Cryptographic Module	`openssh-6.6.1p1-35.el7_3.x86_64`	3028
Oracle Linux 7 OpenSSH Client Cryptographic Module	`openssh-6.6.1p1-35.el7_3.x86_64`	3032
Oracle Linux 7 NSS Cryptographic Module	`nss-softokn-3.16.2.3-14.4.0.1.el7.x86_64`	3143
Oracle Linux 7 Libreswan Cryptographic Module	`libreswan-3.15-8.0.1.el7.x86_64`	3168
Oracle Linux 7 GnuTLS Cryptographic Module	`gnutls-3.3.24-1.0.3.el7.x86_64`, `gmp-6.0.0-12.el7_1.x86_64`, `nettle-2.7.1-8.el7.x86_64`	3169
Oracle Linux 7 `libgcrypt` Cryptographic Module	`libgcrypt-1.5.3-13.el7_3.1.x86_64`	3215

Cryptographic Module Name	Package Version	Certificate Number
Oracle Linux 7 Kernel Crypto API Cryptographic Module	`kernel-3.10.0-862.3.3.0.1.el7.x86_64`	3342
Oracle Linux Unbreakable Enterprise Kernel (UEK 4) Cryptographic Module	`kernel-uek-4.1.12-124.16.4.el7uek.x86_64`	3348

More Information About Modules That Have Received FIPS 140-2 Validation

The site provides the following information for each module:

- Name and description of the module.
- Package version or versions for the module.
- Status of the FIPS 140-2 validation process.

> **Important:**
>
> To achieve compliance with FIPS Publication 140-2, you must use the package version that the Security Policy document specifies for each respective module *only*. You cannot install and use other versions of the cryptographic modules.

- Instructions on how to configure the module for FIPS mode. Refer to Section 10 of the Security Policy document when you install the module to verify that the package was FIPS 140-2 validated and ensure that you correctly enable the module for FIPS mode.

Enabling FIPS Mode on Oracle Linux

Prior to using FIPS validated cryptographic modules on systems that are running Oracle Linux 7, you must enable FIPS mode. The following procedure describes how to configure Oracle Linux to use only those cryptographic algorithms that are FIPS validated.

Unless otherwise noted, the following procedure applies generally to systems that are running an Oracle Linux 7 release that includes support for the enabling of FIPS mode. It is recommended that you update your system to the latest Oracle Linux 7 release that provides this capability. You cannot use FIPS cryptographic modules on Oracle Linux 7 systems that are running an update earlier than Oracle Linux 7.3.

> **Note:**
>
> For more information about enabling FIPS mode in Oracle Linux containers, see the Working With Containers and Images chapter in the Oracle Linux: Oracle Container Runtime for Docker User's Guide.

1. Ensure that the system is running an Oracle Linux 7 release that includes support for validated FIPS cryptographic modules.
2. Ensure that your system is registered with the Unbreakable Linux Network (ULN) and that it is subscribed to the appropriate channel for the Oracle Linux 7 release that you are running. If you are using the Oracle Linux yum server, enable the appropriate repository, or repositories, as required:
   - If you are running Oracle Linux 7.8 on the x86_64 platform, subscribe to the `ol7_x86_64_u8_security_validation` and `ol7_x86_64_latest` ULN channels; or, for the aarch64 platform, subscribe to the `ol7_aarch64_u8_security_validation` and `ol7_aarch64_latest` ULN channels.

     If you are using the Oracle Linux yum server, enable the `ol7_u8_security_validation` and `ol7_latest` yum repositories, for example:

     ```
     sudo yum-config-manager --enable ol7_u8_security_validation ol7_latest
     ```

     See Locations of Packages for FIPS Validated Cryptographic Modules for Oracle Linux 7.8 for additional information.
   - If you are running Oracle Linux 7.3, Oracle Linux 7.5, or Oracle Linux 7.6, subscribe to the `ol7_x86_64_security_validation` ULN channel.

     If you are using the Oracle Linux yum server, enable the `ol7_security_validation` and `ol7_latest` repositories, for example:

     ```
     sudo yum-config-manager --enable ol7_security_validation ol7_latest
     ```
3. Install the `dracut-fips` package.

   ```
   sudo yum install dracut-fips
   ```

   The `dracut-fips` package provides the modules to build a dracut initramfs file system that performs an integrity check.
4. If the system CPU supports AES New Instructions (AES-NI), install the package.
   - Check whether the system supports AES-NI:

     ```
     grep aes /proc/cpuinfo
     ```
   - Install the package.

     ```
     sudo yum install dracut-fips-aesni
     ```
5. Recreate the initramfs file system.

   ```
   sudo dracut -f
   ```
6. Reconfigure the boot loader so that the system boots in FIPS mode:
   a. With appropriate administrative privileges, edit the `/etc/default/grub` file and add the `fips=1` option to the boot loader configuration:

```
GRUB_CMDLINE_LINUX="vconsole.font=latarcyrheb-sun16
rd.lvm.lv=ol/swap rd.lvm.lv=ol/root crashkernel=auto
  vconsole.keymap=uk rhgb quiet fips=1"
```

b. If `/boot` is located on a dedicated partition other than the root partition, you must update the boot loader configuration to use the `boot=UUID=`*`boot_UUID`* option so that the device is mounted at `/boot` when the kernel loads. For example:

```
GRUB_CMDLINE_LINUX="vconsole.font=latarcyrheb-sun16
     rd.lvm.lv=ol/swap rd.lvm.lv=ol/root crashkernel=auto
     vconsole.keymap=uk rhgb quiet
     boot=UUID=69fa3946-dd8d-4870-bf38-0d540eb9e6c6 fips=1"
```

You can determine whether a dedicated block device exists for your boot partition by typing:

```
lsblk -f|grep /boot
```

```
├─nvme0n1p1     vfat     20DC-FE64                               /boot/efi
└─nvme0n1p2     xfs      69fa3946-dd8d-4870-bf38-0d540eb9e6c6    /boot
```

> **Note:**
>
> On systems that are configured to boot with UEFI, `/boot/efi` is always located on a dedicated partition, as it is formatted specifically to meet UEFI requirements. Ignore `/boot/efi` when determining whether or not the `/boot` is located on a dedicated partition.
>
> Only use the `boot=` parameter if `/boot` is located on a dedicated partition. If the parameter is specified incorrectly or points to a non-existent device, the system might not boot.

These steps are required for FIPS to perform kernel validation checks, where it verifies the kernel against the provided HMAC file in the `/boot` directory.

c. Save the changes that you have made to the `/etc/default/grub` file.

7. Rebuild the GRUB configuration.

   - On BIOS-based systems, run the following command:

     ```
     sudo grub2-mkconfig -o /boot/grub2/grub.cfg
     ```

   - On UEFI-based systems, run the following command:

     ```
     sudo grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
     ```

8. Disable prelinking on all libraries and binaries.

   To ensure proper operation of the in-module integrity verification, prelinking must be disabled on all system files and the `prelink` package must not be installed on the system.

   If the `prelink` package is installed, disable prelinking on all libraries and binaries as follows:

   a. Set `PRELINKING=no` in the `/etc/sysconfig/prelink` configuration file.

b. If the libraries were already prelinked, undo the prelink on all of the system files by using the following command:

```
sudo prelink -u -a
```

c. Remove the prelink package from the system:

```
sudo yum remove prelink
```

9. Reboot the system and verify that FIPS is enabled:

```
cat /proc/sys/crypto/fips_enabled

1
```

A response of `1` indicates that FIPS is enabled.

Installing FIPS Validated Cryptographic Modules for Oracle Linux

After you enable FIPS mode on Oracle Linux, you can then install FIPS validated cryptographic modules, as required. For information about where packages for FIPS validated cryptographic modules are located, see Locations of Packages for FIPS Validated Cryptographic Modules for Oracle Linux 7.8.

The following information applies to systems that are running an Oracle Linux 7 release that includes support for installing and enabling FIPS cryptographic modules. It is recommended that you update your system to the latest Oracle Linux 7 release that provides this support.

> **Note:**
>
> You cannot use FIPS cryptographic modules on systems that are running Oracle Linux 7.2 or earlier releases.
>
> If you are installing FIPS cryptographic modules for Oracle Linux 7.8, see Changes to Requirements and Guidance for FIPS Validated Cryptographic Modules for Oracle Linux 7.8 for additional information.

To install FIPS validated cryptographic modules, refer to Section 10 of the Security Policy document for the FIPS module that you plan to install.

The Security Policy document explains how to verify that the package is FIPS 140-2 validated, as well as how to configure the module for FIPS mode. See FIPS Validated Cryptographic Modules for Oracle Linux 7.8, FIPS Validated Cryptographic Modules for Oracle Linux 7.5 and Oracle Linux 7.6, and FIPS Validated Cryptographic Modules for Oracle Linux 7.3 for the certificate number, which provides a link to the NIST FIPS 140 validation page. This page provides details about FIPS certification and the Security Policy document.

7 Oracle Linux Common Criteria Certification

Oracle Linux 7.3 was certified in February 2019 with two sets of certificates under the Swedish Common Criteria Scheme. The first certification demonstrated Exact Conformance against the Protection Profile for General Purpose Operating Systems (OSPP) 4.1. The second certification claimed an evaluation level of Evaluation Assurance Level 1 (EAL1), augmented by flaw remediation.

Oracle Linux 7.6 was certified in May 2021, under the Canadian Common Criteria Scheme, and has demonstrated Exact Conformance against the Protection Profile for General Purpose Operating Systems (OSPP) 4.2.1.

Certificates awarded in Sweden, Canada, and 15 other countries are recognized by 31 countries under the Common Criteria Recognition Arrangement (CCRA).

The certifications include the following evaluated security functionality:

- Security Audit
- Cryptographic support
- Identification and Authentication
- User Data Protection
- Security Management
- Self-protection
- TLS and SSH protocols

> **Important:**
>
> To achieve compliance, the Common Criteria evaluated package set must be selected at installation time, in accordance with the description that's provided in the Common Criteria Guide for the Oracle Linux 7 release that you are running, and installed accordingly.
>
> For more information, see Common Criteria Guide for Oracle Linux 7.3 and Common Criteria Guide for Oracle Linux 7.6.

For additional information about the Common Criteria certification, see the following references:

- External Security Evaluations
- Common Criteria Recognition Arrangement
- Protection Profile for General Purpose Operating Systems

8 Oracle Linux KVM Common Criteria Certification

Oracle Linux KVM and Oracle Linux Virtualization Manager were certified in March 2023 under the Canadian Common Criteria scheme. The certification demonstrates compliance with the following approved Protection Profile and Extended Packages:

- Protection Profile for Virtualization 1.0
- Extended Package for Server Virtualization 1.0
- Extended Package for Secure Shell (SSH) 1.0

Certificates awarded in Canada and 17 other countries are recognized by 31 countries under the Common Criteria Recognition Arrangement (CCRA).

> **Important:**
>
> To achieve compliance, the Common Criteria evaluated package set must be selected at installation time, in accordance with the description that's provided in the Common Criteria Guide for the Oracle Linux 7 and KVM release that you are running, and installed accordingly.
>
> For more information, see the Common Criteria Guide.

For additional information about the Common Criteria certification, see the following references:

- External Security Evaluations
- Common Criteria Recognition Arrangement
- Oracle Linux Common Criteria Certification

For more information about these products, see Oracle Linux: KVM User's Guide and Oracle Linux Virtualization Manager.