Oracle® Linux

Administering SELinux

F22957-10
August 2021

Abstract

Oracle® Linux: Administering SELinux describes SELinux, how the feature works, and includes instructions on configuring and administering SELinux in the Oracle Linux release.

Document generated on: 2021-08-04 (revision: 12311)

Table of Contents

Preface

Oracle® Linux: Administering SELinux provides an overview of the SELinux feature and includes tasks for administering SELinux on Oracle Linux systems.

Note

This guide contains content that was tested against Oracle Linux 8, but generally applies to most Oracle Linux releases, and may also apply to other distributions.

Audience

This document is intended for administrators who need to configure and administer Oracle Linux features. It is assumed that readers are familiar with web technologies and have a general understanding of using the Linux operating system, including knowledge of how to use a text editor such as `emacs` or `vim`, essential commands such as `cd`, `chmod`, `chown`, `ls`, `mkdir`, `mv`, `ps`, `pwd`, and `rm`, and using the `man` command to view manual pages.

Document Organization

The document is organized into the following chapters:

- Chapter 1, *About Administering SELinux in Oracle Linux* describes the Oracle Linux describes the SElinux feature and its components.
- Chapter 2, *Administering SELinux Policies* describes SELinux policies.
- Chapter 3, *Administering SELinux Security Context* describes the security content that SELinux uses.
- Chapter 4, *Administering SELinux Users* describes how to administer SELinux users.
- Chapter 5, *Troubleshooting Access-Denial Messages* describes how to troubleshoot access denial messages in Oracle Linux.

Related Documents

The documentation for this product is available at:

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

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/.

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.

Chapter 1 About Administering SELinux in Oracle Linux

Table of Contents

This chapter describes the SELinux feature and provides tasks for administering SELinux on Oracle Linux systems.

Note

The content in this document was tested against Oracle Linux 8, but generally applies to most Oracle Linux releases, and may also apply to other distributions.

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`. Access to files and devices is based solely on user identity and ownership. Malware or broken software can do anything with files and resources that the user that started the process can do. If the user is `root` or the application is `setuid` or `setgid` to `root`, the process can have `root`-access control over the entire file system.

The National Security Agency created Security Enhanced Linux (SELinux) to provide a finer-grained level of control over files, processes, users and applications in the Linux operating system. 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.

When security-relevant access occurs, such as when a process attempts to open a file, SELinux intercepts the operation in the kernel. If a MAC policy rule allows the operation, it continues; otherwise, SELinux blocks the operation and returns an error to the process. The kernel checks and enforces DAC policy rules before MAC rules, so it does not check SELinux policy rules if DAC rules have already denied access to a resource.

1.1 SELinux Package Descriptions

SELinux contains several packages, each of which contain specific utilities that you can use to administer SELinux on your Oracle Linux systems. Some packages are installed by default, while other packages are optional.

The following table describes the SELinux packages that are installed by default with Oracle Linux.

Package	Description
`policycoreutils`	Provides utilities such as `load_policy`, `restorecon`, `secon`, `setfiles`, `semodule`, `sestatus`, and `setsebool` for operating and managing SELinux.
`libselinux`	Provides the API that SELinux applications use to get and set process and file security contexts, and to obtain security policy decisions.
`python3-libselinux`	Contains Python bindings for developing SELinux applications.
`selinux-policy`	Provides the SELinux Reference Policy, which is used as the basis for other policies, such as the SELinux targeted policy.

Package	Description
selinux-policy-targeted	Provides support for the SELinux targeted policy, where objects outside the targeted domains run under DAC.
libselinux-utils	Provides the avcstat, getenforce, getsebool, matchpathcon, selinuxconlist, selinuxdefcon, selinuxenabled, setenforce, and togglesebool utilities.

The following table describes useful SELinux packages that are not installed by default.

Package	Description
mcstrans	Translates SELinux levels, such as s0-s0:c0.c1023, to an easier-to-read form, such as SystemLow-SystemHigh.
policycoreutils-python-utils	Provides additional Python utilities for operating SELinux, such as audit2allow, audit2why, chcat, and semanage.
selinux-policy-mls	Provides support for the strict Multilevel Security (MLS) policy as an alternative to the SELinux targeted policy.
setroubleshoot	Allows you to view setroubleshoot-server messages by using the sealert command.
setroubleshoot-server	Translates access-denial messages from SELinux into detailed descriptions that you can view on the command line using the sealert command.
setools-console	Provides the Tresys Technology SETools distribution of tools and libraries, which you can use to analyze and query policies, monitor and report audit logs, and manage file context.

Use the dnf command or another suitable package manager to install any additional SELinux packages that you require for your system.

For more information, see the SELinux Project Wiki, the selinux(8) manual page, and other manual pages for the SELinux commands.

1.2 Using SELinux Utilities

The following table describes the utilities that you can use to administer SELinux and information about the packages that contain each utility.

Utility	Package	Description
audit2allow	policycoreutils-python-utils	Generates SELinux policy allow_audit rules from logs of denied operations.
audit2why	policycoreutils-python-utils	Generates SELinux policy don't_audit rules from logs of denied operations.
avcstat	libselinux-utils	Displays statistics for the SELinux Access Vector Cache (AVC).
chcat	policycoreutils-python-utils	Changes or removes the security category for a file or user.
findcon	setools-console	Searches for file context.
fixfiles	policycoreutils	Fixes the security context for file systems.
getenforce	libselinux-utils	Reports the current SELinux mode.
getsebool	libselinux-utils	Reports SELinux boolean values.

Utility	Package	Description
indexcon	setools-console	Indexes file context.
load_policy	policycoreutils	Loads a new SELinux policy into the kernel.
matchpathcon	libselinux-utils	Queries the system policy and displays the default security context that is associated with the file path.
replcon	setools-console	Replaces file context.
restorecon	policycoreutils	Resets the security context on one or more files.
restorecond	policycoreutils	Daemon that watches for file creation and sets the default file context.
sandbox	policycoreutils-python-utils	Runs a command in an SELinux sandbox.
sealert	setroubleshoot-server, setroubleshoot	Acts as the user interface to the setroubleshoot system, which diagnoses and explains SELinux AVC denials and provides recommendations on how to prevent such denials.
seaudit-report	setools-console	Reports from the SELinux audit log.
sechecker	setools-console	Checks SELinux policies.
secon	policycoreutils	Displays the SELinux context from a file, program, or user input.
sediff	setools-console	Compares SELinux polices.
seinfo	setools-console	Queries SELinux policies.
selinuxconlist	libselinux-utils	Displays all SELinux contexts that are reachable by a user.
selinuxdefcon	libselinux-utils	Displays the default SELinux context for a user.
selinuxenabled	libselinux-utils	Indicates whether SELinux is enabled.
semanage	policycoreutils-python-utils	Manages SELinux policies.
semodule	policycoreutils	Manages SELinux policy modules.
semodule_deps	policycoreutils	Displays the dependencies between SELinux policy packages.
semodule_expand	policycoreutils	Expands a SELinux policy module package.
semodule_link	policycoreutils	Links SELinux policy module packages together.
semodule_package	policycoreutils	Creates a SELinux policy module package.
sesearch	setools-console	Queries SELinux policies.
sestatus	policycoreutils	Displays the SELinux mode and the SELinux policy that are in use.
setenforce	libselinux-utils	Modifies the SELinux mode.
setsebool	policycoreutils	Sets SELinux boolean values.
setfiles	policycoreutils	Sets the security context for one or more files.
togglesebool	libselinux-utils	Flips the current value of an SELinux boolean.

1.3 Setting SELinux Modes

SELinux runs in one of three modes:

`Disabled`	The kernel uses only DAC rules for access control. SELinux does not enforce any security policy because no policy is loaded into the kernel.
`Enforcing`	The kernel denies access to users and programs unless permitted by SELinux security policy rules. All denial messages are logged as AVC (Access Vector Cache) denials. This is the default mode that enforces SELinux security policy.
`Permissive`	The kernel does not enforce security policy rules but SELinux sends denial messages to a log file. This allows you to see what actions would have been denied if SELinux were running in enforcing mode. This mode is intended to used for diagnosing the behavior of SELinux.

To display current SELinux mode:

```
getenforce
Enforcing
```

To set the current mode to `Enforcing`:

```
sudo setenforce enforcing
```

To set the current mode to `Permissive`:

```
sudo setenforce permissive
```

The current value that you set for a mode using `setenforce` does not persist across reboots. To configure the default SELinux mode, edit the configuration file for SELinux, `/etc/selinux/config`, and set the value of the `SELINUX` directive to `disabled`, `enforcing`, or `permissive`.

Chapter 2 Administering SELinux Policies

Table of Contents

An SELinux policy describes the access permissions for all users, programs, processes, and files, and for the devices upon which they act. You can configure SELinux to implement either Targeted Policy or Multilevel Security (MLS) Policy. This chapter describes SELinux policies and how to administer them.

2.1 Targeted Policy

Applies access controls to a limited number of processes that are believed to be most likely to be the targets of an attack on the system. Targeted processes run in their own SELinux domain, known as a *confined domain*, which restricts access to files that an attacker could exploit. If SELinux detects that a targeted process is trying to access resources outside the confined domain, it denies access to those resources and logs the denial. Only specific services run in confined domains. Examples are services that listen on a network for client requests, such as `httpd`, `named`, and `sshd`, and processes that run as `root` to perform tasks on behalf of users, such as `passwd`. Other processes, including most user processes, run in an unconfined domain where only DAC rules apply. If an attack compromises an unconfined process, SELinux does not prevent access to system resources and data.

The following table shows examples of SELinux domains.

Domain	Description
`init_t`	`systemd`
`httpd_t`	HTTP daemon threads
`kernel_t`	Kernel threads
`syslogd_t`	`journald` and `rsyslogd` logging daemons
`unconfined_t`	Processes executed by Oracle Linux users run in the unconfined domain

2.2 Multilevel Security (MLS) Policy

Applies access controls to multiple levels of processes with each level having different rules for user access. Users cannot obtain access to information if they do not have the correct authorization to run a process at a specific level. In SELinux, MLS implements the Bell-LaPadula (BLP) model for system security, which applies labels to files, processes and other system objects to control the flow of information between security levels. In a typical implementation, the labels for security levels might range from the most secure, `top secret`, through `secret`, and `classified`, to the least secure, `unclassified`. For example, under MLS, you might configure a program labelled `secret` to be able to write to a file that is labelled `top secret`, but not to be able to read from it. Similarly, you would permit the same program to read from and write to a file labelled `secret`, but only to read `classified` or `unclassified` files. As a result, information that passes through the program can flow upwards through the hierarchy of security levels, but not downwards.

Note

You must install the `selinux-policy-mls` package if you want to be able to apply the MLS policy.

2.3 Setting SELinux Policies

Note

You cannot change the policy type of a running system.

You can configure the default policy type by editing the `/etc/selinux/config` file and setting the value of the the `SELINUXTYPE` directive to `targeted` or `mls`.

2.4 Customizing SELinux Policies

You can customize an SELinux policy by enabling or disabling the members of a set of boolean values. Any changes that you make take effect immediately and do not require a reboot.

To display all of the boolean values and their descriptions, use the following command:

```
semanage boolean -l
SELinux boolean                State  Default Description

ftp_home_dir                   (off  ,  off)
Determine whether ftpd can read and write files in user home directories.
smartmon_3ware                 (off  ,  off)
Determine whether smartmon can support devices on 3ware controllers.
mpd_enable_homedirs            (off  ,  off)
Determine whether mpd can traverse user home directories.
...
```

You can use the `getsebool` and `setsebool` commands to display and set the value of a specific boolean.

```
getsebool boolean
sudo setsebool boolean on|off
```

The following example shows how you to display and set the value of the `ftp_home_dir` boolean:

```
getsebool ftp_home_dir
ftp_home_dir --> off
sudo setsebool ftp_home_dir on
getsebool ftp_home_dir
ftp_home_dir --> on
```

To toggle the value of a boolean, use the `togglesebool` command, as shown in the following example:

```
sudo togglesebool ftp_home_dir
ftp_home_dir: inactive
```

To make the value of a boolean persist across reboots, specify the `-P` option to `setsebool`, for example:

```
sudo setsebool -P ftp_home_dir on
getsebool ftp_home_dir
ftp_home_dir --> on
```

Chapter 3 Administering SELinux Security Context

Table of Contents

Under SELinux, all file systems, files, directories, devices, and processes have an associated security context. For files, SELinux stores a context label in the extended attributes of the file system. The context contains additional information about a system object: the SELinux user, their role, their type, and the security level. SELinux uses this context information to control access by processes, Linux users, and files. This chapter provides information about how to administer SELinux Security Context

You can specify the `-Z` option with certain commands (`ls`, `ps`, and `id`) to display the SELinux context by using the following syntax:

```
SELinux user:Role:Type:Level
```

`SELinux user` — An SELinux user account compliments a regular Linux user account. SELinux maps every Linux user to an SELinux user identity that is used in the SELinux context for the processes in a user session.

`Role` — In the Role-Based Access Control (RBAC) security model, a role acts as an intermediary abstraction layer between SELinux process domains or file types and an SELinux user. Processes run in specific SELinux domains, and file system objects are assigned SELinux file types. SELinux users are authorized to perform specified roles, and roles are authorized for specified SELinux domains and file types. A user's role determines which process domains and file types he or she can access, and hence, which processes and files, he or she can access.

`Type` — A type defines an SELinux file type or an SELinux process domain. Processes are separated from each other by running in their own domains. This separation prevents processes from accessing files that other processes use, and prevents processes from accessing other processes. The SELinux policy rules define the access that process domains have to file types and to other process domains.

`Level` — A level is an attribute of Multilevel Security (MLS) and Multicategory Security (MCS). An MLS range is a pair of sensitivity levels, written as *`low_level-high_level`*. The range can be abbreviated as *`low_level`* if the levels are identical. For example, `s0` is the same as `s0-s0`. Each level has an optional set of security categories to which it applies. If the set is contiguous, it can be abbreviated. For example, `s0:c0.c3` is the same as `s0:c0,c1,c2,c3`.

3.1 Displaying SELinux User Mapping

Display the mapping between SELinux and Linux user accounts by using the `semanage` command:

```
semanage login -l
```

```
Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
```

By default, SELinux maps Linux users other than `root` and the default system-level user, `system_u`, to the Linux `__default__` user, and in turn to the SELinux `unconfined_u` user. The MLS/MCS Range is the security level used by Multilevel Security (MLS) and Multicategory Security (MCS).

3.2 Displaying SELinux Context Information

To display the context information that is associated with files, use the `ls -Z` command:

```
ls -Z
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 config
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 initial-setup-ks.cfg
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 jail
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 team0.cfg
```

To display the context information that is associated with a specified file or directory:

```
ls -Z /etc/selinux/config
-rw-r--r--. root root system_u:object_r:selinux_config_t:s0 /etc/selinux/config
```

To display the context information that is associated with processes, use the `ps -Z` command:

```
ps -Z
LABEL                                             PID  TTY   TIME     CMD
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 3038 pts/0 00:00:00 su
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 3044 pts/0 00:00:00 bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 3322 pts/0 00:00:00 ps
```

To display the context information that is associated with the current user, use the `id -Z` command:

```
id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

3.3 Changing the Default File Type

Under some circumstances, you might need to change the default file type for a file system hierarchy. For example, you might want to use a `DocumentRoot` directory other than `/var/www/html` with `httpd`.

To change the default file type of the directory hierarchy `/var/webcontent` to `httpd_sys_content_t`:

1. Use the `semanage` command to define the file type `httpd_sys_content_t` for the directory hierarchy:

   ```
   sudo /usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/var/webcontent(/.*)?"
   ```

   This command adds the following entry to the file `/etc/selinux/targeted/contexts/files/file_contexts.local`:

   ```
   /var/webcontent(/.*)?     system_u:object_r:httpd_sys_content_t:s0
   ```

2. Use the `restorecon` command to apply the new file type to the entire directory hierarchy.

   ```
   sudo /sbin/restorecon -R -v /var/webcontent
   ```

3.4 Restoring the Default File Type

To restore the default file type of the directory hierarchy `/var/webcontent` after previously changing it to `httpd_sys_content_t`:

1. Use the `semanage` command to delete the file type definition for the directory hierarchy from the file `/etc/selinux/targeted/contexts/files/file_contexts.local`:

```
sudo /usr/sbin/semanage fcontext -d "/var/webcontent(/.*)?"
```

2. Use the `restorecon` command to apply the default file type to the entire directory hierarchy.

```
sudo /sbin/restorecon -R -v /var/webcontent
```

3.5 Relabelling a File System

If you see an error message that contains the string `file_t`, the problem usually lies with a file system having an incorrect context label.

To relabel a file system by using the command line:

1. Create the file `/.autorelabel` and reboot the system.
2. Run the `fixfiles onboot` command, then reboot the system.

Chapter 4 Administering SELinux Users

Table of Contents

As described in Chapter 3, *Administering SELinux Security Context*, each SELinux user account compliments a regular Oracle Linux user account. SELinux maps every Oracle Linux user to an SELinux user identity that is used in the SELinux context for the processes in a user session. This chapter provides tasks for administering SELinux users.

SELinux users form part of a SELinux policy that is authorized for a specific set of roles and for a specific MLS (Multi-Level Security) range, and each Oracle Linux user is mapped to an SELinux user as part of the policy. As a result, Linux users inherit the restrictions and security rules and mechanisms placed on SELinux users. To define the roles and levels of users, the mapped SELinux user identity is used in the SELinux context for processes in a session.

4.1 Displaying Mappings Between SELinux Users and Oracle Linux Users

To display user mapping between SELinux and Oracle Linux user accounts, use the following command:

```
semanage login -l
Login Name   SELinux User     MLS/MCS Range
_default_    unconfined_u     s0-s0:c0.c1023
root         unconfined_u     s0-s0:c0.c1023
system_u     system_u         s0-s0:c0.c1023
```

The MLS/MCS Range column displays the level that is used by MLS and MCS.

By default, Oracle Linux users are mapped to the SELinux user `unconfined_u`.

You can configure SELinux to confine Oracle Linux users by mapping them to SELinux users in confined domains, which have predefined security rules and mechanisms. These security rules and mechanisms are described in the following table.

SELinux User	SELinux Domain	Permit Running su and sudo?	Permit Network Access?	Permit Logging in Using X Window System?	Permit Executing Applications in `$HOME` and `/tmp`?
`guest_u`	`guest_t`	No	Yes	No	No
`staff_u`	`staff_t`	`sudo`	Yes	Yes	Yes
`system_u`	`ssystem_t`	Yes	Yes	Yes	Yes
`user_u`	`user_t`	No	Yes	Yes	Yes
`xguest_x`	`xguest_t`	No	Firefox only	Yes	No

To map an Oracle Linux user *`oluser`* to an SELinux user, such as `user_u`, use the `semanage` command:

```
semanage login -a -s user_u oluser
```

4.2 Configuring the Behavior of Application Execution for Users

To help prevent flawed or malicious applications from modifying a user's files, you can use booleans to specify whether users are permitted to run applications in directories for which they have write access, such as the user's home directory hierarchy and `/tmp`.

To enable Oracle Linux users in the `guest_t` and `xguest_t` domains to execute applications in directories to which they have write access:

```
sudo setsebool -P allow_guest_exec_content on
sudo setsebool -P allow_xguest_exec_content on
```

The following example shows how to prevent users in the `staff_t` and `user_t` domains from executing applications in directories to which they have write access:

```
sudo setsebool -P allow_staff_exec_content off
sudo setsebool -P allow_user_exec_content off
```

For more information, see Section 2.4, “Customizing SELinux Policies”.

Chapter 5 Troubleshooting Access-Denial Messages

This chapter provides information about how to troubleshoot access-denial messages.

The decisions that SELinux makes about allowing and denying access are stored in the Access Vector Cache (AVC). If the auditing service (`auditd`) is not running, SELinux logs AVC denial messages to `/var/log/messages`. Otherwise, the messages are logged to the `/var/log/audit/audit.log` file. If the `setroubleshootd` daemon is running, easier-to-read versions of the denial messages are also written to `/var/log/messages`.

If you have installed the `setroubleshoot` and `setroubleshoot-server` packages, the `auditd` and `setroubleshoot` services are running. If you are using the X Window System, you can also use the `sealert -b` command to run the SELinux Alert Browser, which displays information about SELinux AVC denials. To view the details of the alert, click **Show**. To view a recommended solution, click **Troubleshoot**.

The following example shows how you would search the `/var/log/audit/audit.log` file for messages containing the string `denied`:

```
grep denied /var/log/audit/audit.log
type=AVC msg=audit(1364486257.632:26178): avc:  denied  { read } for
pid=5177 comm="httpd" name="index.html" dev=dm-0 ino=396075
scontext=unconfined_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:acct_data_t:s0 tclass=file
```

The main causes of access-denial problems include the following:

- Context labels for an application or file are incorrect.

  A solution might be to change the default file type of the directory hierarchy. For example, change the default file type from `/var/webcontent` to `httpd_sys_content_t`:

  ```
  sudo /usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/var/webcontent(/.*)?"
  sudo /sbin/restorecon -R -v /var/webcontent
  ```

- A Boolean that configures a security policy for a service is set incorrectly.

  A solution might be to change the value of a Boolean. For example, allow users' home directories to be browsable by turning on `httpd_enable_homedirs`:

  ```
  sudo setsebool -P httpd_enable_homedirs on
  ```

- A service attempts to access a port to which a security policy does not allow access.

  If the service's use of the port is valid, a solution is to use `semanage` to add the port to the policy configuration. For example, allow the Apache HTTP server to listen on port 8000:

  ```
  sudo semanage port -a -t http_port_t -p tcp 8000
  ```

- An update to a package causes an application to behave in a way that breaks an existing security policy.

  You can use the `audit2allow -w -a` command to view the reason why an access denial occurred.

  If you then run the `audit2allow -a -M` *`module`* command, it creates a type enforcement (`.te`) file and a policy package (`.pp`) file. You can use the policy package file with the `semodule -i` *`module`*`.pp` command to stop the error from reoccurring. This procedure is usually intended to allow package updates to function until an amended policy is available. If used incorrectly, it can create potential security holes on your system.