43.2. Introduction to SELinux

Security-Enhanced Linux (SELinux) is a security architecture integrated into the 2.6.**x** kernel using the *Linux Security Modules* (LSM). It is a project of the United States National Security Agency (NSA) and the SELinux community. SELinux integration into Red Hat Enterprise Linux was a joint effort between the NSA and Red Hat.

43.2.1. SELinux Overview

SELinux provides a flexible *Mandatory Access Control* (MAC) system built into the Linux kernel. Under standard Linux *Discretionary Access Control* (DAC), an application or process running as a user (UID or SUID) has the user's permissions to objects such as files, sockets, and other processes. Running a MAC kernel protects the system from malicious or flawed applications that can damage or destroy the system.

SELinux defines the access and transition rights of every user, application, process, and file on the system. SELinux then governs the interactions of these entities using a security policy that specifies how strict or lenient a given Red Hat Enterprise Linux installation should be.

On a day-to-day basis, system users will be largely unaware of SELinux. Only system administrators need to consider how strict a policy to implement for their server environment. The policy can be as strict or as lenient as needed, and is very finely detailed. This detail gives the SELinux kernel complete, granular control over the entire system.

The SELinux Decision Making Process

When a subject, (for example, an application), attempts to access an object (for example, a file), the policy enforcement server in the kernel checks an *access vector cache* (AVC), where subject and object permissions are cached. If a decision cannot be made based on data in the AVC, the request continues to the security server, which looks up the *security context* of the application and the file in a matrix. Permission is then granted or denied, with an `avc: denied` message detailed in `/var/log/messages` if permission is denied. The security context of subjects and objects is applied from the installed policy, which also provides the information to populate the security server's matrix.

Refer to the following diagram:

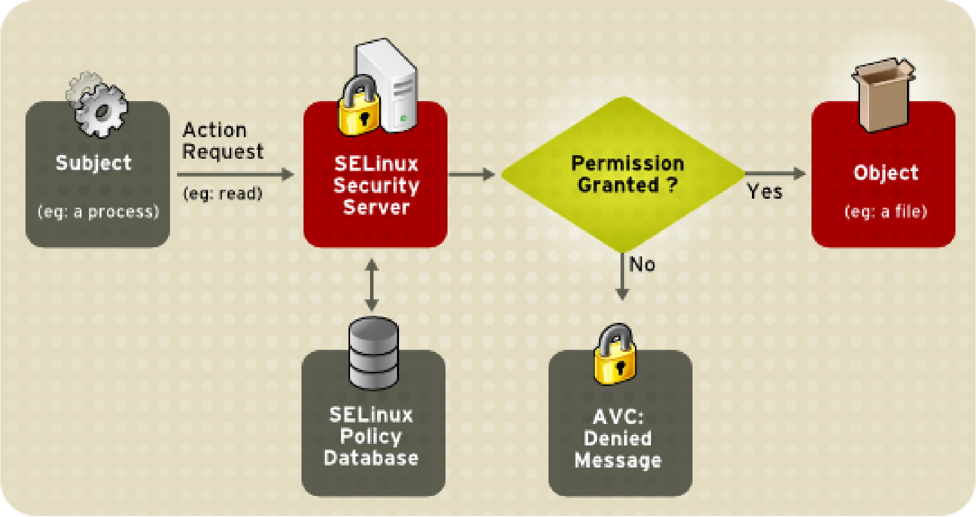

Figure 43.1. SELinux Decision Process

SELinux Operating Modes

Instead of running in *enforcing* mode, SELinux can run in *permissive* mode, where the AVC is checked and denials are logged, but SELinux does not enforce the policy. This can be useful for troubleshooting and for developing or fine-tuning SELinux policy.

For more information about how SELinux works, refer to Section 43.2.3, "Additional Resources".

43.2.2. Files Related to SELinux

The following sections describe SELinux configuration files and related file systems.

43.2.2.1. The SELinux Pseudo-File System

The **/selinux/** pseudo-file system contains commands that are most commonly used by the kernel subsystem. This type of file system is similar to the **/proc/** pseudo-file system.

Administrators and users do not normally need to manipulate this component.

The following example shows sample contents of the **/selinux/** directory:

```
-rw-rw-rw-  1 root root 0 Sep 22 13:14 access
dr-xr-xr-x  1 root root 0 Sep 22 13:14 booleans
--w-------  1 root root 0 Sep 22 13:14 commit_pending_bools
-rw-rw-rw-  1 root root 0 Sep 22 13:14 context
-rw-rw-rw-  1 root root 0 Sep 22 13:14 create
--w-------  1 root root 0 Sep 22 13:14 disable
-rw-r--r--  1 root root 0 Sep 22 13:14 enforce
```

```
-rw-------  1 root root 0 Sep 22 13:14 load
-r--r--r--  1 root root 0 Sep 22 13:14 mls
-r--r--r--  1 root root 0 Sep 22 13:14 policyvers
-rw-rw-rw-  1 root root 0 Sep 22 13:14 relabel
-rw-rw-rw-  1 root root 0 Sep 22 13:14 user
```

For example, running the `cat` command on the `enforce` file reveals either a `1` for enforcing mode or `0` for permissive mode.

43.2.2.2. SELinux Configuration Files

The following sections describe SELinux configuration and policy files, and related file systems located in the `/etc/` directory.

43.2.2.2.1. The `/etc/sysconfig/selinux` Configuration File

There are two ways to configure SELinux under Red Hat Enterprise Linux: using the **Security Level Configuration Tool** (`system-config-selinux`), or manually editing the configuration file (`/etc/sysconfig/selinux`).

The `/etc/sysconfig/selinux` file is the primary configuration file for enabling or disabling SELinux, as well as for setting which policy to enforce on the system and how to enforce it.

> **Note**
>
> The `/etc/sysconfig/selinux` contains a symbolic link to the actual configuration file, `/etc/selinux/config`.

The following explains the full subset of options available for configuration:

`SELINUX=`*`enforcing`*`|`*`permissive`*`|`*`disabled`* — Defines the top-level state of SELinux on a system.

- `enforcing` — The SELinux security policy is enforced.

- `permissive` — The SELinux system prints warnings but does not enforce policy.

  This is useful for debugging and troubleshooting purposes. In permissive mode, more denials are logged because subjects can continue with actions that would otherwise be denied in enforcing mode. For example, traversing a directory tree in permissive mode produces `avc: denied` messages for every directory level read. In enforcing mode, SELinux would have stopped the initial traversal and kept further denial messages from occurring.

- `disabled` — SELinux is fully disabled. SELinux hooks are disengaged from the kernel and the pseudo-file system is unregistered.

  > **Tip**

Actions made while SELinux is disabled may result in the file system no longer having the correct security context. That is, the security context defined by the policy. The best way to relabel the file system is to create the flag file `/.autorelabel` and reboot the machine. This causes the relabel to occur very early in the boot process, before any processes are running on the system. Using this procedure means that processes can not accidentally create files in the wrong context or start up in the wrong context.
It is possible to use the `fixfiles relabel` command prior to enabling SELinux to relabel the file system. This method is not recommended, however, because after it is complete, it is still possible to have processes potentially running on the system in the wrong context. These processes could create files that would also be in the wrong context.

Note

Additional white space at the end of a configuration line or as extra lines at the end of the file may cause unexpected behavior. To be safe, remove unnecessary white space.

`SELINUXTYPE=targeted|strict` — Specifies which policy SELinux should enforce.

- `targeted` — Only targeted network daemons are protected.

Important

The following daemons are protected in the default targeted policy: `dhcpd`, `httpd (apache.te)`, `named`, `nscd`, `ntpd`, `portmap`, `snmpd`, `squid`, and `syslogd`. The rest of the system runs in the unconfined_t domain. This domain allows subjects and objects with that security context to operate using standard Linux security.
The policy files for these daemons are located in `/etc/selinux/targeted/src/policy/domains/program`. These files are subject to change as newer versions of Red Hat Enterprise Linux are released.

Policy enforcement for these daemons can be turned on or off, using Boolean values controlled by the **Security Level Configuration Tool** (`system-config-selinux`).

Setting a Boolean value for a targeted daemon to `0` (zero) disables policy transition for the daemon. For example, you can set `dhcpd_disable_trans` to `0` to prevent `init` from transitioning `dhcpd` from the unconfined_t domain to the domain specified in `dhcpd.te`.

Use the `getsebool -a` command to list all SELinux booleans. The following is an example of using the `setsebool` command to set an SELinux boolean. The `-P` option makes the change permanent. Without this option, the boolean would be reset to `1` at reboot.

```
setsebool -P dhcpd_disable_trans=0
```

- **strict** — Full SELinux protection, for all daemons. Security contexts are defined for all subjects and objects, and every action is processed by the policy enforcement server.

SETLOCALDEFS=0|1 — Controls how local definitions (users and booleans) are set. Set this value to 1 to have these definitions controlled by **load_policy** from files in **/etc/selinux/*<policyname>***. or set it to 0 to have them controlled by **semanage**.

> **Caution**
>
> You should not change this value from the default (0) unless you are fully aware of the impact of such a change.

43.2.2.2.2. The /etc/selinux/ Directory

The **/etc/selinux/** directory is the primary location for all policy files as well as the main configuration file.

The following example shows sample contents of the **/etc/selinux/** directory:

```
-rw-r--r--  1 root root  448 Sep 22 17:34 config
drwxr-xr-x  5 root root 4096 Sep 22 17:27 strict
drwxr-xr-x  5 root root 4096 Sep 22 17:28 targeted
```

The two subdirectories, **strict/** and **targeted/**, are the specific directories where the policy files of the same name (that is, **strict** and **targeted**) are contained.

43.2.2.3. SELinux Utilities

The following are some of the commonly used SELinux utilities:

/usr/sbin/setenforce — Modifies in real-time the mode in which SELinux runs.

For example:

setenforce 1 — SELinux runs in enforcing mode.

setenforce 0 — SELinux runs in permissive mode.

To actually disable SELinux, you need to either specify the appropriate **setenforce** parameter in **/etc/sysconfig/selinux** or pass the parameter **selinux=0** to the kernel, either in **/etc/grub.conf** or at boot time.

/usr/sbin/sestatus -v — Displays the detailed status of a system running SELinux. The following example shows an excerpt of **sestatus -v** output:

```
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 21
Policy from config file:        targeted

Process contexts:
Current context:                user_u:system_r:unconfined_t:s0
Init context:                   system_u:system_r:init_t:s0
/sbin/mingetty                  system_u:system_r:getty_t:s0
```

`/usr/bin/newrole` — Runs a new shell in a new context, or role. Policy must allow the transition to the new role.

> **Note**
>
> This command is only available if you have the **`policycoreutils-newrole`** package installed, which is required for the strict and MLS policies.

`/sbin/restorecon` — Sets the security context of one or more files by marking the extended attributes with the appropriate file or security context.

`/sbin/fixfiles` — Checks or corrects the security context database on the file system.

Refer to the man page associated with these utilities for more information.

Refer to the **`setools`** or **`policycoreutils`** package contents for more information on all available binary utilities. To view the contents of a package, use the following command:

```
rpm -ql <package-name>
```

43.2.3. Additional Resources

Refer to the following resources for more detailed information on SELinux.

43.2.3.1. Installed Documentation

`/usr/share/doc/setools-<`*`version-number`*`>/` All documentation for utilities contained in the **`setools`** package. This includes all helper scripts, sample configuration files, and documentation.

43.2.3.2. Useful Websites

http://www.nsa.gov/selinux/ Homepage for the NSA SELinux development team. Many resources are available in HTML and PDF formats. Although many of these links are not SELinux specific, some concepts may apply.

http://fedora.redhat.com/docs/ Homepage for the Fedora documentation project, which contains Fedora Core specific materials that may be more timely, since the release cycle is much shorter.

http://selinux.sourceforge.net Homepage for the SELinux community.

Prev
Chapter 43. Security and SELinux

Up

Home

Next
43.3. Brief Background and History of SELinux