SElinux: Advanced sestatus usage

I learned something new today! Apparently, **sestatus command** can report security contexts of the key system files – really neat for quickly recognising possible security compromise.

Files and processes in /etc/sestatus.conf

The way this works is you must use the **/etc/sestatus.conf** file which contains list of files and list of processes that are checked for **SElinux** contexts. These are the most common security attack vectors, so SElinux notes them and helps you to quickly confirm their contexts using **sestatus -v** command.

VERY IMPORTANT: at this stage **sestatus command** does NOT highlight or warn you about any non-standard contextual changes. So the only thing is does is show you all the important files you selected and report their current contexts – if some of these have been changed, the task of recognising or fixing this is still on you.

You can add any files and process you like here, but here's the default list in RHEL8:

```
[greys@rhel8 ~]$ cat /etc/sestatus.conf
[files]
/etc/passwd
/etc/shadow
/bin/bash
/bin/login
/bin/sh
/sbin/agetty
/sbin/init
/sbin/mingetty
/usr/sbin/sshd
```

```
/lib/libc.so.6
/lib/ld-linux.so.2
/lib/ld.so.1

[process]
/sbin/mingetty
/sbin/agetty
/usr/sbin/sshd
```

Files and processes contexts with sestatus

```
[greys@rhel8 ~]$ sestatus -v
SELinux status: enabled
SELinuxfs mount: /sys/fs/selinux.png
SELinux root directory: /etc/selinux.png
Loaded policy name: targeted
Current mode: enforcing
Mode from config file: enforcing
Policy MLS status: enabled
Policy deny_unknown status: allowed
Memory protection checking: actual (secure)
Max kernel policy version: 31

Process contexts:
Current context: unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
Init context: system_u:system_r:init_t:s0

File contexts:
Controlling terminal: unconfined_u:object_r:user_devpts_t:s0
/etc/passwd system_u:object_r:passwd_file_t:s0
/etc/shadow system_u:object_r:shadow_t:s0
/bin/bash system_u:object_r:shell_exec_t:s0
/bin/login system_u:object_r:login_exec_t:s0
/bin/sh system_u:object_r:bin_t:s0 ->
system_u:object_r:shell_exec_t:s0
/sbin/agetty system_u:object_r:getty_exec_t:s0
/sbin/init system_u:object_r:bin_t:s0 ->
system_u:object_r:init_exec_t:s0
```

```
/usr/sbin/sshd system_u:object_r:sshd_exec_t:s0
```

See Also

- Disable SElinux
- Check SElinux Status
- Security commands in Unix

SElinux Status

SELinux

This post shows you how to confirm current **SElinux status** before you decide to disable SELinux.

SElinux Enforcing vs Permissive

The most burning question usually is: does my RedHat/CentOS Linux enforce SELinux (and prevent some of my applications from running out of the box) or is it in the permissive state (which means it logs security concerns but doesn’t block anything from running).

Answering this is very easy with the help of the **getenforce command**:

```
[greys@rhel8 ~]$ getenforce
Enforcing
```

SElinux status with sestatus

If you're more curious about the way SELinux is configured, then **sestatus command** will be much more useful:

```
[greys@rhel8 ~]$ sestatus
SELinux status: enabled
SELinuxfs mount: /sys/fs/selinux.png
SELinux root directory: /etc/selinux.png
Loaded policy name: targeted
Current mode: enforcing
Mode from config file: enforcing
Policy MLS status: enabled
Policy deny_unknown status: allowed
Memory protection checking: actual (secure)
Max kernel policy version: 31
```

How to read the sestatus output

Although the output of **sestatus** is fairly standard, you'll appreciate how useful it is once you start making changes to your SELinux policies.

- **Loaded policy name** is useful because you can make SELinux load a strict policy as well, and it's important to understand which one is currently in use.
- **Current mode**: will confirm if SELinux is running in enforcing or permissive mode.
- **Policy MLS status**: must research more! I know MLS is Multi Level Security, but need to understand why it's separate option here.
- **Memory protection checking** – must come back to this as I'm not finding enough information. This is a flag confirming that SElinux still protects certain memory access syscalls in your Linux.

See Also

- How To: Disable SELinux
- How To: Enable SELinux
- **SELinux Reference**