Implementing SELinux



Andrew MallettLinux Author and Trainer

@theurbanpenguin www.theurbanpenguin.com



Module Overview



Understanding SELinux modes

Reading and changing modes

Installing SELinux tools

Understanding the SELinux label and type

Changing file contexts

Restoring file contexts

Debug SELinux violations



SELinux



SELinux is a MAC, Mandatory Access Control, security system originally developed by the National Security Agency



Much of the Kernel module development has been run by both the NSA and Red Hat



Has been a part of Red Hat Enterprise Linux since version RHEL 4



Many services run with root access

SELinux controls access even to processes running as root



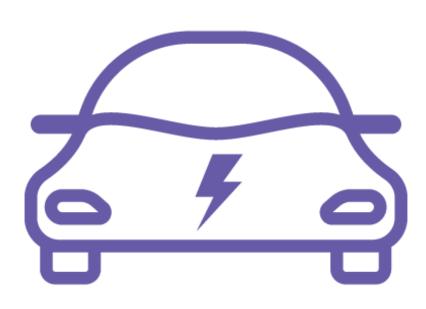
Understanding Security



File system security needs to be adjusted to allow correct access to directories



Firewalls need to be adjusted to control correct access to the network



SELinux is a super-car that used correctly will protect your system, and yes, it needs to be adjusted

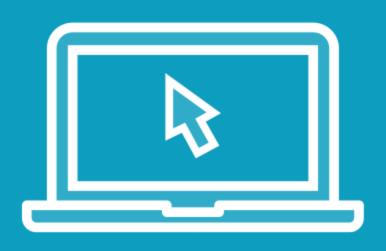


SELinux Modes

SELinux has three operating modes:

- 1. Enforcing: Rules are enforced and violations logged
- 2. Permissive: No rule enforcement, violations logged
- 3. Disabled: SELinux not operational and no logging





Reading the current mode of SELinux

- with getenforce
- with sestatus
- direct from /etc/selinux/config

Changing the SELinux mode



sudo yum install -y policycoreutils
setools setools-console setroubleshoot

Installing Tools

Adding additional tools will help you manage SELinux



```
# getsebool -a
# semanage boolean --list
# setsebool secure_mode_policyload on (-P)
```

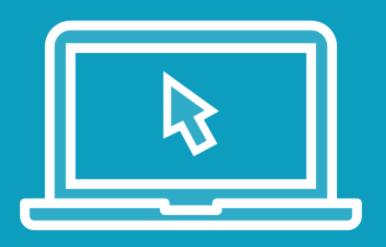
Preventing Runtime Changes to Mode

An administrator may quickly change the SELinux mode to allow something to happen that is not permitted

Setting the Boolean will require a reboot of the system to change the SELinux mode

Use the option -P to persist the change





Preventing runtime changes to the mode



SELinux Type Enforcement

In the main, SELinux works with something called type enforcement. The SELinux type of a source must be compatible with the target SELinux type. If we need to customize content or if we make erroneous changes, operations may not work



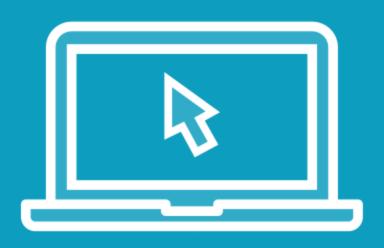
List Contexts

In the main the targeted SELinux policy works with the context of processes, ports and files. The option -Z displays the context with most tools. Processes need to be authorized to access resources such as ports and files

```
$ sudo -i
# ls -Z /etc/shadow
# chcon -t admin_home_t /etc/shadow
# chage -l vagrant
# ausearch -m avc -ts recent
# tail /var/log/messages
# sealert -l <alert-id>
# restorecon -v /etc/shadow
```

/etc/shadow

The /etc/shadow file is a critical system file. Processes can only access this if the correct context is set on the file and matches rules that allow the process access. Taking care not to break your system we can demonstrate SELinux at work.



Managing SELinux Contexts

- take care with your own systems

```
$ man semanage-fcontext
$ sudo ls /etc/selinux/targeted/contexts/files
$ sudo mkdir /staff
$ sudo semanage fcontext -a -e /home /staff
$ sudo restorecon -v /staff
$ ls -ldZ /staff
$ sudo useradd -m -d /staff/u1 u1
$ sudo ls -ldZ /staff/u1
```

File Contexts

The file contexts used by restorecon are part of the current SELinux policy. Relocating user home directories, for example, we can create the top-level directory and store the definition by cloning the configuration of the existing home. This ensures the correct SELinux context on user home directories created below staff.



Setting new home directories



Summary



In this module we have introduced SELinux Modes

- enforcing
- permissive
- disabled
- boolean to prevent runtime changes
- purpose of SELinux
- setting file contexts
- searching policy violations



