

An SELinux Primer

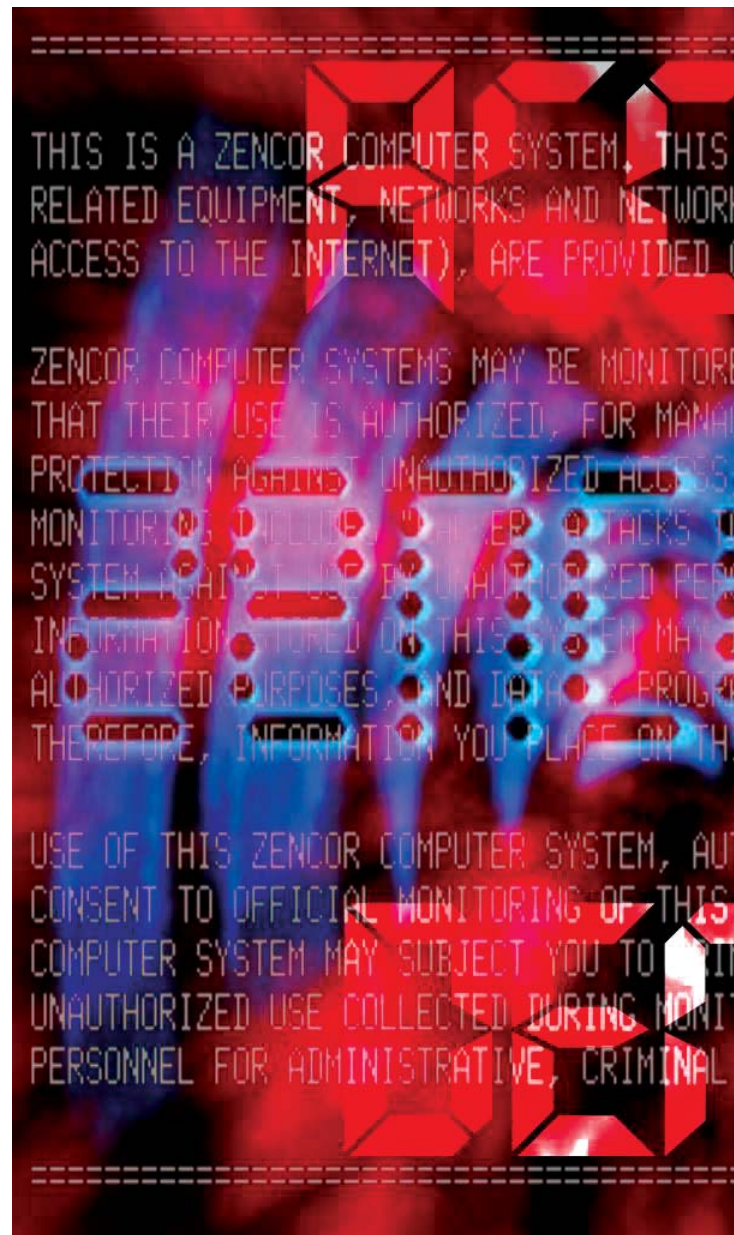
Beginning as a development project, SELinux is now an integral component of RHEL and other major Linux distributions. This primer will help you master this technology that administrators worth their salt should know.

In the world of Linux, SELinux is the new buzzword. Most operating systems use access controls to limit the access a user/process has on other parts of the system such as files, devices, sockets, ports and other processes (called objects in SELinux). The two main types are Discretionary Access Control (DAC) and Mandatory Access Control (MAC). SELinux supplements the traditional DAC mechanism of Linux with MAC. Under SELinux, programs are run inside a sandbox and follow the principle of least privilege, in which programs are limited to a set of necessary operations.

Discretionary Access Control (DAC)

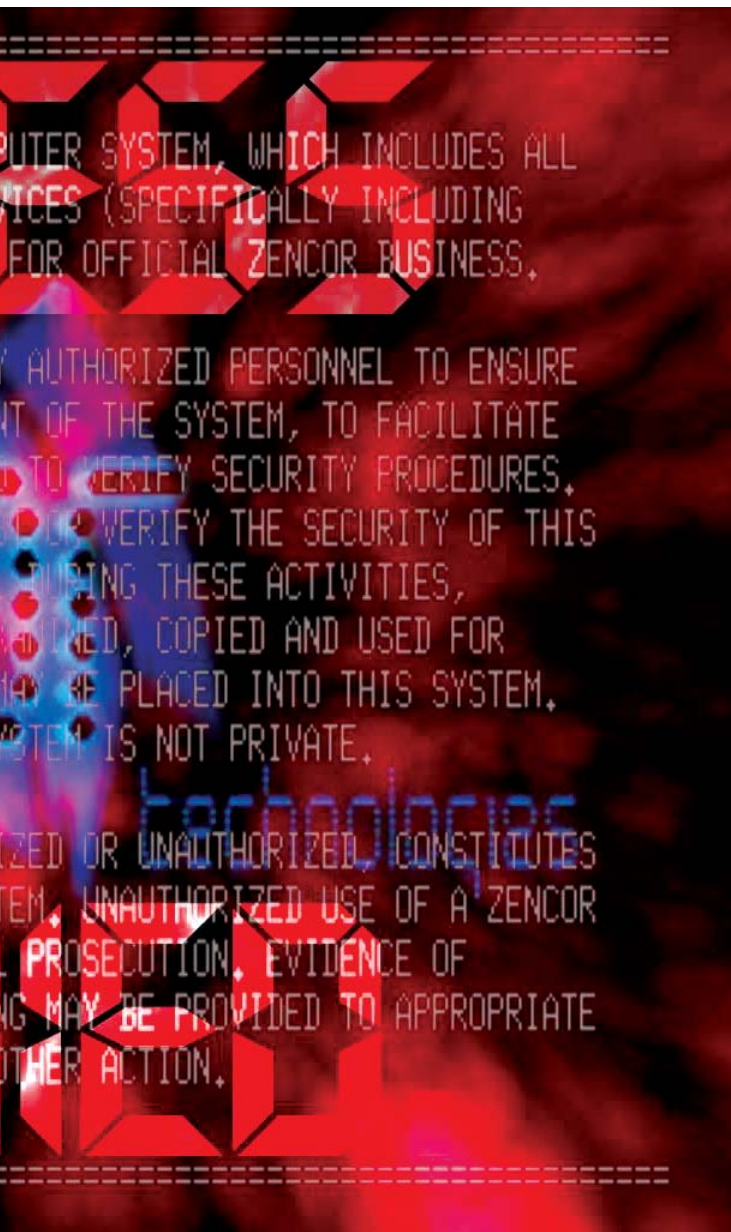
Discretionary Access Control (DAC) is the standard mechanism for Linux security. Under DAC, all processes run with an associated user and group. That process has access to all files and directories that the user and group can access. Thus an errant process could destroy all files that belong to the user!

Under DAC, ownership of a file object provides potentially crippling or risky control over the object. A user can expose a file or directory to a security or confidentiality breach with a misconfigured *chmod* command and an unexpected propagation of access rights. A process started by that user, such as a CGI script, can do anything to the files owned by the user. A compromised Apache HTTP server can perform any operation on files in the Apache group. Malicious or broken software can have root-level access to the entire system, either by running as a root process or using *setuid* or *setgid*.



Mandatory Access Control (MAC)

Also called non-discretionary access control, this framework allows you to define permissions for how all processes (called subjects) interact with other objects. This is done through a security policy defined by the administrator, over all processes and objects. These processes and objects are controlled through the kernel, and security decisions are made on all available information rather than just the user identity. With this model, a process can be granted just the permissions needed to be functional. This follows the principle of least privilege. Under MAC, for example, users who have exposed their data using *chmod* are protected by the fact that their data is a kind only associated with user home directories, and



confined processes cannot touch those files without permission and purpose written into the policy. The two types of access controls are discussed in Figure 1.

SELinux history

SELinux was originally a development project from the National Security Agency (NSA), Secure Computing Corporation (SCC) and others. It is an implementation of the Flask operating system security architecture. As a step in its evolution, SELinux was integrated into the Linux kernel using the Linux Security Modules (LSM) framework. SELinux motivated the creation of LSM, at the suggestion of Linus Torvalds, who wanted a modular approach to security instead of just accepting SELinux into the kernel. SELinux is now a standard component of RHEL

and non-commercial distributions like Fedora, Debian GNU/Linux, Gentoo Linux, etc. Refer to [<http://www.nsa.gov/selinux>] for more definitive information on the history of SELinux.

SELinux architecture

SELinux adds another layer of access control permissions on top of standard file permissions and ACLs, which are defined by the system security policy. Every object (files and other items) and every subject (process) has a security context, with three attributes—a user identity, a role and a type. Collectively, these attributes limit the authority of the subject over the object. Typically, the security context is displayed as a colon-separated triplet in this format:

```
user_identity:role:type
```

To view the security context information associated with objects, you may use commands with the `-Z` option:

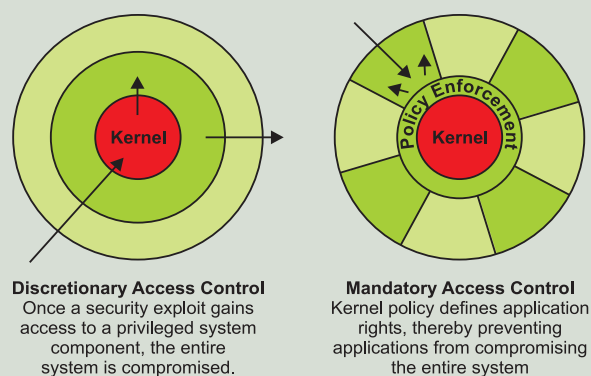
```
[root@server1 /boot]# ls -Z vmlinuz-2.6.9-5.0.3.EL
-rw-r--r-- root root system_u:object_r:boot_t
vmlinuz-2.6.9-5.0.3.EL

[root@server1 /boot]# id -Z
root:system_r:unconfined_t

[root@server1 boot]# ps -AZ | grep httpd
root:system_r:httpd_t 17994 ? 00:00:01 httpd
```

User identity indicates the SELinux user account that is associated with a subject or object. SELinux user identities are different from UNIX identities. They are applied as part of the security label and can be changed in real-time under limited conditions. SELinux uses its own database and a mapping that associates SELinux user identities with Linux users. Roles define a set of permissions a user can be granted. A user can reside only in a single role at any given time. Types or domains are the primary security attributes used for making authorisation decisions.

FIGURE 1: THE TWO TYPES OF ACCESS CONTROL



The SELinux Policy defined in `/etc/selinux/targeted/policy/` controls these important aspects:

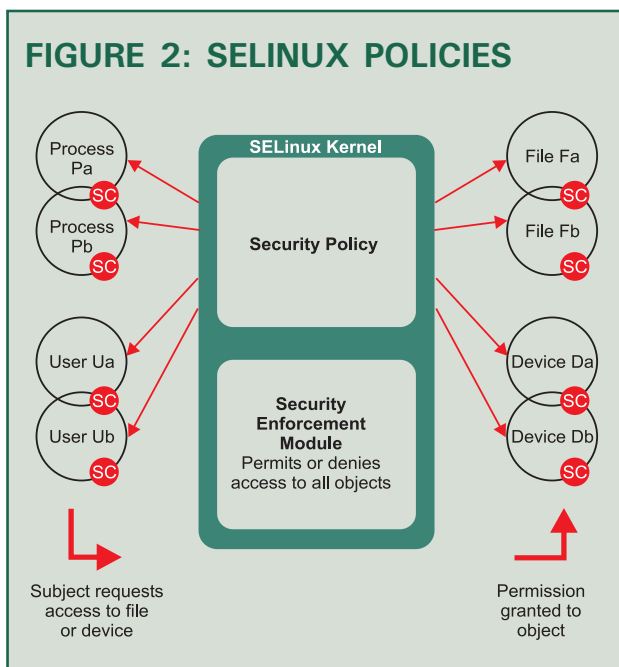
- The particular roles that identities can use
- Which domains roles can enter
- The types that domains can access

The SELinux policy is highly configurable. For Red Hat Enterprise Linux 4, Red Hat supports a single policy—the targeted policy. Under this policy, every subject and object runs in the `unconfined_t` domain except for the specific targeted daemons. The objects on the system that are in the `unconfined_t` domain are allowed by SELinux to have no restrictions and fall back to using standard Linux security, which is DAC. This policy is flexible enough to fit into enterprise infrastructures. The daemons that are part of the targeted policy run in their own domains and are restricted in every operation they perform on the system. This way, daemons that are broken or exploited are limited in the damage that they can do.

In addition to the targeted policy, Red Hat has a strict policy, which does not ship with Red Hat Enterprise Linux. In the strict policy, every subject and object is in a specific security domain, with all interactions and transitions individually considered within the policy rules. This is a much more complex environment. In fact, when logging in as a root user in the GUI, the root user will have less privileges. You'll need to enter the root password for each system configuration just like a normal user.

The policy is written in simple language created specifically for writing security policies. Policy writers use m4 macros to capture common sets of low-level rules. There are a number of m4 macros defined in the existing policy, which are of great assistance in writing new policies. These are illustrated in Figure 2.

FIGURE 2: SELINUX POLICIES



Controlling SELinux

The SELinux policy may be adjusted or disabled through a number of utilities. The easiest to use is the graphical `system-config-securitylevel` tool, which can turn SELinux off, set it to permissive mode, or set it to enforcing mode. It also allows the adjustment of “booleans” which can fine-tune the rules enforced by the policy.

When SELinux is enabled, there are two modes: permissive and enforcing. Permissive mode is the ‘warn-only’ mode. That is, it allows all processes access to the file system using the standard Discretionary Access Control, but it will log all access violations that would have been there, if SELinux had been in the enforcing mode in `/var/log/messages`. The enforcing mode allows SELinux to control access to the system using Mandatory Access Control, and thus enforces the SELinux policy. These modes can be controlled dynamically using the `setenforce` command, and can be permanently set in the file `/etc/sysconfig/selinux`.

`setenforce` is a command-line tool that allows SELinux to the set enforcing mode or permissive mode. To completely disable SELinux, one must use `system-config-securitylevel` to pass `selinux=0` on the kernel line, or in the `/etc/sysconfig/selinux` file.

The kernel option `enforcing=0` can be passed through GRUB at the boot time to set SELinux in warn-only mode; `enforcing=1` sets enforcing mode. The `/selinux` virtual file system is similar to `/proc` and `/sys`. It presents information about the state of SELinux in the kernel to user programs like the ones above. `sestatus` shows the actual SELinux settings.

The contexts of files can be changed using the `chcon` command. It has a `-reference` option, which can be used to copy and apply the contexts from a particular file.

```
[root@server1 /var/www/html]# chcon -t httpd_sys_content_t
index.html
root@server1 /var/www/html]# chcon -reference /var/www/html
index.html
```

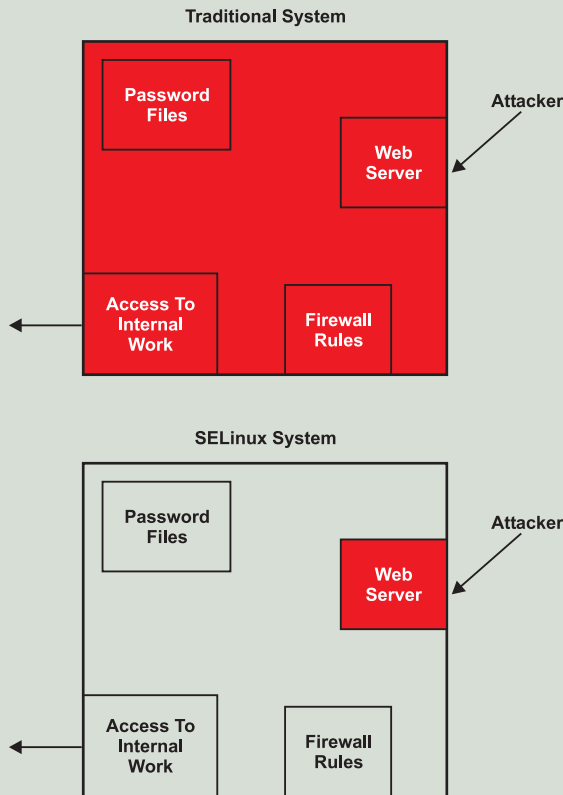
Troubleshooting SELinux

SELinux policy violations are logged to `/var/log/messages`. An error might read like what is shown below:

```
Oct 25 01:38:23 station15 kernel: audit(1109313503.808:0): avc:
denied { read } for pid=4346 exe=/usr/sbin/httpd name=joe
dev=hda2 ino=311297 scontext=root:system_r:httpd_t
tcontext=system_u:object_r:user_home_dir_t tclass=dir
```

In our language, this translates to:

PID 4346, a `/usr/sbin/httpd` process, with the context `root:system_r:httpd_t` was denied read access to a directory named `joe`, which is `inode 311297` on `/dev/hda2`, which has the context `system_u:object_r:user_home_dir_t`.


FIGURE 3: TRADITIONAL AND SELINUX SYSTEMS

Further guidelines

At this point, several questions may arise. Is the process being blocked for legitimate reasons or is it doing something inappropriate? If not, then is the target's context wrong? If so, the correct context needs to be determined and set with *chcon*. If the policy is being too strict, perhaps a "boolean" setting can be adjusted with *system-config-securitylevel* or *setsebool*. In the worst cases, perhaps SELinux can be disabled for just the affected service, or entirely.

Resources that can help troubleshoot SELinux problems include the *Red Hat Enterprise Linux 4: Red Hat SELinux Guide* on [www.redhat.com], and *Understanding and Customizing the Apache SELinux Policy for Fedora Core 3* at [fedora.redhat.com]. The source used to build the SELinux policy is included in the *selinuxtargeted-policy-sources* RPM.

REFERENCES

The Red Hat SELinux Guide: <http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/selg-preface-0011.html>
<http://www.nsa.gov/selinux> 

By: Arun Eapen. The author is a technical consultant—Global Learning Services, Red Hat India Pvt Ltd. He can be reached at aeapen@redhat.com

Want to have an edge over others?

Master LINUX



Read LINUX For You

For more info, log on to: www.linuxforu.com

ASIA'S FIRST
LINUX MAGAZINE

