



SELinux for sysadmins

Linux Day 2009 - Napoli, 24 Ottobre 2009

Gianluca Varisco
Red Hat

What is Security-enhanced Linux?

“NSA Security-enhanced Linux is a set of patches to the Linux kernel and some utilities to incorporate a strong, flexible mandatory access control (MAC) architecture into the major subsystems of the kernel. It provides an enhanced mechanism to enforce the separation of information based on confidentiality and integrity requirements, which allows threats of tampering and bypassing of application security mechanisms to be addressed and enables the confinement of damage that can be caused by malicious or flawed applications. It includes a set of sample security policy configuration files designed to meet common, general-purpose security goals.”

NSA Security-enhanced Linux Team



Discretionary Access Control (DAC)

- Standard **rwX** permissions for **user:group**
 - `-rw----- 1 root root 1404 2008-11-07 09:45 anaconda-ks.cfg`
- Generally controlled by one user; root
 - Has discretion over the system
 - Made decisions for the system
 - Little control given to users

SELinux /1

- Mandatory Access Control (MAC)
- A rule set called the ***policy*** determines how strict the control
- Processes are either **restricted** or **unconfined**
- The policy defines which resources a restricted process is allowed to access
- **Any action that is not explicitly allowed is, by default, denied**

SELinux Security Context

- All files and processes have a *security context*
- The context has several elements, depending on the security needs:
 - **user:role:type:sensitivity:category**
- **User:** root OR user_u (Processes: system_u)
- **Role:** Files -> object_r ; Processes -> system_r
- **Type:** Used by Type Enforcement to specify the nature of the data

- **ls -Z**
- **ps -Z (ps -eZ)**

```
$ ps -ZC bash,sshd
```

```
LABEL          PID TTY      TIME CMD
system_u:system_r:sshd_t:s0-s0:c0.c1023 1709 ? 00:00:00 sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 32019 pts/0 00:00:00
bash
```

SELinux: Targeted Policy

- loaded at **install time**
- Principally uses the type element for **type enforcement**
- The security context can be changed with **chcon** but it is safer to use **restorecon** (with it, the policy determines and applies the object's default context)

SELinux: Management

- Modes: **Enforcing, Permissive, Disabled**
 - `/etc/sysconfig/selinux`
 - **getenforce** and **setenforce 0 | 1**
 - **selinux=0** from GRUB
- Policy adjustments: **Booleans, file contexts, ports, etc.**
 - **system-config-selinux**
 - **getsebool** and **setsebool**
 - **semanage**
- Troubleshooting
 - **setroubleshootd**, **sealert -b** and **sealert -a**

SELinux Troubleshooting

- What is the error?
 - **/var/log/audit/audit.log (AVC denials)**
 - **sealert** analyzes denials
- Is the process doing something it should not?
- Does the target have the right context?
- Does a boolean setting need adjustment?

Real world examples

■ Share home directories through NFS

- [server]# cat /etc/exports
/home 192.168.0.0/24(rw,soft)
- [client]# cat /etc/fstab
...
server:/home /home nfs soft 1 2
...
• [client]# mount /home
Permission denied

DOH!

- Check /var/log/audit/audit.log
- **\$ getsebool -a | grep home**
ftp_home_dir --> off
httpd_enable_homedirs --> on
openvpn_enable_homedirs --> on
samba_create_home_dirs --> off
samba_enable_home_dirs --> off
spamd_enable_home_dirs --> on
use_nfs_home_dirs --> off
use_samba_home_dirs --> off
- setsebool **-P** use_nfs_home_dirs on

Real world examples /2

- **If you want httpd to send email**
 - # setsebool -P httpd_can_sendmail 1
- **Vsftp setup for users to login**
 - # setsebool -P ftp_home_dir 1
- **HTTPd is setup to listen on port 8585**
 - # semanage port -a -t http_port_t -p tcp 8585

Local policy modules generation

- The **audit2allow** utility now has the ability to build policy modules.

```
grep setsebool /var/log/audit/audit.log | audit2allow -M mysemanage
```

Generating type enforcement file: mysemanage.te

Compiling policy

```
checkmodule -M -m -o mysemanage.mod mysemanage.te
```

```
semodule_package -o mysemanage.pp -m mysemanage.mod
```

- In order to load this newly created policy package into the kernel, you are required to execute

```
semodule -i mysemanage.pp
```

Q&A