Stop disabling SElinux

By: Mehdi Mehranfarid
LPI Certified Senior Instructor
Email: mehranfarid@anisa.co.ir

Selinux, Don’t be afraid!

Kick the “disable” habit!

Traditional Unix DAC approach

DAC:

- ✓ You decided how you want to protect and share your data.
- ✓ Typical Linux distributions use DAC (**Discretionary Access Control**).
- ✓ Process with effective UID/GID.
- ✓ chmod, chown, chattr, facl
- ✓ When a user or application is compromised, security is compromised.

Mac approach

MAC:

- ✓ The system decided how the data will be shared.
- ✓ Selinux is an implementation of MAC (**Mandatory Access Control**).
- ✓ The linux kernel has full control of security.
- ✓ Only system administrator is decide what is allowed on the system.
- ✓ A policy enforced by the linux kernel on what processes and are aren’t allowed to do.
- ✓ By default, everything is denied unless specifically enabled.

Selinux

- ✓ Original authors are NSA and Redhats
- ✓ Released the first version to the open source development community under the **GNU GPL** on **December 22, 2000**.
- ✓ Merged into the mainline Linux kernel 2.6.0-test3, released on 8 August 2003.
- ✓ Infested by jargons:

Policies, contex, lable, role, type, sensisivity level, booleans, oh my God.

- ✓ SELinux is a set of kernel modifications (LSM Modules) and user-space tools that have been added to various Linux distributions.

Context

With selinux every thing has a security context.

A process has a context.

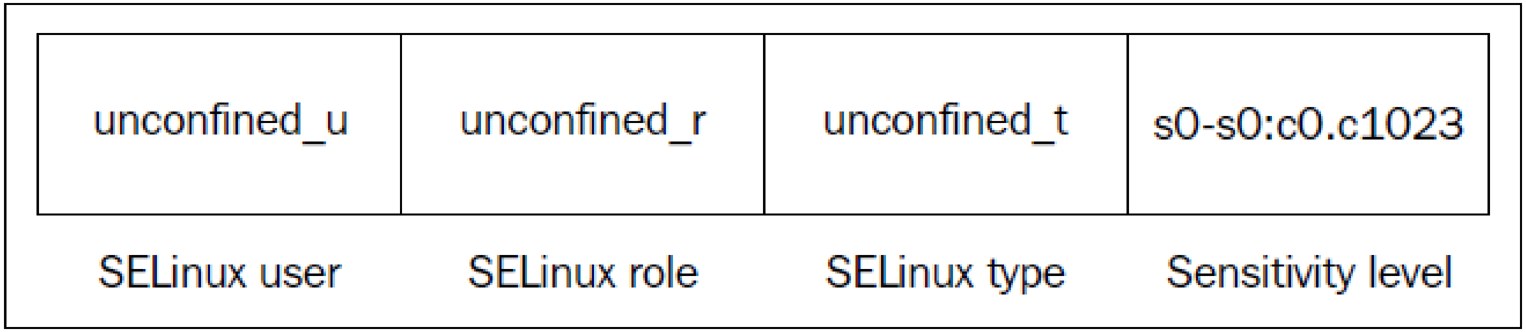
A file has a context.

Database of rules:

Rules allow a process in one context to do operations on an object in another context.

SELinux Contexts

Processes and files are labeled with an SELinux context that contains additional information, such as an SELinux user, role, type, and, optionally, a level.



unconfined_u	unconfined_r	unconfined_t	s0-s0:c0.c1023
SELinux user	SELinux role	SELinux type	Sensitivity level

Context

With selinux every thing has a security context.

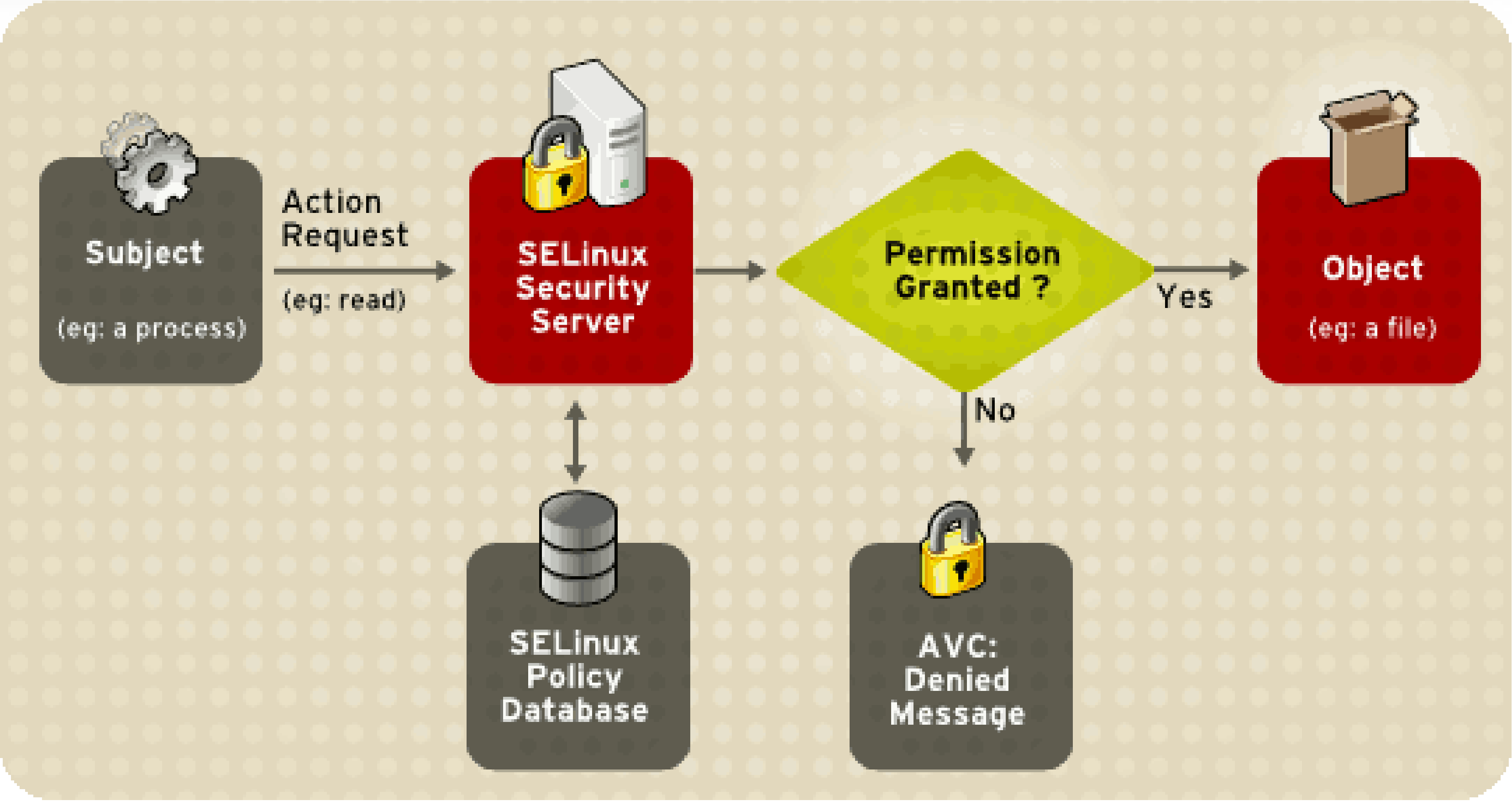
A process has a context.

A file has a context.

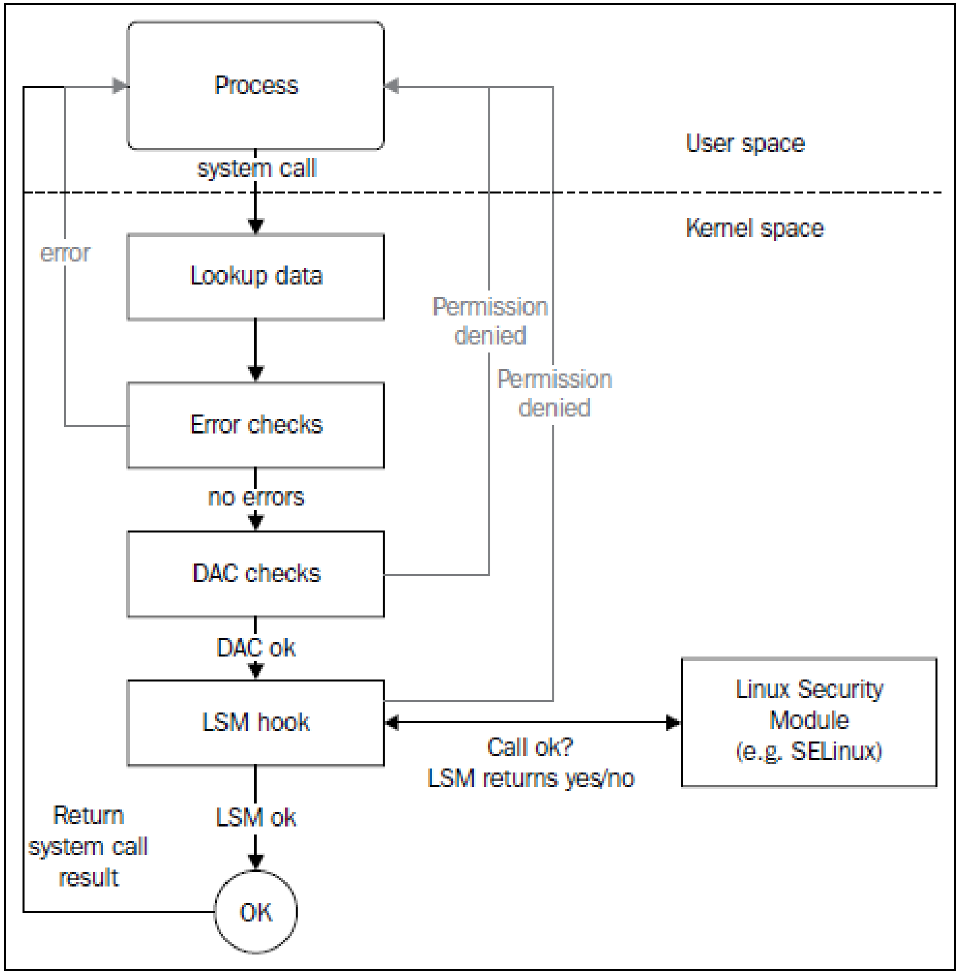
Database of rules:

Rules allow a process in one context to do operations on an object in another context.

SELinux Decision Process



High-level overview of how LSM is integrated in the Linux kernel



SElinux enabled disrtributions

Native support:

RHEL, Centos, Fedora, Oracle Linux

SLES, OpenSuse (almost)

Not native supported, but can be added:

Debian, ubuntu

Gentoo, Arch Linux

…

SElinux policy

SElinux requires a policy to start.

Switching policy requires a system reboot and even relabeling.

Use “sestatus” to see the active policy in use.

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current Mode:                   enforcing
Mode from config file:          enforcing
Policy Version:                 24
Policy from config file:        targeted
```

Set the policy to be loaded at boot in /etc/selinux/config

SELinux Operating Modes

SELinux has three operation modes: enforcing, permissive and diasbled.

✓ In enforcing mode SELinux is fully functional.

✓ In permissive mode, SELinux does not enforce the policy. This can be useful for troubleshooting and for developing or fine-tuning SELinux policy.

✓ In disable mode, SELinux is completely disabled.

Using Security Enhanced Linux

Step 1: Check our system kernel for Selinux support

```
root@anisa# cd /boot
root@anisa# grep –i  selinux config-`uname -r`
```

Step 2: Check selinux current status

```
root@anisa# sestatus
```

Step 3: navigate to selinux main directory

```
root@anisa# cd /etc/selinux
```

Step 4: Check the selinux default state config

```
root@anisa#  cat /etc/selinux/config
```

Using Security Enhanced Linux

Step 5: Change the default selinux mode (It needs system restart to take effect)

```
root@anisa#  system-config-securitylevel
```

Step 6: Change selinux mode at runtime (Just in some distros)

```
 root@anisa#  echo 1 > /selinux/enforce
 root@anisa# sestatus
```

Step 7: change the current selinux mode to permissive

```
 root@anisa#  echo 0 > /selinux/enforce
 root@anisa# sestatus
```

Allowing Access to a Port
(Apache bind on a non-standard port)

Step 1: Create home directory for one site:

```
root@anisa#   mkdir /var/www/lpir_org
root@anisa#   cd  /var/www/lpir_org
root@anisa#   echo “Test content" > index.html
```

Step 2: Configure port based virtual hosting:

Edit file /etc/httpd/conf/httpd.conf and add the following:

```
Listen 8090
<VirtualHost *:8090>
DocumentRoot /var/www/lpir_org
</VirtualHost>
```

Allowing Access to a Port
(Apache bind on a non-standard port)

Step 3: enable selinux and restart server & test:

```
root@anisa#  echo 1 > /selinux/enforce
root@anisa#   service httpd restart          # watch the error messages
```

Step 4: Add a rule to allow

```
root@anisa#   semanage port –l   | grep http
root@anisa#   semanage port -a -t http_port_t -p tcp 8090
root@anisa#   semanage port –l   | grep http
root@anisa#   service httpd restart
root@anisa#   firefox http://127.0.0.1:8090
root@anisa#   semanage port -d -t http_port_t -p tcp 8090
root@anisa#   service httpd restart
```

Disable protection of currently protected deamon

On system with Enforcing mode:

```
root@anisa#  ps –auxZ | grep httpd
root@anisa#  cat /selinux/booleans/httpd_disable_trans
root@anisa#  echo “1 1” > /selinux/Booleans/httpd_disable_trans
root@anisa#  echo “1” > /selinux/commit_pending_bools
root@anisa#  echo $?
root@anisa#  service httpd restart
root@anisa#  ps –auxZ | grep httpd
```

Disable protection of currently protected deamon

now test the public home directory when httpd in run in an unconfined domain (initrc_t).

Go back to previous status:

```
root@anisa#  echo “0 0” > /selinux/Booleans/httpd_disable_trans
root@anisa#  echo “1” > /selinux/commit_pending_bools
root@anisa#  service httpd restart
root@anisa#  ps –auxZ | grep httpd
```

Thanks for your attention

&

Best Regards