



Manage Security Enhanced Linux (SELinux)



Unit objectives

After completing this unit, you should be able to:

- Understand SELinux
- Change SELinux Enforcement Mode
- Controlling SELinux File Contexts

Understand SELinux

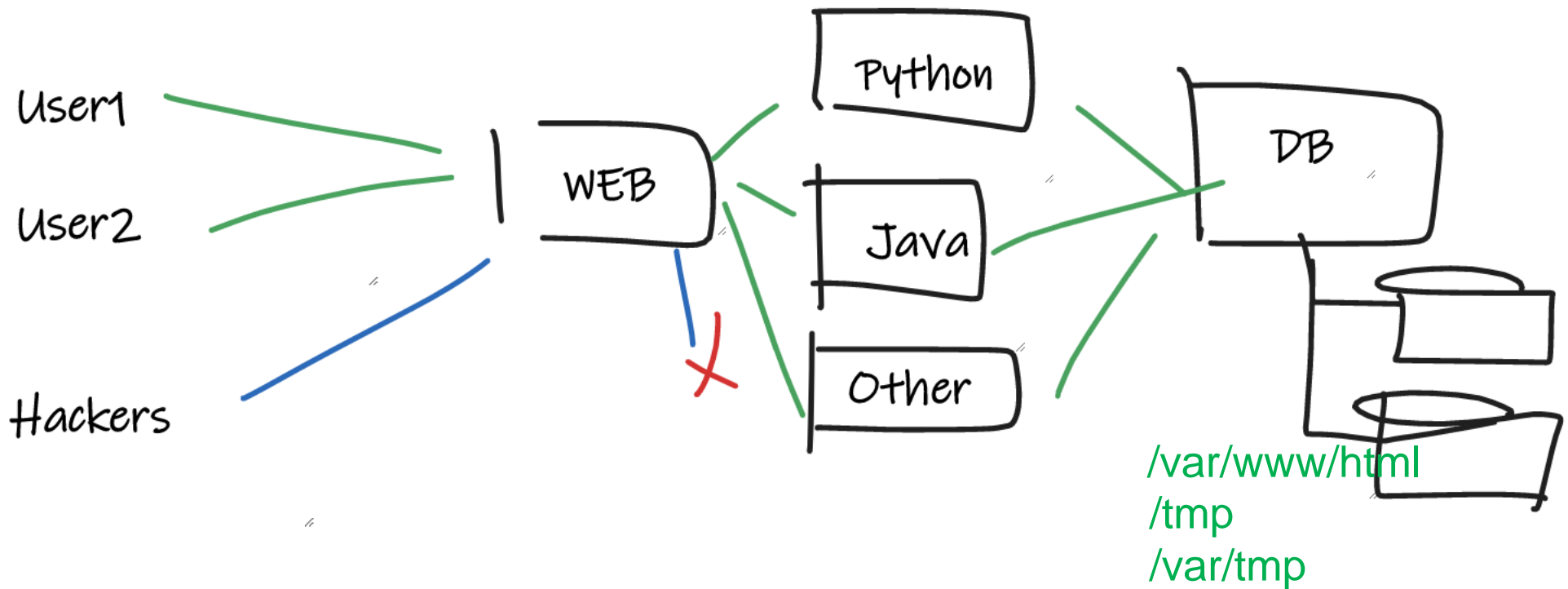
File Permissions / ACLs

- Control user or group access to file and directory
- File / Directory Level
- Control how file is access, not the content

SELinux Settings

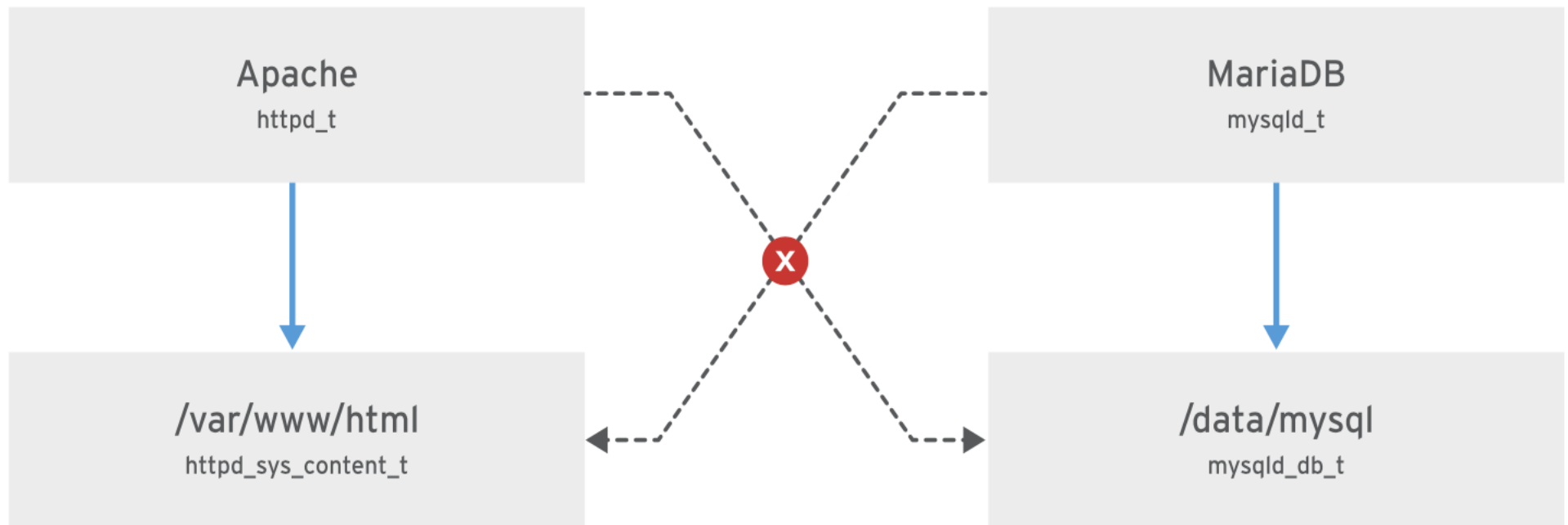
- Control processes or set of programs access to file and directory
- Granular Level – can control content
- Set of policies – determine actions and accesses right
- Place labels
- Policies applies to binaries, executable, configuration, app's data, and network port used by application

Why use SELinux



Basic SELinux security concepts

- **MariaDB** server has context **mysqld_t** and has access to **/data/mysql** with **mysqld_db_t** type context.
- **Apache** server has context **httpd_t** and has access to **/var/www/html** with **httpd_sys_content_t** type context.
- Each data file access by own services but disables access by other services



SELinux Mode

Enforcing

- SELinux enforces access control rules. Enabled by default.

Permissive

- SELinux is active but records warnings of rules violated. Used for testing and troubleshooting

Disabled

- SELinux is turned off entirely. Violations are allowed and not recording whatsoever. Discouraged!

Changing SELinux mode - temporarily

- `getenforce` # get mode
- `setenforce` # switch mode
- `setenforce 0` # disable completely
- `setenforce 1` # same as enforcing

```
[user@host ~]# getenforce
Enforcing
[user@host ~]# setenforce
usage:  setenforce [ Enforcing | Permissive | 1 | 0 ]
[user@host ~]# setenforce 0
[user@host ~]# getenforce
Permissive
[user@host ~]# setenforce Enforcing
[user@host ~]# getenforce
Enforcing
```

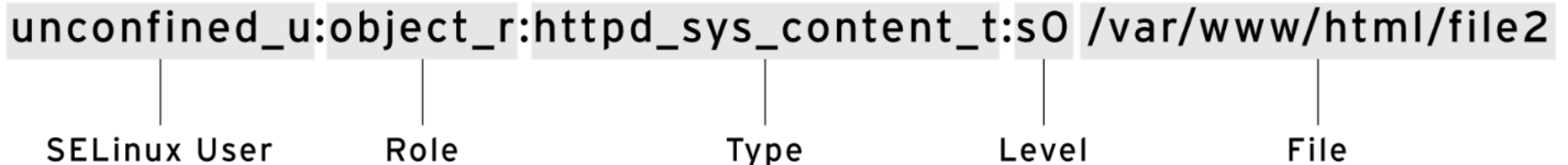
Changing SELinux mode - persistent

- Using /etc/selinux/config
- targetted type # Object access is controlled by SELinux, not by discretionary ccess control. Default
- minimum type # SELinux only protect selected process
- mls # SELinux provide multi level security protection

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes
#               are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```


Basic SELinux security concepts

- Is a layer built on top of other security layer
- Protect user data from system services that have been compromised
 - mitigate against discretionary access control weakness
- Policies - Set of rules determining which process can access file, directories and application ports
- Context – user, role, type ,sensitivity
- Default, implicit denied rule is imposed



The diagram illustrates the components of an SELinux context. It shows a sequence of five components separated by colons: 'unconfined_u', 'object_r', 'httpd_sys_content_t', 's0', and '/var/www/html/file2'. Each component is highlighted with a light gray background. Below each component, a vertical line connects it to a label: 'SELinux User' for 'unconfined_u', 'Role' for 'object_r', 'Type' for 'httpd_sys_content_t', 'Level' for 's0', and 'File' for '/var/www/html/file2'.

SELinux User	Role	Type	Level	File
unconfined_u	object_r	httpd_sys_content_t	s0	/var/www/html/file2

Reveal SELinux contexts

```
[root@host ~]# ps axZ
```

LABEL	PID	TTY	STAT	TIME	COMMAND
system_u:system_r:init_t:s0	1	?	Ss	0:09	/usr/lib/systemd/...
system_u:system_r:kernel_t:s0	2	?	S	0:00	[kthreadd]
system_u:system_r:kernel_t:s0	3	?	S	0:00	[ksoftirqd/0]

...output omitted...

```
[root@host ~]# systemctl start httpd
```

```
[root@host ~]# ps -ZC httpd
```

LABEL	PID	TTY	TIME	CMD
system_u:system_r:httpd_t:s0	1608	?	00:00:05	httpd
system_u:system_r:httpd_t:s0	1609	?	00:00:00	httpd

...output omitted...

```
[root@host ~]# ls -Z /home
```

drwx-----.	root	root	system_u:object_r:lost_found_t:s0	lost+found
drwx-----.	student	student	unconfined_u:object_r:user_home_dir_t:s0	student
drwx-----.	visitor	visitor	unconfined_u:object_r:user_home_dir_t:s0	visitor

```
[root@host ~]# ls -Z /var/www
```

drwxr-xr-x.	root	root	system_u:object_r:httpd_sys_script_exec_t:s0	cgi-bin
drwxr-xr-x.	root	root	system_u:object_r:httpd_sys_content_t:s0	error
drwxr-xr-x.	root	root	system_u:object_r:httpd_sys_content_t:s0	html
drwxr-xr-x.	root	root	system_u:object_r:httpd_sys_content_t:s0	icons

Initial SELinux Context

- Label represents security relevant information
- All processes and files are labelled
- New files inherit context from parent directory
- Inheritancy compromise security

Example: Inherency compromise security

The **ls -Z** command displays the SELinux context of a file. Note the label of the file.

```
[root@host ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/
index.html
```

And the **ls -Zd** command displays the SELinux context of a directory:

```
[root@host ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

Note that the **/var/www/html/index.html** has the same label as the parent directory **/var/www/html/**. Now, create files outside of the **/var/www/html** directory and note their file context:

Example: Inherency compromise security

- /tmp has **unconfined_u:object_r:user_tmp_t:s0** context

```
[root@host ~]# touch /tmp/file1 /tmp/file2
[root@host ~]# ls -Z /tmp/file*
unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
```

Move one of these files to the **/var/www/html** directory, copy another, and note the label of each:

```
[root@host ~]# mv /tmp/file1 /var/www/html/
[root@host ~]# cp /tmp/file2 /var/www/html/
```

```
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

The moved file maintains its original label while the copied file inherits the label from the **/var/www/html** directory. **unconfined_u:** is the user, **object_r:** denotes the role, and **s0** is the level. A sensitivity level of 0 is the lowest possible sensitivity level.

Changing SELinux context of file

- includes "semanage fcontext", restorecon, and chcon
- chcon – discouraged. Changes does not survived thru restorecon or file system relabeled process
- Example: chcon does not survived restorecon

```
[root@host ~]# mkdir /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
[root@host ~]# chcon -t httpd_sys_content_t /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /virtual
[root@host ~]# restorecon -v /virtual
Relabeled /virtual from unconfined_u:object_r:httpd_sys_content_t:s0 to
unconfined_u:object_r:default_t:s0
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

Basic File Context Operations

```
[root@host; ~]# restorecon -Rv /var/www/
Relabeled /var/www/html/file1 from unconfined_u:object_r:user_tmp_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2

[root@host ~]# mkdir /virtual
[root@host ~]# touch /virtual/index.html
[root@host ~]# ls -Zd /virtual/
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/

[root@host ~]# ls -Z /virtual/
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html
[root@host ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'
[root@host ~]# restorecon -RFvv /virtual
[root@host ~]# ls -Zd /virtual/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/
[root@host ~]# ls -Z /virtual/
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```

Basic File Context Operations

semanage fcontext commands

option	description
-a, --add	Add a record of the specified object type
-d, --delete	Delete a record of the specified object type
-l, --list	List records of the specified object type

```
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

```
[root@host ~]# semanage fcontext -l
...output omitted...
/var/www(/.*)?          all files      system_u:object_r:httpd_sys_content_t:s0
...output omitted...
```


Adjusting SELinux Policy with Booleans

- Switches that change behavior of policy
- Can be enabled or disabled only
- Get documentations
 - # dnf -y install selinux-policy-doc selinux-policy-devel
 - # man -k '_selinux'
- Manage thru **setsebool**, **getsebool**, **semanage boolean**

Example: Manage SELinux Boolean

```
[user@host ~]$ getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
...output omitted...
[user@host ~]$ getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off

[user@host ~]$ setsebool httpd_enable_homedirs on
Could not change active booleans. Please try as root: Permission denied
[user@host ~]$ sudo setsebool httpd_enable_homedirs on
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on , off) Allow httpd to enable homedirs
[user@host ~]$ getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
[user@host ~]$ setsebool -P httpd_enable_homedirs on
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on , on) Allow httpd to enable homedirs
```

Example: Manage SELinux Boolean

- List booleans which current state differs from default configuration

```
[user@host ~]$ sudo semanage boolean -l -C
SELinux boolean          State  Default Description
cron_can_relabel        (off  ,   on) Allow cron to can relabel
```

Investigate and Resolving SELinux Issues

- Important: To prevent unauthorized access to files.
- Intentional
 1. Before thinking of making any adjustments, consider that SELinux may be doing its job correctly by prohibiting the attempted access. If a web server tries to access files in **/home**, this could signal a compromise of the service if web content is not published by users. If access should have been granted, then additional steps need to be taken to solve the problem.
- Missconfigured
 2. The most common SELinux issue is an incorrect file context. This can occur when a file is created in a location with one file context and moved into a place where a different context is expected. In most cases, running **restorecon** will correct the issue. Correcting issues in this way has a very narrow impact on the security of the rest of the system.
- Overly restrictive
 3. Another remedy for overly restrictive access could be the adjustment of a Boolean. For example, the **ftpd_anon_write** boolean controls whether anonymous FTP users can upload files. You must turn this boolean on to permit anonymous FTP users to upload files to a server. Adjusting booleans requires more care because they can have a broad impact on system security.

Investigate and Resolving SELinux Issues

- Important: To prevent unauthorized access to files.
 - Bug
4. It is possible that the SELinux policy has a bug that prevents a legitimate access. Since SELinux has matured, this is a rare occurrence. When it is clear that a policy bug has been identified, contact Red Hat support to report the bug so it can be resolved.

Monitor SELinux for any violations

- Install setroubleshoot-server package
- Listen for audit messages in /var/log/audit/audit.log
- Send short message /var/log/messages
 - Look for *UUID* : unique identifiers for violations to SELinux
- sealert -l *UUID*
- sealert -a /var/log/audit/audit.log
- Next slide for sample sequence commands on standard Apache web server.

Diagnose SELinux on standard Apache web server.

```
[root@host ~]# touch /root/file3
[root@host ~]# mv /root/file3 /var/www/html
[root@host ~]# systemctl start httpd
[root@host ~]# curl http://localhost/file3
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /file3
on this server.</p>
</body></html>
```

When getting web page , it shows permission denied. Time to find out why

Check on the logs

```
[root@host ~]# tail /var/log/audit/audit.log
...output omitted...
type=AVC msg=audit(1392944135.482:429): avc: denied { getattr } for
  pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
  scontext=system_u:system_r:httpd_t:s0
  tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
...output omitted...
[root@host ~]# tail /var/log/messages
...output omitted...
Feb 20 19:55:42 host setroubleshoot: SELinux is preventing /usr/sbin/httpd
  from getattr access on the file . For complete SELinux messages. run
  sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
```

Further check, SELinux is preventing access

sealert -l UUID (gather from log)

```
[root@host ~]# sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
SELinux is preventing /usr/sbin/httpd from getattr access on the file .

***** Plugin catchall (100. confidence) suggests *****

If you believe that httpd should be allowed getattr access on the
file by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp

Additional Information:
Source Context          system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:admin_home_t:s0
Target Objects          [ file ]
Source                  httpd
Source Path              /usr/sbin/httpd
Port                    <Unknown>
Host                    servera
Source RPM Packages      httpd-2.4.6-14.el7.x86_64
```

Look for main problem

sealert -l UUID (gather from log)

Raw Audit Messages

```
type=AVC msg=audit(1392944135.482:429): avc: denied { getattr } for
pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
```

```
type=SYSCALL msg=audit(1392944135.482:429): arch=x86_64 syscall=lstat
success=no exit=EACCES a0=7f9fed0edea8 a1=7fff7bffc770 a2=7fff7bffc770
a3=0 items=0 ppid=1608 pid=1609 auid=4294967295 uid=48 gid=48 euid=48
suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295
comm=httpd exe=/usr/sbin/httpd subj=system_u:system_r:httpd_t:s0 key=(null)
```

```
Hash: httpd,httpd_t,admin_home_t,file,getattr
```

Audit messages reveals target file is the main problem.
/var/www/html/file3 does not look like belongs to web server.

Search for recent logs in raw audit messages

```
[root@host ~]# ausearch -m AVC -ts recent
----
time->Tue Apr  9 13:13:07 2019
type=PROCTITLE msg=audit(1554808387.778:4002):
  proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1554808387.778:4002): arch=c000003e syscall=49
  success=no exit=-13 a0=3 a1=55620b8c9280 a2=10 a3=7ffed967661c items=0
  ppid=1 pid=9340 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
  sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
  subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1554808387.778:4002): avc:  denied  { name_bind }
  for  pid=9340 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0
  tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0
```

This raw audit further confirms the problem

The solution

- Simply copy security context using parent directory

```
# restorecon -Rv /var/www/html/
```

- Confirm the rectification

```
# ls -lZ /var/www/html/file3
```

```
# curl http://localhost/file3
```

- Check the default file context of /var/www/html

```
# semanage fcontext -l | grep httpd | grep /var/www/html
```

Quiz

1. How do you verify SELinux is enabled?
 - a) `getenforce 1`
 - b) `setenforce 0`
 - c) `getenforce`
 - d) `setenforce 1`

2. How do you quickly enable SELinux now?
 - a) `getenforce 1`
 - b) `setenforce 0`
 - c) `getenforce`
 - d) `setenforce 1`

3. How do you configure SELinux to permissive mode on every reboot?
 - a) Modify `/etc/security/selinux`, by changing `SELINUX=permissive`
 - b) Modify `/etc/security/selinux`, by changing `SELINUX=enforcing`
 - c) Modify `/etc/selinux/config`, by changing `SELINUX=permissive`
 - d) Modify `/etc/selinux/config`, by changing `SELINUX=enforcing`

4. True or False: Best practice is to disable SELinux as DAC has enough security strength

Quiz - Answer

1. How do you verify if SELinux is enabled?
 - a) `getenforce 1`
 - b) `setenforce 0`
 - c) `getenforce`
 - d) `setenforce 1`

2. How do you quickly enable SELinux now?
 - a) `getenforce 1`
 - b) `setenforce 0`
 - c) `getenforce`
 - d) `setenforce 1`

3. How do you configure SELinux to permissive mode on every reboot?
 - a) Modify `/etc/security/selinux`, by changing `SELINUX=permissive`
 - b) Modify `/etc/security/selinux`, by changing `SELINUX=enforcing`
 - c) Modify `/etc/selinux/config`, by changing `SELINUX=permissive`
 - d) Modify `/etc/selinux/config`, by changing `SELINUX=enforcing`

4. True or False: Best practice is to disable SELinux as DAC has enough security strength

Guided Exercise

Topic	Page number on student-guide.pdf	Time (min)
Changing the SELinux Enforcement Mode	148	10
Controlling SELinux File Contexts	155	10
Adjusting SELinux with Booleans	160	10
Investigating and Resolving SELinux Issues	167	10



Unit summary

Having completed this unit, you should be able to:

- Understand SELinux
- Change SELinux Enforcement Mode