RedHat 2

Day2

Nada Mohamed Ahmed Hassan Eleshmawy

Mansoura Open Source

Date : 14/2/2025

1. Create a script named backup.sh in /usr/local/bin and Set the SUID bit so that the script runs with the permissions of the file owner (root).

SUID: set user id → the script -with this bit- runs with the permissions of the file owner (root)(add only to files).

Set this bit :

chmod u+s /usr/local/bin/backup.sh

Or using

chmod 4755 /usr/local/bin/backup.sh

Remove it using :

chmod u-s /usr/local/bin/backup.sh

Create the file :

```
[nada_mohamed2243@localhost ~]$ sudo touch /usr/local/bin/backup.sh
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ sudo ls /usr/local/bin/
backup.sh
```

Check permission before Set the SUID bit :

```
[nada_mohamed2243@localhost ~]$ sudo ls -dl /usr/local/bin/backup.sh
-rw-r--r--. 1 root root 0 Feb 14 13:01 /usr/local/bin/backup.sh
[nada_mohamed2243@localhost ~]$ sudo ls -dl /usr/local/bin/
drwxr-xr-x. 2 root root 23 Feb 14 13:01 /usr/local/bin/
[nada_mohamed2243@localhost ~]$
```

Set the SUID bit :

chmod u+s /usr/local/bin/backup.sh

check permission After Set the SUID bit :

```
[nada_mohamed2243@localhost ~]$ sudo chmod u+s /usr/local/bin/backup.sh
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ sudo ls -dl /usr/local/bin/backup.sh
-rwSr--r--. 1 root root 0 Feb 14 13:01 /usr/local/bin/backup.sh
[nada_mohamed2243@localhost ~]$
```

The uppercase S instead of lowercase s in `-rwSr--r--` means that the SUID bit is set, but the execute (x) permission is missing for the file owner.

If we add execute permission for the file owner:

chmod u+x /usr/local/bin/backup.sh → we got **lowercase s**

```
[nada_mohamed2243@localhost ~]$ sudo chmod u+x /usr/local/bin/backup.sh
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ sudo ls -dl /usr/local/bin/backup.sh
-rwsr--r--. 1 root root 0 Feb 14 13:01 /usr/local/bin/backup.sh
[nada_mohamed2243@localhost ~]$
```

- If you set SUID (u+s), but the file is not executable, the system cannot run it as another user.
- Adding execute (x) allows it to run with the owner's privileges (root).

---

2. Create a directory named shared_team in /home and Set the SGID bit so that any files created in this directory inherit the group ownership of the directory.

SGID : set group id → any files created in this directory inherit the group ownership of the directory

Set this bit :

chmod g+s /home/shared_team

Or using

chmod 2775 /home/shared_team

Remove it using :

chmod g-s /home/shared_team

Create the directory :

```
[nada_mohamed2243@localhost ~]$ ls /home/
ahmed  nada_mohamed2243  testu
```

sudo mkdir /home/shared_team

```
[nada_mohamed2243@localhost ~]$ sudo mkdir /home/shared_team
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ ls /home/
ahmed  nada_mohamed2243  shared_team  testu
[nada_mohamed2243@localhost ~]$
```

Check permission before Set the SGID bit :

```
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/
[sudo] password for nada_mohamed2243:
drwxr-xr-x. 2 root root 6 Feb 14 15:43 /home/shared_team/
```

Set the SGID bit :

chmod g-s /home/shared_team

check permission After Set the SGID bit :

```
[nada_mohamed2243@localhost ~]$ sudo chmod g+s /home/shared_team
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/
drwxr-sr-x. 2 root root 6 Feb 14 15:43 /home/shared_team/
[nada_mohamed2243@localhost ~]$
```

Test this:

- We create group
- Add a user to it.
- Add the dir to this group
- If we create file in this directory (with **SGID bit**) , it will inherit the group ownership of the directory not the user group

```
[nada_mohamed2243@localhost ~]$ sudo groupadd team
[nada_mohamed2243@localhost ~]$ sudo usermod -aG team ahmed
[nada_mohamed2243@localhost ~]$ sudo chown :team /home/shared_team
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team
drwxr-sr-x. 2 root team 39 Feb 14 15:56 /home/shared_team
[nada_mohamed2243@localhost ~]$ sudo touch /home/shared_team/testSGID3
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/testSGID3
-rw-r--r--. 1 root team 0 Feb 14 16:27 /home/shared_team/testSGID3
[nada_mohamed2243@localhost ~]$ touch testSGIDoutSide
[nada_mohamed2243@localhost ~]$ sudo ls -dl testSGIDoutSide
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 0 Feb 14 16:28 testSGIDoutSide
[nada_mohamed2243@localhost ~]$
```

- If we create file in this directory (without **SGID bit**) , it will inherit the user group

```
[nada_mohamed2243@localhost ~]$ sudo chmod g-s /home/shared_team
[nada_mohamed2243@localhost ~]$ sudo touch /home/shared_team/testSGID4
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/testSGID4
-rw-r--r--. 1 root root 0 Feb 14 16:30 /home/shared_team/testSGID4
[nada_mohamed2243@localhost ~]$
```

- If we create file in this directory (with **SGID bit**) , it will inherit the group ownership of the directory not the user group

```
[nada_mohamed2243@localhost ~]$ sudo chmod g+s /home/shared_team
[nada_mohamed2243@localhost ~]$ sudo touch /home/shared_team/testSGID5
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/testSGID5
-rw-r--r--. 1 root team 0 Feb 14 16:31 /home/shared_team/testSGID5
```

---

3. Set the sticky bit on the shared_team directory so that users can only delete their own files.

sticky bit → Restrict file deletion (add only to directories)

Set this bit :

chmod a+t /home/shared_team

Or using

sudo chmod 1775 /home/shared_team

Remove it using :

chmod a-t /home/shared_team

```
[nada_mohamed2243@localhost ~]$ sudo chmod a+t /home/shared_team/
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/
drwxr-sr-t. 2 root team 90 Feb 14 16:31 /home/shared_team/
[nada_mohamed2243@localhost ~]$
```

Test this:

Before adding sticky bit to the dir :

From another user we try to delete the file which exists inside the dir → we delete it successfully

- We just make sure that the parent has wx permission to delete the files

```
[nada_mohamed2243@localhost ~]$ sudo chmod 777 /home/shared_team
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/
drwxrwsrwx. 2 root team 90 Feb 14 16:31 /home/shared_team/
```

```
[testu@localhost ~]$ ls -dl /home/shared_team
drwxrwsrwx. 2 root team 90 Feb 14 16:31 /home/shared_team
[testu@localhost ~]$ ls -dl /home/shared_team/testSGID5
-rw-r--r--. 1 root team 0 Feb 14 16:31 /home/shared_team/testSGID5
[testu@localhost ~]$ rm /home/shared_team/testSGID5
rm: remove write-protected regular empty file '/home/shared_team/testSGID5'? Y
[testu@localhost ~]$ ls /home/shared_team
testSGID  testSGID2  testSGID3  testSGID4
[testu@localhost ~]$
```

After adding sticky bit to the dir :

From another user we try to delete the file which exists inside the dir → we cannot delete it.

- We create the file again

```
[nada_mohamed2243@localhost ~]$ sudo ls /home/shared_team/
testSGID  testSGID2  testSGID3  testSGID4
[nada_mohamed2243@localhost ~]$ sudo touch /home/shared_team/testSGID5
```

- Add sticky bit

```
[nada_mohamed2243@localhost ~]$ sudo chmod a+t /home/shared_team/
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared_team/
drwxrwsrwt. 2 root team 90 Feb 14 17:19 /home/shared_team/
[nada_mohamed2243@localhost ~]$
```

- Try to remove it from other user

```
[testu@localhost ~]$ ls /home/shared_team
testSGID  testSGID2  testSGID3  testSGID4  testSGID5
[testu@localhost ~]$ rm /home/shared_team/testSGID5
rm: remove write-protected regular empty file '/home/shared_team/testSGID5'? y
rm: cannot remove '/home/shared_team/testSGID5': Operation not permitted
[testu@localhost ~]$
```

4.Create a shared directory named shared where:

add read and write permissions for a group named developers using ACL .

All new files and subdirectories inherit the group developers permissions(use the setgid permission).

Only the owner of a file can delete it (use the sticky bit).

1- Create a shared directory named shared

```
[nada_mohamed2243@localhost ~]$ sudo mkdir /home/shared
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ sudo ls /home/
ahmed  nada_mohamed2243  shared  shared_team  testu
```

2-add read and write permissions for a group named developers using ACL.

```
[nada_mohamed2243@localhost ~]$ sudo groupadd developers
[nada_mohamed2243@localhost ~]$ sudo chown :developers /home/shared
[nada_mohamed2243@localhost ~]$ sudo setfacl -m g:developers:rw /home/shared
[nada_mohamed2243@localhost ~]$ sudo getfacl /home/shared
getfacl: Removing leading '/' from absolute path names
# file: home/shared
# owner: root
# group: developers
user::rwx
group::r-x
group:developers:rw-
mask::rwx
other::r-x
```

3- All new files and subdirectories inherit the group developers permissions(use the setgid permission).

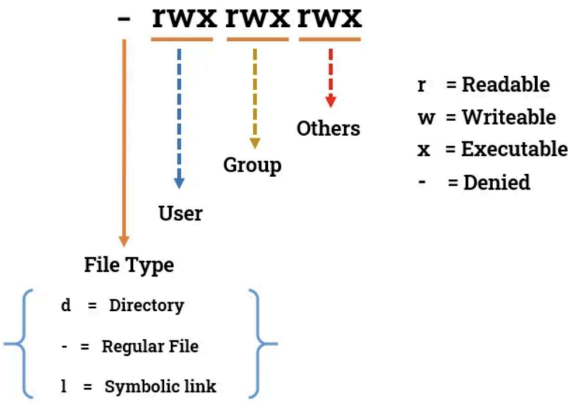
```
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared
drwxrwxr-x+ 2 root developers 6 Feb 14 17:32 /home/shared
[nada_mohamed2243@localhost ~]$ sudo chmod g+s /home/shared
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared
drwxrwsr-x+ 2 root developers 6 Feb 14 17:32 /home/shared
[nada_mohamed2243@localhost ~]$
```

4- Only the owner of a file can delete it (use the sticky bit).

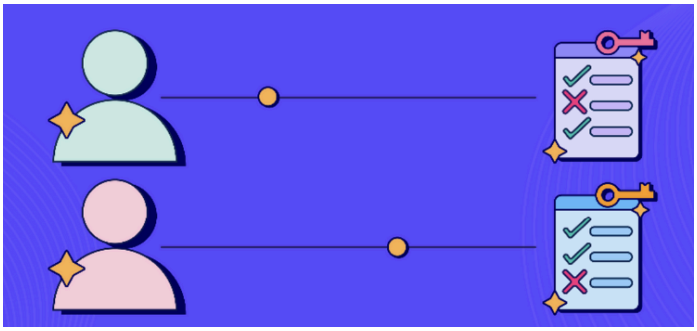
```
[nada_mohamed2243@localhost ~]$ sudo chmod o+t /home/shared
[nada_mohamed2243@localhost ~]$ sudo ls -dl /home/shared
drwxrwsr-t+ 2 root developers 6 Feb 14 17:32 /home/shared
[nada_mohamed2243@localhost ~]$
```

5.What is the difference between traditional Linux permissions and ACLs?

Traditional Linux Permissions: Use `rwx` (read, write, execute) permissions for owner, group, and others.



ACLs (Access Control Lists): allow you to define permissions for specific users or groups beyond the standard owner/group/others model.



6.Create a directory named lab_acls and navigate into it:

mkdir lab_acls

cd lab_acls

```
[nada_mohamed2243@localhost ~]$ mkdir lab_acls
[nada_mohamed2243@localhost ~]$ cd lab_acls
[nada_mohamed2243@localhost lab_acls]$
```

7.Create a file named testfile.txt:

touch testfile.txt

```
[nada_mohamed2243@localhost lab_acls]$ touch testfile.txt
[nada_mohamed2243@localhost lab_acls]$ ls
testfile.txt
```

8.Create two users (alice and bob) and a group (developers):

sudo useradd alice

sudo useradd bob

sudo groupadd developers2

```
[nada_mohamed2243@localhost lab_acls]$ sudo useradd alice
[nada_mohamed2243@localhost lab_acls]$ sudo useradd bob
[nada_mohamed2243@localhost lab_acls]$ sudo groupadd developers
groupadd: group 'developers' already exists
[nada_mohamed2243@localhost lab_acls]$ sudo groupadd developers2
[nada_mohamed2243@localhost lab_acls]$
```

9.Add alice and bob to the developers group:

sudo usermod -aG developers2 alice

sudo usermod -aG developers2 bob

```
[nada_mohamed2243@localhost lab_acls]$ sudo usermod -aG developers2 alice
[nada_mohamed2243@localhost lab_acls]$ sudo usermod -aG developers2 bob
[nada_mohamed2243@localhost lab_acls]$ cat /etc/group
```

```
bob:x:1006:
developers2:x:1007:alice,bob
```

10.View and List the ACL of a file named testfile.txt.

sudo getfacl testfile.txt

```
[nada_mohamed2243@localhost lab_acls]$ ls
testfile.txt
[nada_mohamed2243@localhost lab_acls]$ sudo getfacl testfile.txt
# file: testfile.txt
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rw-
group::r--
other::r--

[nada_mohamed2243@localhost lab_acls]$
```

11.read and write permissions for a user named alice to the file testfile.txt and Verify the changes.

sudo setfacl -m u:alice:rw testfile.txt

sudo getfacl testfile.txt

```
[nada_mohamed2243@localhost lab_acls]$ sudo setfacl -m u:alice:rw testfile.txt
[nada_mohamed2243@localhost lab_acls]$ sudo getfacl testfile.txt
# file: testfile.txt
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rw-
user:alice:rw-
group::r--
mask::rw-
other::r--
```

12.Add execute permission for a group named developers to the file testfile.txt and Verify the changes.

sudo setfacl -m g:developers:x testfile.txt

sudo getfacl testfile.txt

```
[nada_mohamed2243@localhost lab_acls]$ sudo setfacl -m g:developers2:x testfile.txt
[nada_mohamed2243@localhost lab_acls]$ sudo getfacl testfile.txt
# file: testfile.txt
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rw-
user:alice:rw-
group::r--
group:developers2:--x
mask::rwx
other::r--
```

13.Remove the ACL entry for the user alice from the file testfile.txt and Verify the changes.

setfacl -x u:alice testfile.txt

getfacl testfile.txt

```
[nada_mohamed2243@localhost lab_acls]$ sudo setfacl -x u:alice testfile.txt
[nada_mohamed2243@localhost lab_acls]$ sudo getfacl testfile.txt
# file: testfile.txt
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rw-
group::r--
group:developers2:--x
mask::r-x
other::r--
```

14.Set read and execute permissions for bob on all files and subdirectories inside lab_acls.

cd

sudo setfacl -R -m u:bob:rx lab_acls

getfacl lab_acls

The -R flag in setfacl stands for recursive, meaning it applies the ACL changes to the directory and all its files and subdirectories.

```
[nada_mohamed2243@localhost ~]$ sudo setfacl -R -m u:bob:rx lab_acls
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ sudo getfacl lab_acls
# file: lab_acls
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rwx
user:bob:r-x
group::r-x
mask::r-x
other::r-x

[nada_mohamed2243@localhost ~]$
```

15.How does the mask affect the effective permissions of named users and groups?

- The mask limits the maximum permissions a user or group can have.
- Even if ACL grants a user rwx, if the mask is set to r--, the user will only have read access.

- Mask is Recalculated Automatically:
  - Every time you run setfacl, the mask is automatically recalculated to the maximum permissions given to any named user or group.
- We can set the mask manually using :
  - setfacl -m m:r-- testfile.txt
- If you don't want the mask to be automatically recalculated, use -n:
  - setfacl -n -m g:user1:rwx file1.txt

16.Set the mask for the file testfile.txt to r-- and observe how it affects the effective permissions of named users and groups.

sudo setfacl -m m::r-- testfile.txt

sudo getfacl testfile.txt

```
[nada_mohamed2243@localhost lab_acls]$ sudo setfacl -m m::r testfile.txt
[nada_mohamed2243@localhost lab_acls]$ sudo getfacl testfile.txt
# file: testfile.txt
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rw-
user:bob:r-x                    #effective:r--
group::r--
group:developers2:--x           #effective:---
mask::r--
other::r--
```

17.Add read and write permissions for two users, alice and bob, to the file testfile.txt in a single command.

sudo setfacl -m u:alice:rw,u:bob:rw testfile.txt

```
[nada_mohamed2243@localhost lab_acls]$ sudo setfacl -m u:alice:rw,u:bob:rw testfile.txt
[nada_mohamed2243@localhost lab_acls]$ sudo getfacl testfile.txt
# file: testfile.txt
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rw-
user:alice:rw-
user:bob:rw-
group::r--
group:developers2:--x
mask::rwx
other::r--

[nada_mohamed2243@localhost lab_acls]$
```

18.Backup the ACLs of the directory mydir to a file named mydir_acls.txt.

sudo getfacl -R mydir > mydir_acls.txt

Make this about our example lab_acls:

sudo getfacl -R lab_acls > lab_acls_backup.txt

```
[nada_mohamed2243@localhost lab_acls]$ cd
[nada_mohamed2243@localhost ~]$ getfacl -R lab_acls> lab_acls_backup.txt
[nada_mohamed2243@localhost ~]$ cat lab_acls_backup.txt
# file: lab_acls
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rwx
user:bob:r-x
group::r-x
mask::r-x
other::r-x

# file: lab_acls/testfile.txt
# owner: nada_mohamed2243
# group: nada_mohamed2243
user::rw-
user:alice:rw-
user:bob:rw-
group::r--
group:developers2:--x
mask::rwx
other::r--

[nada_mohamed2243@localhost ~]$
```

19.Check the Current SELinux Mode.

getenforce

```
[nada_mohamed2243@localhost ~]$ getenforce
Enforcing
```

- setenforce [Enforcing | Permissive | 1 | 0]

This will return one of the following:

- Enforcing → SELinux is fully enabled and enforcing security policies.
- Permissive → SELinux logs policy violations but doesn't enforce them.
- Disabled → SELinux is turned off.

Or using

sestatus

```
[nada_mohamed2243@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[nada_mohamed2243@localhost ~]$
```

20.Change SELinux mode temporarily.

sudo setenforce [0|1]

- Set SELinux to Permissive mode:

  sudo setenforce 0

```
[nada_mohamed2243@localhost ~]$ sudo setenforce 0
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ getenforce
Permissive
[nada_mohamed2243@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[nada_mohamed2243@localhost ~]$
```

- Set SELinux to Enforcing mode:

  sudo setenforce 1

```
[nada_mohamed2243@localhost ~]$ sudo setenforce 1
[nada_mohamed2243@localhost ~]$ getenforce
Enforcing
[nada_mohamed2243@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[nada_mohamed2243@localhost ~]$
```

21.Change SELinux Mode Permanently.

A- Open the SELinux configuration file:

sudo vi /etc/selinux/config

```
[nada_mohamed2243@localhost ~]$  sudo vi /etc/selinux/config
```

B- Find this line:

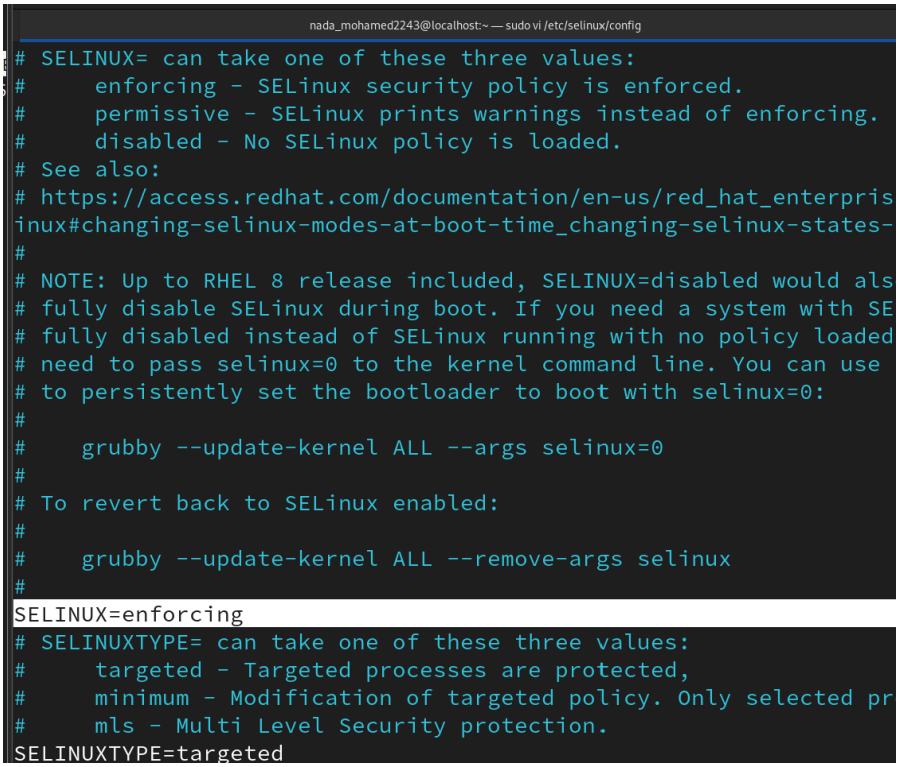
SELINUX=enforcing

It can be :

SELINUX=enforcing (Enforcing mode)

SELINUX=permissive (Permissive mode)

SELINUX=disabled (Disabled mode)

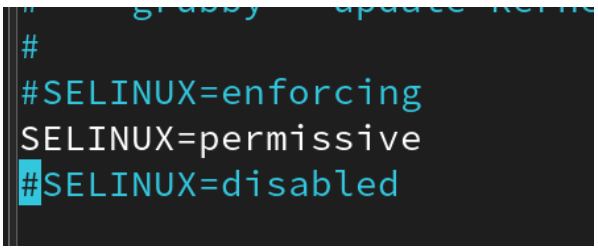


C- Change it to:

SELINUX=enforcing (Enforcing mode)

SELINUX=permissive (Permissive mode)

SELINUX=disabled (Disabled mode)




```
[nada_mohamed2243@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[nada_mohamed2243@localhost ~]$
```

Note :

If we choose Disabled mode (SELINUX=disabled) → we should restart the system

reboot

22.what difference between cp , mv ,cp -a (ContextSwitching).

- cp file1 file2 : Loses SELinux context; new file gets default context (cp loses the context because it creates a new file, which follows the default SELinux policy for the target directory)
- mv file1 file2 : Preserves SELinux context(mv keeps the context because it only updates the file's location, not its attributes)
- cp -a file1 file2 : Preserves SELinux context(cp -a keeps the context by preserving all file attributes, including SELinux labels)

Example:

- get the context of /var/www/html/ :

ls -ldZ /var/www/html/

```
[nada_mohamed2243@localhost ~]$ ls -ldZ /var/www/html/
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Jan 10 20:45 /var/www/html/
[nada_mohamed2243@localhost ~]$
```

- Create files and check there context :

touch file1 file2 file3

```
[nada_mohamed2243@localhost ~]$ touch file1 file2 file3
```

```
[nada_mohamed2243@localhost ~]$ ls -ldZ file*
-r-xr-xr-x. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0  0 Dec 25 13:53 file
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0  0 Feb 14 19:16 file1
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0  0 Feb 14 19:16 file2
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0  0 Feb 14 19:16 file3
-rwxr-xr-x. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0 17 Dec 30 14:44 filename1
-rwxr-xr-x. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0 19 Dec 30 14:45 filename2
[nada_mohamed2243@localhost ~]$
```

- Use cp with file1 ,Use mv with file2 and Use cp -a with file3 :

cp file1 /var/www/html/file1

mv file2 /var/www/html/file2

cp -a file3 /var/www/html/file3

```
[nada_mohamed2243@localhost ~]$ sudo cp file1 /var/www/html/file1
[sudo] password for nada_mohamed2243:
[nada_mohamed2243@localhost ~]$ sudo ls /var/www/html/
file1
[nada_mohamed2243@localhost ~]$ sudo mv file2 /var/www/html/file2
[nada_mohamed2243@localhost ~]$ sudo ls /var/www/html/
file1  file2
[nada_mohamed2243@localhost ~]$ sudo cp -a file3 /var/www/html/file3
[nada_mohamed2243@localhost ~]$ sudo ls /var/www/html/
file1  file2  file3
```

Check the context again after cp,mv,cp -a :

We note that with cp : Loses original context (new file follows directory defaults).

But with mv and cp -a : Preserves original context

```
[nada_mohamed2243@localhost ~]$ sudo ls -ldZ /var/www/html/
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 45 Feb 14 19:29 /var/www/html/
[nada_mohamed2243@localhost ~]$ sudo ls -ldZ /var/www/html/file*
-rw-r--r--. 1 root                 root                 unconfined_u:object_r:httpd_sys_content_t:s0 0 Feb 14 19:28 /var/www/html/file1
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0         0 Feb 14 19:16 /var/www/html/file2
-rw-r--r--. 1 nada_mohamed2243 nada_mohamed2243 unconfined_u:object_r:user_home_t:s0         0 Feb 14 19:16 /var/www/html/file3
[nada_mohamed2243@localhost ~]$
```

Note :

If you accidentally lose the context with `cp`, you can restore it with:

```
restorecon -Rv /var/www/html/file3
```

23.Run Apache web Server on /websites [must SELinuxMode=Enforcing]

1. Show IP

ifconfig → **192.168.112.137**

```
[nada_mohamed2243@localhost ~]$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.112.137  netmask 255.255.255.0  broadcast 192.168.112.255
        inet6 fe80::20c:29ff:fee4:19fc  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e4:19:fc  txqueuelen 1000  (Ethernet)
        RX packets 98005  bytes 137681098 (131.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 44644  bytes 2782501 (2.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 836  bytes 75887 (74.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 836  bytes 75887 (74.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[nada_mohamed2243@localhost ~]$
```

2. Show Mode SElinux

getenforce

```
[nada_mohamed2243@localhost ~]$ getenforce
Enforcing
```

3. create fileHtml

sudo vi /var/www/html/test.html

```
[nada_mohamed2243@localhost ~]$ sudo vi /var/www/html/test.html
```

```
nada_mohamed2243@localhost:~ — sudo vi /var/www/html/test.html
we are in test inside /var/www/html/
~
~
~
```

4. Edit config Apache

sudo vi /etc/httpd/conf/httpd.conf

```
[nada_mohamed2243@localhost ~]$ sudo vi /etc/httpd/conf/httpd.conf
```

DocumentRoot "/websites"
<Directory "/websites">
AllowOverride None
 # Allow open access:
Require all granted

</Directory>

```
<Directory "/websites">
    AllowOverride none
    #allow open access
    Require all granted
</Directory>

<Directory />
    AllowOverride none
    Require all denied
</Directory>
```

5. Restart Service and Make Html File in Website Folder

sudo mkdir /websites

sudo vi /websites/test.html

```
[nada_mohamed2243@localhost ~]$ sudo mkdir /websites
[nada_mohamed2243@localhost ~]$ sudo vi /websites/test.html
```

sudo systemctl restart httpd.service

```
[nada_mohamed2243@localhost ~]$ sudo systemctl restart httpd.service
```

6. Test

sudo setenforce 1

getenforce #Enforcing

```
[nada_mohamed2243@localhost ~]$ sudo setenforce 1
[nada_mohamed2243@localhost ~]$ sudo getenforce
Enforcing
[nada_mohamed2243@localhost ~]$
```

Openbrowser → Forbidden

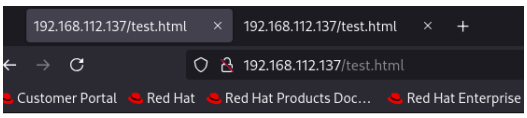
sudo setenforce 0

getenforce #Permissive

Openbrowser → Work

And send notifications in /var/log/messages

sudo cat /var/log/messages



ve are in test inside /var/www/html/

7. Change Context for Website Folders

ls -lZ /websites/

```
[nada_mohamed2243@localhost ~]$ sudo ls -lZ /websites/
total 4
-rwxr-xr-. 1 root root unconfined_u:object_r:default_t:s0 40 Feb 14 23:40 test.
html
[nada_mohamed2243@localhost ~]$
```

ls -ldZ /var/www/html/

drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/

```
[nada_mohamed2243@localhost ~]$ sudo ls -ldZ /var/www/html/
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 84 Feb 14 19:53
 /var/www/html/
[nada_mohamed2243@localhost ~]$
```

sudo chcon -R -t httpd_sys_content_t /websites/

```
[nada_mohamed2243@localhost ~]$ sudo chcon -R -t httpd_sys_content_t /websites/
```

ls -ldZ /websites/

```
[nada_mohamed2243@localhost ~]$ sudo ls -ldZ /websites
drwxr-xr-x. 2 root root unconfined_u:object_r:httpd_sys_content_t:s0 23 Feb 14 2
3:40 /websites
```

drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /websites/

ls -lZ /websites/

```
[nada_mohamed2243@localhost ~]$ sudo ls -Zl /websites
total 4
-rwxr-xr-. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 40 Feb 14 2
3:40 test.html
```

-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 test.html

Openbrowser ==> Enforcing Mode ===> Work

http://192.168.112.137/websites/test.html