HUMBER
Faculty of Applied Sciences & Technology

OPERATING SYSTEMS
CCGC-5000

Module - 06

WE ARE HUMBER

Agenda

Authentic information is available from the given resources in course outline and URL's mentioned from this slides, and this presentation is only supportive document to be read with the given resources and corrected accordingly if required..

- File Permissions
- Modifying file permissions
- Numeric, Symbolic method - chmod
- User permisisons
- umask
- suid, sgid, stickybit
- ACL, chroot
- MAC and DAC, SELinux introduction
- SELinux Key Terms, Definitions
- SELinux Context, Booleans
- Managing SELinux

Required Reading - Course TextBooks:
RHCSA-3,4,21

File Permissions

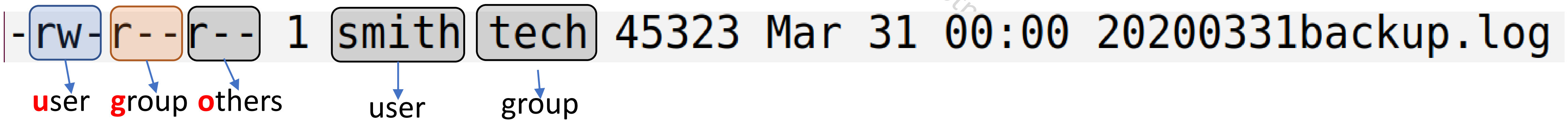
- Access permission to files are read, write and execute, denoted by r, w and x respectively
- **r** is to read permission, **w** is to write permission and **x** to execute permission
- Octal numbers for read is **4**, write is **2** and execute is **1**
- Permission for the files are given to user(owner), group and others
  - User is the one who owns the file with access permissions
  - Group is collection users, who are member of the group provided with access permissions
  - Others are users who is not owner of the file and not member of the group provided with access permissions

File Permissions

- File's Access permissions and ownership (user & group)

```
-rw-r--r-- 1 smith tech 45323 Mar 31 00:00 20200331backup.log
```

- Every file on your system has an accompanying set of permissions based on ownership



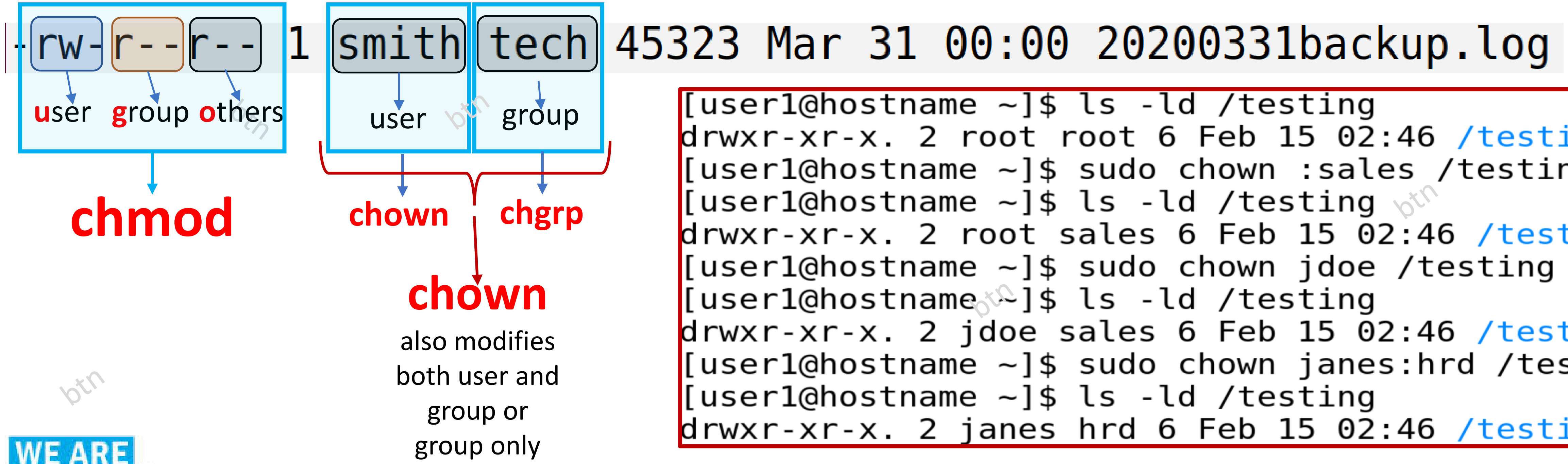
```
-rw-r--r-- 1 smith tech 45323 Mar 31 00:00 20200331backup.log
```

- Access permission specifies **r**,**w**,**x** permissions to user (**u**), group(**g**) and others(**o**) individually as required for the file
- Ownership permission is of two parts the **user** account and **group** account as owned by the file.

Modify File Permissions

- To modify the file access and ownership permissions we use
  - **chmod** to change file access permissions: **chmod** *permissions filename*
  - **chown** to change **user**(owner) **chown** *user filename*
  - **chown** to change **group** **chown** *:group filename*
  - **chown** to change **user** and **group** **chown** *user:group filename*

chgrp command can be used to change group ownership of a file

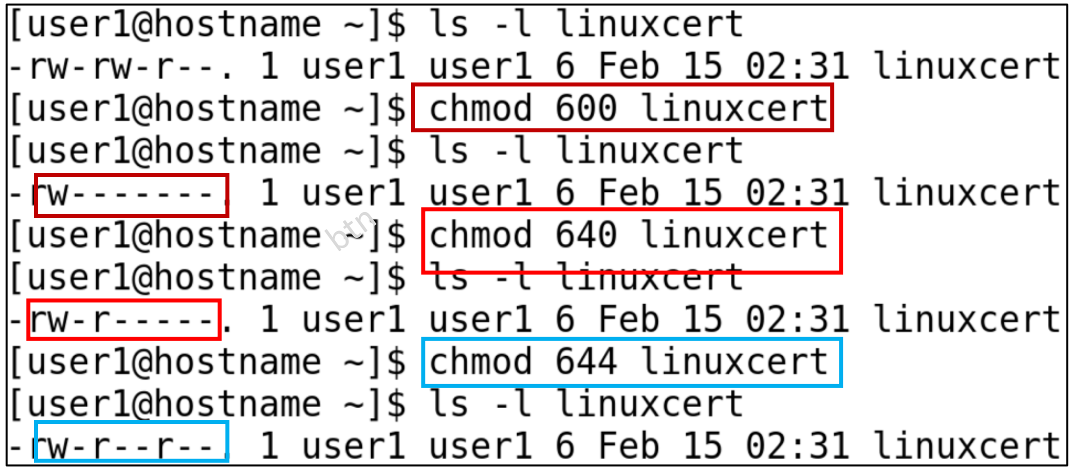

-rw-r--r-- 1 smith tech 45323 Mar 31 00:00 20200331backup.log
user
group
others
user
group
chmod
chown
chgrp
chown
also modifies both user and group or group only

```
[user1@hostname ~]$ ls -ld /testing
drwxr-xr-x. 2 root root 6 Feb 15 02:46 /testing
[user1@hostname ~]$ sudo chown :sales /testing
[user1@hostname ~]$ ls -ld /testing
drwxr-xr-x. 2 root sales 6 Feb 15 02:46 /testing
[user1@hostname ~]$ sudo chown jdoe /testing
[user1@hostname ~]$ ls -ld /testing
drwxr-xr-x. 2 jdoe sales 6 Feb 15 02:46 /testing
[user1@hostname ~]$ sudo chown janes:hrd /testing
[user1@hostname ~]$ ls -ld /testing
drwxr-xr-x. 2 janes hrd 6 Feb 15 02:46 /testing
```

chmod

- r,w,x access permissions modified in 2 ways
- **Numeric** method based on octal values
  - **r** read permission with octal value **4**
  - **w** write permission with octal value **2**
  - **x** execute permission with octal value **1**

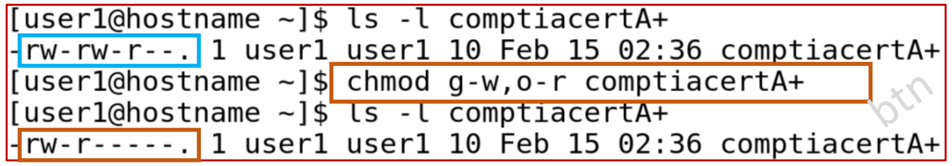
  chmod *octalnumber filename*



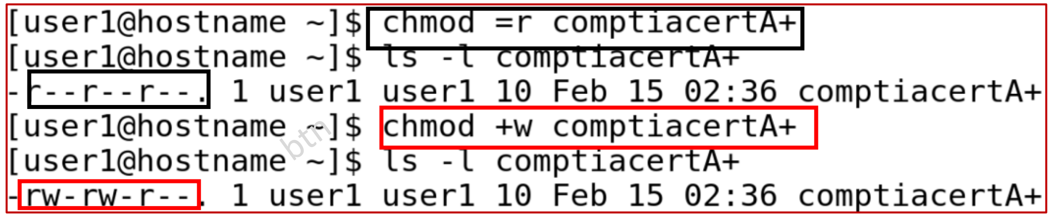
```
[user1@hostname ~]$ ls -l linuxcert
-rw-rw-r--. 1 user1 user1 6 Feb 15 02:31 linuxcert
[user1@hostname ~]$ chmod 600 linuxcert
[user1@hostname ~]$ ls -l linuxcert
-rw-------  1 user1 user1 6 Feb 15 02:31 linuxcert
[user1@hostname ~]$ chmod 640 linuxcert
[user1@hostname ~]$ ls -l linuxcert
-rw-r-----. 1 user1 user1 6 Feb 15 02:31 linuxcert
[user1@hostname ~]$ chmod 644 linuxcert
[user1@hostname ~]$ ls -l linuxcert
-rw-r--r--  1 user1 user1 6 Feb 15 02:31 linuxcert
```

- **Symbolic** method is based on
  - **u** for user
  - **g** for group
  - **o** for others
  - **a** for all (user, group and others)

Directory must have e**X**ecute permission with or without **r** or **w** or **rw** to access the directory even by the owner(user)



```
[user1@hostname ~]$ ls -l comptiacertA+
-rw-rw-r--. 1 user1 user1 10 Feb 15 02:36 comptiacertA+
[user1@hostname ~]$ chmod g-w,o-r comptiacertA+
[user1@hostname ~]$ ls -l comptiacertA+
-rw-r-----. 1 user1 user1 10 Feb 15 02:36 comptiacertA+
```



```
[user1@hostname ~]$ chmod =r comptiacertA+
[user1@hostname ~]$ ls -l comptiacertA+
-r--r--r--. 1 user1 user1 10 Feb 15 02:36 comptiacertA+
[user1@hostname ~]$ chmod +w comptiacertA+
[user1@hostname ~]$ ls -l comptiacertA+
-rw-rw-r--. 1 user1 user1 10 Feb 15 02:36 comptiacertA+
```

- **chmod** command is used in symbolic method by replacing octal number with u or g or o or a as required with + or - or =
- Operator + to add access, - to remove access and = to set exact access

umask

- By default access permissions are assigned to the files during creation.
- These default permissions can be modified with **umask**
- The same numeric values of r,w and x can be used.
- Directory permissions are considered in finding the umask.
- umask calculation (example -1) to have a default permission as rw- rw- r-- (664)when the file is created
  - The complete directory access permission = 777
  - Permission required is calculated for directory = 775 *(for directory it will be rwx rwx r-x)*
  - umask value (subtract the above) = 002
- umask calculation (example - 2) for default permission rw- r-- --- (640)
  - The complete directory access permission = 777
  - Permission required is calculated for directory = 750 *(for directory it will be rwx r-x ---)*
  - umask value (subtract the above) = 027
- For umask there will be four digits, first number is to default suid/sgid/sticky bit.

suid, sgid, stickybit

HUMBER
Faculty of Applied Sciences & Technology

- **suid** (set user ID) set in program files enabling any user to run the file with permissions of owner(user)
- suid by default is turned off in files
- Example, passwd command file

```
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd
```

- For suid files, in the users permission bits, **s** is set as the third bit.
- **chmod** can be used with the octal value 4 for suid.

- **sgid** (set group ID) set in program files enabling any user of the group to run the file with permissions of owner
- sgid by default is turned off in files
- Example, chage command file

```
-rwxr-sr-x 1 root shadow 71816 Mar 22  2019 /usr/bin/chage
```

- For sgid files, in the group permission bits, **s** is set as the third bit.
- **chmod** can be used with the octal 2 for sgid

- **sticky bit** limits who may rename or delete files within a directory and **t** set as third bit in others
- Setting sticky bit files in that directory may be unlinked or renamed only by a super user, the directory owner or the file owner.

```
drwxrwxrwt 18 root root 4096 Jun 10 14:44 /tmp
```

- **chmod** can be used with the octal value 1 for sticky bit.

WE ARE HUMBER

ACL - Access Control List

- ACL gives more control to file permissions and provide specific permission to a user that is not assigned in the given file permissions.
- Command **getfacl** gets files and directories ACL
- Command **setfacl** sets ACL's of files and directories
- **setfacl** is used with options *(refer man setfacl)*

-m modify	
-b delete all	u: user
-x remove	g: group
-R recursive	

- syntax to set acl:

  setfacl -m u:*username*:*permission filename*

  setfacl -m g:*groupname*:*permission filename*
- ACL mask displays the maximum effective permissions

chroot

- chroot is old as UNIX helps to isolate a directory set up temporarily as chroot directory
- It creates a confined space to run programs
- This helps
  - in development and testing purposes,
  - running unstable applications
  - running insecure applications
  - testing deprecated or new packages
- Recommended reading: https://help.ubuntu.com/community/BasicChroot

SELinux

- *Security Enhanced Linux* (SELinux) is an implementation of the *Mandatory Access Control* (MAC) architecture developed by the U.S. *National Security Agency* (NSA) in collaboration with other organizations and the Linux community for flexible, enriched, and granular security controls in Linux.
- MAC is integrated into the Linux kernel as a set of patches using the *Linux Security Modules* (LSM) framework that allows the kernel to support various security implementations, including SELinux.
- MAC provides an added layer of protection above and beyond the standard Linux *Discretionary Access Control* (DAC) security architecture, which includes the traditional file and directory permissions, ACL settings, setuid/setgid bit settings, su/sudo privileges, and so on.
- On DAC systems, access is controlled based on Linux user and group IDs. SELinux policy rules are checked after DAC rules.
- SELinux policy rules are not used if DAC rules deny access first.
- Read Chapter – 21 on SELinux

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/using_selinux/index

SELinux – Key Terms

- A subject is any user or process that accesses an object
- An object is a resource such as a file, directory, hardware device, network interface, port, pipe, or socket that a subject accesses
- An access is an action performed by the subject on an object
- A policy is a defined ruleset that is enforced system-wide and is used to analyze security attributes assigned to subjects and objects
- A context is a tag that SELinux uses to store security attributes for subjects and objects
- Labeling is the mapping between files in a filesystem with their file contexts.

SELinux

- MAC limits the ability of a *subject* (user or process) to access an *object* (file, directory, file system, device, network interface, port, pipe, socket, etc.) in order to reduce or eliminate the potential damage the subject may be able to cause to the system if compromised due to the exploitation of vulnerabilities in service processes, programs, or applications.
- MAC controls are fine-grained; they protect other services in the event one of the services is negotiated.
- To ensure this coarse-grained control, MAC uses a set of defined authorization rules called *policy* to examine security attributes associated with subjects and objects when a subject tries to access an object, and decides whether or not to permit this access attempt.
- These attributes are stored in *contexts* (a.k.a. *labels*) and are applicable to both subjects and objects.

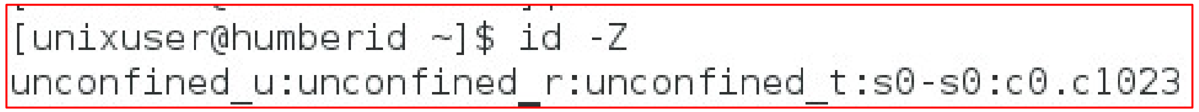
SELinux – Key Terms

- There are several predefined SELinux User identities that exist in the SELinux in the SELinux policy
- A Role is an attribute of the Role-Based Access Control (RBAC) security model that is part of SELinux
- A Type is an attribute of Type Enforcement (TE)
- A Domain determines the type of access that a process has
- Type enforcement identifies and limits a subject’s ability to access domains for process and types for files
- A level is an attribute of Multi-Level Security (MLS) and Multi-Category Security (MCS)

HUMBER
Faculty of Applied Sciences & Technology

SELinux Context & semanage

- Context set on Linux users can be viewed using , id command with –Z option.



```
[unixuser@humberid ~]$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

If an unconfined Linux user executes an application that SELinux policy defines as one that can transition from the `unconfined_t` domain to its own confined domain, the unconfined Linux user is still subject to the restrictions of that confined domain. The security benefit of this is that, even though a Linux user is running unconfined, the application remains confined. Therefore, the exploitation of a flaw in the application can be limited by the policy

- SELinux context for process, **ps –eZ**
- SELinux context for Files **ll –Z**
- Command **semanage** is SE Policy Management tool, *(Refer to semanage man pages)*
- Command **semanage** requires **policycoreutils-python-utils** for RHEL-8 and for RHEL-7 the package required is **policycoreutils-python**
- to view the mapping of users **semanage login -l**
- For context for Ports **semanage port -l**

WE ARE HUMBER

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/using_selinux/index

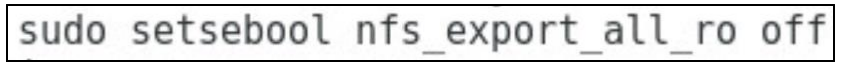
SELinux - seinfo, semodule

- The command **seinfo** allows user to query the components of a SELinux policy.
- The SELinux command **seinfo** requires **setools-console** package
- Familiarize **seinfo** command with various options
- To print users : **seinfo -u**
- To print roles : **seinfo -r**
- Try out more with **seinfo --help** or **seinfo -h**
- The command **semodule** is the tool used to manage SELinux policy modules, including installing, upgrading, listing and removing modules.
- To list the policy modules on a system **semodule -l**
- Refer **semodule** man pages for more information.

SELinux Booleans

- Booleans are on/off switches that are used by SELinux to determine whether to permit an action
- Booleans allow us to immediately activate or deactivate a particular rule in the SELinux policy without the need to recompile or reload it
- Command **getsebool** with **–a** option lists all Booleans along with their current values
- Command **setsebool** is used to enable or disable SELinux Booleans

  setsebool -P *selinux_boolean on/off*



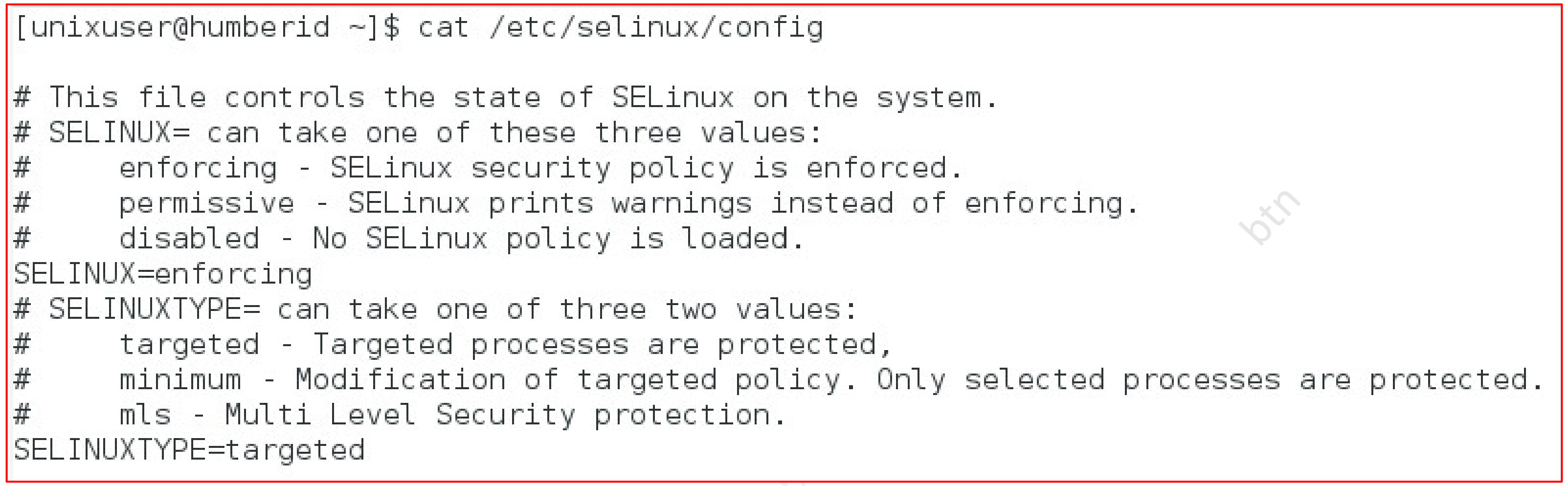
```
sudo setsebool nfs_export_all_ro off
```

- To permanently set the selinux boolean **–P** options is used with **setsebool**
- Alternatively **sestatus –b** & **seinfo -b** command lists SELinux Booleans
- Command **semanage boolean –l** may be used to list all the Booleans with a short description for each and their current settings

Managing SELinux

- **/etc/selinux/config** file is one of the key configuration files that controls the SELinux activation mode and sets its default type.
- The SELinux directive in the files sets the activation mode for SELinux
- **Enforcing** activates it and allows or denies actions based on policy rules
- **Permissive** activates SELinux, but permits all actions, it records all security violations; however it does not taken any action.
- **Disabled**, disables SELinux
- Command **getenforce** displays the current SELinux mode
- Command **setenforce** used to alter SELinux operating state, but this change is lost when system starts again.
- To make it persistent, need to edit the /etc/selinux/config file update the SELINUX directive and restart the system.

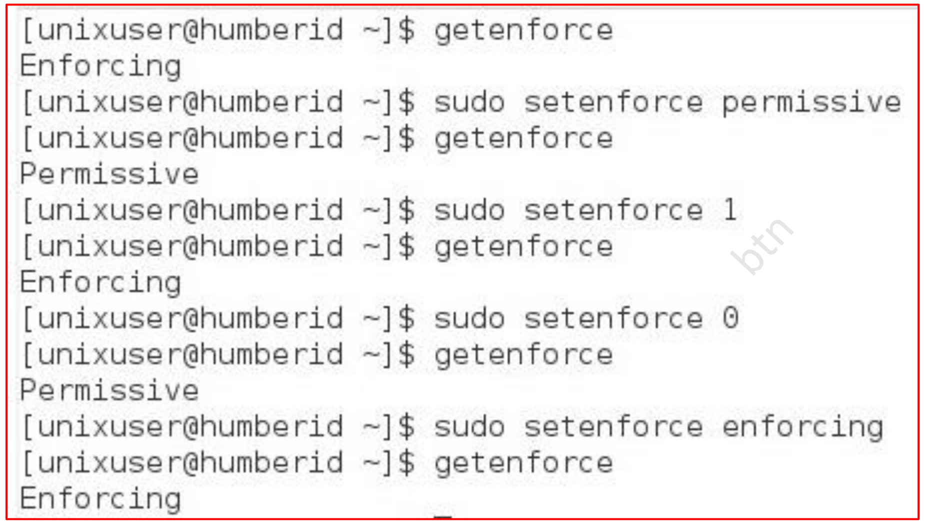
Managing SELinux



```
[unixuser@humberid ~]$ cat /etc/selinux/config

# 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 three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

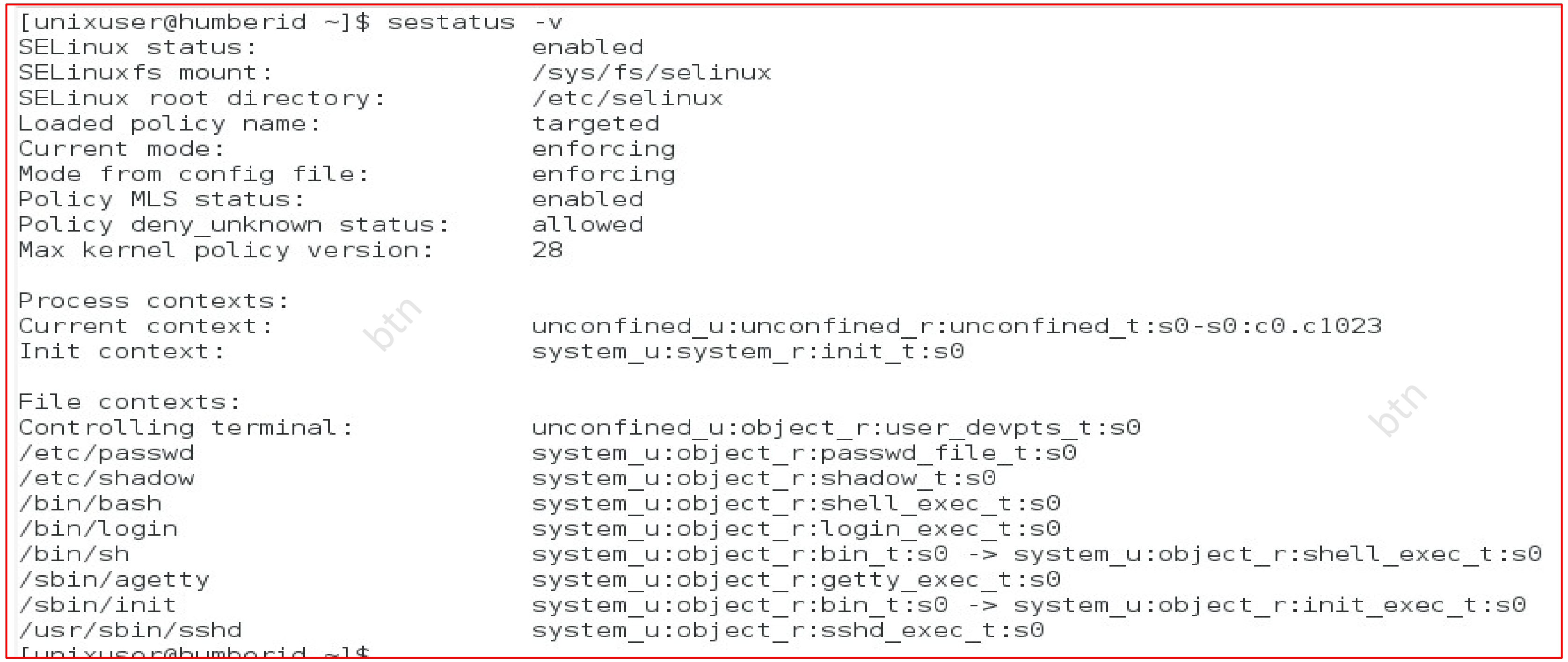
Note that when setting to Permissive, 0 (zero) can be used and to set as Enforcing, 1 can be used with **setenforce** command



```
[unixuser@humberid ~]$ getenforce
Enforcing
[unixuser@humberid ~]$ sudo setenforce permissive
[unixuser@humberid ~]$ getenforce
Permissive
[unixuser@humberid ~]$ sudo setenforce 1
[unixuser@humberid ~]$ getenforce
Enforcing
[unixuser@humberid ~]$ sudo setenforce 0
[unixuser@humberid ~]$ getenforce
Permissive
[unixuser@humberid ~]$ sudo setenforce enforcing
[unixuser@humberid ~]$ getenforce
Enforcing
```

Managing SELinux

- To query the running state of SELinux **sestatus** command can be used
- To report on security contexts set on files and process that are listed in /etc/sestatus.conf file **sestatus –v** can be used



```
[unixuser@humberid ~]$ sestatus -v
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
Max kernel policy version:      28

Process contexts:
Current context:                unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context:                   system_u:system_r:init_t:s0

File contexts:
Controlling terminal:           unconfined_u:object_r:user_devpts_t:s0
/etc/passwd                     system_u:object_r:passwd_file_t:s0
/etc/shadow                     system_u:object_r:shadow_t:s0
/bin/bash                       system_u:object_r:shell_exec_t:s0
/bin/login                      system_u:object_r:login_exec_t:s0
/bin/sh                         system_u:object_r:bin_t:s0 -> system_u:object_r:shell_exec_t:s0
/sbin/agetty                    system_u:object_r:getty_exec_t:s0
/sbin/init                      system_u:object_r:bin_t:s0 -> system_u:object_r:init_exec_t:s0
/usr/sbin/sshd                  system_u:object_r:sshd_exec_t:s0
[unixuser@humberid ~]$
```

SELinux user mapping

- When creating new Linux user and to map the user to SELinux user staff_u,

```
useradd –Z staff_u username
```

- To map an existing user to SELinux user user_u

```
semanage login –a –s user_u username
```

Modify SELinux Contexts

- To modify the context on the file to user_u and public_content_t

```
chcon –vu user_u –t public_content_t filename
```

- To add a new context to the SELinux policy to ensure it survives file system relabeling (if SELinux is disabled and then re-enabled)

```
semanage fcontext –a -t public_content_t "filename (/.*)?"
restorecon -Rv "filename"
```

- Use semanage and restorecon commands to add a file context to the SELinux policy and then apply it. This will prevent the context on file to reset to the original value should SELinux relabeling has to occur

SELinux alerts

- SELinux generates alerts for system activities when it runs in enforcing or permissive mode.
- Alerts are written to **/var/log/audit/audit.log** provided **auditd** daemon is running
- Alternatively it is written to **/var/log/messages** if auditd is inactive
- SELinux runs service daemon **setroubleshootd** that analyse the audit data to identify the potential cause of the rejection
- Command **sealert** is client interface to setroubleshoot service and reads the data and displays it for assessment.
- Package **setroubleshoot-server** is required to be installed.