SELinux is Security Enhanced Linux

Use of it:

It helps for access control

This we are doing through permission read,write,execute

There are 2 types of security controle

1. Discretionary access control when we use chmod we use read write access permission as per needed i.e discretionary access control
2. Mandatory—
   In mandatory control, there are certain policies. These policies decide what is right or wrong
   No user will decide whether to give access or not
   When you start any service on your machine it access files or directories on your machine.
   Example. When you start FTP or HTTP service they access some folder or files on your machine

   To check whether FTP or http are installed on your machine
   $ rpm -q httpd
   $rpm -q ftpd

   To check whether security is on
   $getenforcing
   Enforcing

   To check yum repo is configured or not

$yum repolist

To check yum repo mounted at sr0 or not

mount |grep sr0

To install httpd and ftpd

$ yum -y install vsftpd httpd

For httpd service the default folder is /var/www so by default it access all files in this folder

For Ftp service the default directory is /var/ftp -------It uploads or download files from this folder

For ftp service and for httpd service it is predefined that they can access data from above mentioned folders. Its predefined in SELinux security policy

But if we try to change it and we ask ftpd /httpd service to read data from these different folder or if it tries to read data from some users home folder then SELinux will not allow to read that folder

Actually SELinux is not decide that http service will access /var/www, but it specifies that this service will access the folder which has some specific label

In SELinux there is a concept called as context

1. Context of all files and folders
2. Context of all services running on your machine
3. Context of all users created on your system

Basically, context is a label

When you install Linux then the files and folders on your machine or users created on your machine is assigned one label.

To see the label the command is

$ls -ldZ

-Z is used to see SELinux label

It looks as shown in the figure

To see the label of file use command

$ls -lZ myfile.txt

Each label is divided into 4 parts

The first part

Unconfined_u ------- indicates user

Object_r ---------shows role

admin_home_t ------- shows type admin_home indicates that this file is in admin home folder

s0 ----indicates level

If you open a file from some other user then the label will show user_home

```
[root@server5 ~]# ls -ldZ /home/sumit
drwx------. sumit sumit unconfined_u:object_r:user_home_dir_t:s0 /home/sumit
[root@server5 ~]#
```

Every folder has its own context and usually all files in same folder are in same context(means label)

Eaxample

For /boot ------- it is boot

For /var it is var_t

All files in these folders will have same context

```
[root@server5 ~]# ls -lZ /root/file10.txt
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 /root/file10.txt
[root@server5 ~]#
[root@server5 ~]# ls -ldZ /home/sumit
drwx------. sumit sumit unconfined_u:object_r:user_home_dir_t:s0 /home/sumit
[root@server5 ~]#
[root@server5 ~]# ls -ldZ /boot
dr-xr-xr-x. root root system_u:object_r:boot_t:s0       /boot
[root@server5 ~]#
[root@server5 ~]#
[root@server5 ~]#
[root@server5 ~]# ls -ldZ /var
drwxr-xr-x. root root system_u:object_r:var_t:s0        /var
[root@server5 ~]#
```

Even for all services running on your system also has its own context(i.e label)

Lets start httpd service

$systemctl start httpd

$ps -efZ |grep httpd

```
[root@server5 ~]# systemctl start httpd
[root@server5 ~]#
[root@server5 ~]# ps -efZ | grep httpd
system_u:system_r:httpd_t:s0    root      4883     1  1 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4884  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4885  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4886  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4887  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
```

If you check the folder /var/www or files in the folder all will have similar context

```
[root@server5 ~]# systemctl start httpd
[root@server5 ~]#
[root@server5 ~]# ps -efZ | grep httpd
system_u:system_r:httpd_t:s0    root      4883     1  1 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4884  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4885  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4886  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4887  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4888  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 4896 28378  0 16:36 pts/0 00:00:00 grep --color=auto
httpd
[root@server5 ~]# ls -ldZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www
[root@server5 ~]# ls -ldZ /var/www/html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html
[root@server5 ~]#
```

So SELinux defined which process will access which folder or file is predefined so the service with label httpd_t can access folder with label httpd_sys_content is predefined.

Like this many labels are accessible to service with label httpd_t. Like if you check /etc/httpd/

Its label is https_config_t , this labe

```
root@server5 ~]# systemctl start httpd
root@server5 ~]#
root@server5 ~]# ps -efZ | grep httpd
ystem_u:system_r:httpd_t:s0    root      4883     1  1 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
ystem_u:system_r:httpd_t:s0    apache    4884  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
ystem_u:system_r:httpd_t:s0    apache    4885  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
ystem_u:system_r:httpd_t:s0    apache    4886  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
ystem_u:system_r:httpd_t:s0    apache    4887  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
ystem_u:system_r:httpd_t:s0    apache    4888  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
nconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 4896 28378  0 16:36 pts/0 00:00:00 grep --color=auto
ttpd
root@server5 ~]# ls -ldZ /var/www
rwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www
root@server5 ~]# ls -ldZ /var/www/html
rwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html
root@server5 ~]#
root@server5 ~]# ls -ldZ /etc/httpd/
rwxr-xr-x. root root system_u:object_r:httpd_config_t:s0 /etc/httpd/
root@server5 ~]#
```

For httpd there is a log file also, you may check label for it

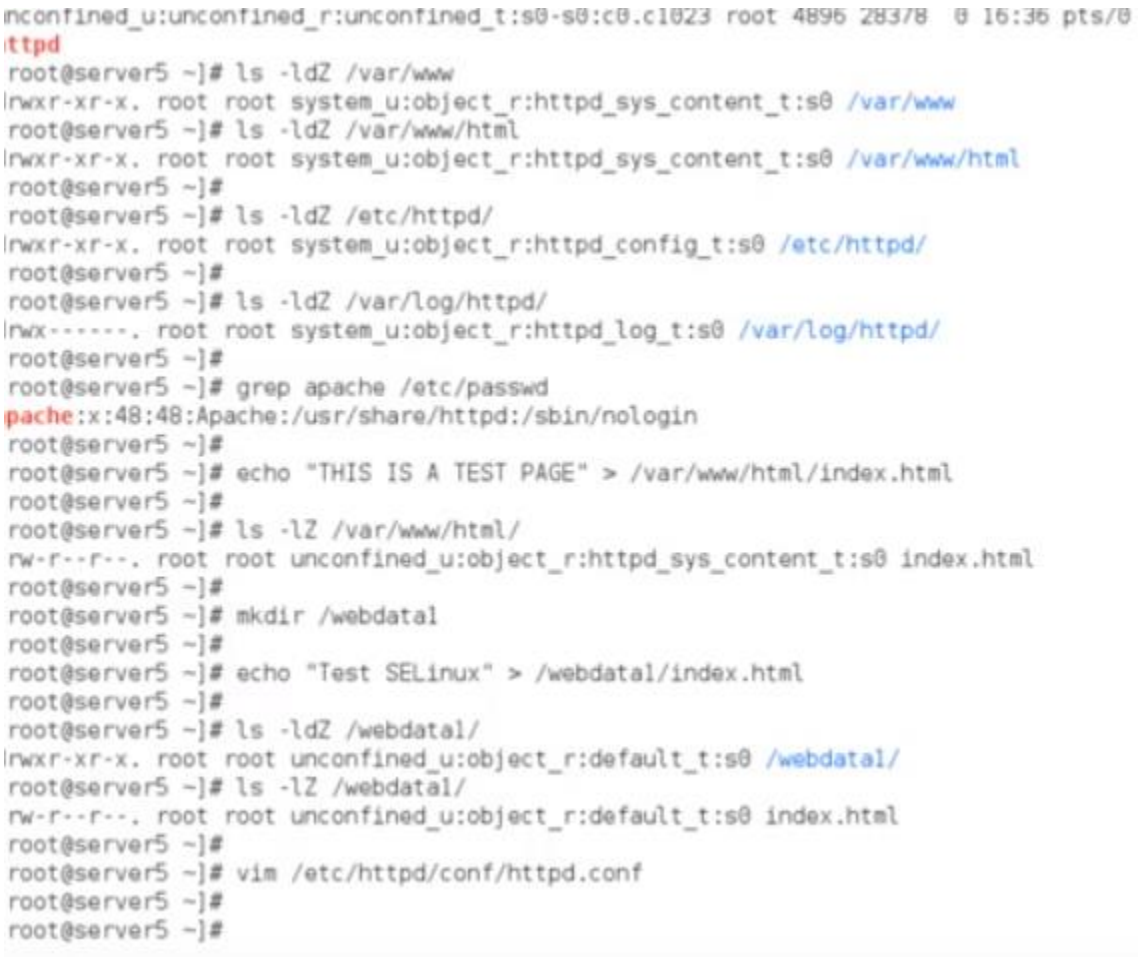
```
[root@server5 ~]# systemctl start httpd
[root@server5 ~]#
[root@server5 ~]# ps -efZ | grep httpd
system_u:system_r:httpd_t:s0    root      4883     1  1 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4884  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4885  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4886  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4887  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4888  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 4896 28378  0 16:36 pts/0 00:00:00 grep --color=auto
httpd
[root@server5 ~]# ls -ldZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www
[root@server5 ~]# ls -ldZ /var/www/html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html
[root@server5 ~]#
[root@server5 ~]# ls -ldZ /etc/httpd/
drwxr-xr-x. root root system_u:object_r:httpd_config_t:s0 /etc/httpd/
[root@server5 ~]#
[root@server5 ~]# ls -ldZ /var/log/httpd/
drwx------. root root system_u:object_r:httpd_log_t:s0 /var/log/httpd/
[root@server5 ~]#
```

But it will not be able to access a folder with label admin_home_t or user)folder_t

If you see httpd files are accessible to user apache and if anyone assigns read ,write execute permission to apache user for any admin directory. Still apache user will nort be able to access those files because SELinux security will stop the user from accessing.

```
[root@server5 ~]# systemctl start httpd
[root@server5 ~]#
[root@server5 ~]# ps -efZ | grep httpd
system_u:system_r:httpd_t:s0    root      4883     1  1 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4884  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4885  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4886  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4887  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache    4888  4883  0 16:36 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 4896 28378  0 16:36 pts/0 00:00:00 grep --color=auto
httpd
[root@server5 ~]# ls -ldZ /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www
[root@server5 ~]# ls -ldZ /var/www/html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html
[root@server5 ~]#
[root@server5 ~]# ls -ldZ /etc/httpd/
drwxr-xr-x. root root system_u:object_r:httpd_config_t:s0 /etc/httpd/
[root@server5 ~]#
[root@server5 ~]# ls -ldZ /var/log/httpd/
drwx------. root root system_u:object_r:httpd_log_t:s0 /var/log/httpd/
[root@server5 ~]#
[root@server5 ~]# grep apache /etc/passwd
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
[root@server5 ~]#
[root@server5 ~]# echo "THIS IS A TEST PAGE" > /var/www/html/index.html
[root@server5 ~]#
[root@server5 ~]# ls -lZ /var/www/html/
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index.html
[root@server5 ~]#
```

If we create index.html file in /var/www/html/index.html



How to use semanage

How to Use Semanage Command for SELinux Policy

Updated November 8, 2018LINUX COMMANDS

Semanage is a tool used to configure certain elements of SELinux policy without modifying or recompiling policy sources. This includes mapping Linux usernames to SELinux user identities and security context mappings for objects like network ports, interfaces, and hosts.

By default, SELinux only allows known services to bind to known ports. If we want to modify a service to use a non-default port we will need to modify the port type with the semanage command.

semanage command ----- how to list, create/add and delete port types on RPM-based distributions like CentOS and RedHat.

Listing Ports with Semanage

The basic command for listing all ports is

```
# semanage port -l

SELinux Port Type              Proto    Port Number



afs3_callback_port_t           tcp      7001

afs3_callback_port_t           udp      7001

afs_bos_port_t                 udp      7007

afs_fs_port_t                  tcp      2040

afs_fs_port_t                  udp      7000, 7005

afs_ka_port_t                  udp      7004

afs_pt_port_t                  tcp      7002

afs_pt_port_t                  udp      7002

...
```

To list port numbers of a specific service like http, use this command:

```
# semanage port -l | grep -w http_port_t



http_port_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

Similarly for mysqld

```
# semanage port -l | grep -w mysqld_port_t

mysqld_port_t                  tcp      1186, 3306, 63132-63164
```

To find port names with a specific port number in it, use this command:

```
# semanage port -l | grep 53

apertus_ldp_port_t              tcp      539

apertus_ldp_port_t              udp      539

dns_port_t                      tcp      53

dns_port_t                      udp      53
```

Creating or Adding Ports with Semanage

In this example, we will create a new port for http and assign it to tcp port 2222. The `-a` option is to add a new port, the `-t` option specifies the SELinux type, and the `-p` option is to specify the protocol to use (in this case tcp).

```
# semanage port -a -t http_port_t -p tcp 2222
```

to view the newly created port, we use the command list command with the `-C` option to show only customizations.

```
# semanage port -lC
```

SELinux Port Type Proto Port Number

```
http_port_t                     tcp      2222
```

To assign a range of ports numbers to a specific port, use the command:

```
# semanage port -a -t http_port_t -p tcp 2223-2225
```

Now, we can see the port range here.

```
# semanage port -lC

SELinux Port Type               Proto    Port Number



http_port_t                     tcp      2223-2225
```

If you try to add another entry with the same values like you used before, you get the error:

```
ValueError: Port tcp/2222 already defined
```

To override an existing port that was already created, use the `-m` option to modify:

```
# semanage port -m -t unreserved_port_t -p tcp 2222
```

Now if we list all ports we will see the change.

```
# semanage port -lC

SELinux Port Type              Proto    Port Number



unreserved_port_t              tcp      2222
```

Deleting Ports with Semanage

We use the option `-d` to delete a port record. To delete unreserved_port_t on tcp port 2222, we use the command:

```
# semanage port -d -t unreserved_port_t -p tcp 2222
```

To delete a range of ports, use the command:

```
# semanage port -d -t http_port_t -p tcp 2223-2225
```

If you run the customized list command and it returns nothing, then the entry has been removed.

Using Semanage-Permmissive

Semanage permissive is used to add or remove SELinux Policy permissive modules.

To list all permissive modules, use the `-l` option:

```
# semanage permissive -l
```

```
Customized Permissive Types




Builtin Permissive Types



sanlk_resetd_t

hsqldb_t

systemd_hwdb_t

blkmapd_t

ipmievd_t

targetd_t
```

To create httpd_t a permissive domain, use the `-a` option:

```
# semanage permissive -a httpd_t
```

Now, let's check all permissive modules:

```
# semanage permissive -l



Customized Permissive Types



httpd_t



Builtin Permissive Types



```

```
sanlk_resetd_t

hsqldb_t

systemd_hwdb_t

blkmapd_t

ipmievd_t
```

To delete a permissive type we just created, we use the `-d` option.

```
# semanage permissive -d httpd_t



libsemanage.semanage_direct_remove_key: Removing last permissive_
httpd_t module (no other permissive_httpd_t module exists at anot
her priority).
```