SELinux Lab Exercises

1. **Basic Concepts**
2. **Files and Processes**
    a. Creating test user accounts
    b. SELinux File Contexts
    c. SELinux Process Contexts
    d. How Processes Access Resources
3. **Users**
    a. SELinux Users
    b. Restricting Switched User Access
    c. Restricting Permissions to Run Scripts
    d. Restricting Access to Services
    e. SELinux Audit Logs

1. Basic Concepts

- Installing Apache service

  [s1mtech@localhost ~]$ sudo yum install httpd

- Start the daemon manually

  [s1mtech@localhost ~]$ sudo service httpd start

- Check the status of the service

  [s1mtech@localhost ~]$ sudo service httpd status

  Observe service name, status and pid

- Installing SFTP service

  [s1mtech@localhost ~]$ sudo yum install vsftpd

- Start the daemon manually

  [s1mtech@localhost ~]$ sudo service vsftpd start

- Check the status of the service

  [s1mtech@localhost ~]$ sudo service vsftpd status

  Observe Service name, status and pid

- Check what SELinux packages are installed on your CentOS 7 system

  [s1mtech@localhost ~]$ sudo rpm -qa | grep selinux

  How many packages are there?

- Check the current SELinux mode

  [s1mtech@localhost ~]$ sudo getenforce

- check the current SELinux status

  [s1mtech@localhost ~]$ sudo sestatus

- View the the main configuration file for SELinux

  [s1mtech@localhost ~]$ sudo cat /etc/selinux/config

- Try changing the SELinux mode to permissive

  [s1mtech@localhost ~]$ sudo setenforce permissive
- Check the current SELinux status

  [s1mtech@localhost ~]$ sudo sestatus
- Try changing the SELinux mod to enforcing

  [s1mtech@localhost ~]$ sudo setenforce enforcing
- check the current SELinux status

  [s1mtech@localhost ~]$ sudo sestatus

- List the SELinux policy modules currently loaded into memory

  [s1mtech@localhost ~]$ sudo semodule -l | less

- List the policy files

  [s1mtech@localhost ~]$ ls –l /etc/selinux/targeted/modules/active/modules/
- To list the active policy

  [s1mtech@localhost ~]$ ls -l /etc/selinux/targeted/modules/active/modules/

- List the SELinux booleans

  [s1mtech@localhost ~]$ sudo semanage boolean -l | less

- Check the current value of a boolean

  [s1mtech@localhost ~]$ sudo getsebool ftpd_anon_write
- Change the value of the boolean

  [s1mtech@localhost ~]$ sudo setsebool ftpd_anon_write on
- Check the new value of the boolean

  [s1mtech@localhost ~]$ sudo getsebool ftpd_anon_write

2. Files and Processes

a. **Creating test user accounts**

- Create four user accounts
  - regularuser
  - switcheduser
  - guestuser
  - Restricteduser

```
[s1mtech@localhost ~]$ sudo useradd -c "Regular User" regularuser
[s1mtech@localhost ~]$ sudo passwd regularuser
[s1mtech@localhost ~]$ sudo useradd -c "Switched User" switcheduser
[s1mtech@localhost ~]$ sudo passwd switcheduser
[s1mtech@localhost ~]$ sudo useradd -c "Guest User" guestuser
[s1mtech@localhost ~]$ sudo passwd guestuser
[s1mtech@localhost ~]$ sudo useradd -c "Restricted Role User" restricteduser
[s1mtech@localhost ~]$ sudo passwd restricteduser
```

b. **SELinux File Contexts**

- Normal listing of the files

  [s1mtech@localhost ~]$ ls -l /etc/*.conf
- Show the security context of the files

  [s1mtech@localhost ~]$ ls -Z /etc/*.conf
- Check the type contexts for user home directories

  [s1mtech@localhost ~]$ ls -Z /home

c. **SELinux Process Contexts**

- Check the process security contexts

  [s1mtech@localhost ~]$ ps -efZ | grep 'httpd\|vsftpd'

d. **How Processes Access Resources?**

- Create a file within the default home directory of the web server and check its context:

  [s1mtech@localhost ~]$ sudo touch /var/www/html/index.html

  [s1mtech@localhost ~]$ ls -Z /var/www/html/*

- Use the sesearch command to check the type of access allowed for the httpd daemon

  [s1mtech@localhost ~]$ sudo sesearch --allow --source httpd_t --target httpd_sys_content_t --class file

- Edit the content of the file /var/www/html/index.html

  [s1mtech@localhost ~]$ sudo vi /var/www/html/index.html

```
<html>
        <title>
                This is a test web page
        </title>
        <body>
                <h1>This is a test web page</h1>
        </body>
</html>
```

- Change the permission of the /var/www/ folder and its contents, followed by a restart of the httpd daemon

  [s1mtech@localhost ~]$ chmod -R 755 /var/www

  [s1mtech@localhost ~]$ service httpd restart

- Access the html using firefox http://localhost/index.html

  Or

  [s1mtech@localhost ~]$ wget http://localhost/index.html

- Now change the file type to **var_t**

  [s1mtech@localhost ~]$ sudo chcon --type var_t /var/www/html/index.html

- Confirm the type change

  [s1mtech@localhost ~]$ ls -Z /var/www/html/

- Access the html again using firefox http://localhost/index.html

  Or

  [s1mtech@localhost ~]$ wget http://localhost/index.html

- To make things work again, let's change the file type with the restorecon command

  [s1mtech@localhost ~]$ sudo restorecon -v /var/www/html/index.html

3. Users

a. **SELinux Users**

- When SELinux is enforced, each regular Linux user account is mapped to an SELinux user account

  [s1mtech@localhost ~]$ sudo semanage login -l

- To see what SELinux users are available in the system

  [s1mtech@localhost ~]$ sudo semanage user –l

- To know the security context of root

  [s1mtech@localhost ~]$ sudo id –Z

- Start four new SSH sessions with the four users you created from separate terminal windows

  [s1mtech@localhost ~]$ ssh regularuser@localhost

  [s1mtech@localhost ~]$ ssh switcheduser@localhost

  [s1mtech@localhost ~]$ ssh guestuser@localhost

  [s1mtech@localhost ~]$ ssh restricteduser@localhost

- Check the security context of regular user

  [regularuser@localhost ~]$ id -Z

We had seen the list of a number of SELinux users before:

- **guest_u**: This user doesn't have access to X-Window system (GUI) or networking and can't execute su / sudo command.
- **xguest_u**: This user has access to GUI tools and networking is available via Firefox browser.
- **user_u**: This user has more access than the guest accounts (GUI and networking), but can't switch users by running su or sudo.
- **staff_u**: Same rights as user_u, except it can execute sudo command to have root privileges.
- **system_u**: This user is meant for running system services and not to be mapped to regular user accounts.

b. **Restricting Switched User Access**

- First check the user's ability to switch to another account

  [regularuser@localhost ~]$ su – switcheduser

- Go to s1mtech terminal and map regularuser to user_u

  [s1mtech@localhost ~]$ sudo semanage login -a -s user_u regularuser

- Go back to regularuser's terminal window and switch back from switcheduser

  [switcheduser@localhost ~]$ logout

- Try to change to switcheduser again

  [regularuser@localhost ~]$ su – switcheduser

- regularuser is now mapped to user_u

  [regularuser@localhost ~]$ id –Z

c. **Restricting Permissions to Run Scripts**

- SELinux allows users mapped to the guest_t account to execute scripts from their home directories

  [s1mtech@localhost ~]$ sudo getsebool allow_guest_exec_content

- Change the SELinux user mapping for the guestuser account

  [s1mtech@localhost ~]$ sudo semanage login -a –s guest_u guestuser

- Verify the action by running the semanage login –l

  [s1mtech@localhost ~]$ sudo semanage login –l

- Verify that you are in the guestuser home directory:

  [guestuser@localhost ~]$ pwd

- Create the script:

  [guestuser@localhost ~]$ vi myscript.sh

- Script contents:

  #!/bin/bash echo "This is a test script"

- Make the script executable:

  [guestuser@localhost ~]$ chmod u+x myscript.sh

- Execute the script

  [guestuser@localhost ~]$ ~/myscript.sh

- Go back to the root terminal window and change the boolean setting allow_guest_exec_content to off and verify it

  [s1mtech@localhost ~]$ sudo setsebool allow_guest_exec_content off

  [s1mtech@localhost ~]$ sudo getsebool allow_guest_exec_content

- Going back to the console logged in as guestuser we try to run the script again

[guestuser@localhost ~]$ ~/myscript.sh

- Check the denial in the log

  [s1mtech@localhost ~]$ sudo grep "SELinux is preventing" /var/log/messages

- Run the sealert command with the ID for more information

  [s1mtech@localhost ~]$ sudo sealert -l 8343a9d2-ca9d-49db-9281-3bb03a76b71a (Replace this id with that of yours)

d. **Restricting Access to Services**

- Stop the httpd daemon

  [s1mtech@localhost ~]$ sudo service httpd stop

- Go to the terminal of restricted user and check the security context

  [restricteduser@localhost ~]$ id –Z

- Try to start the http daemon

  [restricteduser@localhost ~]$ service httpd start

- Go to the terminal of s1mtech and add restricted user to /etc/sudoers file

  [s1mtech@localhost ~]$ sudo visudo

- Add the following line, save and exit

  restricteduser ALL=(ALL) ALL

- Now restricted user can start and stop httpd service with sudo privileges

  [restricteduser@localhost ~]$ sudo service httpd start

- Stop the service

  [restricteduser@localhost ~]$ sudo service httpd stop

- Map the restricteduser to the SELinux user_r account

  [s1mtech@localhost ~]$ sudo semanage login -a -s user_u restricteduser

- Show the roles user_u can assume

  [s1mtech@localhost ~]$ sudo seinfo -uuser_u –x

- Show the domains the user_r role is authorized to enter

  [s1mtech@localhost ~]$ sudo seinfo -ruser_r –x

- Check whether user_r can enter in to httpd_t domain

  [s1mtech@localhost ~]$ sudo seinfo -ruser_r -x | grep httpd

- Go back to the restricteduser account's terminal window and try to start the httpd daemon now

  [restricteduser@localhost ~]$ sudo service httpd start

e. **SELinux Audit Logs**

- Error messages are logged in specific files and they can provide detailed information about access denials
- In a CentOS 7 system you can look at two files:

  /var/log/audit/audit.log

  /var/log/messages

- Try to look at all the error messages related to the httpd daemon

  [s1mtech@localhost ~]$ sudo ausearch -m avc -c httpd

- type=AVC and avc: AVC stands for *Access Vector Cache*. SELinux caches access control decisions for resource and processes. This cache is known as the Access Vector Cache (AVC). That's why SELinux access denial messages are also known as "AVC denials". These two fields of information are saying the entry is coming from an AVC log and it's an AVC event.
- denied { getattr }: The permission that was attempted and the result it got. In this case the get attribute operation was denied.
- pid=10204. This is the process id of the process that attempted the access.
- comm: The process id by itself doesn't mean much. The comm attribute shows the process command. In this case it's httpd. Immediately we know the error is coming from the web server.
- path: The location of the resource that was accessed. In this case it's a file under /www/html/index.html.
- dev and ino: The device where the target resource resides and its inode address.
- scontext: The security context of the process. We can see the source is running under the httpd_t domain.
- tcontext: The security context of the target resource. In this case the file type is default_t.
- tclass: The class of the target resource. In this case it's a file.

[s1mtech@localhost ~]$ sudo cat /var/log/messages | grep "SELinux is preventing"

[s1mtech@localhost ~]$ sudo sealert -l e9e6c6d8-f217-414c-a14e-4bccb70cfbce

References:

https://www.digitalocean.com/community/tutorials/an-introduction-to-selinux-on-centos-7-part-1-basic-concepts

https://www.digitalocean.com/community/tutorials/an-introduction-to-selinux-on-centos-7-part-2-files-and-processes#top

https://www.digitalocean.com/community/tutorials/an-introduction-to-selinux-on-centos-7-part-3-users