Lab Setup – I am using Ubuntu 20.04 as referenced in the video because certspotter was not working on the apt install command, using the SeedLabs Ubuntu 20.04 prebuilt VM I was able to get this command working to move forward with the lab.

A screenshot of a computer

Description automatically generated

**1. AppArmor with myscript.sh**

1. myscript.sh

gedit myscript.sh

chmod 777 myscript.sh

A screenshot of a computer

Description automatically generated

Here is the current shell code for this step:

A screen shot of a computer

Description automatically generated

1. Profile Creation

aa-genprof /home/seed/lab7/myscript.sh

A screenshot of a computer

Description automatically generated

During profile creation, touch, cat and rm were given inherited permissions and /bin/bash was allowed to run.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Profile generation is complete for the program myscript.sh.

The program is able to run properly because it is running within the confines of the profile, and it can be seen in the other terminal that the profile for the program was generated successfully.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Change the owner’s right from w to r.

A screenshot of a computer

Description automatically generated

apparmor-parser:

A screen shot of a computer

Description automatically generated

Executing the shell script with updated permissions change:

A screenshot of a computer code

Description automatically generated

With the permission change and the AppArmor profile updated, it is observed that the shell code is encountering permission denied when executing as seen above. This means that the change in the AppArmor profile and updating the profile was successfully enforced on the myscript.sh shell code.

(4) Creating a new file using touch

myscript.sh updated version:

A screenshot of a computer

Description automatically generated

Execution of myscript.sh:

A screen shot of a computer

Description automatically generated

The new file cannot be created because touch is being enforced inside of AppArmor. It is observed in the above screenshot, “touch: cannot touch ‘test.txt’ : Permission denied”, that the AppArmor profile enforced the denial of the permission even though this command in the script was added after the profile creation. By running ls we can confirm that the file was not created.

(5) Change profile right of touch or rm

In the profile I will be changing the profile lines:

/usr/bin/rm mrix,

/usr/bin/touch mrix,

To:

/usr/bin/rm mr,

/usr/bin/touch mr,

I am modifying the ability of those commands to execute by removing the x and I permission. At this point I will also have to add back in the w permission so that the script can manage these files. This will allow me to observe the outcome of the above permission changes.

owner /home/\*/lab7/test.txt r,

To:

owner /home/\*/lab7/test.txt rw,

Profile:

A screenshot of a computer program

Description automatically generated

myscript.sh: A screenshot of a computer

Description automatically generated

Reloading AppArmor profile:

A screen shot of a computer

Description automatically generated

Executing myscript.sh:

A screenshot of a computer

Description automatically generated

Looking at the outcome of the script when executed, I can see that the touch command has been denied to execute in lines 4 and 9 which means that the permission change from mrix to mr has successfully changed the ability of touch to execute. When cat is run it is still able to read from the mydata.data file since the sentence added was added using the echo command earlier in the shell code. It is also seen that the change of mrix to mr for the permissions of the rm command were successful. The permission to execute rm was denied. This is confirmed by using ls to see if the file mydata.data is in the folder, which as seen in the above screenshot, the file has not been removed by the shell script.

**2.** AppArmor and Ping Command

Ping Command w/o AppArmor Profile:

A screenshot of a computer

Description automatically generated

Profile Creation:

aa-autodep ping

A screenshot of a computer

Description automatically generated

ping command is in complain mode:

A black line with white text

Description automatically generated

Setting ping to enforce mode:

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Testing ping again:

A screen shot of a computer

Description automatically generated

In the screenshot it can be observed that even as the root user, I am unable to use the ping command as permission is being denied with the enforced AppArmor profile.

Changing the permission of ping to allow in the profile:

A screenshot of a computer program

Description automatically generated

Testing ping with updated profile:

A screenshot of a computer

Description automatically generated

The ping command can now successfully be executed.

AppArmor profile for ping:

A screenshot of a computer

Description automatically generated

The ping command is now secured using AppArmor and allowed to be executed through the profile permissions.

**Conclusion:**

In this lab we created and secured a shell script, myscript.sh, and modified its behavior, observing how AppArmor profiles can effectively enforce or relax specific restrictions on file and command access. This exercise highlighted the granular control AppArmor offers, as even slight changes in profile permissions directly impacted the execution outcomes of our script. Additionally, our experimentation with the ping command provided an insightful perspective on how system commands can be secured or allowed using AppArmor, demonstrating its potential in a real-world security context. This lab emphasized the importance of understanding and managing application-level permissions in enhancing system security.