

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.

```
root@VM: /home/seed
Unpacking apparmor-utils (2.13.3-7ubuntu5) ...
Selecting previously unselected package certspotter.
Preparing to unpack .../3-certspotter_0.9-2_amd64.deb ...
Unpacking certspotter (0.9-2) ...
Selecting previously unselected package libapparmor-perl:amd64.
Preparing to unpack .../4-libapparmor-perl_2.13.3-7ubuntu5_amd64.deb ...
Unpacking libapparmor-perl:amd64 (2.13.3-7ubuntu5) ...
Selecting previously unselected package apparmor-easyprof.
Preparing to unpack .../5-apparmor-easyprof_2.13.3-7ubuntu5_all.deb ...
Unpacking apparmor-easyprof (2.13.3-7ubuntu5) ...
Selecting previously unselected package apparmor-notify.
Preparing to unpack .../6-apparmor-notify_2.13.3-7ubuntu5_all.deb ...
Unpacking apparmor-notify (2.13.3-7ubuntu5) ...
Setting up python3-libapparmor (2.13.3-7ubuntu5) ...
Setting up libapparmor-perl:amd64 (2.13.3-7ubuntu5) ...
Setting up certspotter (0.9-2) ...
Setting up apparmor-notify (2.13.3-7ubuntu5) ...
Setting up python3-apparmor (2.13.3-7ubuntu5) ...
Setting up apparmor-easyprof (2.13.3-7ubuntu5) ...
Setting up apparmor-utils (2.13.3-7ubuntu5) ...
Processing triggers for man-db (2.9.1-1) ...
root@VM:/home/seed#
```

1. AppArmor with myscript.sh

(1) myscript.sh

gedit myscript.sh

chmod 777 myscript.sh

```
myscript.sh
/home/seed/lab7
1 #!/bin/bash
2
3 echo "good thing" > mydata.data
4 touch mydata.data
5 cat mydata.data
6 ls -l mydata.data
7 rm mydata.data

root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# gedit myscript.sh
```

Here is the current shell code for this step:

```

Open  ▼  [+]
```

myscript.sh
/home/seed/lab7

```

Save  ≡  -  □  ✕

1 #!/bin/bash
2
3 echo "good thing" > mydata.data
4 touch mydata.data
5 cat mydata.data
6 ls -l mydata.data
7 rm mydata.data
8 |
```

(2) Profile Creation

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

```

root@VM: /etc/apparmor.d
```

<https://gitlab.com/apparmor/apparmor/wikis/Profiles>

Profiling: /home/seed/lab7/myscript.sh

Please start the application to be profiled in another window and exercise its functionality now.

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.

Profile: /home/seed/lab7/myscript.sh
Execute: /usr/bin/cat
Severity: unknown

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish

```

root@VM: /home/seed/lab7# ./myscript.sh
good thing
-rw-r--r-- 1 root root 11 Oct 23 14:26 mydata.data
root@VM: /home/seed/lab7#
```

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

```

root@VM: /etc/apparmor.d
```

allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.

Profile: /home/seed/lab7/myscript.sh
Execute: /usr/bin/touch
Severity: 3

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish

Profile: /home/seed/lab7/myscript.sh
Execute: /usr/bin/cat
Severity: unknown

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish

Profile: /home/seed/lab7/myscript.sh
Execute: /usr/bin/rm
Severity: unknown

(I)nherit / (C)hild / (N)amed / (X) ix On / (D)eny / Abo(r)t / (F)inish

COMPSCI 750 – Lab 7 AppArmor

Gunnar Yonker

```
root@VM: /etc/apparmor.d

Profiling: /home/seed/lab7/myscript.sh

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
Setting /home/seed/lab7/myscript.sh to enforce mode.

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Finished generating profile for /home/seed/lab7/myscript.sh.
root@VM:/etc/apparmor.d#
```

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.

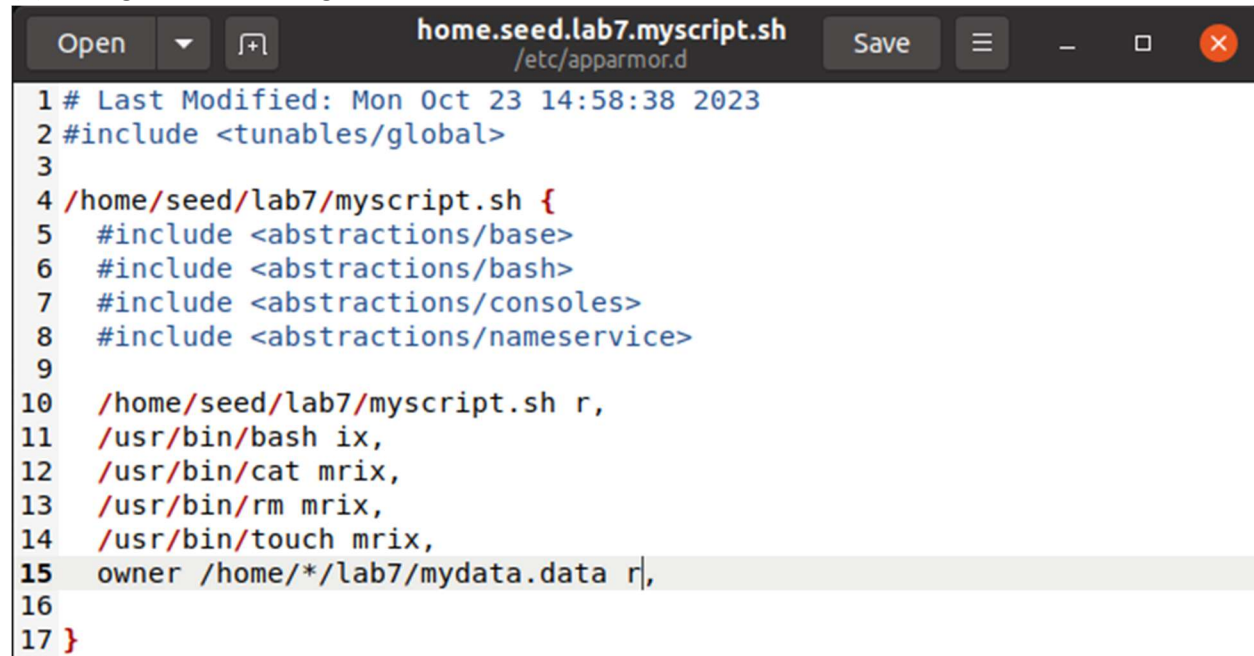
```
root@VM:/etc/apparmor.d# ls
abstractions      usr.bin.man
disable           usr.lib.libreoffice.program.oosplash
force-complain    usr.lib.libreoffice.program.senddoc
home.seed.lab7.myscript.sh  usr.lib.libreoffice.program soffice.bin
local             usr.lib.libreoffice.program.xpdfimport
lsb_release       usr.lib.snapd.snap-confine.real
nvidia_modprobe   usr.sbin.cups-browsed
sbin.dhclient      usr.sbin.cupsd
tunables           usr.sbin.ippusbxd
usr.bin.evince     usr.sbin.rsyslogd
usr.bin.firefox    usr.sbin.tcpdump
root@VM:/etc/apparmor.d#
```

```
root@VM:/home/seed/lab7# ./myscript.sh
good thing
-rw-r--r-- 1 root root 11 Oct 23 14:59 mydata.data
root@VM:/home/seed/lab7#
```

```
Open  home.seed.lab7.myscript.sh  Save
      /etc/apparmor.d

1 # Last Modified: Mon Oct 23 14:58:38 2023
2 #include <tunables/global>
3
4 /home/seed/lab7/myscript.sh {
5   #include <abstractions/base>
6   #include <abstractions/bash>
7   #include <abstractions/consoles>
8   #include <abstractions/nameservice>
9
10  /home/seed/lab7/myscript.sh r,
11  /usr/bin/bash ix,
12  /usr/bin/cat mrix,
13  /usr/bin/rm mrix,
14  /usr/bin/touch mrix,
15  owner /home/*/lab7/mydata.data rw,
16
17 }
```

(3) Change the owner's right from w to r.



```

1 # Last Modified: Mon Oct 23 14:58:38 2023
2 #include <tunables/global>
3
4 /home/seed/lab7/myscript.sh {
5   #include <abstractions/base>
6   #include <abstractions/bash>
7   #include <abstractions/consoles>
8   #include <abstractions/namespace>
9
10  /home/seed/lab7/myscript.sh r,
11  /usr/bin/bash ix,
12  /usr/bin/cat mrx,
13  /usr/bin/rm mrx,
14  /usr/bin/touch mrx,
15  owner /home/*/lab7/mydata.data r,
16
17 }

```

apparmor-parser:

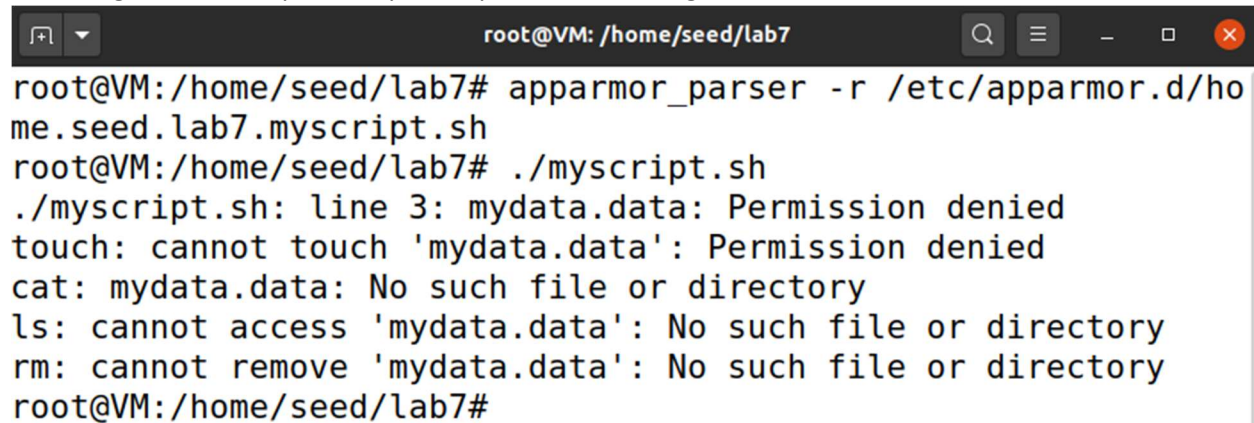


```

root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# apparmor_parser -r /etc/apparmor.d/home.seed.lab7.myscript.sh
root@VM:/home/seed/lab7#

```

Executing the shell script with updated permissions change:



```

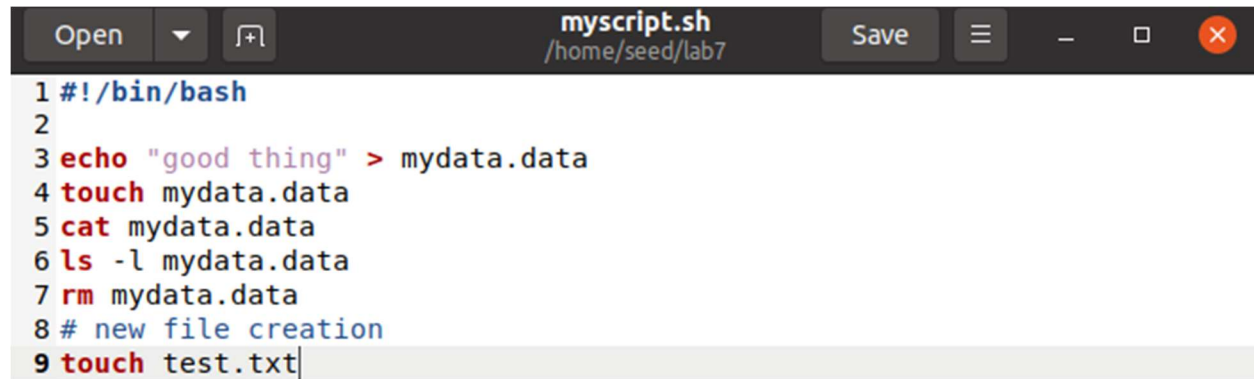
root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# apparmor_parser -r /etc/apparmor.d/home.seed.lab7.myscript.sh
root@VM:/home/seed/lab7# ./myscript.sh
./myscript.sh: line 3: mydata.data: Permission denied
touch: cannot touch 'mydata.data': Permission denied
cat: mydata.data: No such file or directory
ls: cannot access 'mydata.data': No such file or directory
rm: cannot remove 'mydata.data': No such file or directory
root@VM:/home/seed/lab7#

```

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 myscrip.sh shell code.

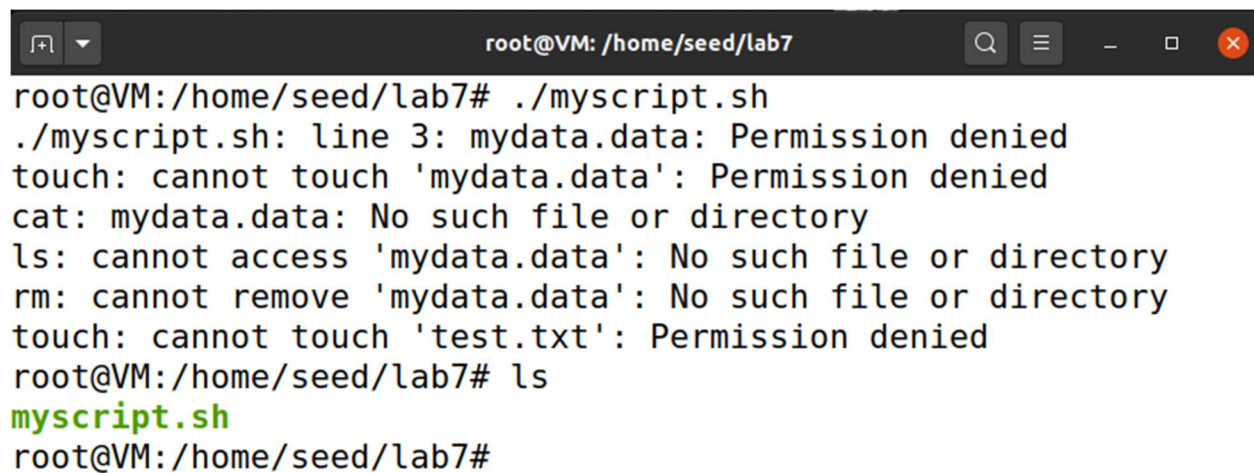
(4) Creating a new file using touch

myscript.sh updated version:

A screenshot of a text editor window titled 'myscript.sh' with the path '/home/seed/lab7'. The editor contains a shell script with the following lines:

```
1 #!/bin/bash
2
3 echo "good thing" > mydata.data
4 touch mydata.data
5 cat mydata.data
6 ls -l mydata.data
7 rm mydata.data
8 # new file creation
9 touch test.txt
```

Execution of myscript.sh:

A screenshot of a terminal window with the prompt 'root@VM: /home/seed/lab7'. The user runs './myscript.sh', which results in several 'Permission denied' errors for the touch, cat, ls, and rm commands. The user then runs 'ls', which shows 'myscript.sh' as a file. The terminal output is as follows:

```
root@VM:/home/seed/lab7# ./myscript.sh
./myscript.sh: line 3: mydata.data: Permission denied
touch: cannot touch 'mydata.data': Permission denied
cat: mydata.data: No such file or directory
ls: cannot access 'mydata.data': No such file or directory
rm: cannot remove 'mydata.data': No such file or directory
touch: cannot touch 'test.txt': Permission denied
root@VM:/home/seed/lab7# ls
myscript.sh
root@VM:/home/seed/lab7#
```

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,
```

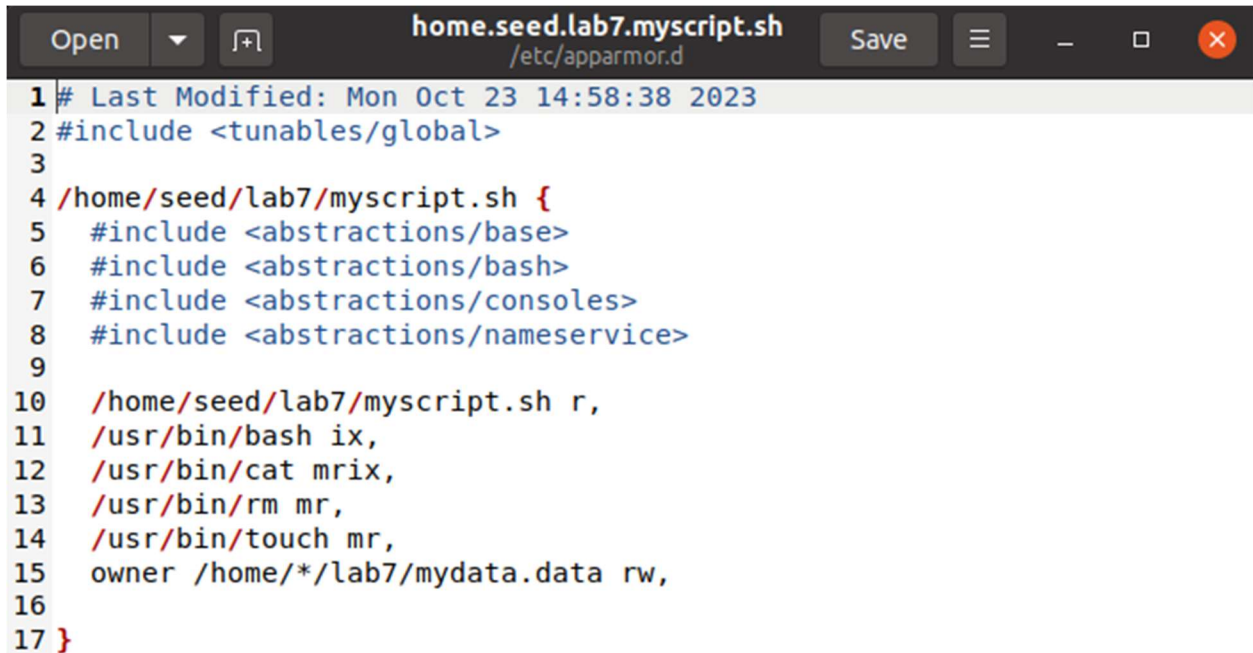
I am modifying the ability of those commands to execute by removing the x and l permission. At this point I will also have to add back in the w permission so that the script can manage these files. This will allow me to observe the outcome of the above permission changes.

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

To:

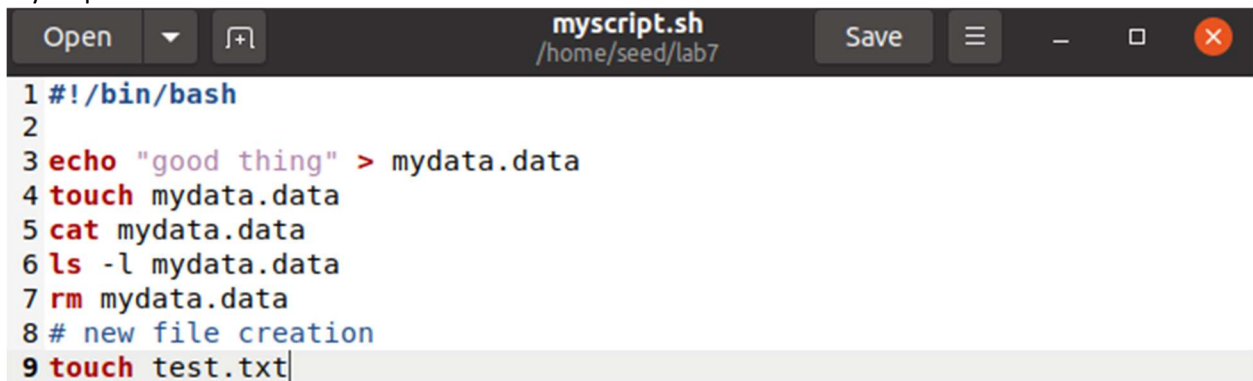
owner /home/*/lab7/test.txt rw,

Profile:



```
1 # Last Modified: Mon Oct 23 14:58:38 2023
2 #include <tunables/global>
3
4 /home/seed/lab7/myscript.sh {
5   #include <abstractions/base>
6   #include <abstractions/bash>
7   #include <abstractions/containers>
8   #include <abstractions/nameservice>
9
10  /home/seed/lab7/myscript.sh r,
11  /usr/bin/bash ix,
12  /usr/bin/cat mrx,
13  /usr/bin/rm mr,
14  /usr/bin/touch mr,
15  owner /home/*/lab7/mydata.data rw,
16
17 }
```

myscript.sh:



```
1 #!/bin/bash
2
3 echo "good thing" > mydata.data
4 touch mydata.data
5 cat mydata.data
6 ls -l mydata.data
7 rm mydata.data
8 # new file creation
9 touch test.txt
```

Reloading AppArmor profile:

```
root@VM: /home/seed/lab7# apparmor_parser -r /etc/apparmor.d/home.seed.lab7.myscript.sh
root@VM: /home/seed/lab7#
```

Executing myscript.sh:

```
root@VM: /home/seed/lab7# apparmor_parser -r /etc/apparmor.d/home.seed.lab7.myscript.sh
root@VM: /home/seed/lab7# ./myscript.sh
./myscript.sh: line 4: /usr/bin/touch: Permission denied
good thing
-rw-r--r-- 1 root root 11 Oct 23 15:30 mydata.data
./myscript.sh: line 7: /usr/bin/rm: Permission denied
./myscript.sh: line 9: /usr/bin/touch: Permission denied
root@VM: /home/seed/lab7# ls
mydata.data  myscript.sh
root@VM: /home/seed/lab7#
```

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:

```
root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.033 ms
^C
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2044ms
rtt min/avg/max/mdev = 0.033/0.035/0.039/0.003 ms
root@VM:/home/seed/lab7#
```

Profile Creation:

aa-autodep ping

```
root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# aa-autodep ping
Writing updated profile for /usr/bin/ping.
root@VM:/home/seed/lab7#
```

ping command is in complain mode:

```
6 profiles are in complain mode.
/home/seed/lab7/myscript.sh//null-/usr/bin/cat
/home/seed/lab7/myscript.sh//null-/usr/bin/rm
/home/seed/lab7/myscript.sh//null-/usr/bin/touch
/usr/bin/ping
libreoffice-oopslash
libreoffice-soffice
```

Setting ping to enforce mode:

```
root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# aa-enforce /bin/ping
Setting /usr/bin/ping to enforce mode.
root@VM:/home/seed/lab7#
```



```

36 profiles are in enforce mode.
  /home/seed/lab7/myscript.sh
  /snap/snapd/20290/usr/lib/snapd/snap-confine
  /snap/snapd/20290/usr/lib/snapd/snap-confine//mount-namespace-ca
pture-helper
  /usr/bin/evince
  /usr/bin/evince-previewer
  /usr/bin/evince-previewer//sanitized_helper
  /usr/bin/evince-thumbnailer
  /usr/bin/evince//sanitized_helper
  /usr/bin/man
  /usr/bin/ping

```

Testing ping again:



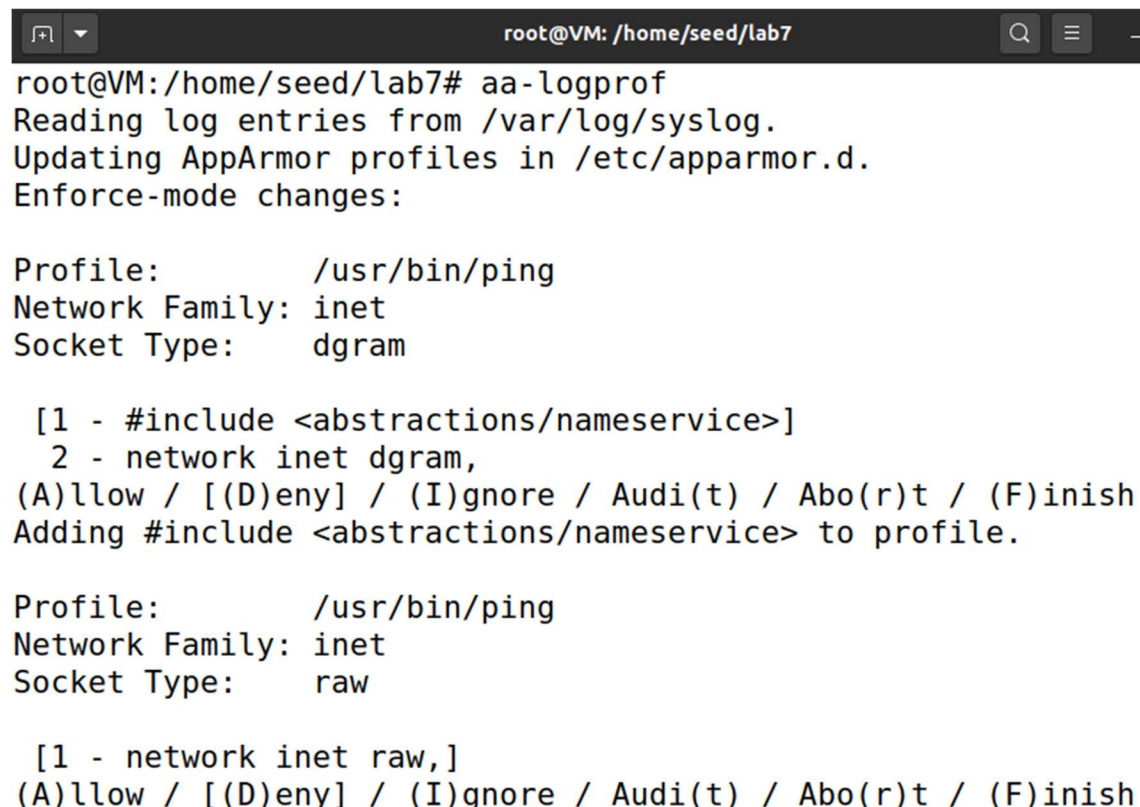
```

root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# ping 127.0.0.1
ping: socket: Permission denied
root@VM:/home/seed/lab7#

```

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:



```

root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# aa-logprof
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.
Enforce-mode changes:

Profile:      /usr/bin/ping
Network Family: inet
Socket Type:  dgram

[1 - #include <abstractions/nameservice>]
2 - network inet dgram,
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
Adding #include <abstractions/nameservice> to profile.

Profile:      /usr/bin/ping
Network Family: inet
Socket Type:  raw

[1 - network inet raw,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish

```

Testing ping with updated profile:

```
root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.033 ms
^C
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.027/0.031/0.033/0.003 ms
root@VM:/home/seed/lab7#
```

The ping command can now successfully be executed.

AppArmor profile for ping:

```
root@VM: /home/seed/lab7
root@VM:/home/seed/lab7# cat /etc/apparmor.d/usr.bin.ping
# Last Modified: Mon Oct 23 15:52:06 2023
#include <tunables/global>

/usr/bin/ping {
    #include <abstractions/base>
    #include <abstractions/nameservice>

    /usr/bin/ping mr,
}
root@VM:/home/seed/lab7#
```

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

COMPSCI 750 – Lab 7 AppArmor

Gunnar Yonker

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.