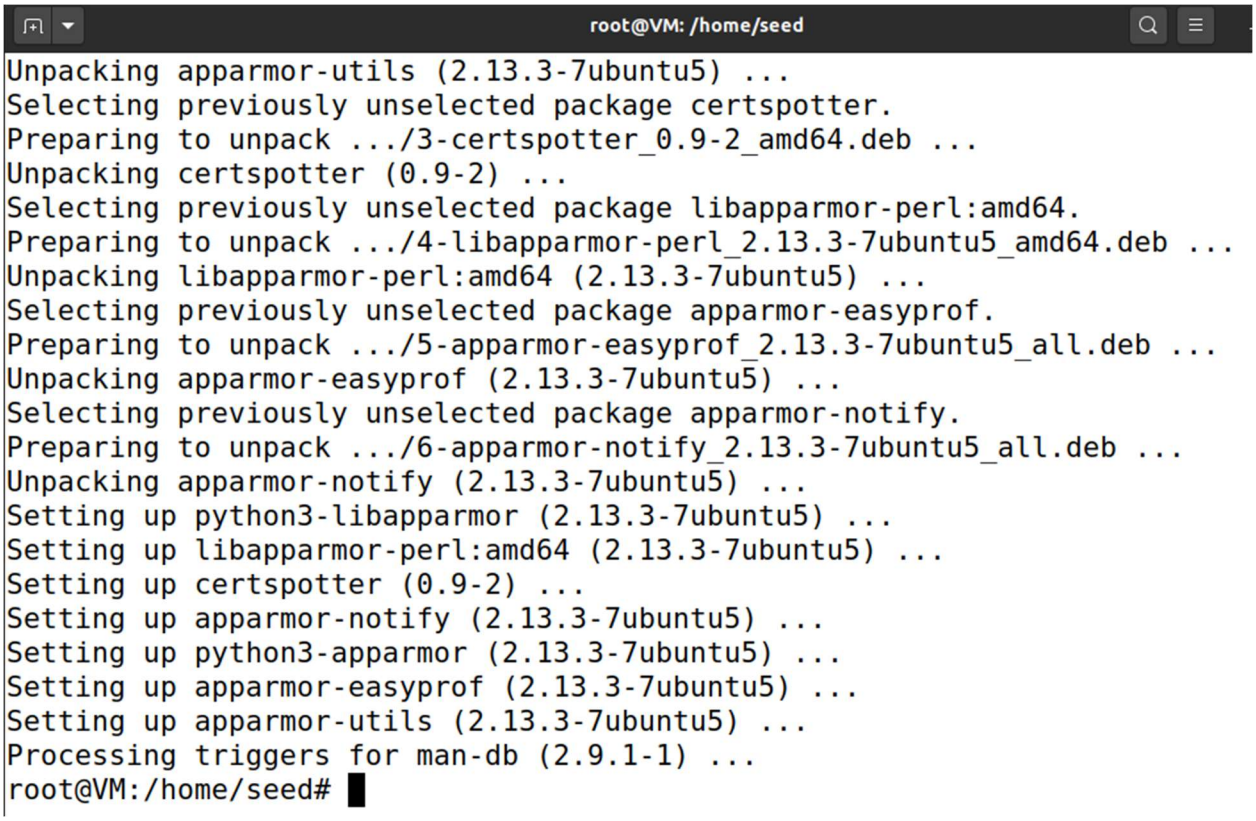
COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker

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.

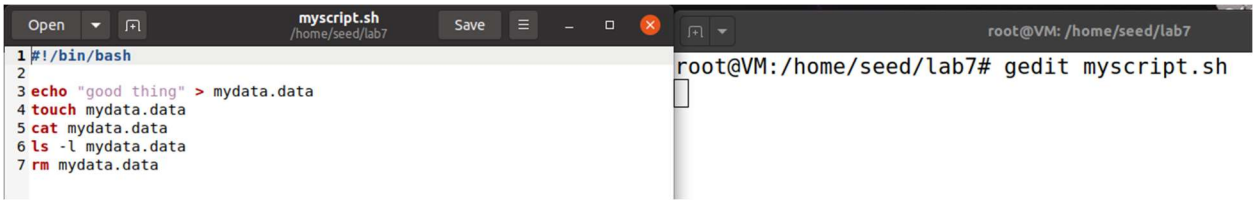


1. AppArmor with myscript.sh

(1) myscript.sh

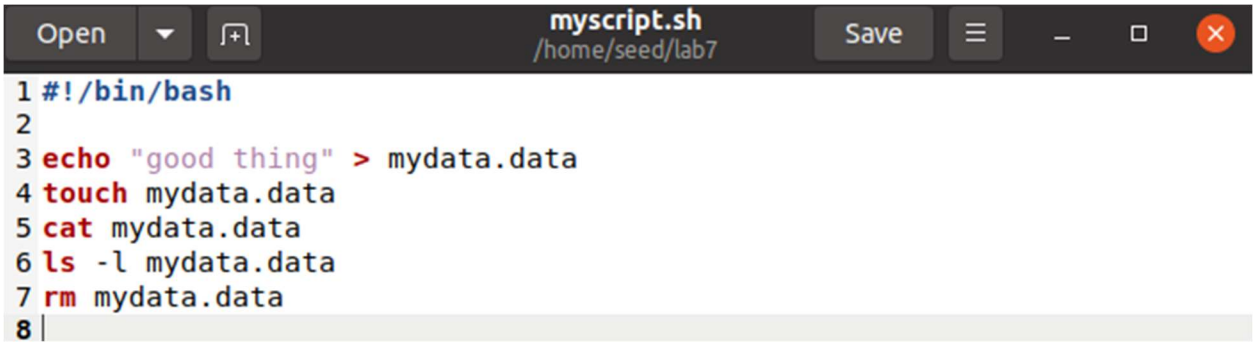
gedit myscript.sh

chmod 777 myscript.sh



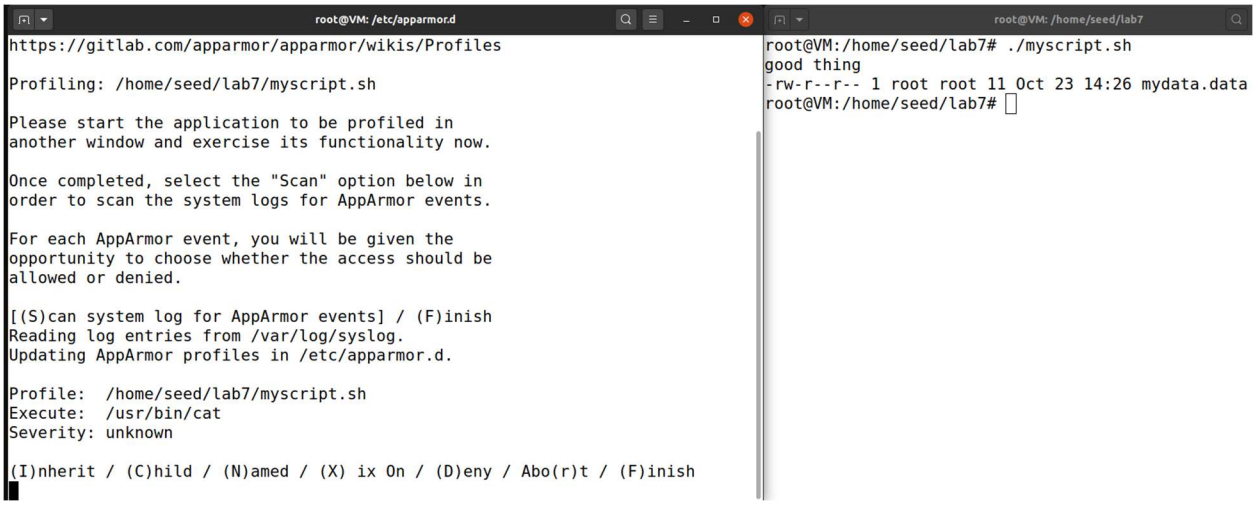
COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker

Here is the current shell code for this step:



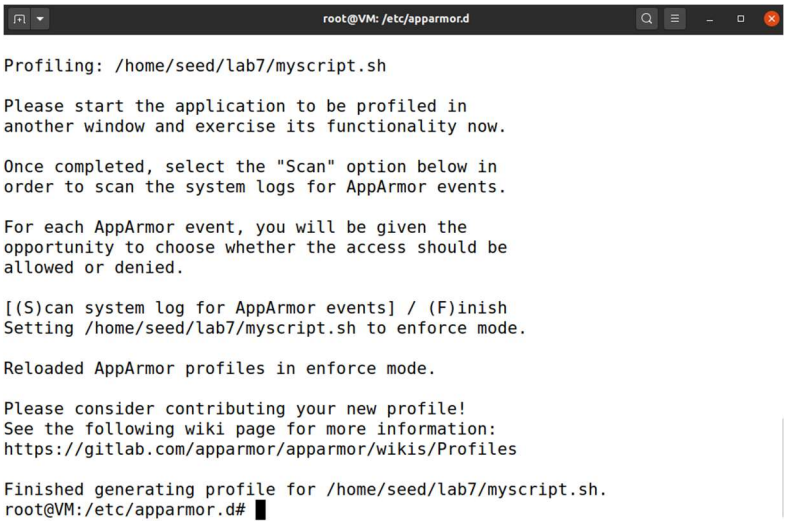
(2) Profile Creation

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



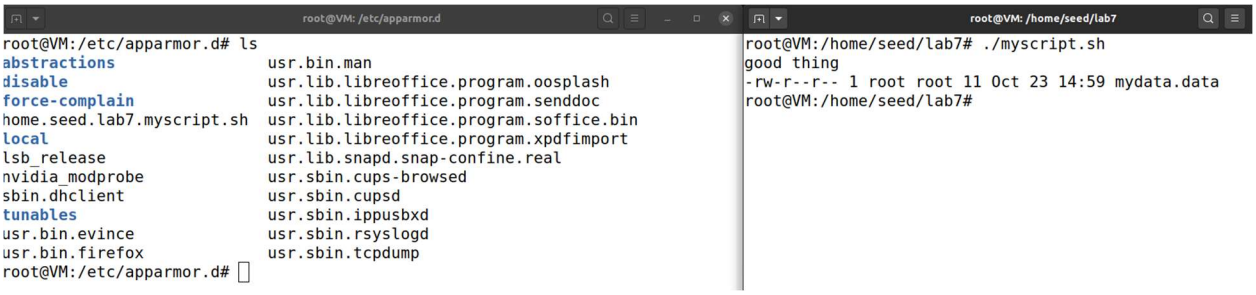
During profile creation, touch, cat and rm were given inherited permissions and /bin/bash was allowed to run.

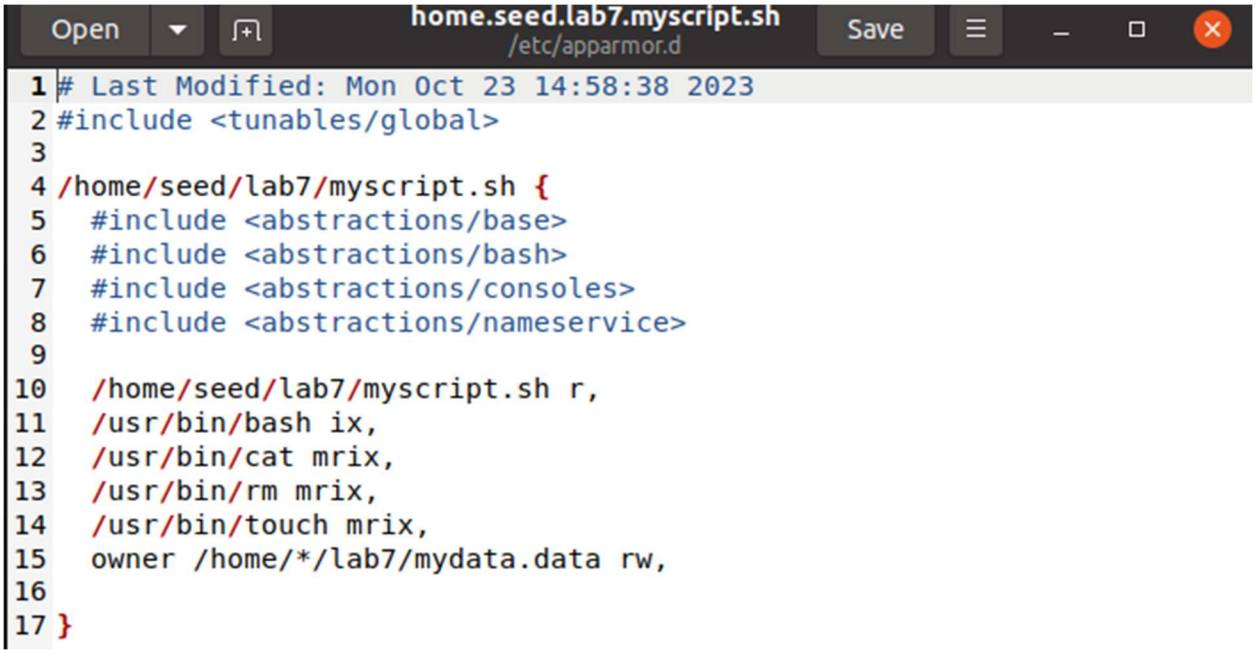
COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker



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.





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

home.seed.lab7.myscript.sh
/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 r,
16
17 }
```

apparmor-parser:

root@VM: /home/seed/lab7

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

Executing the shell script with updated permissions change:

root@VM: /home/seed/lab7

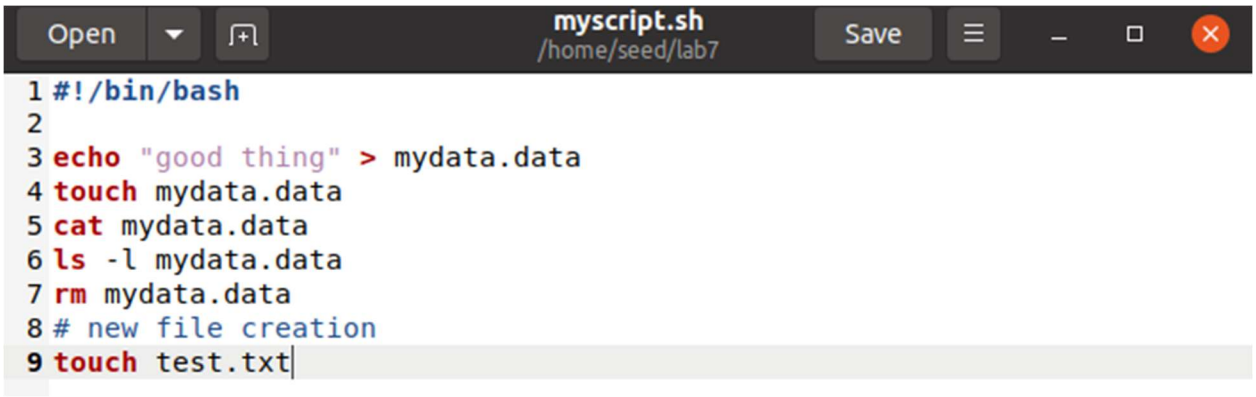
```
root@VM:/home/seed/lab7# apparmor_parser -r /etc/apparmor.d/ho
me.seed.lab7.myscript.sh
root@VM:/home/seed/lab7# ./myscript.sh
./myscript.sh: line 3: mydata.data: Permission denied
touch: cannot touch 'mydata.data': Permission denied
cat: mydata.data: No such file or directory
ls: cannot access 'mydata.data':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

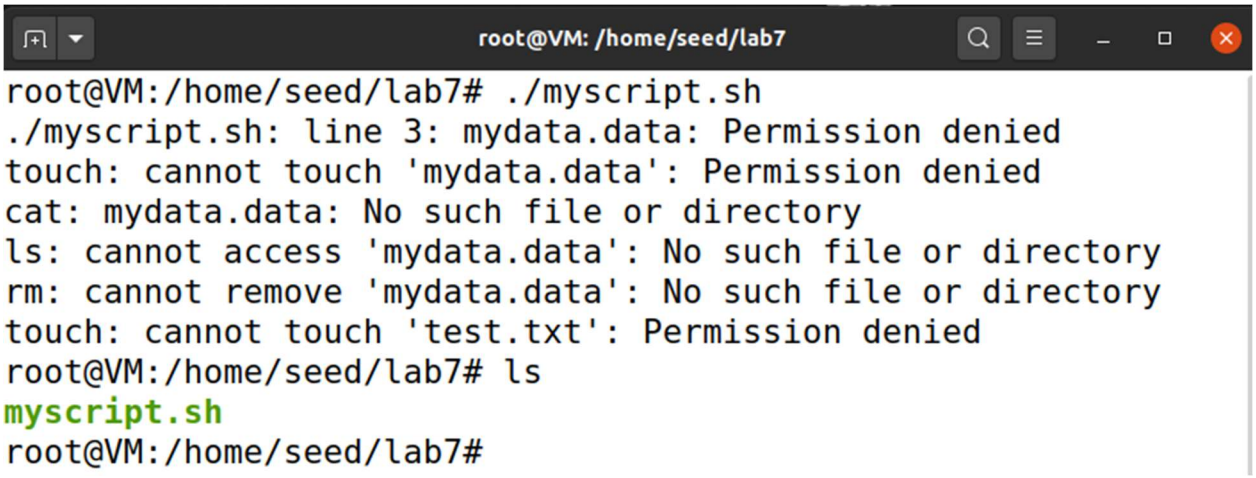
COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker

(4) Creating a new file using touch

myscript.sh updated version:



Execution of myscript.sh:



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,

COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker

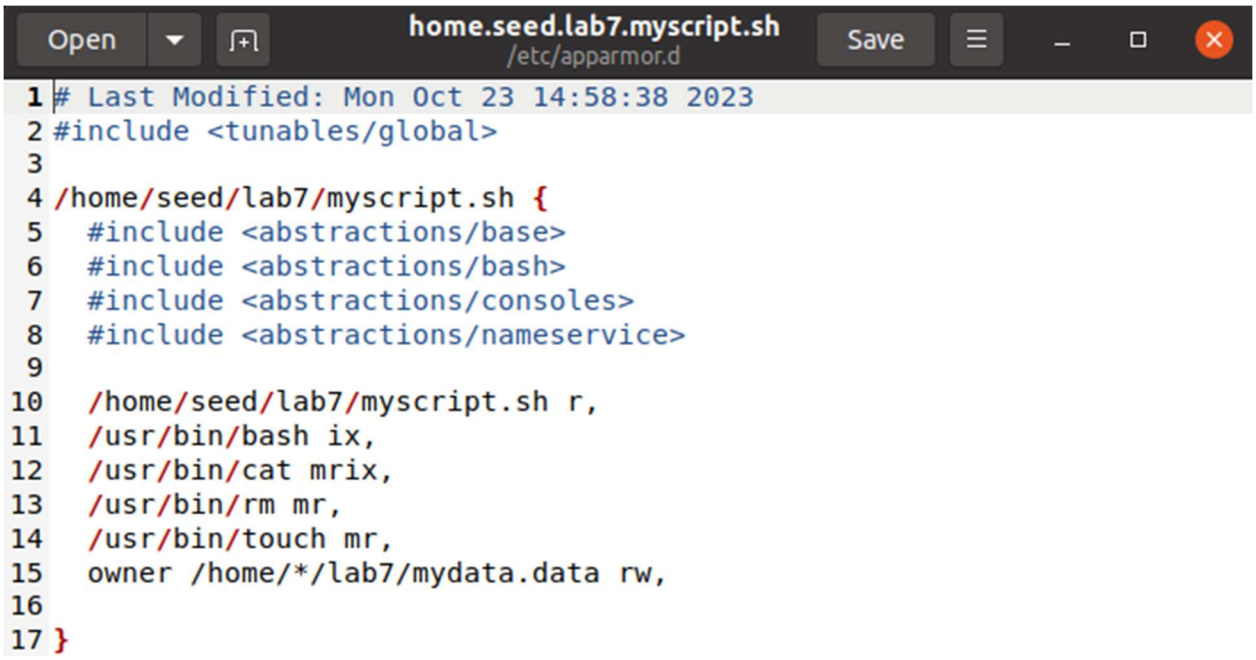
I am modifying the ability of those commands to execute by removing the x and I permission. At this point I will also have to add back in the w permission so that the script can manage these files. This will allow me to observe the outcome of the above permission changes.

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

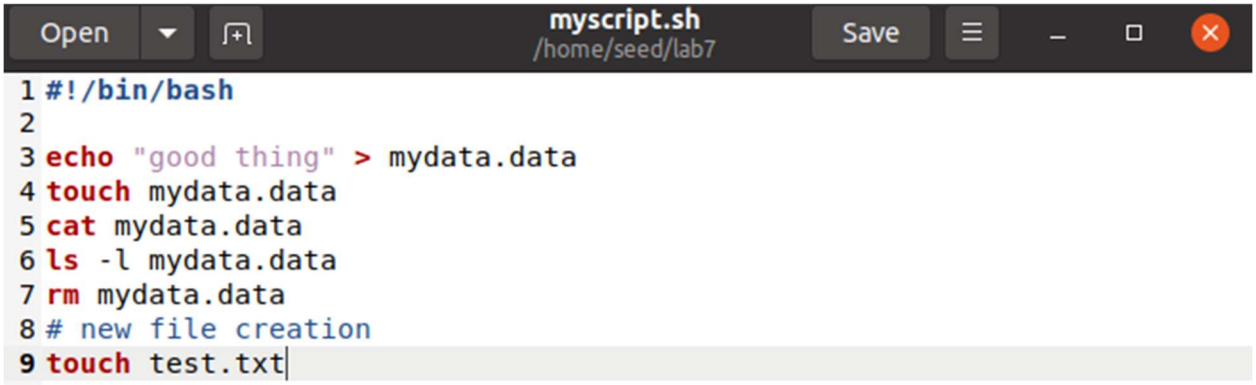
To:

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

Profile:

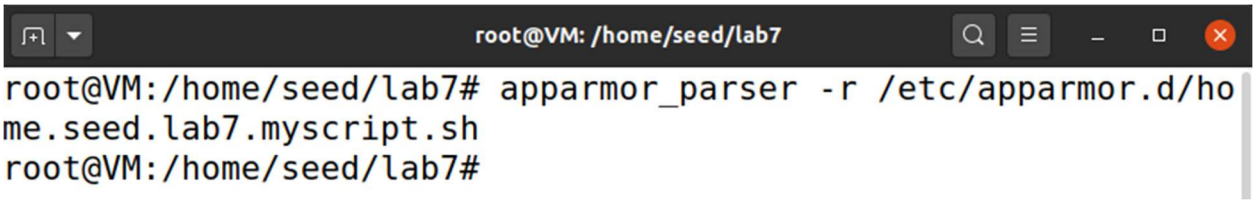


myscript.sh:

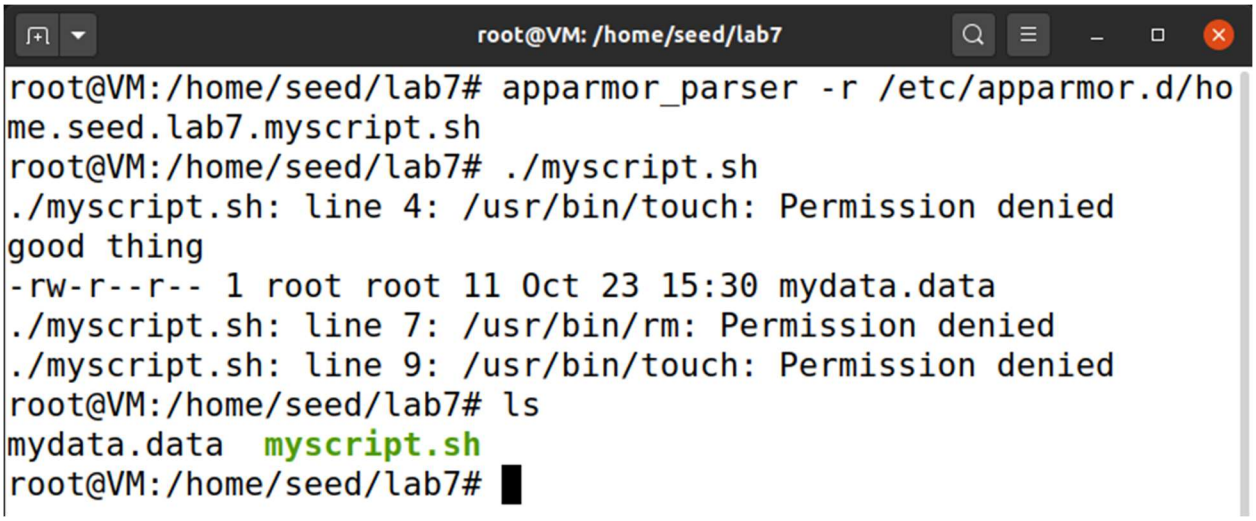


COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker

Reloading AppArmor profile:



Executing myscript.sh:

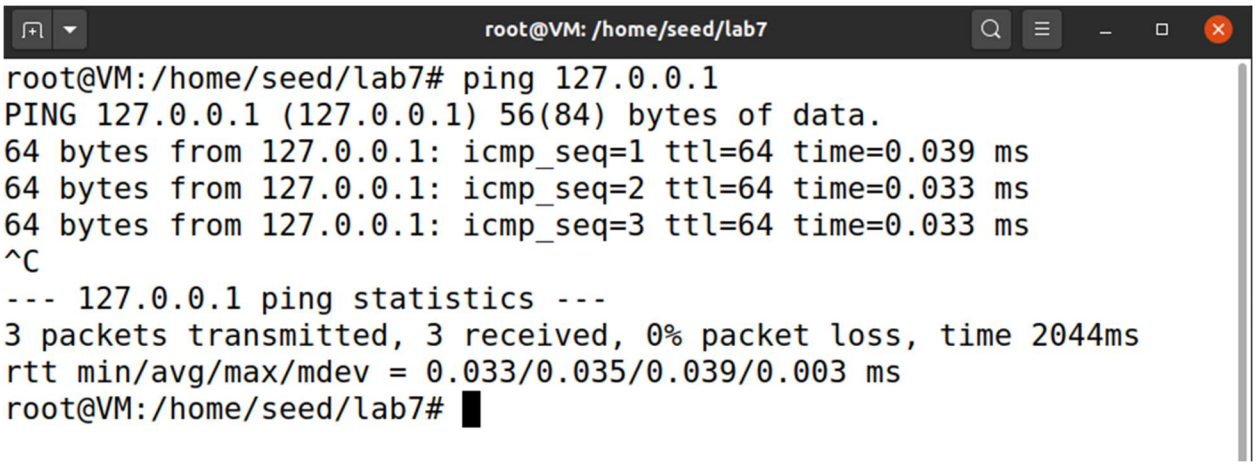


Looking at the outcome of the script when executed, I can see that the touch command has been denied to execute in lines 4 and 9 which means that the permission change from mrix to mr has successfully changed the ability of touch to execute. When cat is run it is still able to read from the mydata.data 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.

COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker

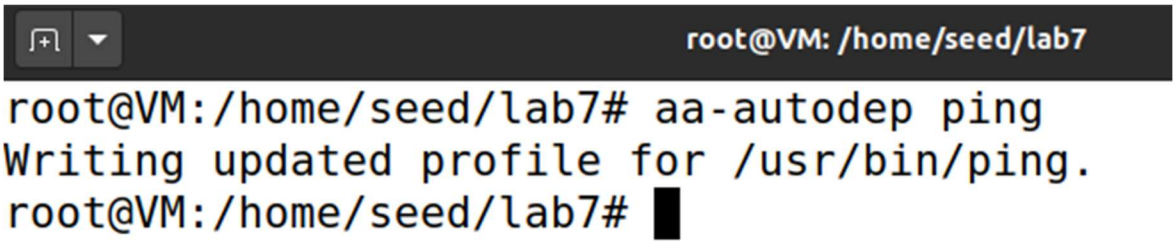
2. AppArmor and Ping Command

Ping Command w/o AppArmor Profile:

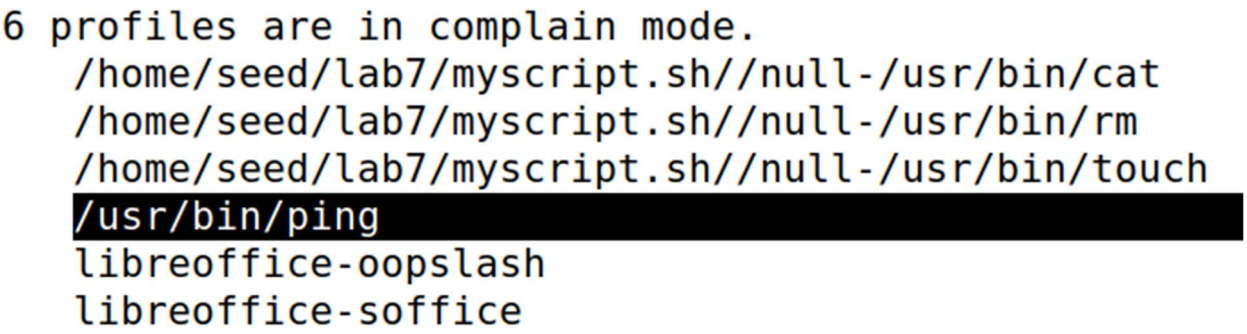


Profile Creation:

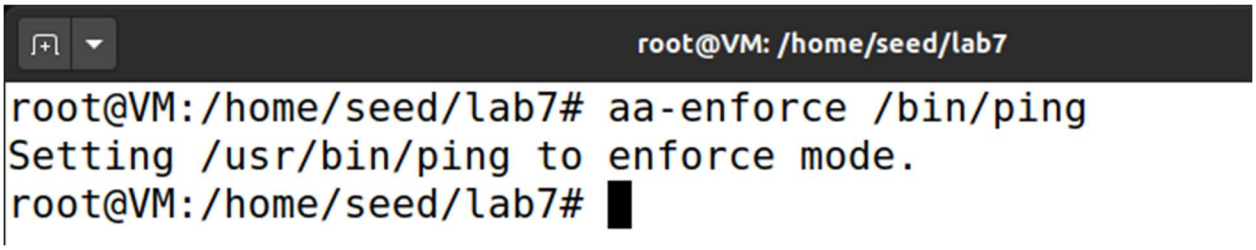
aa-autodep ping



ping command is in complain mode:



Setting ping to enforce mode:



```
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#
```

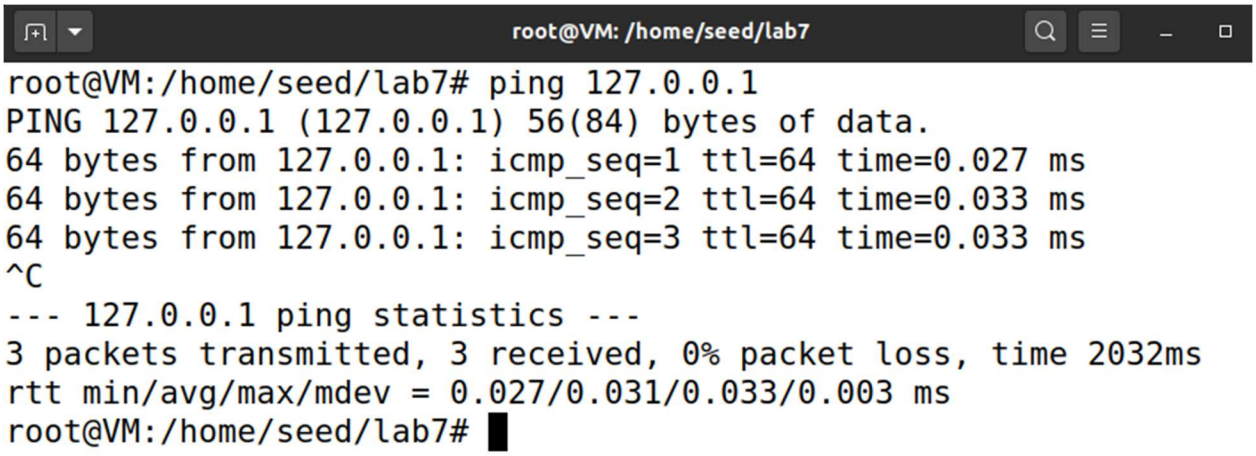
In the screenshot it can be observed that even as the root user, I am unable to use the ping command as permission is being denied with the enforced AppArmor profile.

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
```

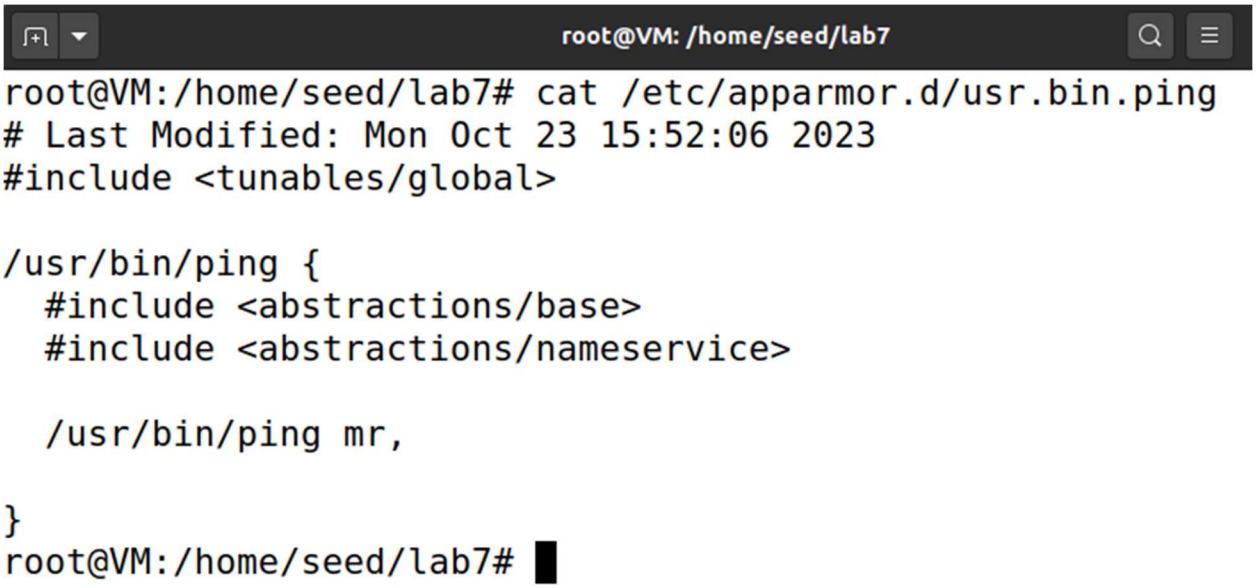
COMPSCI 750 – Lab 7 AppArmor
Gunnar Yonker

Testing ping with updated profile:



The ping command can now successfully be executed.

AppArmor profile for ping:



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.