

Phase 0 – Password management

Define/describe PAM:

It is the purpose of the Linux-PAM project to separate the development of privilege granting software from the development of secure and appropriate authentication schemes. This is accomplished by providing a library of functions that an application may use to request that a user be authenticated.

On **myServer** find 3 daemons (from /usr/sbin) that are PAM aware and 3 that are not:

PAM Aware	Unaware
sshd	student@myserver:/usr/sbin\$ sudo ldd fdisk grep libpam.so
atd	student@myserver:/usr/sbin\$ sudo ldd ldconfig grep libpam.so
cron	student@myserver:/usr/sbin\$ sudo ldd rtmon grep libpam.so
	student@myserver:/usr/sbin\$ sudo ldd sshd grep libpam.so libpam.so.0 => /lib/x86_64-linux-gnu/libpam.so.0 (0x00007f900ed9c000)
	student@myserver:/usr/sbin\$ sudo ldd atd grep libpam.so libpam.so.0 => /lib/x86_64-linux-gnu/libpam.so.0 (0x00007f2fbbf11000)
	student@myserver:/usr/sbin\$ sudo ldd cron grep libpam.so libpam.so.0 => /lib/x86_64-linux-gnu/libpam.so.0 (0x00007fdd32d26000)
	student@myserver:/usr/sbin\$ _

On **myServer** what is currently configured in *pam.conf*? Why?

student@myserver:~\$ cat /etc/pam.conf
-----#
/etc/pam.conf
-----#
#
NOTE

#
NOTE: Most program use a file under the /etc/pam.d/ directory to setup their
PAM service modules. This file is used only if that directory does not exist.
-----#
#
Format:
serv. module ctrl module [path] ...[args..]
name type flag

Pam.d is the used folder for pam modules

Carefully read the *pam* configuration file for *login*. What does it include and what are its session parameters:?

included
Common-auth
Common-account
Common-session
Common-password

Session Variable	Required?
Pam_selinux.so	no
Pam_loginuid.so	Yes
Pam_env.so	yes
Pam_keyinit.so	no
Pan_motd.so	No
Pam_lastlog	no

Still on **myServer**, try to log in as root. Does it work? Why or why not? (hint: Only the shadow knows)

```
student@myserver:~$ su root
Password:
su: Authentication failure
student@myserver:~$ _
```

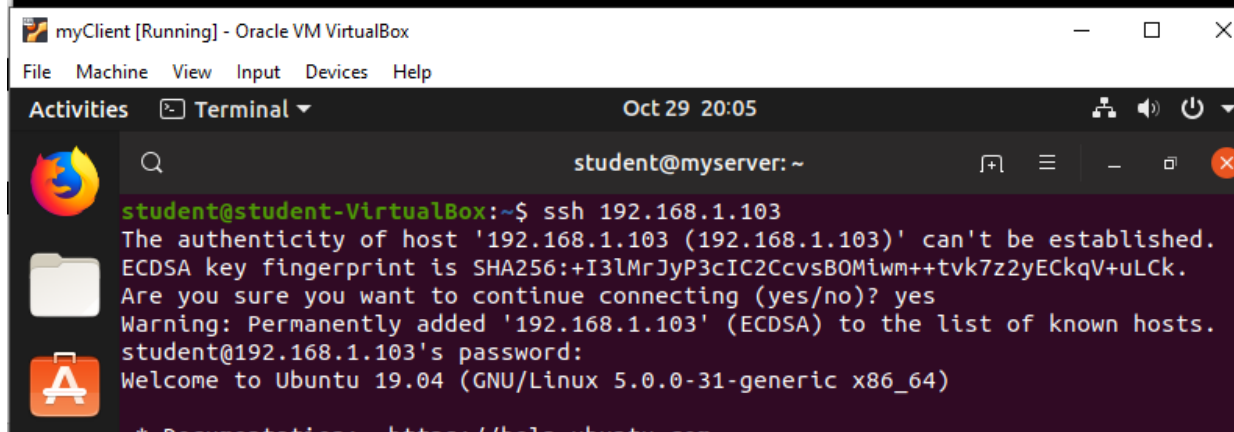
Root account is not active by default

Do what you need to do to log in as root. As always **P@ssw0rd** Show your successful login here:

```
student@myserver:~$ sudo passwd root
New password:
Retype new password:
passwd: password updated successfully
student@myserver:~$ sudo -i
root@myserver:~# _
```

Now go to myClient. ssh into **myServer** as student and show that it works here:

```
student@myserver:~$ sudo ufw disable
Firewall stopped and disabled on system startup
student@myserver:~$
```



```
student@student-VirtualBox:~$ ssh 192.168.1.103
The authenticity of host '192.168.1.103 (192.168.1.103)' can't be established.
ECDSA key fingerprint is SHA256:+I3lMrJyP3cIC2CcvsBOMiwm++tvk7z2yEckqV+uLck.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.103' (ECDSA) to the list of known hosts.
student@192.168.1.103's password:
Welcome to Ubuntu 19.04 (GNU/Linux 5.0.0-31-generic x86_64)

* Documentation:  https://help.ubuntu.com
```

Now try to ssh as **root** Does it work:

```
student@student-VirtualBox:~$ ssh root@192.168.1.103
root@192.168.1.103's password:
Permission denied, please try again.
root@192.168.1.103's password: █
```

Now return to **myServer**. Find where in the sshd configuration it uses PAM to authenticate remote users:

```
# PAM configuration for the Secure Shell service

# Standard Unix authentication.
@include common-auth
```

Extra Credit: Allow root access through ssh (show proof and explain how):

Now we will set the password complexity requirements. Please list your complexity requirements and have your instructor verify that they will be adequate:

```
student@myserver:~$ sudo nano /etc/pam.d/common-password
```

```
# here are the per-package modules (the "Primary" block)
password    [success=1 default=ignore]      pam_unix.so obscure sha512 minlen=6
# here's the fallback if no module succeeds
password    requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password    required                       pam_permit.so
# and here are more per-package modules (the "Additional" block)
```

Now look at the pam_unix man page; which options will be important? Are they all there?:

```
obscure
  Enable some extra checks on password strength. These checks are based on the "obscure"
  checks in the original shadow package. The behavior is similar to the pam_cracklib
  module, but for non-dictionary-based checks. The following checks are implemented:

  Palindrome
    Verifies that the new password is not a palindrome of (i.e., the reverse of) the
    previous one.

  Case Change Only
    Verifies that the new password isn't the same as the old one with a change of
    case.

  Similar
    Verifies that the new password isn't too much like the previous one.

  Simple
    Is the new password too simple? This is based on the length of the password and
    the number of different types of characters (alpha, numeric, etc.) used.

  Rotated
    Is the new password a rotated version of the old password? (E.g., "billy" and
    "illyb")
```

Create a user account named **jim** and set his password to **newpass.**:

```
student@myserver:~$ sudo adduser jim
```

Change the password complexity requirements and log in as **jim**. Show you cannot set a weak password:

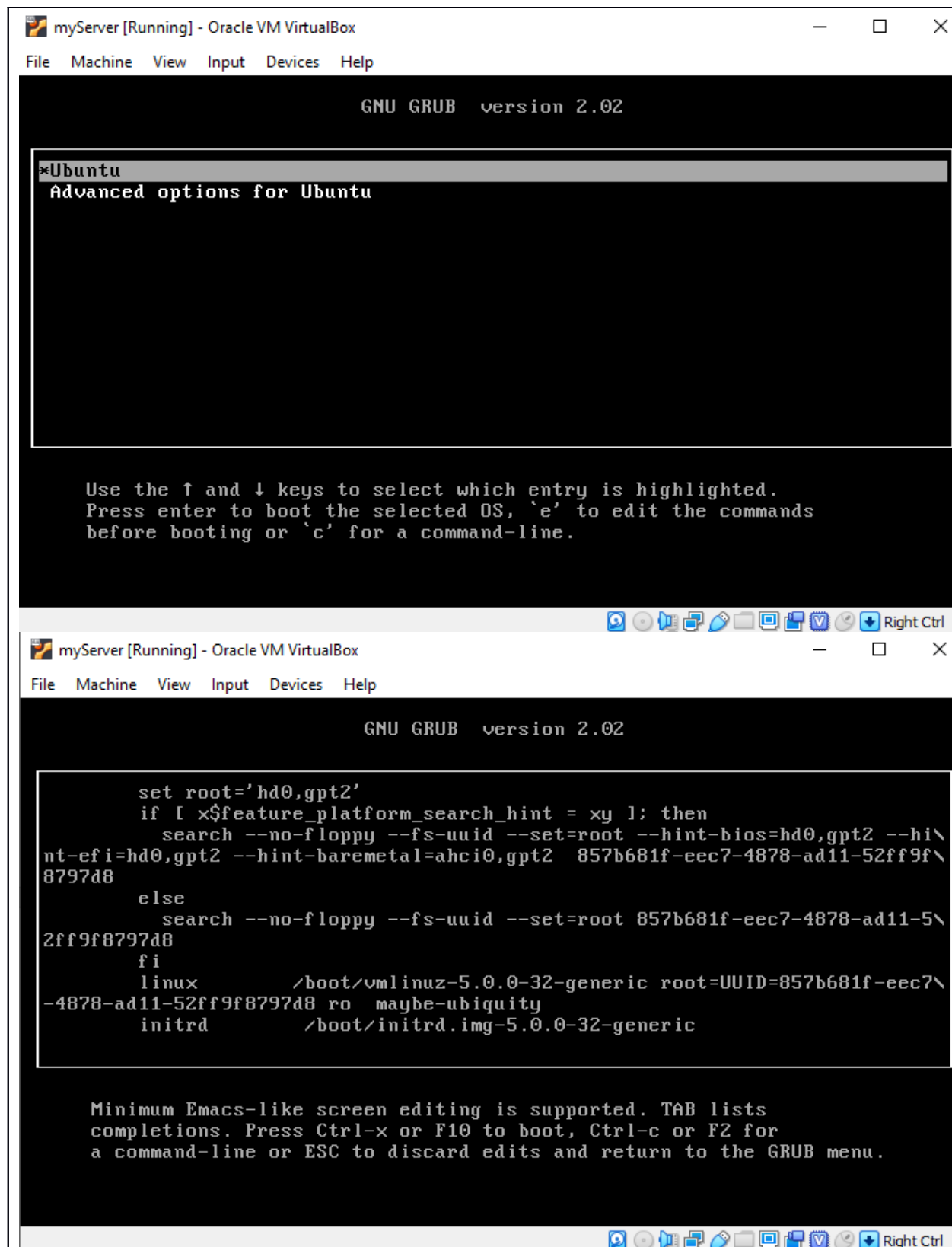
```
jim@myserver:~$ passwd
Changing password for jim.
Current password:
New password:
Retype new password:
You must choose a longer password
New password: _
```

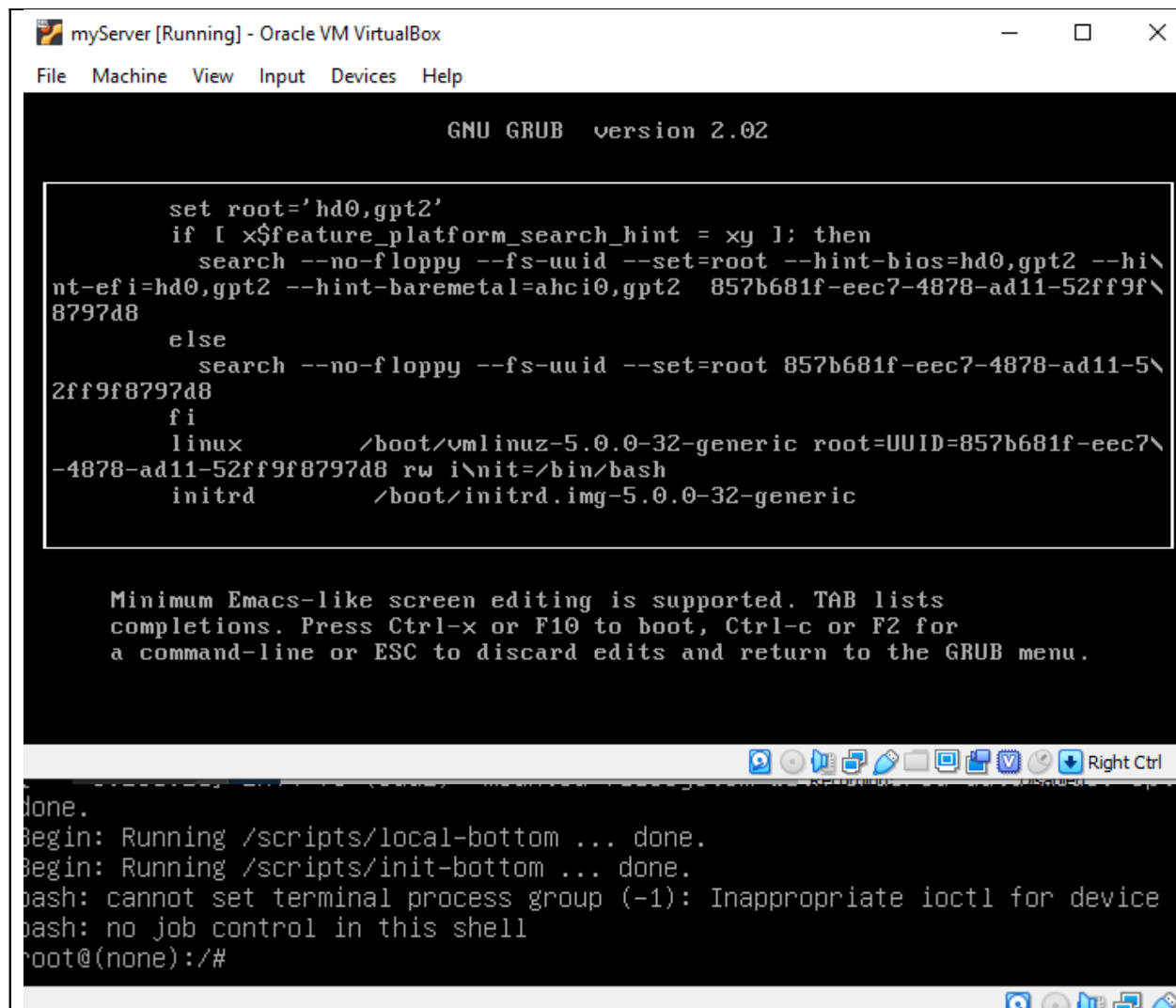
Attempted to type in "password" char=8 min was 10

```
# here are the per-package modules (the "Primary" block)
password      [success=1 default=ignore]      pam_unix.so obscure sha512 minlen=10
```

Phase 1 – Secure the boot

Show the steps and results booting **myServer** to single user mode:





```
GNU GRUB version 2.02

set root='hd0,gpt2'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2 857b681f-eec7-4878-ad11-52ff9f8797d8
else
  search --no-floppy --fs-uuid --set=root 857b681f-eec7-4878-ad11-52ff9f8797d8
fi
linux      /boot/vmlinuz-5.0.0-32-generic root=UUID=857b681f-eec7-4878-ad11-52ff9f8797d8 rw init=/bin/bash
initrd     /boot/initrd.img-5.0.0-32-generic

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for
a command-line or ESC to discard edits and return to the GRUB menu.

done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none):/#
```

Explain why the fact that you can do this is a problem:

Anybody can do this and gain root access bypassing any security

Change the GRUB configuration so it will **countdown 15 seconds** before choosing the default boot option and show the steps here:

```
root@(none):/# nano /etc/default/grub_

GRUB_DEFAULT=0
GRUB_TIMEOUT=15_
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="maybe-ubiquity"
GRUB_CMDLINE_LINUX=""
```

```

root@(none):/# update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/50-curtin-settings.cfg'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.0.0-32-generic
Found initrd image: /boot/initrd.img-5.0.0-32-generic
Found linux image: /boot/vmlinuz-5.0.0-31-generic
Found initrd image: /boot/initrd.img-5.0.0-31-generic
done
root@(none):/# _

```

Change GRUB to have a password to access the boot menu; set the password to **P@ssw0rd**
Show the steps here:

```

root@(none):/# grub-mkpasswd-pbkdf2
Enter password:
Reenter password:
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.20C28A20DE2D7DC1332962AF59A4A32A2EA1F971
3F52C947D466825B110F7C9F9AA40A1B7776ED4B1666D90AC87AF697FE18F83BB0FA0CDD0A33A2B59448.BD182A832E821
6E28260B329FAC060F9B87E7B99D9591FBCCD23DAB5765713E1E5F1DCD1A85E38F8F277AF88FDD984CB80404A69EF8D65B7
834AA91A142F6D
root@(none):/#

root@(none):/# grub-mkpasswd-pbkdf2 >> /etc/grub.d/40_custom

#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.

set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.42079138C1359EBE092054A94E790A237D080C33ECAC25587FF623

student@myserver:~$ sudo grub-mkconfig -o /boot/grub/grub.cfg
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/50-curtin-settings.cfg'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.0.0-32-generic
Found initrd image: /boot/initrd.img-5.0.0-32-generic
Found linux image: /boot/vmlinuz-5.0.0-31-generic
Found initrd image: /boot/initrd.img-5.0.0-31-generic
done
student@myserver:~$ _

```

pressing 'e' on Ubuntu prompts:

```

Enter username:
root
Enter password:
_

```

Phase 2 – AppArmor

On **myServer** how many AppArmor profiles are loaded, being enforced, and/or complaining:

```
student@myserver:~$ sudo apparmor_status_  
49 profiles are loaded.  
30 profiles are in enforce mode.  
/sbin/dhclient  
/snap/core/7917/usr/lib/snapd/snap-confine  
/snap/core/7917/usr/lib/snapd/snap-confine//mount-namespace-capture-helper  
/usr/bin/man  
/usr/lib/NetworkManager/nm-dhcp-client.action  
/usr/lib/NetworkManager/nm-dhcp-helper  
/usr/lib/connman/scripts/dhclient-script  
/usr/lib/snapd/snap-confine  
/usr/lib/snapd/snap-confine//mount-namespace-capture-helper  
/usr/sbin/tcpdump  
chromium_browser//browser_java  
chromium_browser//browser_openjdk  
chromium_browser//sanitized_helper  
man_filter  
man_groff  
nvidia_modprobe  
nvidia_modprobe//kmod  
snap-update-ns.core  
snap-update-ns.lxd  
snap.core.hook.configure  
snap.lxd.activate  
snap.lxd.benchmark  
snap.lxd.buginfo  
snap.lxd.check-kernel  
snap.lxd.daemon  
snap.lxd.hook.configure  
snap.lxd.hook.install  
snap.lxd.lxc  
snap.lxd.lxd  
snap.lxd.migrate  
19 profiles are in complain mode.  
/usr/sbin/dnsmasq
```

How many are not confined? Which ones? What potential security issues might arise?

```
0 processes are unconfined but have a profile defined.
```

Write a bash script named **thumpthump.sh** that will prove connectivity with **myServer**, **myGateway**, and **8.8.8.8** in order waiting 30 seconds between each and save the output to **/tmp/heartbeat.log**


```
#!/bin/bash
while true;
do
ping -c 2 192.168.1.103 >> /tmp/heartbeat.log
sleep 30s
ping -c 2 192.168.1.1 >> /tmp/heartbeat.log
sleep 30s
ping -c 2 8.8.8.8 >> /tmp/heartbeat.log
sleep 30s
done
```

Have your instructor verify your script before moving on.

Using the command (**sleep 15 ; ./thumpthump.sh**) & to run it in the background, generate an AppArmor profile for your script:

```
student@myserver:~$ ./thumpthump.sh &
[1] 1223
student@myserver:~$ ps
  PID TTY          TIME CMD
 1013 tty1        00:00:00 bash
 1223 tty1        00:00:00 thumpthump.sh
 1225 tty1        00:00:00 sleep
 1226 tty1        00:00:00 ps
student@myserver:~$ sudo aa-autodep thumpthump.sh
Writing updated profile for /home/student/thumpthump.sh.
student@myserver:~$ sudo apt-get install apparmor-utils
```

What line in the profile gives the script permission to run ping:

```
but some applications depend on the presence
of LD_PRELOAD or LD_LIBRARY_PATH.

[(Y)es] / (N)o
Writing updated profile for /usr/bin/sleep.
Complain-mode changes:

Profile: /home/student/thumpthump.sh
Path: /usr/bin/ping
New Mode: r
Severity: unknown

[1 - /usr/bin/ping r,]
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / Abo(r)t / (F)ini
h
Adding /usr/bin/ping r, to profile.
Enforce-mode changes:

Profile: /home/student/thumpthump.sh
Path: /usr/bin/sleep
New Mode: r
Severity: unknown

[1 - /usr/bin/sleep r,]
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / Abo(r)t / (F)ini
h
Adding /usr/bin/sleep r, to profile.

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

[1 - /home/student/thumpthump.sh]
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)lean profiles / A
o(r)t
Writing updated profile for /home/student/thumpthump.sh.
student@myserver:~$ _
student@myserver:~$ sudo cat /etc/apparmor.d/home.student.thumpthump.sh
# Last Modified: Wed Nov 6 04:41:09 2019
#include <tunables/global>

/home/student/thumpthump.sh {
    #include <abstractions/base>
    #include <abstractions/bash>
    #include <abstractions/consoles>
    #include <abstractions/user-tmp>

    /home/student/thumpthump.sh r,
    /usr/bin/bash ix,
    /usr/bin/ping Px,
    /usr/bin/ping r,
    /usr/bin/sleep Px,
    /usr/bin/sleep r,
    /usr/lib/x86_64-linux-gnu/ld-*.so mr,
}
student@myserver:~$
```

Force the profile into the **complain** state. What impact does that have?

```
student@myserver:~$ sudo aa-complain thumpthump.sh_
```

Force the profile into the **audit** state. What impact does that have?

```
student@myserver:~$ sudo aa-logprof
```

Force the profile back into **enforce**. Now edit your script to write some message to open a listening socket using **nc**

```
student@myserver:~$ sudo aa-enforce thumpthump.sh
Setting /home/student/thumpthump.sh to enforce mode.
student@myserver:~$ cat thumpthump.sh
#!/bin/bash
while true;
do
ping -c 2 192.168.1.103 >> /tmp/heartbeat.log
sleep 30s
ping -c 2 192.168.1.1 >> /tmp/heartbeat.log
sleep 30s
ping -c 2 8.8.8.8 >> /tmp/heartbeat.log
sleep 30s
netcat -z -n -v 192.168.1.103 1-1000
done
student@myserver:~$ _
```

What happens when you run **thumpthump.sh** now? How can you make this 'legit' again?

```
./thumpthump.sh: line 10: /usr/bin/netcat: Permission denied
student@myserver:~$ sudo aa-complain thumpthump.sh
student@myserver:~$ sudo aa-logprof_
```

Allow netcat

Phase 3.5 - Apache

Earlier you found that **apache2** was not confined. Fix this and demonstrate that here (note, create the profile yourself):

```
student@myserver:~$ sudo aa-genprof /etc/apache2
Writing updated profile for /etc/apache2.
Setting /etc/apache2 to complain mode.

Before you begin, you may wish to check if a
profile already exists for the application you
wish to confine. See the following wiki page for
more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Profiling: /etc/apache2

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 /etc/apache2 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 /etc/apache2.
student@myserver:~$ sudo aa-enforce
5 profiles are in enforce mode.
/etc/apache2
/etc/apache2//DEFAULT_URI
/etc/apache2//HANDLING_UNTRUSTED_INPUT
```

If you have multiple vhosts that required different permissions, how would you modify the profile? What tool?

26 Confining Users with `pam_apparmor`

An AppArmor profile applies to an executable program; if a portion of the program needs different access permissions than other portions need, the program can change hats via `change_hat` to a different role, also known as a subprofile. The `pam_apparmor` PAM module allows applications to confine authenticated users into subprofiles based on group names, user names, or a default profile. To accomplish this, `pam_apparmor` needs to be registered as a PAM session module.

The package `pam_apparmor` is not installed by default, you can install it using YaST or **zypper**. Details about how to set up and configure `pam_apparmor` can be found in `/usr/share/doc/packages/pam_apparmor/README` after the package has been installed. For details on PAM, refer to [Chapter 2, Authentication with PAM](#).