Cybersecurity

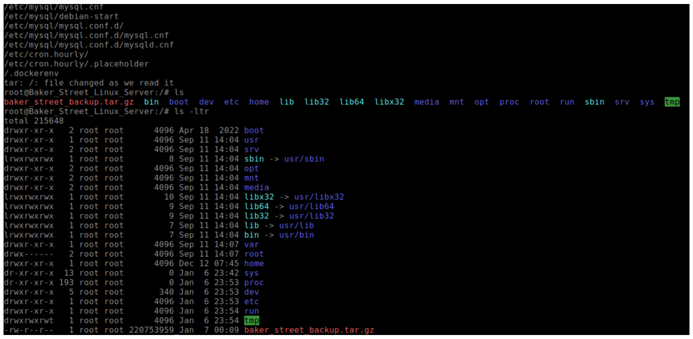
Project 1 Hardening Summary and Checklist

OS Information
Generated by David Contreras

Customer	Baker Street Corporation
Hostname	**Baker_Street_Linux_Server**
OS Version	**Ubuntu 22.04.5 LTS**
Memory information	**total used free shared buff/cache available** **Mem: 3.7Gi 707Mi 602Mi 26Mi 2.5Gi 2.8Gi**
Uptime information	**00:04:00 up 21 min, 0 users, load average: 0.04, 0.05, 0.08**

Checklist

Completed	Activity	Script(s) used / Tasks completed / Screenshots
☑	OS backup	*sudo tar -cvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run /*



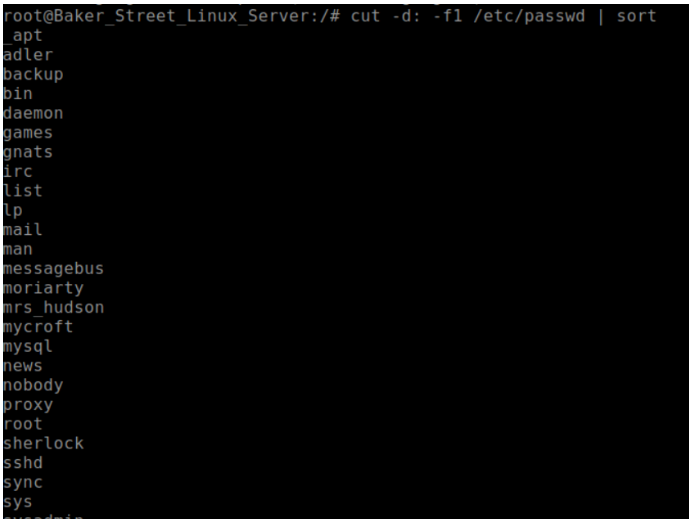
```
/etc/mysql/mysql.cnf
/etc/mysql/debian-start
/etc/mysql/mysql.conf.d/
/etc/mysql/mysql.conf.d/mysql.cnf
/etc/mysql/mysql.conf.d/mysqld.cnf
/etc/cron.hourly/
/etc/cron.hourly/.placeholder
/.dockerenv
tar: /: file changed as we read it
root@Baker_Street_Linux_Server:/# ls
baker_street_backup.tar.gz  bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp
root@Baker_Street_Linux_Server:/# ls -ltr
total 215648
drwxr-xr-x   2 root root      4096 Apr 18  2022 boot
drwxr-xr-x   1 root root      4096 Sep 11 14:04 usr
drwxr-xr-x   2 root root      4096 Sep 11 14:04 srv
lrwxrwxrwx   1 root root         8 Sep 11 14:04 sbin -> usr/sbin
drwxr-xr-x   2 root root      4096 Sep 11 14:04 opt
drwxr-xr-x   2 root root      4096 Sep 11 14:04 mnt
drwxr-xr-x   2 root root      4096 Sep 11 14:04 media
lrwxrwxrwx   1 root root        10 Sep 11 14:04 libx32 -> usr/libx32
lrwxrwxrwx   1 root root         9 Sep 11 14:04 lib64 -> usr/lib64
lrwxrwxrwx   1 root root         9 Sep 11 14:04 lib32 -> usr/lib32
lrwxrwxrwx   1 root root         7 Sep 11 14:04 lib -> usr/lib
lrwxrwxrwx   1 root root         7 Sep 11 14:04 bin -> usr/bin
drwxr-xr-x   1 root root      4096 Sep 11 14:07 var
drwx------   2 root root      4096 Sep 11 14:07 root
drwxr-xr-x   1 root root      4096 Dec 12 07:45 home
dr-xr-xr-x  13 root root         0 Jan  6 23:42 sys
dr-xr-xr-x 193 root root         0 Jan  6 23:53 proc
drwxr-xr-x   5 root root       340 Jan  6 23:53 dev
drwxr-xr-x   1 root root      4096 Jan  6 23:53 etc
drwxr-xr-x   1 root root      4096 Jan  6 23:54 run
drwxrwxrwt   1 root root      4096 Jan  6 23:54 tmp
-rw-r--r--   1 root root 220753959 Jan  7 00:09 baker_street_backup.tar.gz
```

☑

Auditing users and groups

List current users in the system from first column in system file and sorted
cut -d: -f1 /etc/passwd | *sort*

Remove terminated users
Command : *sudo userdel -r <uname>*

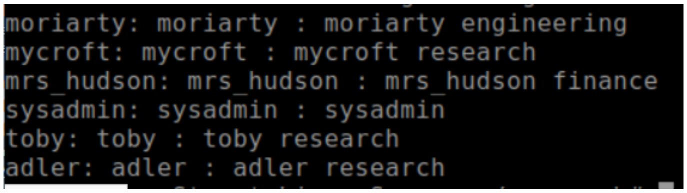


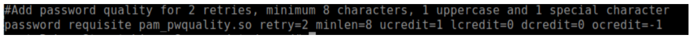
```
root@Baker_Street_Linux_Server:/# cut -d: -f1 /etc/passwd | sort
_apt
adler
backup
bin
daemon
games
gnats
irc
list
lp
mail
man
messagebus
moriarty
mrs_hudson
mycroft
mysql
news
nobody
proxy
root
sherlock
sshd
sync
sys
```

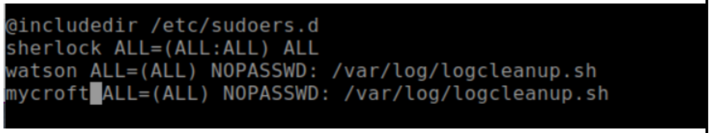
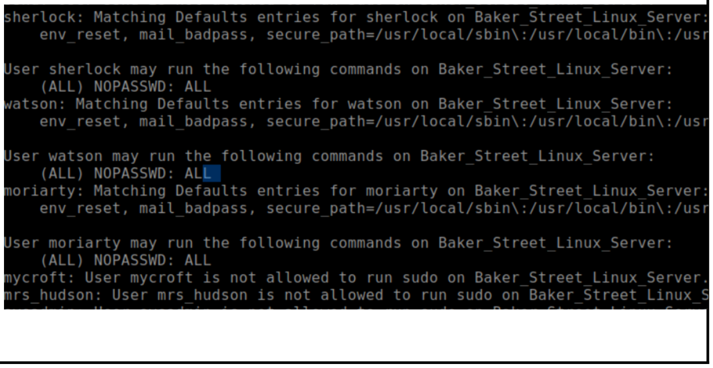
Lock users in temporary leave
Command: *sudo passwd -l <user>* in this case moriarty and mrs_hudson it adds the ! after their name in the /etc/shadow file

moriarty:!$y$j9T$YA/ARBF8VH7Q6k7TnHq0i0$FOY9g4Bgd9hJMBhS3I8D90tJ.8bkhYYiQoRCMe0RSz9:20095:0:99999:7:::
mycroft:$y$j9T$B6No6qOx00QtjZ2IZ9msX.$kfXNioTa1uX5q3CYZI/PN61t5SnBbvUDIlplJcVyyk9:20095:0:99999:7:::

		mrs_hudson:!:20069:0:99999:7::: ● Unlock any users who are employed. Unlock: *sudo passwd -u toby* Force change on first logon:sudo passwd -e toby Set temporary password: sudo passwd toby Check user :*sudo chage -l toby* Same for adler. Verify /etc/shadow directory ● Move all the employees who were in the marketing department to a new group called **research**. Create this group if it doesn't exist. I verified users accounts and no one is in the marketing group. Script to verify users and groups https://github.com/davconbel/CYBER-PT-EAST-OCTOBER-102124-MTTH-CONS/blob/main/src/main/bash/module4/day2_activity3.sh Remove marketing group: **sudo groupdel marketing** Check if research group exists: *cat /etc/group* | *grep research* Create research group: *sudo addgroup research* **Added users toby,adler and mycroft to research**

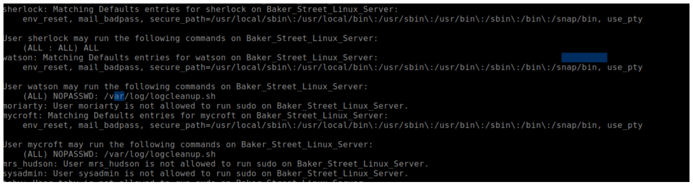
		Command used: *sudo usermod -aG research <username>*
		
		`moriarty: moriarty : moriarty engineering`
		`mycroft: mycroft : mycroft research`
		`mrs_hudson: mrs_hudson : mrs_hudson finance`
		`sysadmin: sysadmin : sysadmin`
		`toby: toby : toby research`
		`adler: adler : adler research`

☑	Updating and enforcing password policies	Line added to /etc/pam.d/common-password file password requisite pam_pwquality.so retry=2 minlen=8 ucredit=-1 lcredit=0 dcredit=0 ocredit=-1
		
		`#Add password quality for 2 retries, minimum 8 characters, 1 uppercase and 1 special character`
		`password requisite pam_pwquality.so retry=2 minlen=8 ucredit=1 lcredit=0 dcredit=0 ocredit=-1`

☑	Updating and enforcing sudo permissions	Using sudo visudo
		
		`@includedir /etc/sudoers.d`
		`sherlock ALL=(ALL:ALL) ALL`
		`watson ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh`
		`mycroft ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh`
		Before changes sudo -lU output
		
		`sherlock: Matching Defaults entries for sherlock on Baker_Street_Linux_Server:`
		`    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr`
		
		`User sherlock may run the following commands on Baker_Street_Linux_Server:`
		`    (ALL) NOPASSWD: ALL`
		`watson: Matching Defaults entries for watson on Baker_Street_Linux_Server:`
		`    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr`
		
		`User watson may run the following commands on Baker_Street_Linux_Server:`
		`    (ALL) NOPASSWD: ALL`
		`moriarty: Matching Defaults entries for moriarty on Baker_Street_Linux_Server:`
		`    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr`
		
		`User moriarty may run the following commands on Baker_Street_Linux_Server:`
		`    (ALL) NOPASSWD: ALL`
		`mycroft: User mycroft is not allowed to run sudo on Baker_Street_Linux_Server.`
		`mrs_hudson: User mrs_hudson is not allowed to run sudo on Baker_Street_Linux_S`

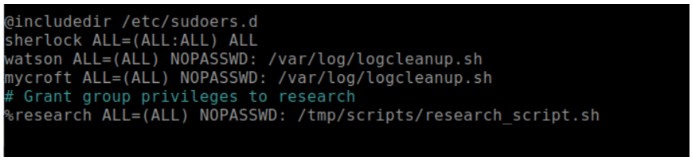
<table>
<tr>
<td></td>
<td></td>
<td>

After changes sudo -lu output



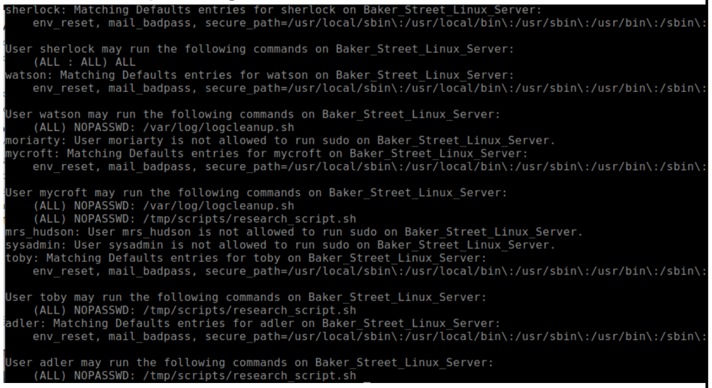
```
sherlock: Matching Defaults entries for sherlock on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User sherlock may run the following commands on Baker_Street_Linux_Server:
    (ALL : ALL) ALL
watson: Matching Defaults entries for watson on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User watson may run the following commands on Baker_Street_Linux_Server:
    (ALL) NOPASSWD: /var/log/logcleanup.sh
moriarty: User moriarty is not allowed to run sudo on Baker_Street_Linux_Server.
mycroft: Matching Defaults entries for mycroft on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User mycroft may run the following commands on Baker_Street_Linux_Server:
    (ALL) NOPASSWD: /var/log/logcleanup.sh
mrs_hudson: User mrs_hudson is not allowed to run sudo on Baker_Street_Linux_Server.
sysadmin: User sysadmin is not allowed to run sudo on Baker_Street_Linux_Server.
toby: User toby is not allowed to run sudo on Baker_Street_Linux_Server.
```

Changes to allowed research group to execute script



```
@includedir /etc/sudoers.d
sherlock ALL=(ALL:ALL) ALL
watson ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh
mycroft ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh
# Grant group privileges to research
%research ALL=(ALL) NOPASSWD: /tmp/scripts/research_script.sh
```

What sudo privileges look for users.

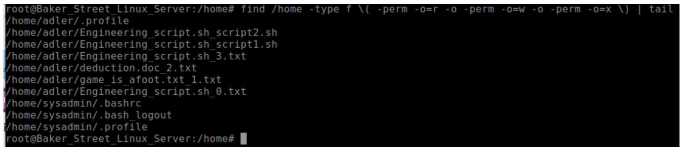


```
sherlock: Matching Defaults entries for sherlock on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:

User sherlock may run the following commands on Baker_Street_Linux_Server:
    (ALL : ALL) ALL
watson: Matching Defaults entries for watson on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:

User watson may run the following commands on Baker_Street_Linux_Server:
    (ALL) NOPASSWD: /var/log/logcleanup.sh
moriarty: User moriarty is not allowed to run sudo on Baker_Street_Linux_Server.
mycroft: Matching Defaults entries for mycroft on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:

User mycroft may run the following commands on Baker_Street_Linux_Server:
    (ALL) NOPASSWD: /var/log/logcleanup.sh
    (ALL) NOPASSWD: /tmp/scripts/research_script.sh
mrs_hudson: User mrs_hudson is not allowed to run sudo on Baker_Street_Linux_Server.
sysadmin: User sysadmin is not allowed to run sudo on Baker_Street_Linux_Server.
toby: Matching Defaults entries for toby on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:

User toby may run the following commands on Baker_Street_Linux_Server:
    (ALL) NOPASSWD: /tmp/scripts/research_script.sh
adler: Matching Defaults entries for adler on Baker_Street_Linux_Server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:

User adler may run the following commands on Baker_Street_Linux_Server:
    (ALL) NOPASSWD: /tmp/scripts/research_script.sh
```

</td>
</tr>
<tr>
<td>☑</td>
<td>Validating and updating permissions on files and directories</td>
<td>

69 Files found using command:

find /home -type f \(-perm -o=r -o -perm -o=w -o -perm -o=x \)



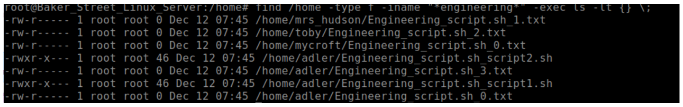
```
root@Baker_Street_Linux_Server:/home# find /home -type f \( -perm -o=r -o -perm -o=w -o -perm -o=x \) | tail
/home/adler/.profile
/home/adler/Engineering_script.sh_script2.sh
/home/adler/Engineering_script.sh_script1.sh
/home/adler/Engineering_script.sh_3.txt
/home/adler/deduction.doc_2.txt
/home/adler/game_is_afoot.txt_1.txt
/home/adler/Engineering_script.sh_0.txt
/home/sysadmin/.bashrc
/home/sysadmin/.bash_logout
/home/sysadmin/.profile
root@Baker_Street_Linux_Server:/home# 
```

Remove world permissions using compound command:
find /home -type f \(-perm -o=r -o -perm -o=w -o -perm -o=x \) -exec chmod o-rwx {} \;

</td>
</tr>
</table>

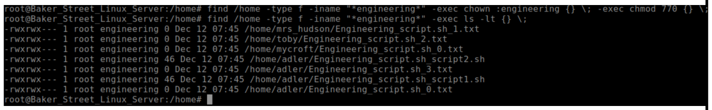
Engineering scripts (scripts with the word ‘engineering’ in the filename): Only members of the engineering group can view, edit, or execute.

Files listed for engineering case insensitive



```
root@Baker_Street_Linux_Server:/home# find /home -type f -iname "*engineering*" -exec ls -lt {} \;
-rw-r----- 1 root root 0 Dec 12 07:45 /home/mrs_hudson/Engineering_script.sh_1.txt
-rw-r----- 1 root root 0 Dec 12 07:45 /home/toby/Engineering_script.sh_2.txt
-rw-r----- 1 root root 0 Dec 12 07:45 /home/mycroft/Engineering_script.sh_0.txt
-rwxr-x--- 1 root root 46 Dec 12 07:45 /home/adler/Engineering_script.sh_script2.sh
-rw-r----- 1 root root 0 Dec 12 07:45 /home/adler/Engineering_script.sh_3.txt
-rwxr-x--- 1 root root 46 Dec 12 07:45 /home/adler/Engineering_script.sh_script1.sh
-rw-r----- 1 root root 0 Dec 12 07:45 /home/adler/Engineering_script.sh_0.txt
```

Command to update files:
*find /home -type f -iname "*engineering*" -exec chown :engineering {} \; -exec chmod 770 {} \;*
Results after command:



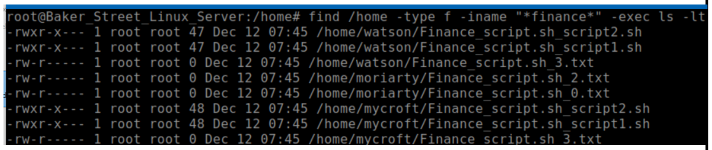
```
root@Baker_Street_Linux_Server:/home# find /home -type f -iname "*engineering*" -exec chown :engineering {} \; -exec chmod 770 {} \;
root@Baker_Street_Linux_Server:/home# find /home -type f -iname "*engineering*" -exec ls -lt {} \;
-rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/mrs_hudson/Engineering_script.sh_1.txt
-rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/toby/Engineering_script.sh_2.txt
-rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/mycroft/Engineering_script.sh_0.txt
-rwxrwx--- 1 root engineering 46 Dec 12 07:45 /home/adler/Engineering_script.sh_script2.sh
-rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/adler/Engineering_script.sh_3.txt
-rwxrwx--- 1 root engineering 46 Dec 12 07:45 /home/adler/Engineering_script.sh_script1.sh
-rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/adler/Engineering_script.sh_0.txt
root@Baker_Street_Linux_Server:/home# 
```

- **Research scripts:** Only members of the research group can view, edit, or execute.

No files found for research

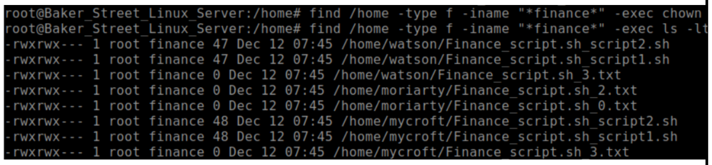
- **Finance scripts:** Only members of the finance group can view, edit, or execute.

List of files found

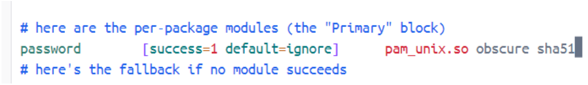
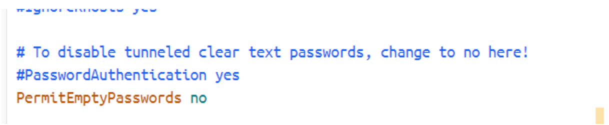
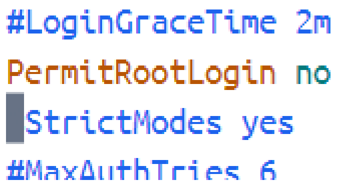


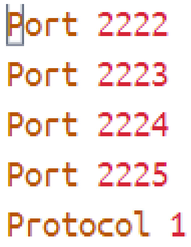
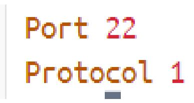
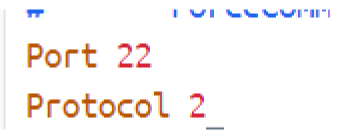
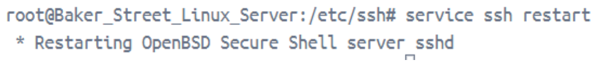
```
root@Baker_Street_Linux_Server:/home# find /home -type f -iname "*finance*" -exec ls -lt
-rwxr-x--- 1 root root 47 Dec 12 07:45 /home/watson/Finance_script.sh_script2.sh
-rwxr-x--- 1 root root 47 Dec 12 07:45 /home/watson/Finance_script.sh_script1.sh
-rw-r----- 1 root root 0 Dec 12 07:45 /home/watson/Finance_script.sh_3.txt
-rw-r----- 1 root root 0 Dec 12 07:45 /home/moriarty/Finance_script.sh_2.txt
-rw-r----- 1 root root 0 Dec 12 07:45 /home/moriarty/Finance_script.sh_0.txt
-rwxr-x--- 1 root root 48 Dec 12 07:45 /home/mycroft/Finance_script.sh_script2.sh
-rwxr-x--- 1 root root 48 Dec 12 07:45 /home/mycroft/Finance_script.sh_script1.sh
-rw-r----- 1 root root 0 Dec 12 07:45 /home/mycroft/Finance_script.sh_3.txt
```

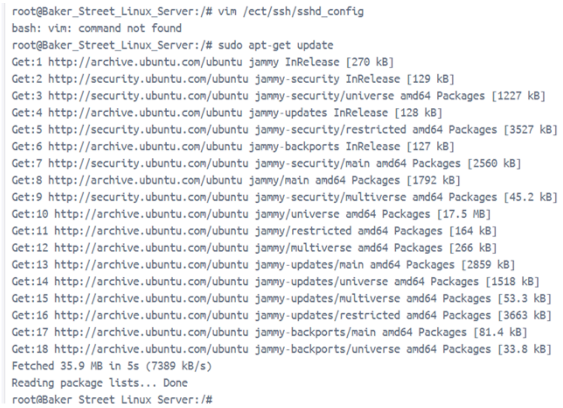
After change:



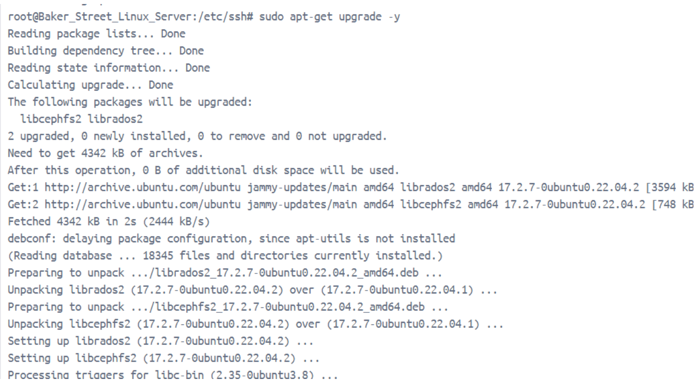
```
root@Baker_Street_Linux_Server:/home# find /home -type f -iname "*finance*" -exec chown
root@Baker_Street_Linux_Server:/home# find /home -type f -iname "*finance*" -exec ls -lt
-rwxrwx--- 1 root finance 47 Dec 12 07:45 /home/watson/Finance_script.sh_script2.sh
-rwxrwx--- 1 root finance 47 Dec 12 07:45 /home/watson/Finance_script.sh_script1.sh
-rwxrwx--- 1 root finance 0 Dec 12 07:45 /home/watson/Finance_script.sh_3.txt
-rwxrwx--- 1 root finance 0 Dec 12 07:45 /home/moriarty/Finance_script.sh_2.txt
-rwxrwx--- 1 root finance 0 Dec 12 07:45 /home/moriarty/Finance_script.sh_0.txt
-rwxrwx--- 1 root finance 48 Dec 12 07:45 /home/mycroft/Finance_script.sh_script2.sh
-rwxrwx--- 1 root finance 48 Dec 12 07:45 /home/mycroft/Finance_script.sh_script1.sh
-rwxrwx--- 1 root finance 0 Dec 12 07:45 /home/mycroft/Finance_script.sh_3.txt
```

☑	Optional: Updating password hashing configuration	To update the password hashing configuration so it can use the latest algorithm sha-512 we need to edit the file /etc/pam.d/common-password. And ensure it has the following line for pam_unix.so. password [success=1 default=ignore] pam_unix.so obscure sha512  # here are the per-package modules (the "Primary" block) password [success=1 default=ignore] pam_unix.so obscure sha51 # here's the fallback if no module succeeds
☑	Auditing and securing SSH	Configure SSH to **not** allow the ability to: □ SSH with empty passwords We need to edit sshd_congig file and set the flag PermitEmptyPassword to no Command: sudo nano /etc/ssh/sshd_config  #IgnoreRhosts yes # To disable tunneled clear text passwords, change to no here! #PasswordAuthentication yes PermitEmptyPasswords no □ SSH with the root user We need to edit sshd_congig file and set the flag PermitRootLogin to no Command: sudo nano /etc/ssh/sshd_config  #LoginGraceTime 2m PermitRootLogin no StrictModes yes #MaxAuthTries 6 □ SSH with any other ports besides 22 In the file sshd_config multiple ports are defined.

		
		`Port 2222`
		`Port 2223`
		`Port 2224`
		`Port 2225`
		`Protocol 1`
		We need to delete entries and only leave Port 22 defined.
		
		`Port 22`
		`Protocol 1`
		Now to enable Protocol 2 we just change the line after after Port
		
		`Port 22`
		`Protocol 2`
		Restart service
		
		`root@Baker_Street_Linux_Server:/etc/ssh# service ssh restart`
		` * Restarting OpenBSD Secure Shell server sshd`

☑	Reviewing and updating system packages	☐ Run *apt update* to update your package manager to make sure it has the latest version of all packages.
		
		`root@Baker_Street_Linux_Server:/# vim /ect/ssh/sshd_config`
		`bash: vim: command not found`
		`root@Baker_Street_Linux_Server:/# sudo apt-get update`
		`Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]`
		`Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]`
		`Get:3 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1227 kB]`
		`Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]`
		`Get:5 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3527 kB]`
		`Get:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]`
		`Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2560 kB]`
		`Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]`
		`Get:9 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [45.2 kB]`
		`Get:10 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]`
		`Get:11 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]`
		`Get:12 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]`
		`Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2859 kB]`
		`Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1518 kB]`
		`Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [53.3 kB]`
		`Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [3663 kB]`
		`Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [81.4 kB]`
		`Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [33.8 kB]`
		`Fetched 35.9 MB in 5s (7389 kB/s)`
		`Reading package lists... Done`
		`root@Baker Street Linux Server:/#`

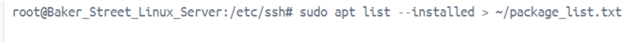
☐ Next, run *apt upgrade -y* to update all already installed packages to the latest versions.



```
root@Baker_Street_Linux_Server:/etc/ssh# sudo apt-get upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  libcephfs2 librados2
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 4342 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 librados2 amd64 17.2.7-0ubuntu0.22.04.2 [3594 kB
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcephfs2 amd64 17.2.7-0ubuntu0.22.04.2 [748 kB
Fetched 4342 kB in 2s (2444 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
(Reading database ... 18345 files and directories currently installed.)
Preparing to unpack .../librados2_17.2.7-0ubuntu0.22.04.2_amd64.deb ...
Unpacking librados2 (17.2.7-0ubuntu0.22.04.2) over (17.2.7-0ubuntu0.22.04.1) ...
Preparing to unpack .../libcephfs2_17.2.7-0ubuntu0.22.04.2_amd64.deb ...
Unpacking libcephfs2 (17.2.7-0ubuntu0.22.04.2) over (17.2.7-0ubuntu0.22.04.1) ...
Setting up librados2 (17.2.7-0ubuntu0.22.04.2) ...
Setting up libcephfs2 (17.2.7-0ubuntu0.22.04.2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
```

☐ Create a file called package_list.txt, which contains all installed packages.

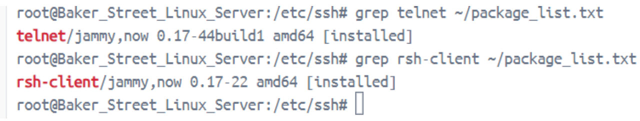
Run command apt list –installed and redirect the output to file package_list.txt
Command: sudo apt list --installed > ~/package_list.txt



```
root@Baker_Street_Linux_Server:/etc/ssh# sudo apt list --installed > ~/package_list.txt
```

☐ Identify if any of the following packages are on the list as having these could introduce a security issue:
   ☐ telnet
   ☐ Rsh-client

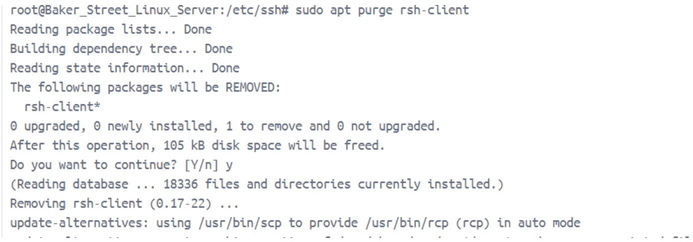
Grep the package_list.txt file previously created and grep for telnet and rsh-client



```
root@Baker_Street_Linux_Server:/etc/ssh# grep telnet ~/package_list.txt
telnet/jammy,now 0.17-44build1 amd64 [installed]
root@Baker_Street_Linux_Server:/etc/ssh# grep rsh-client ~/package_list.txt
rsh-client/jammy,now 0.17-22 amd64 [installed]
root@Baker_Street_Linux_Server:/etc/ssh#
```

- □ If they are on the list, remove those packages.

We have two options; to only remove the package using apt remove, or remove package and configuration files using apt purge. I am going to use the purge option for this packages.

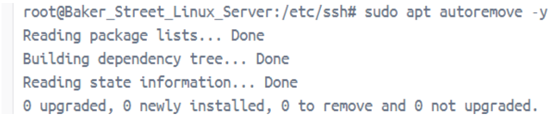


```
root@Baker_Street_Linux_Server:/etc/ssh# sudo apt purge rsh-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  rsh-client*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 105 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 18336 files and directories currently installed.)
Removing rsh-client (0.17-22) ...
update-alternatives: using /usr/bin/scp to provide /usr/bin/rcp (rcp) in auto mode
```

We can update the package_list.txt file again. Telnet is considered a security issue because of the lack of encryption and plain text communication.

Ssh-client is considered a security issue because it is outdated and insecure due to its lack of encryption and weak authentication methods.

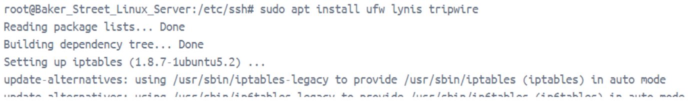
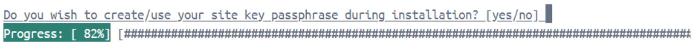
- □ Remove all unnecessary dependencies of those packages with *apt autoremove -y*.



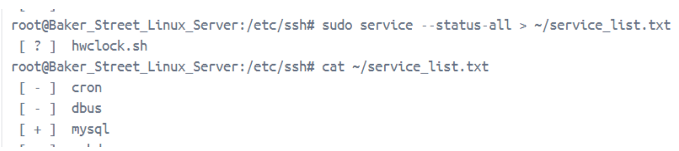
```
root@Baker_Street_Linux_Server:/etc/ssh# sudo apt autoremove -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

- □ Add the following packages:
  - □ ufw
  - □ lynis
  - □ Tripwire

I am going to installed all 3 packages in a single command.
Command: sudo apt install ufw lynis tripwire

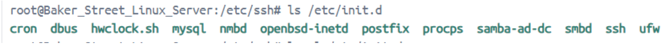
		
		`root@Baker_Street_Linux_Server:/etc/ssh# sudo apt install ufw lynis tripwire`
		`Reading package lists... Done`
		`Building dependency tree... Done`
		`Setting up iptables (1.8.7-1ubuntu5.2) ...`
		`update-alternatives: using /usr/sbin/iptables-legacy to provide /usr/sbin/iptables (iptables) in auto mode`
		`update-alternatives: using /usr/sbin/ip6tables-legacy to provide /usr/sbin/ip6tables (ip6tables) in auto mode`
		
		`Do you wish to create/use your site key passphrase during installation? [yes/no]`
		`Progress: [ 82%] [##########################################################################################`
		□ Once the packages have been installed, research and document the hardening features these packages can provide.
		The `ufw` **(Uncomplicated Firewall)** package in Ubuntu provides a simple interface for managing firewall rules. It includes several features that can be configured to **harden the security** of your system.
		Lynis is a powerful **security auditing tool** for Unix-based systems, including Linux. It helps assess vulnerabilities, harden system configurations, and ensure compliance with security standards.
		Tripwire is a powerful file integrity monitoring tool used to enhance system security by detecting unauthorized changes to files and directories. It's widely used for intrusion detection and system auditing.

☑	Disabling unnecessary services	□ Run the command to list out all services. Output this into a file called service_list.txt. Since systemctl is not available in the docker image we need to use the service command. Command: *sudo service --status-all > ~/service_lists.txt*



```
root@Baker_Street_Linux_Server:/etc/ssh# sudo service --status-all > ~/service_list.txt
 [ ? ]  hwclock.sh
root@Baker_Street_Linux_Server:/etc/ssh# cat ~/service_list.txt
 [ - ]  cron
 [ - ]  dbus
 [ + ]  mysql
```

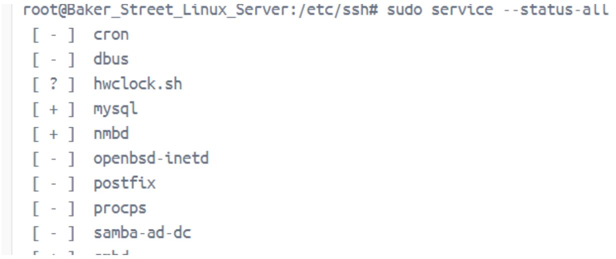
Another way will be to list the /etc/init.d directory



```
root@Baker_Street_Linux_Server:/etc/ssh# ls /etc/init.d
cron  dbus  hwclock.sh  mysql  nmbd  openbsd-inetd  postfix  procps  samba-ad-dc  smbd  ssh  ufw
```

- ☐ Identify if any of the following services are running:
  - ☐ mysql
  - ☐ Samba

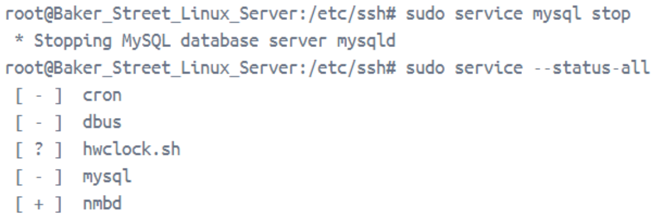
From the service –status-all command we can see that mysql has the + sign which indicates is running and samba is stopped since it has the - sign



```
root@Baker_Street_Linux_Server:/etc/ssh# sudo service --status-all
 [ - ]  cron
 [ - ]  dbus
 [ ? ]  hwclock.sh
 [ + ]  mysql
 [ + ]  nmbd
 [ - ]  openbsd-inetd
 [ - ]  postfix
 [ - ]  procps
 [ - ]  samba-ad-dc
```

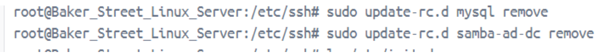
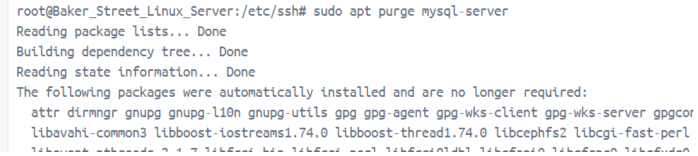
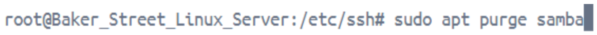
- ☐ If any of the above services are running,
  - ☐ Stop them
  - ☐ Disable them
  - ☐ Remove them

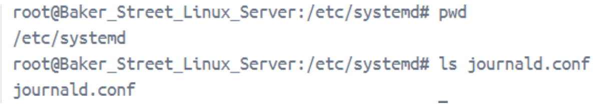
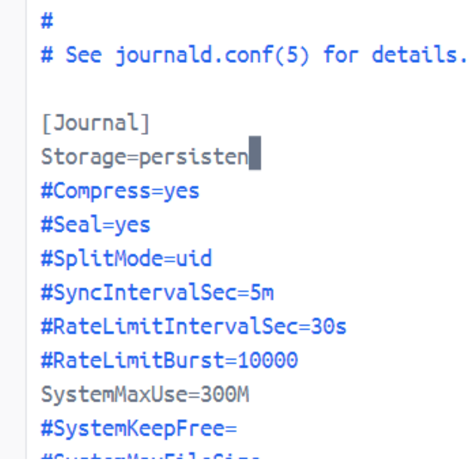
We know the mysql service is running so we can stopped:
Command: sudo service mysql stop

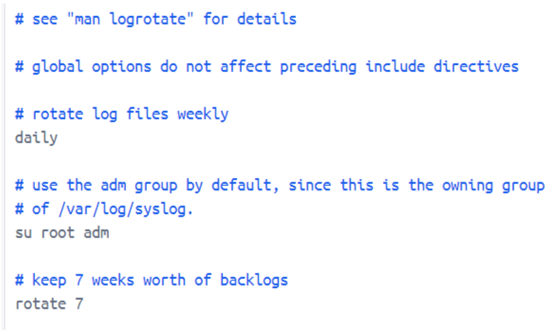


```
root@Baker_Street_Linux_Server:/etc/ssh# sudo service mysql stop
 * Stopping MySQL database server mysqld
root@Baker_Street_Linux_Server:/etc/ssh# sudo service --status-all
 [ - ]  cron
 [ - ]  dbus
 [ ? ]  hwclock.sh
 [ - ]  mysql
 [ + ]  nmbd
```

To disable the service without using systemctl we use

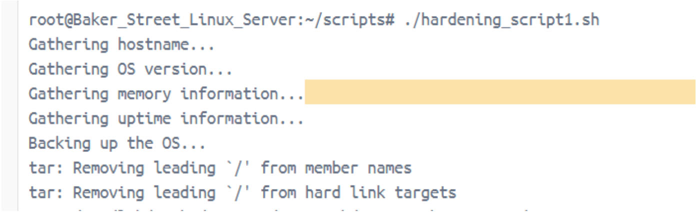
		sudo update-rc.d <service-name> remove  root@Baker_Street_Linux_Server:/etc/ssh# sudo update-rc.d mysql remove root@Baker_Street_Linux_Server:/etc/ssh# sudo update-rc.d samba-ad-dc remove Remove samba and mysql-server packages.  root@Baker_Street_Linux_Server:/etc/ssh# sudo apt purge mysql-server Reading package lists... Done Building dependency tree... Done Reading state information... Done The following packages were automatically installed and are no longer required: attr dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgcor libavahi-common3 libboost-iostreams1.74.0 libboost-thread1.74.0 libcephfs2 libcgi-fast-perl  root@Baker_Street_Linux_Server:/etc/ssh# sudo apt purge samba

□	Enabling and configuring logging	□ Access the *journald.conf* file located */etc/systemd/*.
		
		root@Baker_Street_Linux_Server:/etc/systemd# pwd
		/etc/systemd
		root@Baker_Street_Linux_Server:/etc/systemd# ls journald.conf
		journald.conf
		□ Use nano to edit the following settings in the file. Be sure to uncomment the lines!
		□ Set "**storage=persistent**"
		□ Set "**systemMaxUse=300M**"
		
		#
		# See journald.conf(5) for details.
		
		[Journal]
		Storage=persisten
		#Compress=yes
		#Seal=yes
		#SplitMode=uid
		#SyncIntervalSec=5m
		#RateLimitIntervalSec=30s
		#RateLimitBurst=10000
		SystemMaxUse=300M
		#SystemKeepFree=

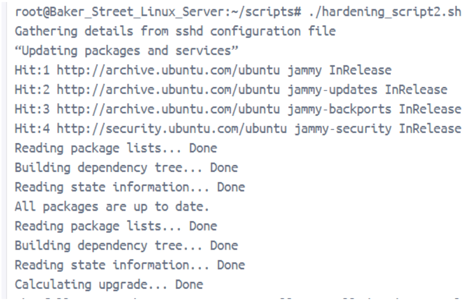
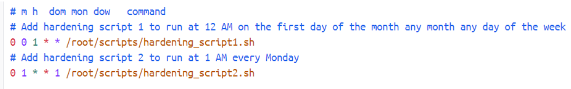
		☐ To prevent logs from taking up too much space, you will need to configure log rotation. (Use the following guide to assist: https://linux.die.net/man/8/logrotate) ☐ Edit the file: /etc/logrotate.conf with the following settings: ☐ Change the log rotation from weekly to daily. ☐ Rotate out the logs after 7 days.  # see "man logrotate" for details # global options do not affect preceding include directives # rotate log files weekly daily # use the adm group by default, since this is the owning group # of /var/log/syslog. su root adm # keep 7 weeks worth of backlogs rotate 7

☑	Scripts created	First script file:
		
		#!/bin/bash
		
		# Variable for the report output file, choose an output file name
		REPORT_FILE="system_hardening.txt"
		
		touch $REPORT_FILE
		# Output the hostname
		echo "Gathering hostname..."
		echo "Hostname: $(hostname)" >> $REPORT_FILE
		printf "\n" >> $REPORT_FILE
		
		# Output the OS version
		echo "Gathering OS version..."
		echo "OS Version: $(uname -a)" >> $REPORT_FILE
		printf "\n" >> $REPORT_FILE
		
		# Output memory information

		echo "Gathering memory information..." echo "Memory Information: $(free -h)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Output uptime information echo "Gathering uptime information..." echo "Uptime Information: $(uptime)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Backup the OS echo "Backing up the OS..." sudo tar -cvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run / >> $REPORT_FILE echo "OS backup completed." >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Output the sudoers file to the report echo "Gathering sudoers file..." echo "Sudoers file:$(getent group sudo)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Script to check for files with world permissions and update them echo "Checking for files with world permissions..." find /home -type f \(-perm -o=r -o -perm -o=w -o -perm -o=x \) -exec chmod o-rwx {} \; echo "World permissions have been removed from any files found." >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Find specific files and update their permissions echo "Updating permissions for specific scripts..." # Engineering scripts - Only members of the engineering group

		echo "Updating permissions for Engineering scripts." find /home -type f -iname "*engineering*" -exec chown :engineering {} \; -exec chmod 770 {} \; echo "Permissions updated for Engineering scripts." >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Research scripts - Only members of the research group echo "Updating permissions for Research scripts..." find /home -type f -iname "*research*" -exec chown :research {} \; -exec chmod 770 {} \; echo "Permissions updated for Research scripts" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Finance scripts - Only members of the finance group echo "Updating permissions for Finance scripts" # Placeholder for command to update permissions find /home -type f -iname "*finance*" -exec chown :finance {} \; -exec chmod 770 {} \; echo "Permissions updated for Finance scripts." >> $REPORT_FILE printf "\n" >> $REPORT_FILE echo "Script execution completed. Check $REPORT_FILE for details." It runs  root@Baker_Street_Linux_Server:~/scripts# ./hardening_script1.sh Gathering hostname... Gathering OS version... Gathering memory information... Gathering uptime information... Backing up the OS... tar: Removing leading `/' from member names tar: Removing leading `/' from hard link targets

		Second script file:
		
		#!/bin/bash
		
		# Variable for the report output file, choose a NEW output file name
		REPORT_FILE="system_configuration_hardening.txt"
		
		# Output the sshd configuration file
		echo "Gathering details from sshd configuration file"
		
		echo "sshd configuration file:$(cat /etc/ssh/sshd_config)" >> $REPORT_FILE
		printf "\n" >> $REPORT_FILE
		
		# Update packages and services
		echo “Updating packages and services”
		
		apt update
		
		apt upgrade -y
		
		echo "Packages have been updated and upgraded" >> $REPORT_FILE
		printf "\n" >> $REPORT_FILE
		
		echo "Installed Packages:$(apt list --installed)" >> $REPORT_FILE
		printf "\n" >> $REPORT_FILE
		
		echo “Printing out logging configuration data”
		
		echo "journald.conf file data: $(cat /etc/systemd/journald.conf)" >> $REPORT_FILE
		printf "\n" >> $REPORT_FILE
		
		echo "logrotate.conf file data:$(cat /etc/logrotate.conf)" >> $REPORT_FILE
		printf "\n" >> $REPORT_FILE
		
		echo "Script execution completed. Check $REPORT_FILE for details."

		 root@Baker_Street_Linux_Server:~/scripts# ./hardening_script2.sh Gathering details from sshd configuration file “Updating packages and services” Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease Hit:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease Reading package lists... Done Building dependency tree... Done Reading state information... Done All packages are up to date. Reading package lists... Done Building dependency tree... Done Reading state information... Done Calculating upgrade... Done It runs
☑	Scripts scheduled with cron	Use command contab -e to edit cron file and add the following: # Add hardening script 1 to run at 12 AM on the first day of the month any month any day of the week 0 0 1 * * /root/scripts/hardening_script1.sh # Add hardening script 2 to run at 1 AM every Monday 0 1 * * 1 /root/scripts/hardening_script2.sh  # m h dom mon dow command # Add hardening script 1 to run at 12 AM on the first day of the month any month any day of the week 0 0 1 * * /root/scripts/hardening_script1.sh # Add hardening script 2 to run at 1 AM every Monday 0 1 * * 1 /root/scripts/hardening_script2.sh