Cybersecurity

Project 1 Hardening Summary and Checklist

OS Information

Customer	Baker Street Corporation
Hostname	Baker_Street_Linux_Server
OS Version	x86_64-pc-linux-gnu
Memory information	root@Baker_Street_Linux_Server:/# free total used free shared buff/cache available Mem: 16182804 1423676 10602464 210968 4156664 14195376 Swap: 0 0 0
Uptime information	root@Baker_Street_Linux_Server:/# uptime 00:07:14 up 21 min, 0 users, load average: 0.52, 0.64, 0.49

Checklist

Completed	Activity	Script(s) used / Tasks completed / Screenshots
☑	OS backup	**Part 1** The first part of server hardening is taking inventory on our server. We will need to collect data on the Host, OS, memory and uptime. After that we will back the whole thing up so that any changes we make can be rolled back if needed. In our backup we do not include the proc, tmp, mnt, sys, dev, and run folders. This is because those folders contain only temporary or runtime data that does not need to be backed up. echo -e $MACHTYPE

```
root@Baker_Street_Linux_Server:/# echo -e $MACHTYPE
x86_64-pc-linux-gnu
```

hostname -s

```
root@Baker_Street_Linux_Server:/# hostname -s
Baker_Street_Linux_Server
```

free

```
root@Baker_Street_Linux_Server:/# free
              total        used        free      shared  buff/cache   available
Mem:       16182804     1423676    10602464      210968     4156664    14195376
Swap:              0           0           0
```

uptime

```
root@Baker_Street_Linux_Server:/# uptime
 00:07:14 up 21 min,  0 users,  load average: 0.52, 0.64, 0.49
```

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/dbus-1/
/etc/dbus-1/system.d/
/etc/dbus-1/session.d/
/etc/python3.10/
/etc/python3.10/sitecustomize.py
/boot/
/media/
/lib32
/sbin
/.dockerenv
tar: /: file changed as we read it
root@Baker_Street_Linux_Server:/# history
    1  sudo tar -cvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backu
p.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=
/dev --exclude=/run /
    2  history
```

```
root@Baker_Street_Linux_Server:/# ls
baker_street_backup.tar.gz  dev   lib    libx32  opt   run   sys  var
bin                         etc   lib32  media   proc  sbin  tmp
boot                        home  lib64  mnt     root  srv   usr
```

☑	Auditing users and groups	**Part 2** Next we will audit our users and groups. In this scenario, we have 4 employees who are no longer with the company. We will delete their accounts with the deluser command. We use the --remove-home as a best practice to ensure the home directory and related files are removed. These files could contain sensitive data and another user with the correct permissions could take advantage of this. This is important in reducing the attack surface. Next we have some users on leave. We do not want someone to be able to access their account or sensitive files while they are not in use. We will use passwd -l to lock the account of the user. Doing this sets appends a ! to the beginning of the password hash making the string invalid. For our

active users and when the employees on leave return we will use passwd -u to ensure the account is not locked and the user can log in to complete their work. *notably when unlocking user accounts toby and adler we receive a message indicating they do not have passwords set. We can set the password on their accounts to a temporary one and set it to expire so they are prompted to create a new one upon successful log in.

Finally, we were instructed to move everyone in marketing to research. Unfortunately, the project environment was not prepared correctly so we will use our imagination and assume these poor employees were the ones let go. However, to demonstrate my understanding we would have created a new group called research and used usermod to add the desired users into that group. We would then delete the marketing group.

```
root@Baker_Street_Linux_Server:/home# ls
adler    irene     mary      mrs_hudson  sherlock  toby
gregson  lestrade  moriarty  mycroft     sysadmin  watson
```

deluser --remove-home irene

```
Removing files ...
Removing user `lestrade' ...
Warning: group `lestrade' has no more members.
Done.
root@Baker_Street_Linux_Server:/home# deluser --remove-home irene
Looking for files to backup/remove ...
Removing files ...
Removing user `irene' ...
Warning: group `irene' has no more members.
Done.
root@Baker_Street_Linux_Server:/home# deluser --remove-home mary
Looking for files to backup/remove ...
Removing files ...
Removing user `mary' ...
Warning: group `mary' has no more members.
Done.
root@Baker_Street_Linux_Server:/home# deluser --remove-home gregson
Looking for files to backup/remove ...
Removing files ...
Removing user `gregson' ...
Warning: group `gregson' has no more members.
Done.
root@Baker_Street_Linux_Server:/home#
```

passwd -l moriarty

```
root@Baker_Street_Linux_Server:/home# passwd -l moriarty
passwd: password expiry information changed.
root@Baker_Street_Linux_Server:/home# passwd -l mrs_hudson
passwd: password expiry information changed.
```

```
root@Baker_Street_Linux_Server:/home# cat /etc/shadow | grep moriarty
moriarty:!$y$j9T$VLx9.NPPbGh1FzjVTeoqd/$sSwx1RTe0iq3rf0/C/JlG3ms5PmSpPTUkUuH4IRc
916:20145:0:99999:7:::
```

passwd -u sherlock

```
root@Baker_Street_Linux_Server:/home# passwd -u sherlock
passwd: password expiry information changed.
root@Baker_Street_Linux_Server:/home# passwd -u watson
passwd: password expiry information changed.
root@Baker_Street_Linux_Server:/home# passwd -u mycroft
passwd: password expiry information changed.
root@Baker_Street_Linux_Server:/home# passwd -u toby
passwd: unlocking the password would result in a passwordless account.
You should set a password with usermod -p to unlock the password of this account
.
root@Baker_Street_Linux_Server:/home# passwd -u adler
passwd: unlocking the password would result in a passwordless account.
You should set a password with usermod -p to unlock the password of this account
.
```

groups *

```
root@Baker_Street_Linux_Server:/home# groups *
adler : adler
moriarty : moriarty engineering
mrs_hudson : mrs_hudson finance
mycroft : mycroft
sherlock : sherlock engineering
sysadmin : sysadmin
toby : toby
watson : watson engineering
```

addgroup research

```
root@Baker_Street_Linux_Server:/home# addgroup research
Adding group `research' (GID 1004) ...
Done.
```

For some reason my marketing group is empty. I would have used:
usermod -G research username

delgroup marketing

```
root@Baker_Street_Linux_Server:/home# delgroup marketing
Removing group `marketing' ...
Done.
```

☑	Updating and enforcing password policies	**Part 3** Next we will enforce a stricter password policy for our users. For this scenario we will enforce a minimum of 8 characters with 1 special character, one uppercase, and 2 retries. However, best practices would likely have number requirements as well and in recent times a password length of 12 characters. nano /etc/pam.d/common-password password requisite pam_pwquality.so minlen=8 ocredit=1 ucredit=1 retry=2

```
# here are the per-package modules (the "Primary" block)
password        [success=1 default=ignore]      pam_unix.so obscure yescrypt
password        requisite                       pam_pwquality.so minlen=8 ocredit=1 ucredit=1 retry=2
# 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
```

☑ Updating and enforcing sudo permissions

Part 4

Sudo permissions are very critical as they prevent users from using commands and files they should not have access to. Additionally, they allow users to have escalated privileges when necessary to complete tasks. I have displayed the before and after screenshots of the sudoers file. We have removed sudo permissions from everyone except sherlock. Sherlock was changed to still require a password. Additionally, watson and moriarty now only have permissions to run a specific script as sudo.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
sherlock ALL=(ALL) NOPASSWD:ALL
watson ALL=(ALL) NOPASSWD:ALL
moriarty ALL=(ALL) NOPASSWD:ALL
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
%research ALL=(ALL) /tmp/scripts/research_script.sh
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
sherlock ALL=(ALL:ALL) ALL
watson ALL=(ALL) /var/log/logcleanup.sh
moriarty ALL=(ALL) /var/log/logcleanup.sh
```

☑	Validating and updating permissions on files and directories	Part 5
		Permissions on files are also important. I will show the before and after of our file trees to show changes. We will first remove all world permissions. This reduces our attack surface by ensuring people without privileges cannot access any files. We are tasked with changing the files ownership to specific groups and to ensure that group is the only one able to read, write, or execute the scripts.
		`total 8`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 Engineering_script.sh_0.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 Engineering_script.sh_3.txt`
		`-rwxr-xr-x 1 root root 46 Dec 12 07:45 Engineering_script.sh_script1.sh`
		`-rwxr-xr-x 1 root root 46 Dec 12 07:45 Engineering_script.sh_script2.sh`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_1.txt`
		
		`./moriarty:`
		`total 8`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 Finance_script.sh_0.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 Finance_script.sh_2.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 elementary.txt_1.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_3.txt`
		`-rwxr-xr-x 1 root root 49 Dec 12 07:45 game_is_afoot.txt_script1.sh`
		`-rwxr-xr-x 1 root root 49 Dec 12 07:45 game_is_afoot.txt_script2.sh`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 my_file.txt`
		
		`./mrs_hudson:`
		`total 8`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 Engineering_script.sh_1.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_0.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 elementary.txt_3.txt`
		`-rwxr-xr-x 1 root root 51 Dec 12 07:45 elementary.txt_script1.sh`
		`-rwxr-xr-x 1 root root 51 Dec 12 07:45 elementary.txt_script2.sh`
		
		`./mycroft:`
		`total 8`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 Engineering_script.sh_0.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 Finance_script.sh_3.txt`
		`-rwxr-xr-x 1 root root 48 Dec 12 07:45 Finance_script.sh_script1.sh`
		`-rwxr-xr-x 1 root root 48 Dec 12 07:45 Finance_script.sh_script2.sh`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_1.txt`
		`-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt`

```
./sherlock:
total 8
-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_3.txt
-rwxr-xr-x 1 root root 49 Dec 12 07:45 deduction.doc_script1.sh
-rwxr-xr-x 1 root root 49 Dec 12 07:45 deduction.doc_script2.sh
-rw-r--r-- 1 root root  0 Dec 12 07:45 elementary.txt_0.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_1.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_2.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 my_file.txt

./sysadmin:
total 0

./toby:
total 8
-rw-r--r-- 1 root root  0 Dec 12 07:45 Engineering_script.sh_2.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_1.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 elementary.txt_0.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 elementary.txt_3.txt
-rwxr-xr-x 1 root root 45 Dec 12 07:45 elementary.txt_script1.sh
-rwxr-xr-x 1 root root 45 Dec 12 07:45 elementary.txt_script2.sh

./watson:
total 8
-rw-r--r-- 1 root root  0 Dec 12 07:45 Finance_script.sh_3.txt
-rwxr-xr-x 1 root root 47 Dec 12 07:45 Finance_script.sh_script1.sh
-rwxr-xr-x 1 root root 47 Dec 12 07:45 Finance_script.sh_script2.sh
-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_0.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_1.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt
-rw-r--r-- 1 root root  0 Dec 12 07:45 my_file.txt
```

chmod /home/ -R o= *

```
root@Baker_Street_Linux_Server:/home# chmod -R o= *
root@Baker_Street_Linux_Server:/home# ls -lR
.:
total 64
drwxr-x--- 1 adler      adler      4096 Dec 12 07:45 adler
drwxr-x--- 1 moriarty   moriarty   4096 Dec 12 07:45 moriarty
drwxr-x--- 1 mrs_hudson mrs_hudson 4096 Dec 12 07:45 mrs_hudson
drwxr-x--- 1 mycroft    mycroft    4096 Dec 12 07:45 mycroft
drwxr-x--- 1 sherlock   sherlock   4096 Dec 12 07:45 sherlock
drwxr-x--- 1 sysadmin   sysadmin   4096 Dec 12 07:45 sysadmin
drwxr-x--- 1 toby       toby       4096 Dec 12 07:45 toby
drwxr-x--- 1 watson     watson     4096 Dec 12 07:45 watson

./adler:
total 8
-rw-r----- 1 root root  0 Dec 12 07:45 Engineering_script.sh_0.txt
-rw-r----- 1 root root  0 Dec 12 07:45 Engineering_script.sh_3.txt
-rwxr-x--- 1 root root 46 Dec 12 07:45 Engineering_script.sh_script1.sh
-rwxr-x--- 1 root root 46 Dec 12 07:45 Engineering_script.sh_script2.sh
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt
-rw-r----- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_1.txt

./moriarty:
total 8
-rw-r----- 1 root root  0 Dec 12 07:45 Finance_script.sh_0.txt
-rw-r----- 1 root root  0 Dec 12 07:45 Finance_script.sh_2.txt
-rw-r----- 1 root root  0 Dec 12 07:45 elementary.txt_1.txt
-rw-r----- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_3.txt
-rwxr-x--- 1 root root 49 Dec 12 07:45 game_is_afoot.txt_script1.sh
-rwxr-x--- 1 root root 49 Dec 12 07:45 game_is_afoot.txt_script2.sh
-rw-r----- 1 root root  0 Dec 12 07:45 my_file.txt

./mrs_hudson:
total 8
-rw-r----- 1 root root  0 Dec 12 07:45 Engineering_script.sh_1.txt
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_0.txt
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt
-rw-r----- 1 root root  0 Dec 12 07:45 elementary.txt_3.txt
-rwxr-x--- 1 root root 51 Dec 12 07:45 elementary.txt_script1.sh
-rwxr-x--- 1 root root 51 Dec 12 07:45 elementary.txt_script2.sh
```

```
./mycroft:
total 8
-rw-r----- 1 root root  0 Dec 12 07:45 Engineering_script.sh_0.txt
-rw-r----- 1 root root  0 Dec 12 07:45 Finance_script.sh_3.txt
-rwxr-x--- 1 root root 48 Dec 12 07:45 Finance_script.sh_script1.sh
-rwxr-x--- 1 root root 48 Dec 12 07:45 Finance_script.sh_script2.sh
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_1.txt
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt

./sherlock:
total 8
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_3.txt
-rwxr-x--- 1 root root 49 Dec 12 07:45 deduction.doc_script1.sh
-rwxr-x--- 1 root root 49 Dec 12 07:45 deduction.doc_script2.sh
-rw-r----- 1 root root  0 Dec 12 07:45 elementary.txt_0.txt
-rw-r----- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_1.txt
-rw-r----- 1 root root  0 Dec 12 07:45 game_is_afoot.txt_2.txt
-rw-r----- 1 root root  0 Dec 12 07:45 my_file.txt

./sysadmin:
total 0

./toby:
total 8
-rw-r----- 1 root root  0 Dec 12 07:45 Engineering_script.sh_2.txt
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_1.txt
-rw-r----- 1 root root  0 Dec 12 07:45 elementary.txt_0.txt
-rw-r----- 1 root root  0 Dec 12 07:45 elementary.txt_3.txt
-rwxr-x--- 1 root root 45 Dec 12 07:45 elementary.txt_script1.sh
-rwxr-x--- 1 root root 45 Dec 12 07:45 elementary.txt_script2.sh

./watson:
total 8
-rw-r----- 1 root root  0 Dec 12 07:45 Finance_script.sh_3.txt
-rwxr-x--- 1 root root 47 Dec 12 07:45 Finance_script.sh_script1.sh
-rwxr-x--- 1 root root 47 Dec 12 07:45 Finance_script.sh_script2.sh
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_0.txt
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_1.txt
-rw-r----- 1 root root  0 Dec 12 07:45 deduction.doc_2.txt
-rw-r----- 1 root root  0 Dec 12 07:45 my_file.txt
```

Part 5 - Rationale

The instructions are to update specifically the engineering and finance scripts so that only members of the group can access it. Therefore I changed the owner of specifically those files to the respective group as due to lab issues finance scripts were in directories belonging to engineers and vice versa. By executing the below commands I changed the owners of the scripts to the appropriate group and gave that group the appropriate permissions.

```
find  -iname '*engineering*' -exec chown :engineering {} +
find  -iname '*engineering*' -exec chmod g=wrx {} +


find  -iname '*finance*' -exec chown :finance {} +
```

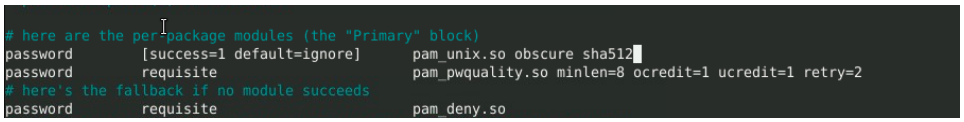
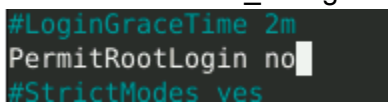
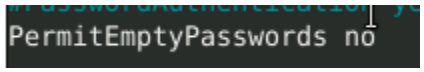
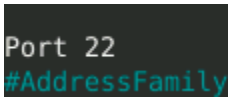
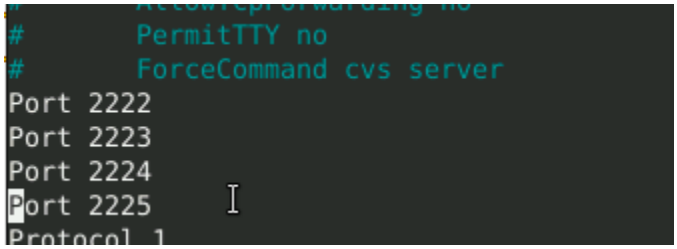
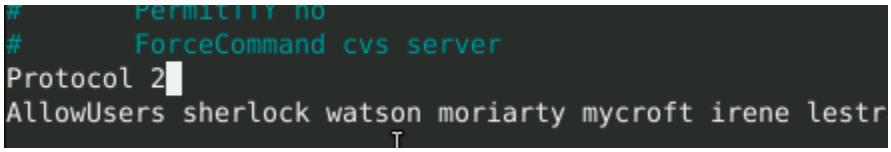
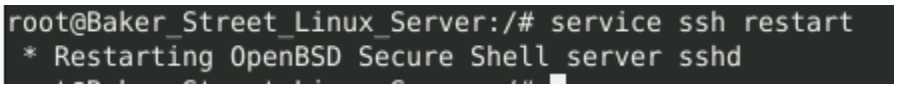
find -iname '*finance*' -exec chmod g=wrx {} +

```
root@Baker_Street_Linux_Server:/home# find  -iname '*engineering*' -exec chmod g=wrx {} +
root@Baker_Street_Linux_Server:/home# find  -iname '*finance*' -exec chown :finance {} +
root@Baker_Street_Linux_Server:/home# find  -iname '*finance*' -exec chmod g=wrx {} +

root@Baker_Street_Linux_Server:/home# ls -lR
.:
total 64
drwxr-x--- 1 adler      root 4096 Feb 27 03:22 adler
-rw-r--r-- 1 root       root    0 Feb 27 02:57 chgrp
-rw-r--r-- 1 root       root    0 Feb 27 02:44 find
drwxr-x--- 1 moriarty   root 4096 Dec 12 07:45 moriarty
drwxr-x--- 1 mrs_hudson root 4096 Dec 12 07:45 mrs_hudson
drwxr-x--- 1 mycroft    root 4096 Dec 12 07:45 mycroft
drwxr-x--- 1 sherlock   root 4096 Feb 27 03:26 sherlock
drwxr-x--- 1 sysadmin   root 4096 Dec 12 07:45 sysadmin
drwxr-x--- 1 toby       root 4096 Dec 12 07:45 toby
drwxr-x--- 1 watson     root 4096 Feb 27 03:19 watson

./adler:
total 8
-rw-rwx--- 1 root engineering  0 Dec 12 07:45 Engineering_script.sh_0.txt
-rw-rwx--- 1 root engineering  0 Dec 12 07:45 Engineering_script.sh_3.txt
-rwxrwx--- 1 root engineering 46 Dec 12 07:45 Engineering_script.sh_script1.sh
-rwxrwx--- 1 root engineering 46 Dec 12 07:45 Engineering_script.sh_script2.sh
-rw-r----- 1 root root         0 Dec 12 07:45 deduction.doc_2.txt
-rw-r----- 1 root root         0 Dec 12 07:45 game_is_afoot.txt_1.txt

./moriarty:
total 8
-rw-rwx--- 1 root finance  0 Dec 12 07:45 Finance_script.sh_0.txt
-rw-rwx--- 1 root finance  0 Dec 12 07:45 Finance_script.sh_2.txt
-rw-r----- 1 root root     0 Dec 12 07:45 elementary.txt_1.txt
-rw-r----- 1 root root     0 Dec 12 07:45 game_is_afoot.txt_3.txt
-rwxr-x--- 1 root root    49 Dec 12 07:45 game_is_afoot.txt_script1.sh
-rwxr-x--- 1 root root    49 Dec 12 07:45 game_is_afoot.txt_script2.sh
-rw-r----- 1 root root     0 Dec 12 07:45 my_file.txt
```

I did a cat * for each user's home directory and found no passwords or file content??? Seems like a productive group of employees at this company… Only including 1 screenshot as the rest were the same and redundant. This is due to a lab environment issue. In a larger org where using cat to view everyone's files is not feasible we could grep files for keywords like “password”, or we could directly search the plaintext password to see if it is stored anywhere.

```
root@Baker_Street_Linux_Server:/home/watson# cat *
#!/bin/bash
echo 'This is a script for watson'
#!/bin/bash
echo 'This is a script for watson'
```

☑	Optional: Updating password hashing configuration	**Optional Task** We are already using the yes script. After doing some research I was able to determine this is a strong hashing configuration so I reverted the changes made in the screenshot below. However, to demonstrate how you would update hashing I edited the line to use sha512 as the hashing algorithm. According to my research sha 512 is not as strong as yesscript.

		
☑	Auditing and securing SSH	**Part 6** To audit our ssh settings we will nano into sshd_config. Within this file We need to uncomment certain lines and change their controls. We do not want root login, or empty passwords. We will only accept connections on port 22. We want to be using protocol 2. To finalize this we will restart the ssh server. nano /etc/ssh/sshd_config   We will uncomment port 22 and delete the cheeky ports at the bottom of the file    service ssh restart 
☑	Reviewing and updating system packages	**Part 7** System packages need to be kept up to date to reduce known vulnerabilities. We will use update and upgrade to make sure we are running the newest version of all packages. Then we will list all of these packages to submit in our report. Finally, we will remove packages that impose a security risk and add some packages that will help harden the server. apt update

```
root@Baker_Street_Linux_Server:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1533 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [3824 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [53.3 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2955 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [35.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [81.4 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2645 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [45.2 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1235 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3676 kB]
Fetched 36.4 MB in 26s (1421 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

apt upgrade -y

```
root@Baker_Street_Linux_Server:/# apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  dmsetup libc-bin libc6 libcap2 libcap2-bin libcephfs2 libdevmapper1.02.1 libgnutls30 libgssapi-krb5-2 libk5crypto3 libkrb5-3
  libpam0g libpython3.10 libpython3.10-minimal libpython3.10-stdlib librados2 libseccomp2 libssl3 libtasn1-6 libxml2 mysql-clie
  openssh-server openssh-sftp-server openssl python3.10 python3.10-minimal
38 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 45.3 MB of archives.
After this operation, 52.2 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6 amd64 2.35-0ubuntu3.9 [3235 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc-bin amd64 2.35-0ubuntu3.9 [706 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpam0g amd64 1.4.0-11ubuntu2.5 [59.8 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpam-modules-bin amd64 1.4.0-11ubuntu2.5 [37.4 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpam-modules amd64 1.4.0-11ubuntu2.5 [280 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpython3.10 amd64 3.10.12-1~22.04.9 [1949 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libssl3 amd64 3.0.2-0ubuntu1.19 [1905 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3.10 amd64 3.10.12-1~22.04.9 [508 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpython3.10-stdlib amd64 3.10.12-1~22.04.9 [1850 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3.10-minimal amd64 3.10.12-1~22.04.9 [2263 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpython3.10-minimal amd64 3.10.12-1~22.04.9 [815 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.41-0ubuntu0.22.04.1 [2716 kB]
```

touch package_list.txt
apt list --installed > package_list.txt
cat package_list.txt

```
root@Baker_Street_Linux_Server:/# apt list --installed > package_list.txt

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

root@Baker_Street_Linux_Server:/# cat package_list.txt
Listing...
adduser/jammy,now 3.118ubuntu5 all [installed]
apt/jammy-updates,now 2.4.13 amd64 [installed]
attr/jammy,now 1:2.5.1-1build1 amd64 [installed,automatic]
base-files/jammy-updates,now 12ubuntu4.7 amd64 [installed]
base-passwd/jammy,now 3.5.52build1 amd64 [installed]
bash/jammy-updates,jammy-security,now 5.1-6ubuntu1.1 amd64 [installed]
bsdutils/jammy-updates,jammy-security,now 1:2.37.2-4ubuntu3.4 amd64 [installed]
ca-certificates/jammy-updates,jammy-security,now 20240203~22.04.1 all [installed,automatic]
```

grep 'telnet\|rsh-client' package_list.txt

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

apt remove rsh-client
apt remove telnet

```
root@Baker_Street_Linux_Server:/# apt remove 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 ... 16312 files and directories currently installed.)
Removing rsh-client (0.17-22) ...
update-alternatives: using /usr/bin/scp to provide /usr/bin/rcp (rcp) in
update-alternatives: warning: skip creation of /usr/share/man/man1/rcp.1.
update-alternatives: using /usr/bin/ssh to provide /usr/bin/rsh (rsh) in
update-alternatives: warning: skip creation of /usr/share/man/man1/rsh.1.
update-alternatives: using /usr/bin/slogin to provide /usr/bin/rlogin (rl
update-alternatives: warning: skip creation of /usr/share/man/man1/rlogin
root@Baker_Street_Linux_Server:/# apt remove telnet
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  telnet
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 158 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 16301 files and directories currently installed.)
Removing telnet (0.17-44build1) ...
```

apt autoremove -y

```
root@Baker_Street_Linux_Server:/# 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.
```

Telnet and rsh-client transmit data and passwords in plaintext which in today's age is a massive security issue.

apt -y install ufw lynis tripwire

```
root@Baker_Street_Linux_Server:/# apt -y install ufw lynis tripwire
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpio iptables libip6tc2 libnetfilter-conntrack3 libnfnetlink0 libnftnl11 menu postfix ssl-cert
Suggested packages:
  libarchive1 firewalld kmod nftables dnsutils apt-listbugs debsecan debsums samhain aide fail2ba
  sasl2-bin | dovecot-common resolvconf postfix-cdb mail-reader postfix-mta-sts-resolver postfix-
The following NEW packages will be installed:
  cpio iptables libip6tc2 libnetfilter-conntrack3 libnfnetlink0 libnftnl11 lynis menu postfix ssl
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 4543 kB of archives.
After this operation, 23.7 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 ssl-cert all 1.1.2 [17.4 kB]
```

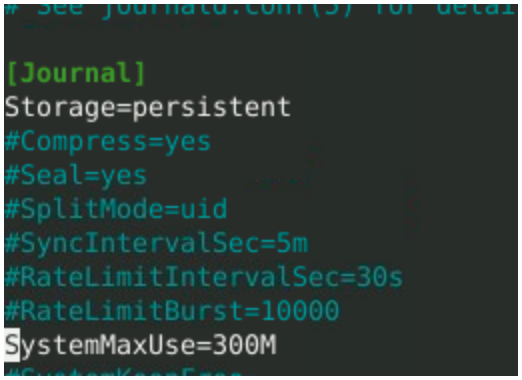
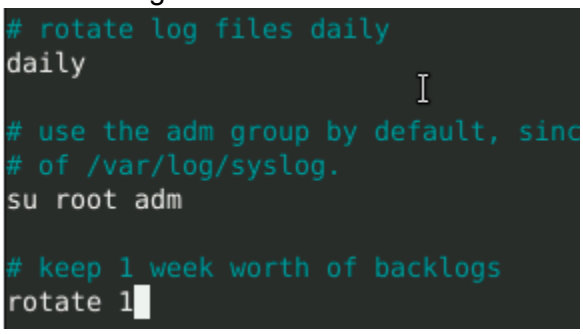
UFW is an uncomplicated firewall that simplifies configuring the firewall. This helps ensure the firewall is configured properly

Linus can run scans and provide feedback on your server's hardening. It can be used to address vulnerabilities.

		Tripwire is a host based intrusion detection system that is used to alert admin to any unauthorized changes.

☑	Disabling unnecessary services	**Part 8**
		Here we will compile our services into a text document for reporting purposes. Mysql, and samba need to be removed. If these services are not needed then they should be removed to reduce the attack surface and free up resources.
		
		service –status-all
		hwclock.sh status was not able to be determined. So it does not get written to the file.
		`root@Baker_Street_Linux_Server:/# service --status-all > service_list.txt`
		`[ ? ] hwclock.sh`
		`root@Baker_Street_Linux_Server:/# cat service_list.txt`
		`[ - ] cron`
		`[ - ] dbus`
		`[ - ] mysql`
		`[ + ] nmbd`
		`[ - ] openbsd-inetd`
		`[ - ] postfix`
		`[ - ] procps`
		`[ - ] samba-ad-dc`
		`[ + ] smbd`
		`[ - ] ssh`
		`[ - ] ufw`
		`root@Baker_Street_Linux_Server:/# service mysql stop`
		`* Stopping MySQL database server mysqld`
		`root@Baker_Street_Linux_Server:/# service mysql disable`
		`Usage: /etc/init.d/mysql start|stop|restart|reload|force-reload|status`
		`root@Baker_Street_Linux_Server:/# apt remove mysql`
		`Reading package lists... Done`
		`Building dependency tree... Done`
		`Reading state information... Done`
		`E: Unable to locate package mysql`
		`root@Baker_Street_Linux_Server:/# service samba-ad-dc stop`
		`* Stopping Samba AD DC daemon samba`
		`root@Baker_Street_Linux_Server:/# apt remove samba-ad-dc`
		`Reading package lists... Done`
		`Building dependency tree... Done`
		`Reading state information... Done`
		`E: Unable to locate package samba-ad-dc`

☑	Enabling and configuring logging	**Part 9** Here we want to configure our logs to be created daily, be persistent and not take up too much space on the system. To do this we will set limitations to size as well as rotate the logs on a weekly basis. nano /etc/systemd/journald.conf

		 nano /etc/logrotate.conf 
☑	Scripts created	**Part 10** Automating tasks can greatly impact productivity and allow analysts to focus on more important tasks. Here I will be automating all of the tasks we completed prior in two separate scripts. I have pasted the content of the scripts and included some screenshots of the output of bot the echo lines showing the script is working and the report files that were created to hold our system information. #!/bin/bash # Variable for the report output file, choose an output file name REPORT_FILE="report1.txt" # Output the hostname echo "Gathering hostname..." # Placeholder for command to get the hostname echo "Hostname: $(hostname -s)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE

		# Output the OS version echo "Gathering OS version..." # Placeholder for command to get the OS version echo "OS Version: $(uname -o)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Output memory information echo "Gathering memory information..." # Placeholder for command to get memory info echo "Memory Information: $(free -h)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Output uptime information echo "Gathering uptime information..." # Placeholder for command to get uptime info echo "Uptime Information: $(uptime)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Backup the OS echo "Backing up the OS..." # Placeholder for command to back 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 / echo "OS backup completed." >> $REPORT_FILE printf "\n" >> $REPORT_FILE

		# Output the sudoers file to the report echo "Gathering sudoers file..." # Placeholder for command to output sudoers file echo "Sudoers file:$(visudo -V)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Script to check for files with world permissions and update them echo "Checking for files with world permissions..." chmod -R o= * # Placeholder for command to find and update files with world permissions 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."

Placeholder for command to update permissions

find -iname '*engineering*' -exec chown :engineering {} +
find -iname '*engineering*' -exec chmod g=wrx {} +

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..."

Placeholder for command to update permissions

#Place command here to only allow members of ^`^|research ^`^} group to view, edit, and execute all research scripts. See above script for syntax.

#echo "Permissions updated for Research scripts" >> $REPORT_FILE

#printf "\n" >> $REPORT_FILE

```
root@Baker_Street_Linux_Server:/# ./hardening_script1.sh
Gathering hostname...
Gathering OS version...
Gathering memory information...
Gathering uptime information...
Backing up the OS...
```

```
chmod: changing permissions of 'sys/module/drm/sections/.parainstructions': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/.text.unlikely': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/.retpoline_sites': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/__ksymtab_strings': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/__ksymtab_gpl': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/__srcu_struct_ptrs': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/.altinstr_aux': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/__bpf_raw_tp_map': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/.rodata.str1.8': Read-only file system
chmod: changing permissions of 'sys/module/drm/sections/.ref.data': Read-only file system
Updating permissions for specific scripts...
Updating permissions for Engineering scripts.
Updating permissions for Finance scripts
Script execution completed. Check report1.txt for details.
```

report1.txt

```
  GNU nano 6.2

Hostname: Baker_Street_Linux_Server

OS Version: GNU/Linux

Memory Information:                total        used        free      shared  buff/cache   available
Mem:            15Gi       1.0Gi        10Gi       185Mi       4.2Gi        13Gi
Swap:             0B          0B          0B

Uptime Information:  00:45:10 up  1:02,  0 users,  load average: 0.19, 0.31, 0.33

OS backup completed.

Sudoers file:visudo version 1.9.9
visudo grammar version 48

World permissions have been removed from any files found.

Permissions updated for Engineering scripts.

Permissions updated for Finance scripts.
```

```
root@Baker_Street_Linux_Server:/# nano hardening_script2.sh
root@Baker_Street_Linux_Server:/# chmod +x hardening_script2.sh
root@Baker_Street_Linux_Server:/# ./hardening_script2.sh
Gathering details from sshd configuration file
Updating packages and services
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3694 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [3830 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1235 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2652 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2958 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1533 kB]
Fetched 16.3 MB in 12s (1355 kB/s)
Reading package lists... 49%
```

```
root@Baker_Street_Linux_Server:/# cat report2.txt
sshd configuration file:
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO
```

```
#!/bin/bash

# Variable for the report output file, choose a NEW output file name

REPORT_FILE="report2.txt"


# Output the sshd configuration file

echo "Gathering details from sshd configuration file"

# Placeholder for command to get the 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"


# Placeholder for command to update packages

apt update

# Placeholder for command to upgrade packages
```

		apt upgrade -y echo "Packages have been updated and upgraded" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Placeholder for command to list all installed packages echo "Installed Packages:$(apt list --installed)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE echo ^`^|Printing out logging configuration data ^`^} # Placeholder for command to display logging data echo "journald.conf file data: $(cat /etc/systemd/journald.conf)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Placeholder for command to display logrotate data

		echo "logrotate.conf file data:$(cat /etc/logrotate.conf)" >> $REPORT_FILE printf "\n" >> $REPORT_FILE echo "Script execution completed. Check $REPORT_FILE for details."
☑	Scripts scheduled with cron	**Part 13** To further automation so that we do not need to run scripts daily we will use crontabs to ensure the scripts are run routinely. Doing this ensures the task is always completed which is necessary if 3rd party auditing demands it. I have placed this cron schedule in the root users crontab. This means the root user will be the one running the scripts. Alternatively, we could have placed these crons in the cron directories so that they would be system wide. crontab -e 0 0 1 * * /hardening_script1 0 0 * * 1 /hardening_script2