

## **Project Overview: Linux Server Hardening for BSC**

**This project demonstrates a comprehensive approach to hardening a Linux server for The Baker Street Corporation (BSC), an organization that manages sensitive and confidential data. As a security professional, my task was to audit, secure, and automate key aspects of the server to minimize potential vulnerabilities and ensure long-term system integrity.**

### **Objectives and Scope**

**The project was completed over three phases:**

#### **Day 1: User, Group, and File Security**

- Conducted a system inventory and created a full backup.
- Audited users and groups to identify misconfigurations or unnecessary accounts.
- Enforced strong password policies and expiration rules.
- Reviewed and restricted sudo permissions.
- Validated and corrected file and directory permissions to prevent unauthorized access.

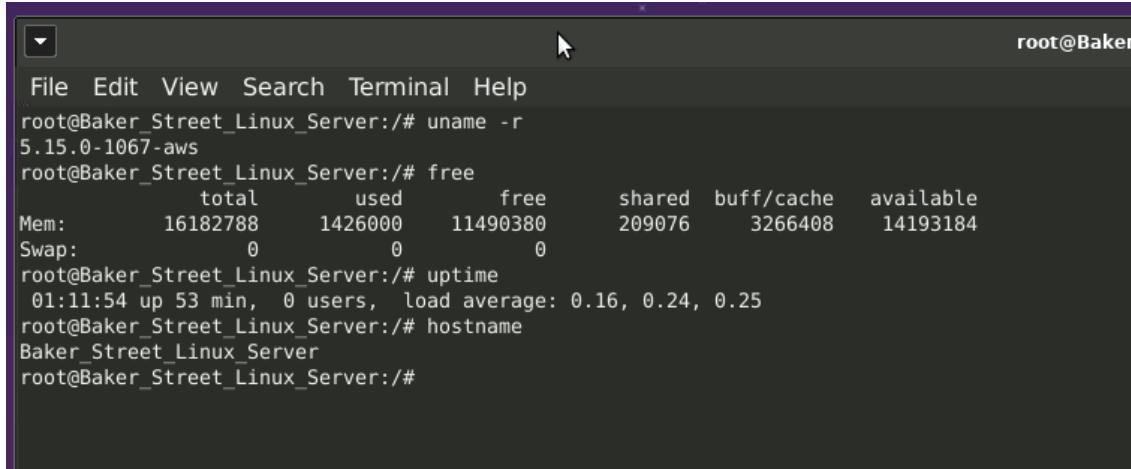
#### **Day 2: System Configuration Hardening**

- Audited and secured SSH settings to reduce remote access risk.
- Updated and patched system packages.
- Identified and disabled unnecessary services.
- Enabled and configured system logging to support future auditing and incident response.

#### **Day 3: Automation and Reporting**

- Developed scripts to automate the security tasks completed in Days 1 and 2.
- Scheduled the scripts to ensure ongoing enforcement of hardening policies.
- Compiled a final summary report detailing all hardening actions taken.

**Day 1 Part 1:** I ran the commands Hostname to see the host name. I ran uname – r to get the OS version. I ran the command free for the memory. Finally, I ran the uptime to show how long I have been in the terminal.



A screenshot of a terminal window titled "root@Baker". The window shows a series of Linux commands run by the root user on a server named "Baker\_Street\_Linux\_Server". The commands and their outputs are as follows:

- uname -r: 5.15.0-1067-aws
- free:

	total	used	free	shared	buff/cache	available
Mem:	16182788	1426000	11490380	209076	3266408	14193184
Swap:	0	0	0			
- uptime: 01:11:54 up 53 min, 0 users, load average: 0.16, 0.24, 0.25
- hostname: Baker\_Street\_Linux\_Server
- exit command: root@Baker\_Street\_Linux\_Server:~#

In Part 1, I ran the backup command given in the activity file to back up all files and directories. When it finished, I used the command ls—ahl to make sure the backup was completed. baker\_street\_backup.Tar.gz was listed in red, which let me know that the backup was completed.

- The command i used to backup the OS : -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/ufw/applications.d/
/etc/ufw/applications.d/samba
/etc/ufw/applications.d/openssh-server
/etc/ca-certificates.conf
/etc/perl/
/etc/perl/Net/
/etc/perl/Net/libnet.cfg
/etc/etheratypes
/etc/cron.hourly/
/etc/cron.hourly/.placeholder
/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
root@Baker_Street_Linux_Server:/# ls -ahl
total 211M
drwxr-xr-x 1 root root 4.0K Feb 25 00:58 .
drwxr-xr-x 1 root root 4.0K Feb 25 00:58 ..
-rw-r--r-- 1 root root 0 Feb 25 00:20 .dockerenv
-rw-r--r-- 1 root root 211M Feb 25 00:59 baker_street_backup.tar.gz
lnwxrwxrwx 1 root root 7 Sep 11 14:04 bin -> usr/bin
drwxr-xr-x 2 root root 4.0K Apr 18 2022 boot
drwxr-xr-x 5 root root 340 Feb 25 00:20 dev
drwxr-xr-x 1 root root 4.0K Feb 25 00:20 etc
drwxr-xr-x 1 root root 4.0K Dec 12 07:45 home
lnwxrwxrwx 1 root root 7 Sep 11 14:04 lib -> usr/lib
lnwxrwxrwx 1 root root 9 Sep 11 14:04 lib32 -> usr/lib32
lnwxrwxrwx 1 root root 9 Sep 11 14:04 lib64 -> usr/lib64
lnwxrwxrwx 1 root root 10 Sep 11 14:04 libx32 -> usr/libx32
drwxr-xr-x 2 root root 4.0K Sep 11 14:04 media
drwxr-xr-x 2 root root 4.0K Sep 11 14:04 mnt
drwxr-xr-x 2 root root 4.0K Sep 11 14:04 opt
dr-xr-xr-x 332 root root 0 Feb 25 00:20 proc
drwx----- 2 root root 4.0K Sep 11 14:07 root
drwxr-xr-x 1 root root 4.0K Feb 25 00:20 run
lnwxrwxrwx 1 root root 8 Sep 11 14:04 sbin -> usr/sbin
drwxr-xr-x 2 root root 4.0K Sep 11 14:04 srv
dr-xr-xr-x 13 root root 0 Feb 25 00:18 sys
drwxrwxrwt 1 root root 4.0K Feb 25 00:20 tmp
drwxr-xr-x 1 root root 4.0K Sep 11 14:04 usr
drwxr-xr-x 1 root root 4.0K Sep 11 14:07 var
root@Baker_Street_Linux_Server:/# █

```

**Part 2:** I removed all the files and directories of all the employees who have been terminated, which are Mary, Gregson, Irene, and Lestrade. Then, I locked the employees' accounts on temporary leave, which were mrs\_hudson and Moriarty. I then checked all the groups to make sure none of the employees was in the marketing department, which none of them were. I then checked the groups and went in and deleted the marketing group, which was closed earlier this year.

- When I deleted the user who had been terminated, I used the userdel -r (username) to make sure all the files and directories were deleted as well. To make sure the users were deleted, I ran the command groups with the names of terminated users Mary, Gregson, and Lestrade. (no need for me to use sudo if I'm using root, my apologies)

```
root@Baker_Street_Linux_Server:/ # File Edit View Search Terminal Help
root@Baker_Street_Linux_Server:/ # groups gregson
groups: 'gregson': no such user
root@Baker_Street_Linux_Server:/ # groups mary
mary : mary finance
root@Baker_Street_Linux_Server:/ # sudo userdel -r mary
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
userdel: mary mail spool (/var/mail/mary) not found
root@Baker_Street_Linux_Server:/ # groups mary
groups: 'mary': no such user
root@Baker_Street_Linux_Server:/ # sudo userdel -r irene
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
userdel: irene mail spool (/var/mail/irene) not found
root@Baker_Street_Linux_Server:/ # groups irene
groups: 'irene': no such user
root@Baker_Street_Linux_Server:/ # sudo userdel -r lestrade
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
userdel: lestrade mail spool (/var/mail/lestrade) not found
root@Baker_Street_Linux_Server:/ # groups lestrade
groups: 'lestrade': no such user
root@Baker_Street_Linux_Server:/ #
```

- I ran the command `passwd -S (username)` to check all the status of all employees listed. The terminated employees are shown below with “user doesn't exist” and we have two employees on the bottom that are locked out, which are Toby and Adler.
- The employees on temporary leave were Moriarty and mrs\_hudson. I ran the command `passwd -l (moriarty)(mrs_hudson)` to lock their accounts. As you can see in the screenshot below, they have a L after I ran the command `passwd -S (username)` to show their account has been locked.

```
root@Baker_Street_Linux_Server: /  
File Edit View Search Terminal Help  
[sudo] password for sysadmin:  
project1_v4  
sysadmin@ip-10-0-1-33:~$ sudo docker exec -it project1_v4 /bin/bash  
root@Baker_Street_Linux_Server:/# groups mary  
groups: 'mary': no such user  
root@Baker_Street_Linux_Server:/# groups gregson  
groups: 'gregson': no such user  
<root@Baker_Street_Linux_Server:/# groups lestrade  
groups: 'lestrade': no such user  
root@Baker_Street_Linux_Server:/# ls  
baker_street_backup.tar.gz etc lib64 opt sbin tmp  
bin home libx32 proc service_list.txt usr  
boot lib media root srv var  
dev lib32 mnt run sys  
root@Baker_Street_Linux_Server:/# ls -U  
bin lib64 mnt root tmp etc media baker_street_backup.tar.gz  
home libx32 lib usr sys proc lib32 service_list.txt  
var opt dev srv run boot sbin  
root@Baker_Street_Linux_Server:/# cd usr  
root@Baker_Street_Linux_Server:/usr# ls  
bin include lib32 libexec local share  
games lib lib64 libx32 sbin src  
root@Baker_Street_Linux_Server:/usr#  
root@Baker_Street_Linux_Server:/usr# groups lestrade  
groups: 'lestrade': no such user  
root@Baker_Street_Linux_Server:/usr# groups irene  
groups: 'irene': no such user  
root@Baker_Street_Linux_Server:/usr# groups mary  
groups: 'mary': no such user  
root@Baker_Street_Linux_Server:/usr# cd ..  
root@Baker_Street_Linux_Server:/# passwd -S sherlock  
sherlock P 03/05/2025 0 99999 7 -1  
root@Baker_Street_Linux_Server:/# passwd -S watsib  
passwd: user 'watsib' does not exist  
root@Baker_Street_Linux_Server:/# passwd -S watson  
watson P 03/05/2025 0 99999 7 -1  
root@Baker_Street_Linux_Server:/# passwd -S mycroft  
mycroft P 03/05/2025 0 99999 7 -1  
root@Baker_Street_Linux_Server:/# passwd -S moriarty  
moriarty P 03/05/2025 0 99999 7 -1  
root@Baker_Street_Linux_Server:/# passwd -S lestrade  
passwd: user 'lestrade' does not exist  
root@Baker_Street_Linux_Server:/# passwd -S irene  
passwd: user 'irene' does not exist  
root@Baker_Street_Linux_Server:/# passwd -S mrs_hudson  
mrs_hudson L 12/12/2024 0 99999 7 -1  
root@Baker_Street_Linux_Server:/# passwd -S mary  
passwd: user 'mary' does not exist  
root@Baker_Street_Linux_Server:/# passwd -S gregson  
passwd: user 'gregson' does not exist  
root@Baker_Street_Linux_Server:/# passwd -S toby  
toby L 12/12/2024 0 99999 7 -1  
root@Baker_Street_Linux_Server:/# passwd -S adler  
adler L 12/12/2024 0 99999 7 -1  
root@Baker_Street_Linux_Server:/#
```

- I nano into /etc/shadow to remove the “!” by their hash to unlock the user account. With me deleting the “!” in their hash, they don't have a password. I ran the command passwd -S (toby and adler) and they have NP which means no password.

root@Baker\_Street\_Linux\_Server: /etc

```
GNU nano 6.2                                         shadow
root:*:19977:0:99999:7:::
daemon:*:19977:0:99999:7:::
bin:*:19977:0:99999:7:::
sys:*:19977:0:99999:7:::
sync:*:19977:0:99999:7:::
games:*:19977:0:99999:7:::
man:*:19977:0:99999:7:::
lp:*:19977:0:99999:7:::
mail:*:19977:0:99999:7:::
news:*:19977:0:99999:7:::
uucp:*:19977:0:99999:7:::
proxy:*:19977:0:99999:7:::
www-data:*:19977:0:99999:7:::
backup:*:19977:0:99999:7:::
list:*:19977:0:99999:7:::
irc:*:19977:0:99999:7:::
gnats:*:19977:0:99999:7:::
nobody:*:19977:0:99999:7:::
_apt:*:19977:0:99999:7:::
systemd-network:*:20069:0:99999:7:::
systemd-resolve:*:20069:0:99999:7:::
mysql:!:20069:0:99999:7:::
messagebus:*:20069:0:99999:7:::
systemd-timesync:*:20069:0:99999:7:::
syslog:*:20069:0:99999:7:::
sshd:*:20069:0:99999:7:::
sherlock:$y$j9T$MqZdAYKngl0Mlf7fVLYPh/$VrCwtlE0M2PX65snRxxYhp0U40FCVuhh3tLzhn5E4UD:20152:0:99999:7:::
watson:$y$j9T$/A7e17.kETxLkeXvPxORM/$DKNJltaqRYQNSIf6DLxQmWb1t97n8u0PuDQpvqFvLUC:20152:0:99999:7:::
moriarty:$y$j9T$dKbj8u0LR.KYAbc3063ME1$gKwXtTc8k/ej15E8iYpEeo9MBu4IKmHFOHJ/u6b.6M3:20152:0:99999:7:::
mycroft:$y$j9Ts7qpLMqC2Yjm2Ly634lAsB0$vU0ztDwqJaDbcNHVyRyRR56fn7014AysUgMXGVL7q6:20152:0:99999:7:::
mrs_hudson:!:20069:0:99999:7:::
sysadmin:!:20069:0:99999:7:::
toby:!:20069:0:99999:7:::
adler:!:20069:0:99999:7:::
postfix:*:20146:0:99999:7:::
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

File Edit View Search Terminal Help

shadow

```
root:*:19977:0:99999:7:::  
daemon:*:19977:0:99999:7:::  
bin:*:19977:0:99999:7:::  
sys:*:19977:0:99999:7:::  
sync:*:19977:0:99999:7:::  
games:*:19977:0:99999:7:::  
man:*:19977:0:99999:7:::  
lp:*:19977:0:99999:7:::  
mail:*:19977:0:99999:7:::  
news:*:19977:0:99999:7:::  
uucp:*:19977:0:99999:7:::  
proxy:*:19977:0:99999:7:::  
www-data:*:19977:0:99999:7:::  
backup:*:19977:0:99999:7:::  
list:*:19977:0:99999:7:::  
irc:*:19977:0:99999:7:::  
gnats:*:19977:0:99999:7:::  
nobody:*:19977:0:99999:7:::  
_apt:*:19977:0:99999:7:::  
systemd-network:*:20069:0:99999:7:::  
systemd-resolve:*:20069:0:99999:7:::  
mysql!:20069:0:99999:7:::  
messagebus:*:20069:0:99999:7:::  
systemd-timesync:*:20069:0:99999:7:::  
syslog*:20069:0:99999:7:::  
sshd*:20069:0:99999:7:::  
sherlock:$y$j9T$MqZdAYKngl0MLf7fVLYPh/$VrCwtLE0M2PX65snRxxYhp0U40FCVuuh3tLzhn5E4UD:20152:0:99999:7:::  
watson:$y$j9T$/A7e17.KETxLkeXvPxORM/$DKNJltaqRYQNSIf6DLxQmWbt97n8u0PuDQpvqFvLUC:20152:0:99999:7:::  
moriarty:$y$j9T$dkbj8u0LR.KYAbc3063ME1$gKwxtTc8k/ej15E81YpEeo9MBu4IKmHFOHJ/u6b.6M3:20152:0:99999:7:::  
mycroft:$y$j9T$7qpLMqC2Yjm2Wy634lAsB0$vU0ztdDWqJaDbcNHVyRyRR56fn7014AysUgMXGVL7q6:20152:0:99999:7:::  
mrs_hudson!:20069:0:99999:7:::  
sysadmin!:20069:0:99999:7:::  
toby::20069:0:99999:7:::  
adler::20069:0:99999:7:::  
postfix:*:20146:0:99999:7:::
```

[ Read 35 lines ]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

```
root@Baker_Street_Linux_Server:~# passwd -S toby
toby NP 12/12/2024 0 99999 7 -1
root@Baker_Street_Linux_Server:~# passwd -S adler
adler NP 12/12/2024 0 99999 7 -1
root@Baker_Street_Linux_Server:~#
```

The screenshot shows a terminal window titled "root@Baker\_Street\_Linux\_Server: /etc". The window contains the contents of the /etc/group file, which lists various system groups and their members. A cursor is visible in the middle of the list. The bottom of the window shows the nano editor's command bar with various keyboard shortcuts.

```
File Edit View Search Terminal Help
GNU nano 6.2
fax:x:21:
voice:x:22:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:
audio:x:29:
dip:x:30:
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
mysql:x:104:
crontab:x:105:
messagebus:x:106:
systemd-timesync:x:107:
syslog:x:108:
rdma:x:109:
_ssh:x:110:
sambashare:x:111:
sherlock:x:1000:
watson:x:1001:
moriarty:x:1002:
mycroft:x:1003:
mrs_hudson:x:1006:
sysadmin:x:1008:
toby:x:1010:
adler:x:1011:
engineering:x:1012:sherlock,watson,moriarty
finance:x:1013:mrs_hudson
ssl-cert:x:112:
postfix:x:113:
postdrop:x:114:
research:x:1014:
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo  
^X Exit ^R Read File ^H Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

- I nano into /etc/group to see if there is a marketing department and there is not one.  
I created the research group.

**Part 3:** I ran nano /etc/pam.d/common-password to edit this file. While in the file, I added a comment saying here are the new password requirements.

- So, I added the new available settings of the new password requirements, which were minlen=8, ocredit=-1, retry=2, uccredit= -1. The screenshot will provide evidence of how it was entered into the file.

The screenshot shows a terminal window titled "root@Baker\_Street\_Linux\_Server: /etc/pam.d". The window contains the contents of the file "/etc/pam.d/common-password". The file is a configuration script for password authentication, using the PAM (Pluggable Authentication Modules) framework. It includes sections for "Primary" and "Additional" modules, and specifies requirements like minimum length, special characters, and uppercase letters. The terminal window has a dark theme and includes standard window controls (minimize, maximize, close) at the top right.

```
root@Baker_Street_Linux_Server: /etc/pam.d
File Edit View Search Terminal Help
GNU nano 6.2                                         common-password *
#for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

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

# here are the new password requirements

# minimum length of 8 character
password      requisite                  pam_pwquality.so minlen = 8
# at least one special character
password      requisite                  pam_pwquality.so ocredit = -1
# two retries allowed
password      requisite                  pam_pwquality.so retry = 2
# at least one uppercase letter
password      requisite                  pam_pwquality.so ucredit = -1

# end of pam-auth-update config
```

## Part 4:

all evidence is provided below in the screenshot.

- I nano into /etc/sudoers to make changes to this file. The first change was giving Sherlock full sudo permissions.
- I then gave Watson and Mycroft sudo privileges to run the following script /var/log/logcleanup.sh
- I gave all the employees in the research group sudo privileges to run /tmp/scripts/research\_script.sh.

```
# "sudo scp" or "sudo rsync" should be able to use your SSH agent.  
Defaults: %sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"  
  
# Ditto for GPG agent  
Defaults: %sudo env_keep += "GPG_AGENT_INFO"  
  
# Host alias specification  
  
# User alias specification  
  
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
sherlock ALL=(ALL:ALL) ALL  
# Members of the admin group may gain root privileges  
%admin  ALL=(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  
  
# Sudo privileges for /var/log/logcleanup.sh  
watson ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh  
mycroft ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh  
  
# allow members of the research group have sudo privileges to run the following script  
%research ALL=(ALL) NOPASSWD: /tmp/scripts/research_script.sh
```

## Part 5:

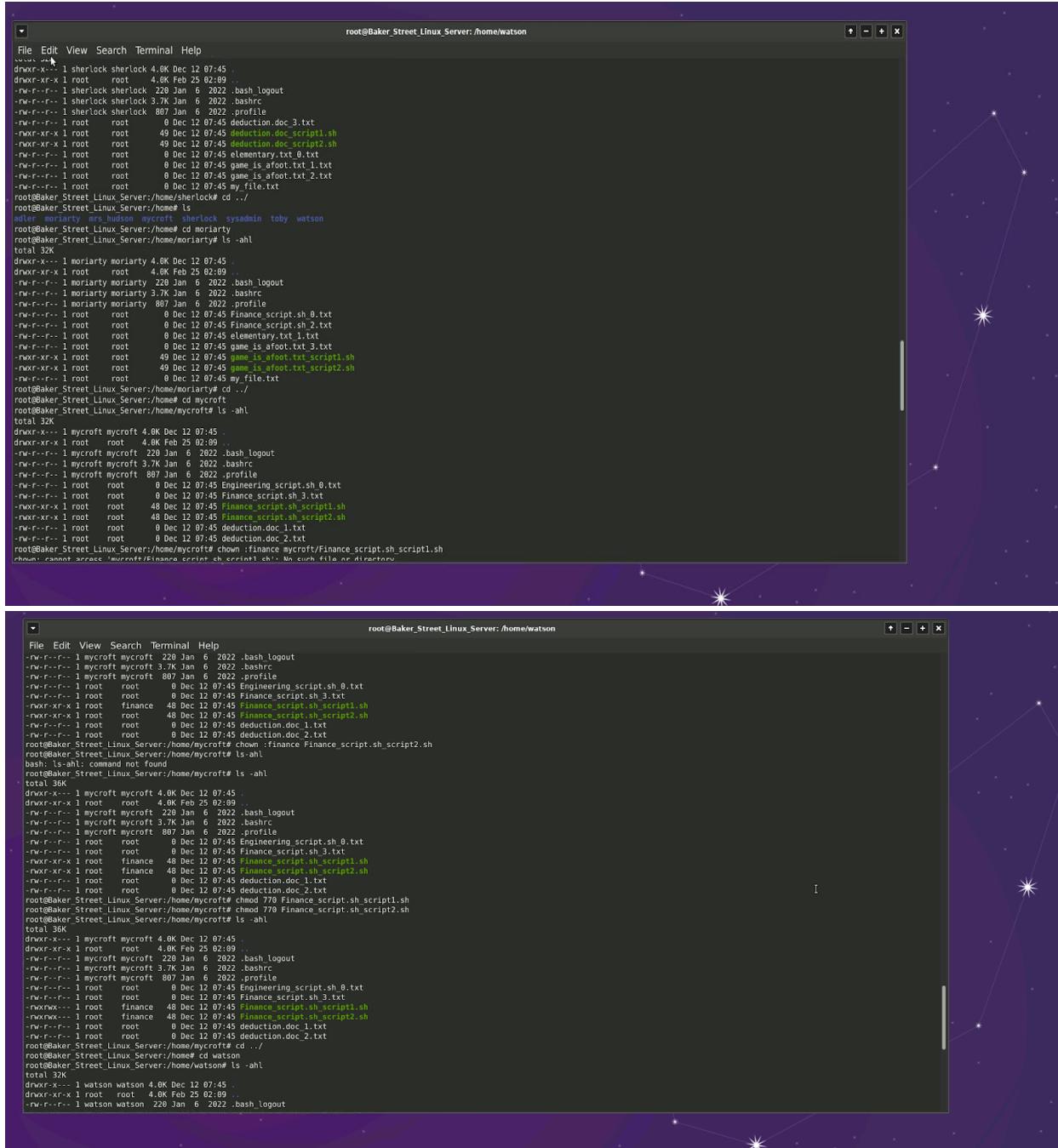
- I went to the home directory, ran the ls command to see all the users/employees. I would change the directory into one of the user/employee and use the ls -ahl command to get a long listing of everything in their directory.
  - I found certain scripts in certain user groups that they were not a part of. I had to change the ownership (chown) and change the permissions to make sure the right people had the right access to those scripts.

- I went to watson home directory and ran the ls -ahl command to see the full listing. I saw he had the finance\_script.sh\_script1&2.sh listed.

```
root@Baker_Street_Linux_Server:/home/watson# chown :finance Finance_script.sh_script1.sh
root@Baker_Street_Linux_Server:/home/watson# chmod 770 Finance_script.sh_script1.sh
root@Baker_Street_Linux_Server:/home/watson# ls -ahl
total 36K
drwxr-x--- 1 watson watson 4.0K Dec 12 07:45 .
drwxr-xr-x 1 root   root   4.0K Mar  7 01:23 ..
-rw-r--r-- 1 watson watson 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 watson watson 3.7K Jan  6 2022 .bashrc
-rw-r--r-- 1 watson watson 807 Jan  6 2022 .profile
-rw-r--r-- 1 root   root   0 Dec 12 07:45 Finance_script.sh_3.txt
-rwxrwx--- 1 root   finance 47 Dec 12 07:45 Finance_script.sh_script1.sh
-rwxrwx--- 1 root   finance 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
root@Baker_Street_Linux_Server:/home/watson# cd ..
```

```
root@Baker_Street_Linux_Server:/home/watson# chown :finance Finance_script.sh_script2.sh
root@Baker_Street_Linux_Server:/home/watson# chmod 770 Finance_script.sh_script2.sh
root@Baker_Street_Linux_Server:/home/watson# ls -ahl
total 36K
drwxr-x--- 1 watson watson 4.0K Dec 12 07:45 .
drwxr-xr-x 1 root   root   4.0K Mar  7 01:23 ..
-rw-r--r-- 1 watson watson 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 watson watson 3.7K Jan  6 2022 .bashrc
-rw-r--r-- 1 watson watson 807 Jan  6 2022 .profile
-rw-r--r-- 1 root   root   0 Dec 12 07:45 Finance_script.sh_3.txt
-rwxrwx--- 1 root   finance 47 Dec 12 07:45 Finance_script.sh_script1.sh
-rwxrwx--- 1 root   finance 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
root@Baker_Street_Linux_Server:/home/watson#
```

- I changed ownership (chown) to the finance group (I used the command chown :finance Finance\_script.sh\_script1.sh). Used the same command for the second script.
- I then had to change the permissions (chmod) to 770 to read, write, execute so all members of the finance group can read, write, execute. Command used (chmod 770 Finance\_script.sh\_script1.sh. I did the same thing for the second script as well.



```

root@Baker_Street_Linux_Server:/home/watson
File Edit View Search Terminal Help
...> 1 root      4.0K Dec 12 07:45
drwxr-x--- 1 root      4.0K Feb 25 02:09 .
drwxr-x--- 1 root      4.0K Feb 25 02:09 ..
-rw-r--r-- 1 root      220 Jan 6 2022 .bash_logout
-rw-r--r-- 1 root      3.7K Jan 6 2022 .bashrc
-rw-r--r-- 1 root      807 Jan 6 2022 .profile
-rw-r--r-- 1 root      0 Dec 12 07:45 deduction.doc
-rw-r--r-- 1 root      49 Dec 12 07:45 deduction.doc_script1.sh
-rw-r--r-- 1 root      49 Dec 12 07:45 deduction.doc_script2.sh
-rw-r--r-- 1 root      0 Dec 12 07:45 elementary.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 game_is_afoot.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 game_is_afoot.txt.2.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 my_file.txt
root@Baker_Street_Linux_Server:/home/sherlock# cd ..
root@Baker_Street_Linux_Server:/home/ls
moriarty mrs_hudson mycroft sherlock sysadmin toby watson
root@Baker_Street_Linux_Server:/home# cd moriarty
root@Baker_Street_Linux_Server:/home/moriarty# ls -ahl
total 2K
drwxr-x--- 1 moriarty moriarty 4.0K Dec 12 07:45 .
drwxr-x--- 1 moriarty moriarty 4.0K Feb 25 02:09 ..
-rw-r--r-- 1 moriarty moriarty 220 Jan 6 2022 .bash_logout
-rw-r--r-- 1 moriarty moriarty 3.7K Jan 6 2022 .bashrc
-rw-r--r-- 1 moriarty moriarty 807 Jan 6 2022 .profile
-rw-r--r-- 1 root      0 Dec 12 07:45 Finance.script.sh.0.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 Finance.script.sh.2.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 elementary.txt.1.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 game_is_afoot.txt.3.txt
-rwxr-x--- 1 root      49 Dec 12 07:45 game_is_afoot.txt.script1.sh
-rwxr-x--- 1 root      49 Dec 12 07:45 game_is_afoot.txt.script2.sh
-rw-r--r-- 1 root      0 Dec 12 07:45 my_file.txt
root@Baker_Street_Linux_Server:/home/moriarty# cd ..
root@Baker_Street_Linux_Server:/home/mycroft
root@Baker_Street_Linux_Server:/home/mycroft# ls -ahl
total 32K
drwxr-x--- 1 mycroft mycroft 4.0K Dec 12 07:45 .
drwxr-x--- 1 root      4.0K Feb 25 02:09 ..
-rw-r--r-- 1 mycroft mycroft 220 Jan 6 2022 .bash_logout
-rw-r--r-- 1 mycroft mycroft 3.7K Jan 6 2022 .bashrc
-rw-r--r-- 1 mycroft mycroft 807 Jan 6 2022 .profile
-rw-r--r-- 1 root      0 Dec 12 07:45 Engineering.script.sh.0.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 Finance.script.sh.script1.sh
-rwxr-x--- 1 root      48 Dec 12 07:45 finance.script.sh.script1.sh
-rwxr-x--- 1 root      0 Dec 12 07:45 deduction.doc.1.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 deduction.doc.2.txt
root@Baker_Street_Linux_Server:/home/mycroft# chown :finance mycroft/Finance_script.sh.script1.sh
chown: cannot access '/mycroft/Finance_script.sh.script1.sh': No such file or directory
root@Baker_Street_Linux_Server:/home/watson
File Edit View Search Terminal Help
...> 1 mycroft mycroft 220 Jan 6 2022 .bash_logout
-rw-r--r-- 1 mycroft mycroft 3.7K Jan 6 2022 .bashrc
-rw-r--r-- 1 mycroft mycroft 807 Jan 6 2022 .profile
-rw-r--r-- 1 root      0 Dec 12 07:45 Engineering.script.sh.0.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 Finance.script.sh.3.txt
-rwxr-x--- 1 root      48 Dec 12 07:45 Finance.script.sh.script1.sh
-rwxr-x--- 1 root      48 Dec 12 07:45 Finance.script.sh.script2.sh
-rw-r--r-- 1 root      0 Dec 12 07:45 deduction.doc.1.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 deduction.doc.2.txt
root@Baker_Street_Linux_Server:/home/mycroft# chown :finance Finance.script.sh.script2.sh
root@Baker_Street_Linux_Server:/home/watson
File Edit View Search Terminal Help
...> 1 watson watson 4.0K Dec 12 07:45
drwxr-x--- 1 root      4.0K Feb 25 02:09 .
drwxr-x--- 1 watson watson 220 Jan 6 2022 .bash_logout
-rw-r--r-- 1 mycroft mycroft 3.7K Jan 6 2022 .bashrc
-rw-r--r-- 1 mycroft mycroft 807 Jan 6 2022 .profile
-rw-r--r-- 1 root      0 Dec 12 07:45 Engineering.script.sh.0.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 Finance.script.sh.3.txt
-rwxr-x--- 1 root      48 Dec 12 07:45 Finance.script.sh.script1.sh
-rwxr-x--- 1 root      48 Dec 12 07:45 Finance.script.sh.script2.sh
-rw-r--r-- 1 root      0 Dec 12 07:45 deduction.doc.1.txt
-rw-r--r-- 1 root      0 Dec 12 07:45 deduction.doc.2.txt
root@Baker_Street_Linux_Server:/home/watson# cd ..
root@Baker_Street_Linux_Server:/home/watson
root@Baker_Street_Linux_Server:/home/watson# ls -ahl
total 32K
drwxr-x--- 1 watson watson 4.0K Dec 12 07:45
drwxr-x--- 1 root      4.0K Feb 25 02:09 ..
-rw-r--r-- 1 watson watson 220 Jan 6 2022 .bash_logout

```

- I, cd into the adler directory ran the command ls -ahl to see the permissions and files and directories. I saw he had a script in there called Engineering\_script.sh\_script1.sh and script2.sh. I then used the command chown and chmod to change the permissions and change ownership.

- The command I used was chown :engineering Engineering\_script.sh\_script.sh1. I did the same thing for script2.sh
- Next command was chmod 770 Engineering\_script.sh\_script1.sh i did the same thing for script2.sh
- These commands allowed people in the engineering group to read, write, and execute the scripts if they are assigned to these groups.

```
root@Baker_Street_Linux_Server:/home/watson
File Edit View Search Terminal Help
/home/sysadmin/.profile
root@Baker_Street_Linux_Server:# cd home
root@Baker_Street_Linux_Server:/home# ls
adler moriarty mrs_hudson mycroft sherlock sysadmin toby watson
root@Baker_Street_Linux_Server:/home# ls -ahl
total 48K
drwxr-xr-x 1 root      root      4.0K Feb 25 02:09 .
drwxr-xr-x 1 root      root      4.0K Feb 25 00:58 ..
drwxr-xr-x 1 adler     adler     4.0K Dec 12 07:45 adler
drwxr-xr-x 1 moriarty  moriarty  4.0K Dec 12 07:45 moriarty
drwxr-xr-x 1 mrs_hudson mrs_hudson 4.0K Dec 12 07:45 mrs_hudson
drwxr-xr-x 1 mycroft  mycroft  4.0K Dec 12 07:45 mycroft
drwxr-xr-x 1 sherlock  sherlock  4.0K Dec 12 07:45 sherlock
drwxr-xr-x 2 sysadmin  sysadmin  4.0K Dec 12 07:45 sysadmin
drwxr-xr-x 1 toby     toby     4.0K Dec 12 07:45 toby
drwxr-xr-x 1 watson   watson   4.0K Dec 12 07:45 watson
root@Baker_Street_Linux_Server:/home# ls -ahl adler
total 36K
drwxr-xr-x 1 adler adler  4.0K Dec 12 07:45 .
drwxr-xr-x 1 root    root   4.0K Feb 25 02:09 ..
-rw-r--r-- 1 adler adler 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 adler adler 3.7K Jan  6 2022 .bashrc
-rw-r--r-- 1 adler adler 807 Jan  6 2022 .profile
-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
-rw-r--r-- 1 root    root   0 Dec 12 07:45 Engineering_script.sh_script1.sh
-rw-r--r-- 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 deduction.doc 2.txt
-rw-r--r-- 1 root    root   0 Dec 12 07:45 game_is_afoot.txt 1.txt
root@Baker_Street_Linux_Server:/home# chown :engineering adler/Engineering_script.sh_script1.sh
root@Baker_Street_Linux_Server:/home# ls -ahl adler
total 36K
drwxr-xr-x 1 adler adler  4.0K Dec 12 07:45 .
drwxr-xr-x 1 root    root   4.0K Feb 25 02:09 ..
-rw-r--r-- 1 adler adler 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 adler adler 3.7K Jan  6 2022 .bashrc
-rw-r--r-- 1 adler adler 807 Jan  6 2022 .profile
-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
-rw-r--r-- 1 root    root   46 Dec 12 07:45 Engineering_script.sh_script1.sh
-rw-r--r-- 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 deduction.doc 2.txt
-rw-r--r-- 1 root    root   0 Dec 12 07:45 game_is_afoot.txt 1.txt
root@Baker_Street_Linux_Server:/home# chmod 770 adler/Engineering_script.sh_script1.sh
root@Baker_Street_Linux_Server:/home# ls -ahl adler
total 36K
drwxr-xr-x 1 adler adler  4.0K Dec 12 07:45 .

root@Baker_Street_Linux_Server:/home/watson
File Edit View Search Terminal Help
/home/sysadmin/.profile
root@Baker_Street_Linux_Server:/home# cd /root/adler/
root@Baker_Street_Linux_Server:/root/adler# chmod 770 Engineering_script.sh_script2.sh
root@Baker_Street_Linux_Server:/home/adler# ls
Engineering_script.sh 0.txt Engineering_script.sh 3.txt Engineering_script.sh_script1.sh Engineering_script.sh_script2.sh deduction.doc_2.txt game_is_afoot.txt 1.txt
root@Baker_Street_Linux_Server:/home/adler# ls -ahl
total 36K
drwxr-xr-x 1 adler adler  4.0K Dec 12 07:45 .
drwxr-xr-x 1 root    root   4.0K Feb 25 02:09 ..
-rw-r--r-- 1 adler adler 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 adler adler 3.7K Jan  6 2022 .bashrc
-rw-r--r-- 1 adler adler 807 Jan  6 2022 .profile
-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
-rw-r--r-- 1 root    root   46 Dec 12 07:45 Engineering_script.sh_script1.sh
-rw-r--r-- 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 deduction.doc 2.txt
-rw-r--r-- 1 root    root   0 Dec 12 07:45 game_is_afoot.txt 1.txt
root@Baker_Street_Linux_Server:/home/adler# cd ..
root@Baker_Street_Linux_Server:/home# ls
adler moriarty mrs_hudson mycroft sherlock sysadmin toby watson
root@Baker_Street_Linux_Server:/home# chmod 770 mrs_hudson
root@Baker_Street_Linux_Server:/home/mrs_hudson# ls
Engineering_script.sh 1.txt deduction.doc 0.txt deduction.doc_2.txt elementary.txt_3.txt elementary.txt_script1.sh elementary.txt_script2.sh
root@Baker_Street_Linux_Server:/home/mrs_hudson# ls -ahl
total 32K
drwxr-xr-x 1 mrs_hudson mrs_hudson 4.0K Dec 12 07:45 .
drwxr-xr-x 1 root    root   4.0K Feb 25 02:09 ..
-rw-r--r-- 1 mrs_hudson mrs_hudson 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 mrs_hudson mrs_hudson 3.7K Jan  6 2022 .bashrc
-rw-r--r-- 1 mrs_hudson mrs_hudson 807 Jan  6 2022 .profile
-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
-rw-r--r-- 1 root    root   51 Dec 12 07:45 elementary.txt_script1.sh
-rw-r--r-- 1 root    root   51 Dec 12 07:45 elementary.txt_script2.sh
root@Baker_Street_Linux_Server:/home/mrs_hudson# cd ..
root@Baker_Street_Linux_Server:/home# ls
adler moriarty mrs_hudson mycroft sherlock sysadmin toby watson
root@Baker_Street_Linux_Server:/home# cd sherlock
root@Baker_Street_Linux_Server:/home/sherlock# ls -ahl
total 32K
drwxr-xr-x 1 sherlock sherlock 4.0K Dec 12 07:45 .
drwxr-xr-x 1 root    root   4.0K Feb 25 02:09 ..

```

## **Day 2: System Configuration Hardening**

- Audited and secured SSH settings to reduce remote access risk.
- Updated and patched system packages.
- Identified and disabled unnecessary services.
- Enabled and configured system logging to support future auditing and incident response.

### Day 2 Part 1

#### Auditing and Securing SSH:

I ran the command `nano /etc/ssh/sshd_config`. While in this file I made the changes needed.

- I disable empty password,
- disable root login,
- enable ssh protocol 2.

The screenshots will show the work that I did. Once I was done with that, I made sure I saved everything. Then I ran the command `service ssh status` to restart ssh (last

screenshot).

root@Baker\_Street\_Linux\_Server: /etc/ssh

```
GNU nano 6.2                                     sshd config
allowAgentForwarding yes
allowTcpForwarding yes
#AllowUser nobody
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes
#PermitTTY yes
#PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PIDFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem    sftp    /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anonyvs
#   X11Forwarding no
#   AllowTcpForwarding no
#   PermitTTY no
#   ForceCommand cvs server

Port 2222
Port 2223
Port 2224
Port 2225
Protocol 2
AllowUsers sherlock watson moriarty mycroft irene lestrade
```

File Edit View Search Terminal Help

Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket Previous Back Read File Replace Paste Justify Go To Line Redo Copy Where Was Next Forward Left Prev Word Right Next Word

root@Baker\_Street\_Linux\_Server: /etc/ssh

```
GNU nano 6.2                                     sshd config
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreRhosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
#KbdInteractiveAuthentication no

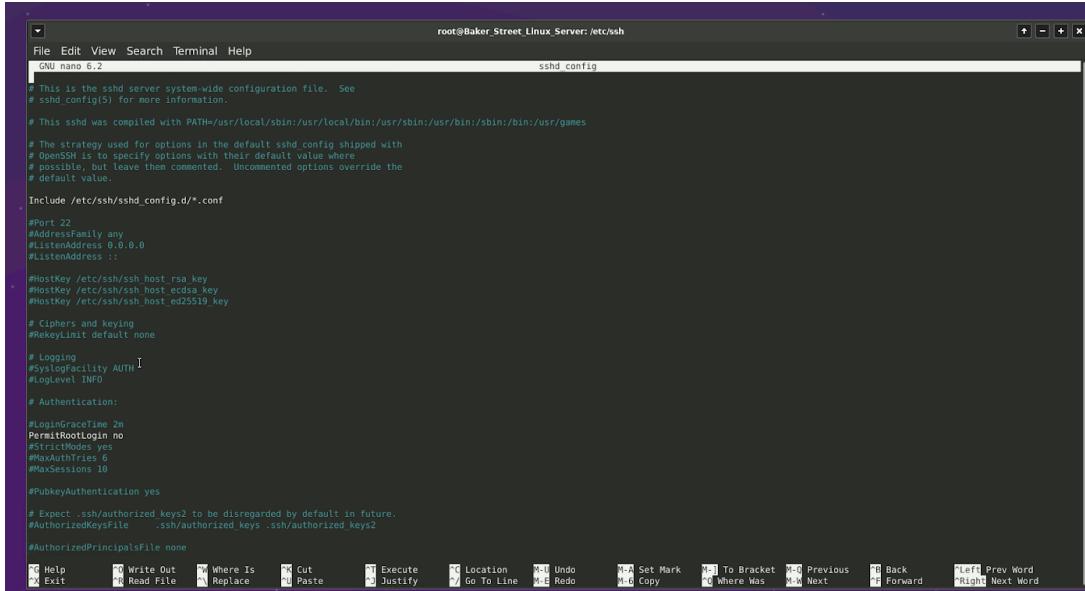
# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetTSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you want PAM to bypass all password-related checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
#UsePAM yes
```

File Edit View Search Terminal Help

Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket Previous Back Read File Replace Paste Justify Go To Line Redo Copy Where Was Next Forward Left Prev Word Right Next Word



```

root@Baker_Street_Linux_Server:/etc/ssh
# 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:/usr/games

# The strategy used for options with their default value where
# 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
#KeyExchange none

# Logging
#LogLevel AUTH
#LogLevel INFO

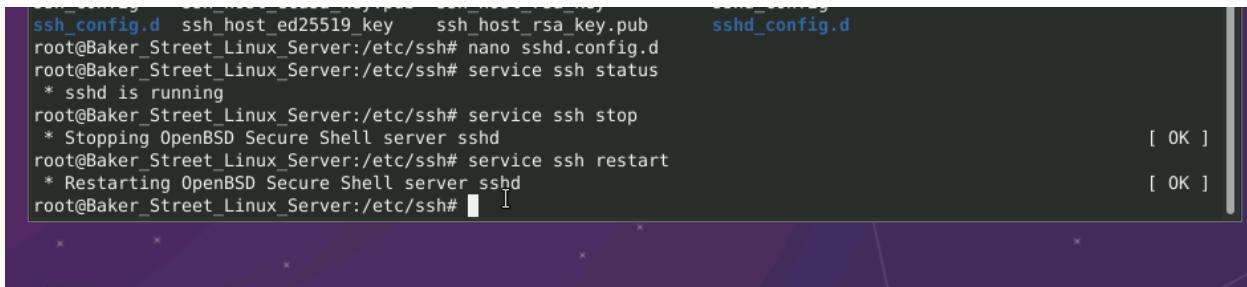
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
#AuthorizedPrincipalsFile none

#Help
#Exit
#WriteOut
#ReadFile
#Replace
#Cut
#Paste
#Execute
#Justify
#Location
#Undo
#SetMark
#ToBracket
#WhereWas
#Next
#Back
#Forward
#Left
#PrevWord
#Right
#NextWord

```

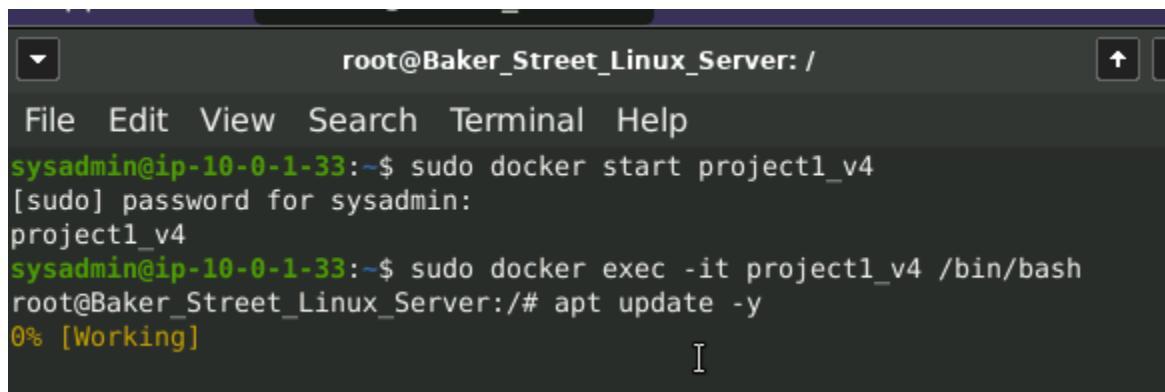


```

sshd_config.d ssh_host_ed25519_key ssh_host_rsa_key.pub      sshd_config.d
root@Baker_Street_Linux_Server:/etc/ssh# nano sshd.config.d
root@Baker_Street_Linux_Server:/etc/ssh# service ssh status
 * sshd is running
root@Baker_Street_Linux_Server:/etc/ssh# service ssh stop
 * Stopping OpenBSD Secure Shell server sshd
root@Baker_Street_Linux_Server:/etc/ssh# service ssh restart
 * Restarting OpenBSD Secure Shell server sshd
[ OK ]
root@Baker_Street_Linux_Server:/etc/ssh#

```

- I ran the command apt update to make sure it has the version of all packages.



```

root@Baker_Street_Linux_Server: /
File Edit View Search Terminal Help
sysadmin@ip-10-0-1-33:~$ sudo docker start project1_v4
[sudo] password for sysadmin:
project1_v4
sysadmin@ip-10-0-1-33:~$ sudo docker exec -it project1_v4 /bin/bash
root@Baker_Street_Linux_Server:/# apt update -y
0% [Working]

```

- I ran the command apt upgrade -y to update all already installed packages to the latest version.

```
2 kB]
Fetched 16.7 MB in 26s (646 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@Baker_Street_Linux_Server:/# apt upgrade -y
Reading package lists... Done
```

- I then removed the telnet package and rsh-client package.

```
Setting up libgssapi-krb5-2:amd64 (1.19.2-2ubuntu0.6) ...
Processing triggers for libc-bin (2.35-0ubuntu3.9) ...
root@Baker_Street_Linux_Server:/# apt remove telnet
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'telnet' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Baker_Street_Linux_Server:/# apt remove rsh-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'rsh-client' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Baker_Street_Linux_Server:/# █
```

- I did some online research to find out why telnet and rsh-client needed to be removed. The reason why we removed telnet was because any username and password can be easily intercepted by hackers or attacks. Rsh-client was removed because of unencrypted information over the network, which makes us vulnerable to spoofing attacks.

- I ran the command `apt autoremove -y` to clean up dependencies, remove disk space, and remove old files that are no longer needed to be on the system.

```
update-alternatives: using /usr/bin/scp to provide /usr/bin/rcp (rcp) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/rcp.1.gz because as
update-alternatives: using /usr/bin/ssh to provide /usr/bin/rsh (rsh) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/rsh.1.gz because as
update-alternatives: using /usr/bin/slogin to provide /usr/bin/rlogin (rlogin) in auto
update-alternatives: warning: skip creation of /usr/share/man/man1/rlogin.1.gz because
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.
root@Baker Street Linux Server:~#
```

- I ran the command apt install and added the following packages ufw, lynis, and tripwire.

```
File Edit View Search terminal Help
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Baker_Street_Linux_Server:~# apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcap2 libcurl3 libcurl3-nss libufw libufw-core

  Created symlink /etc/systemd/system/timers.target.wants/ufw.timer → /lib/systemd/
Setting up menu (2.1.47ubuntu4) ...
Processing triggers for menu (2.1.47ubuntu4) ...
root@Baker_Street_Linux_Server:~# apt install tripwire
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpio postfix ssl-cert

Suggested packages:
  lynis

  0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Baker_Street_Linux_Server:/# apt install lynis
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lynis is already the newest version (3.0.7-1).
```

- UFW can block incoming traffic, deny and limit traffic for firewall rules and can log network traffic to help detect and analyze attacks. Lynis scans the system and checks for vulnerabilities within the system. Tripwire has multiple functions of monitoring and unauthorized changes.

I ran the top command to see the current services running. I found some services running in the background that needed to be killed. Those PID numbers were 205 and 58.

```

root@Baker_Street_Linux_Server:~# top
File Edit View Search Terminal Help
%Cpu(s): 2.1 us, 0.7 sy, 0.0 ni, 95.8 id, 0.3 wa, 0.0 hi, 0.0 sl, 1.1 st
Mem Mem : 15803.5 total, 12295.3 free, 1436.7 used, 2071.6 buff/cache
Mib Swap : 0.0 total, 0.0 free, 0.0 used, 13818.2 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
205 mysql 20 0 2514460 476992 44496 S 1.0 2.9 0:14.13 mysqld
1 root 20 0 4364 3116 2872 S 0.0 0.0 0:00.04 start_services.
58 mysql 20 0 2514460 476992 1588 S 0.0 0.0 0:00.00 mysqld_safe
59 root 20 0 4628 3932 3224 S 0.0 0.0 0:00.00 mysqld
614 root 20 0 81136 16388 13464 S 0.0 0.1 0:00.00 mysqld
623 root 20 0 79000 9240 6448 S 0.0 0.1 0:00.00 smbd-notifyd
624 root 20 0 78992 4304 1512 S 0.0 0.0 0:00.00 cleanupd
632 root 20 0 65436 8788 6588 S 0.0 0.1 0:00.01 nmbd
641 root 20 0 15432 3796 2172 S 0.0 0.0 0:00.00 sshd
648 root 20 0 2824 1098 912 S 0.0 0.0 0:00.04 tail
668 root 20 0 7368 3512 2952 R 0.0 0.0 0:00.00 top

I

root@Baker_Street_Linux_Server:~# kill -9 205 58
root@Baker_Street_Linux_Server:~# top

```

- I also ran the ps aux command and killed PID numbers 304,302,303,293,310.

```

root@Baker_Street_Linux_Server:/etc/init.d# ls -ahl
total 64K
drwxr-xr-x 1 root root 4.0K Feb 27 00:33 .
drwxr-xr-x 1 root root 4.0K Feb 27 01:33 ..
-rw-r--r-- 1 root root 3.0K Mar 17 2021 cron
-rw-r--r-- 1 root root 3.1K Jun 28 2021 dbus
-rw-r--r-- 1 root root 1.8K Feb 28 2022 hwclock.sh
-rw-r--r-- 1 root root 5.5K Jun 14 2023 mysql
-rw-r--r-- 1 root root 1.9K Jan 5 2024 nmbd
-rw-r--r-- 1 root root 2.4K Dec 26 2016 opensbsd-inetd
-rw-r--r-- 1 root root 3.1K Mar 30 2023 postfix
-rw-r--r-- 1 root root 959 Feb 25 2022 procps
-rw-r--r-- 1 root root 2.3K Jan 5 2023 samba-ad-dc
-rw-r--r-- 1 root root 1.9K Mar 13 2024 smbd
-rw-r--r-- 1 root root 4.0K Mar 13 2024 tail
drwxr-xr-x 1 root root 2.1K Sep 19 2021 lvs
root@Baker_Street_Linux_Server:/etc/init.d# ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.4 4364 3112 ? Ss Feb26 0:00 /bin/bash /usr/local/bin/start_services.sh
root 293 0.0 0.1 81168 16856 ? Ss Feb26 0:00 /usr/sbin/nmbd -D
root 382 0.0 0.0 79028 9288 ? Ss Feb26 0:00 /usr/sbin/nmbd -D
root 303 0.0 0.0 79028 4384 ? Ss Feb26 0:00 /usr/sbin/nmbd -D
root 304 0.0 0.1 80960 20268 ? Ss Feb26 0:00 /usr/lib/x86_64-linux-gnu/samba/smbd-bggd --ready-signal-fd=46 --parent-watch-fd=12 --debuglevel=0 -F
root 310 0.0 0.0 65364 8904 ? Ss Feb26 0:00 /usr/sbin/nmbd -D
root 326 0.0 0.0 2824 1056 ? Ss Feb26 0:00 tail -f /dev/null
root 327 0.0 0.0 4628 3868 pts/0 Ss Feb26 0:00 /bin/bash
root 454 0.0 0.0 15432 3844 ? Ss 00:09 0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root 2656 0.0 0.0 15432 3844 ? Ss 02:08 0:00 ps aux
root@Baker_Street_Linux_Server:/etc/init.d# ps aux | grep samba
root 304 0.0 0.1 80960 20268 ? Ss Feb26 0:00 /usr/lib/x86_64-linux-gnu/samba/smbd-bggd --ready-signal-fd=46 --parent-watch-fd=12 --debuglevel=0 -F
root 2658 0.0 0.0 3472 1592 pts/0 S+ 02:08 0:00 grep --color=auto samba
root@Baker_Street_Linux_Server:/etc/init.d# file mysql
bash: file: command not found
root@Baker_Street_Linux_Server:/etc/init.d# kill 304
root@Baker_Street_Linux_Server:/etc/init.d# kill 302
root@Baker_Street_Linux_Server:/etc/init.d# kill 303
root@Baker_Street_Linux_Server:/etc/init.d# kill 293
root@Baker_Street_Linux_Server:/etc/init.d# kill 310
root@Baker_Street_Linux_Server:/etc/init.d# ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 0.0 0.0 4364 3112 ? Ss Feb26 0:00 /bin/bash /usr/local/bin/start_services.sh
root 326 0.0 0.0 2824 1056 ? Ss Feb26 0:00 tail -f /dev/null
root 327 0.0 0.0 4628 3868 pts/0 Ss Feb26 0:00 /bin/bash
root 454 0.0 0.0 15432 3844 ? Ss 00:09 0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root 2656 0.0 0.0 15432 3844 ? Ss 02:17 0:00 ps aux
root@Baker_Street_Linux_Server:/etc/init.d#

```

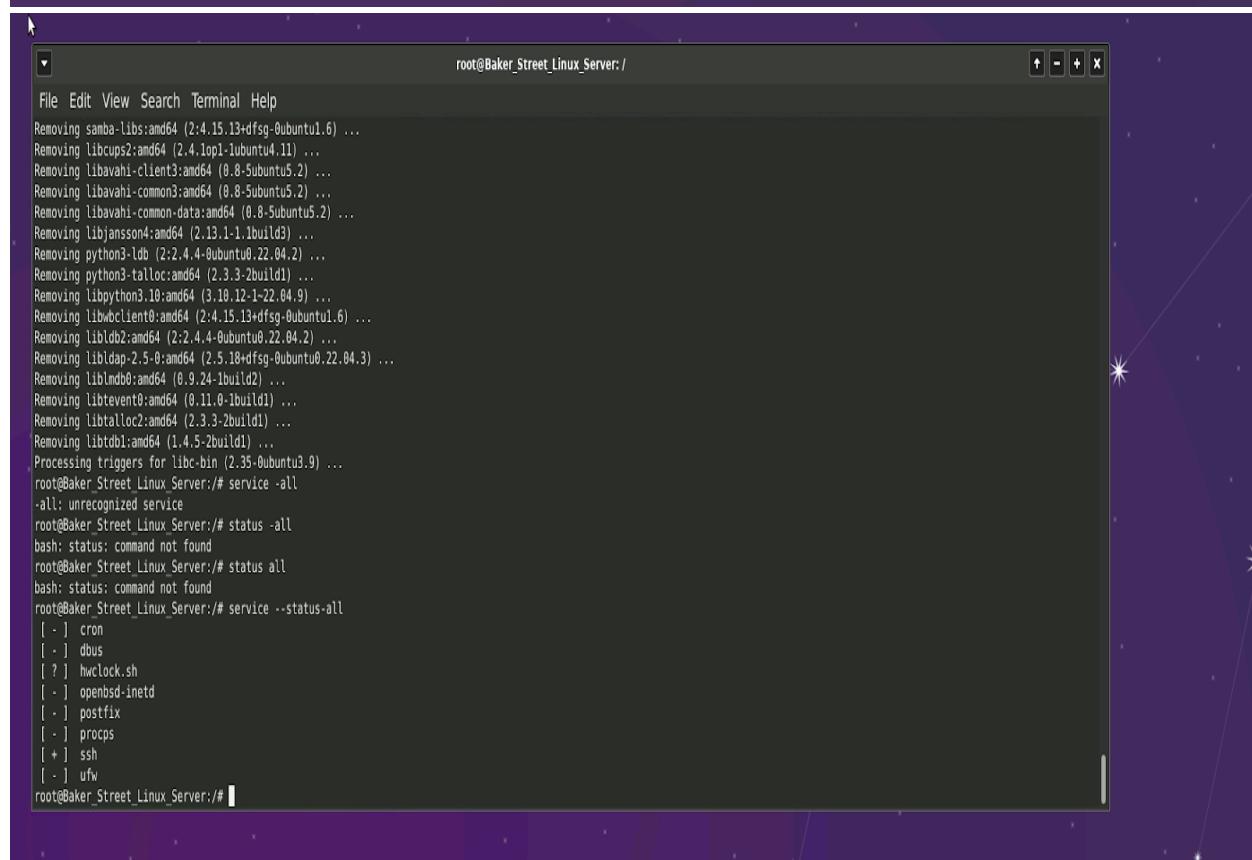
I then did some research because we couldn't use the systemctl to remove mysql and samba, so we needed to use the service command to remove it.

- After conducting my research, I ran the command `apt-get purge -y samba 2>/dev/null` which allowed me to remove samba.
- I then ran the same command and changed it to mysql to remove mysql. I then ran `service -status -all` to check and see if samba and mysql had been removed.

```
e 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Baker_Street_Linux_Server:/# apt-get purge -y samba 2>/dev/null
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'samba' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Baker_Street_Linux_Server:/#
```

```
root@Baker_Street_Linux_Server:/# apt-get purge -y mysql 2>/dev/null
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
root@Baker_Street_Linux_Server:/#
```

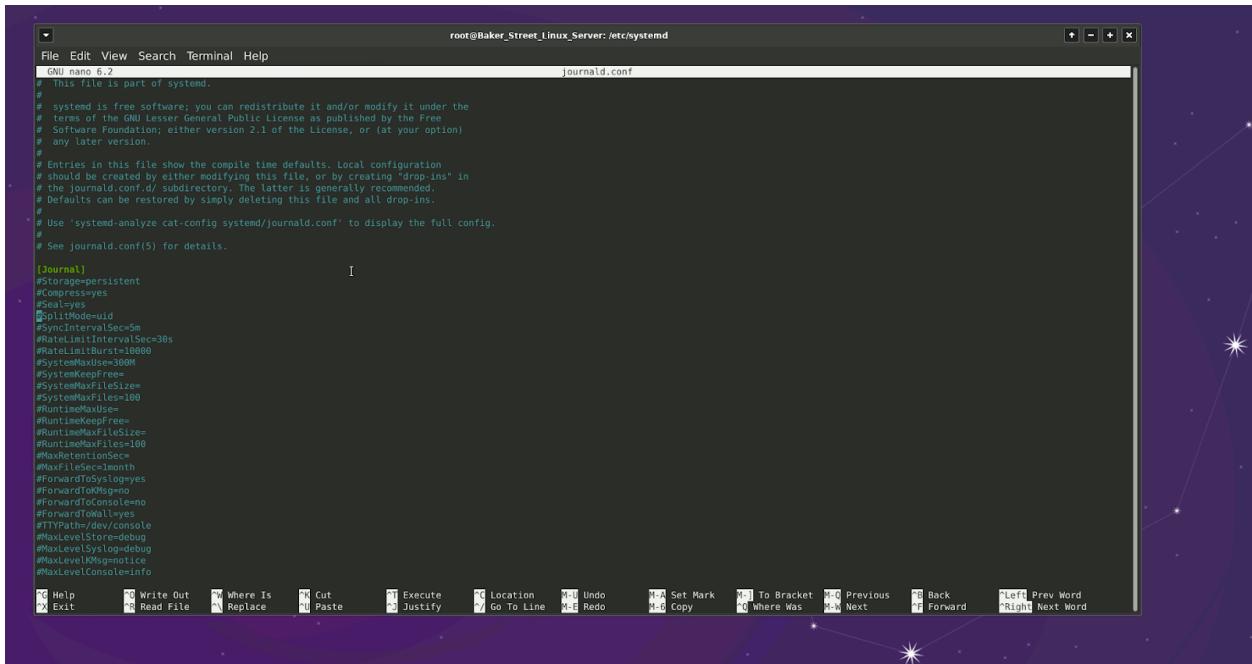
  


The screenshot shows a terminal window titled "root@Baker\_Street\_Linux\_Server:/". It displays the following command and its output:

```
File Edit View Search Terminal Help
Removing samba-libs:amd64 (2:4.15.13+dfsg-0ubuntu1.6) ...
Removing libcurl2:amd64 (2.4.10+1~ubuntu4.11) ...
Removing libavahi-client3:amd64 (0.8-5ubuntu5.2) ...
Removing libavahi-common3:amd64 (0.8-5ubuntu5.2) ...
Removing libavahi-common-data:amd64 (0.8-5ubuntu5.2) ...
Removing libjsonnson4:amd64 (2.13.1-1~build3) ...
Removing python3-ldb (2:2.4.4-0ubuntu0.22.04.2) ...
Removing python3-talloc:amd64 (2.3.3-2build1) ...
Removing libpython3.10:amd64 (3.10.12-1~22.04.9) ...
Removing libwbclient0:amd64 (2:4.15.13+dfsg-0ubuntu1.6) ...
Removing libldb2:amd64 (2:2.4.4-0ubuntu0.22.04.2) ...
Removing libldap-2.5-8:amd64 (2.5.18+dfsg-0ubuntu0.22.04.3) ...
Removing libldb8:amd64 (0.9.24-1build2) ...
Removing libtevent8:amd64 (0.11.0-1build1) ...
Removing libtalloc2:amd64 (2.3.3-2build1) ...
Removing libtdb1:amd64 (1.4.5-2build1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.9) ...
root@Baker_Street_Linux_Server:/# service -all
-all: unrecognized service
root@Baker_Street_Linux_Server:/# status -all
bash: status: command not found
root@Baker_Street_Linux_Server:/# status all
bash: status: command not found
root@Baker_Street_Linux_Server:/# service --status-all
[ - ] cron
[ - ] dbus
[ ? ] hwclock.sh
[ - ] openbsd-inetd
[ - ] postfix
[ - ] procps
[ + ] ssh
[ - ] ufw
root@Baker_Street_Linux_Server:/#
```

I ran the command nano /etc/systemd/journald.conf to make changes.

- The changes I made was set “storage=persistent
  - This will save logs locally on the machine
- The other change I made was systemMaxuse=300
  - This configures the max disk space logs can utilize.



The screenshot shows a terminal window titled "root@Baker\_Street\_Linux\_Server: /etc/systemd". The window contains the contents of the /etc/systemd/journald.conf file. The file includes comments about its purpose and how to modify it. It defines a [Journal] section with parameters like storage=persistent, systemMaxuse=300, and RuntimeMaxUse=30s. The bottom of the screen shows the nano editor's menu bar with options like Help, Write Out, Read File, etc., and a status bar showing various keyboard shortcuts.

```
# This file is part of systemd.
#
# Copyright © 2011-2012 Red Hat, Inc.
#
# systemd is free software; you can redistribute it and/or modify it under the
# terms of the GNU General Public License as published by the Free
# Software Foundation; either version 2.1 of the License, or (at your option)
# any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the journald.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# Use 'systemctl-analyze cat-config systemd/journald.conf' to display the full config.
#
# See journald.conf(5) for details.

[Journal]
#Storage=persistent
#Compress=yes
#Seal=yes
#SyncInterval=5ms
#RelimitIntervalSec=5s
#RelimitBurst=10000
#SystemMaxuse=300M
#SystemKeepFree=
#SystemMaxFileSize=
#SystemMaxFileCount=100
#RuntimeMaxuse=
#RuntimeKeepFree=
#RuntimeMaxFileSize=
#RuntimeMaxFileCount=100
#WtmpMaxAgeSec=3600
#WtmpXFileSec=3600
#ForwardToSyslog=yes
#ForwardToMsgno=
#ForwardToConsole=
#ForwardToAll=yes
#ForwardToTTY=tty0
#LogLevel=debug
#LogLevelSyslog=debug
#LogLevelKmsg=info
#LogLevelConsole=info
```

- Next, I ran the command nano /etc/logrotate.conf. The changes I made were changing the log rotation from weekly **to daily** and rotating out the **logs after 7 days**.

```
root@Baker_Street_Linux_Server: /  
File Edit View Search Terminal Help  
san      GNU nano 6.2          /etc/logrotate.conf *  
# 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 4 weeks worth of backlogs  
rotate 7  
  
# create new (empty) log files after rotating old ones  
create  
  
# use date as a suffix of the rotated file  
#dateext  
  
# uncomment this if you want your log files compressed  
#compress  
  
# packages drop log rotation information into this directory  
include /etc/logrotate.d  
  
# system-specific logs may also be configured here.  
ima  
  
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo  
^X Exit      ^R Read File   ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
```

I created a script located in the home directory that was named `hardening_script1.sh` into the home directory. The changes I made are listed below.

- The changes I made were to list all the commands. hostname command, OS command, uname -r, the free command and uptime.
- For the backup I entered the command we used earlier for the backup which was `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 /`
- I placed and displayed the sudoers command, which was `/etc/sudoers`
- I placed the command to show how to remove all world permissions, which was `chmod -R o-000`, or you can use `o-rwx`

- Showed the updating permissions of the engineering scripts. Which is listed in the screenshot listed below.
  - There are no members listed in the research group. I added a comment in the script saying there was no one in the research group.

```
File Edit View Search Terminal Help
GNU nano 6.2                               hardening_script1.sh *
#!/bin/bash
# Variable for the report output file, choose an output file name
REPORT_FILE="script 1.txt"
# Output the hostname
echo "Gathering hostname..."
# Placeholder for command to get the hostname
echo "Hostname: $(hostname)" >> $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 -r)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
# Output memory information
echo "Gathering memory information..."
# Placeholder for command to get memory info
echo "Memory Information: $(free)" >> $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 -cvzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc -->
echo "OS backup completed." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
# Output the sudoers file to the report
echo "Gathering sudoers file..."
# Placeholder for command to output sudoers file
echo "Sudoers file:$(/etc/sudoers)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
# Script to check for files with world permissions and update them
echo "Checking for files with world permissions..."
```

```
root@Baker_Street_Linux_Server: /home
File Edit View Search Terminal Help
GNU nano 6.2                               hardening_script1.sh *
# you can use either of the two commands
chmod -R o-rwx /home/
chmod -R o-000 /home/
# 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
chmod 770 Engineering_script.sh_script1.sh
chmod 770 Engineering_script.sh_script2.sh
Here is the example command for the engineering group:
find -iname '*engineering*' -exec chown :engineering {} +
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 r>
there was no user in the research group
echo "Permissions updated for Research scripts" >> $REPORT_FILE
#no one was listed in the research group
printf "\n" >> $REPORT_FILE
# Finance scripts - Only members of the finance group
echo "Updating permissions for Finance scripts"
# Placeholder for command to update permissions
chmod 770 Finance_script.sh_script1.sh
chmod 770 Finance_script.sh_script2.sh
echo "Permissions updated for Finance scripts." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
echo "Script execution completed. Check $REPORT_FILE for details."

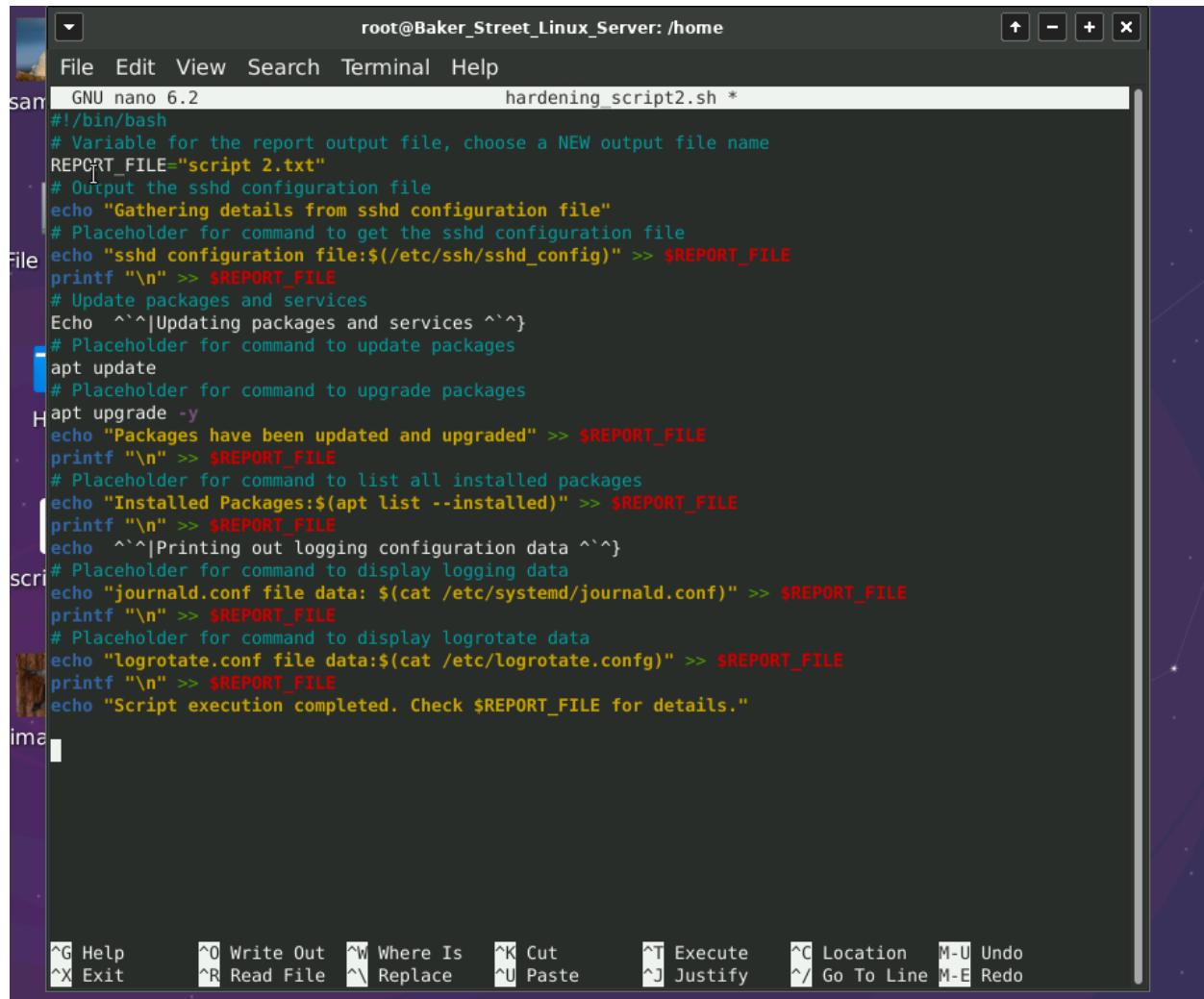
```

^G Help      ^O Write Out      ^W Where Is      ^K Cut      ^T Execute      ^C Location      M-U Undo  
^X Exit      ^R Read File      ^\ Replace      ^U Paste      ^J Justify      ^/ Go To Line      M-E Redo

For the second script, the changes I made were

- Report\_file, which I named it as script2.txt
- I showed the command I used for the sshd configuration files, which was
  - /etc/ssh/sshd\_config
- I showed the command I used to update packages, which was
  - Apt update
- Showed the command to upgrade packages which was
  - Apt upgrade -y
- The command I used to show the installed packages was apt list –installed

- For the journald.conf and the logrotate.conf I made sure I used cat to make sure all the information displayed what was actually in that file.
  - Command I used was
    - cat /etc/logrotate.conf
    - cat /etc/systemd/journal.conf



```

root@Baker_Street_Linux_Server: /home
File Edit View Search Terminal Help
san  GNU nano 6.2           hardening_script2.sh *
#!/bin/bash
# Variable for the report output file, choose a NEW output file name
REPCRT_FILE="script 2.txt"
# Output the sshd configuration file
echo "Gathering details from sshd configuration file"
# Placeholder for command to get the sshd configuration file
echo "ssh configuration file:$(/etc/ssh/sshd_config)" >> $REPCRT_FILE
printf "\n" >> $REPCRT_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" >> $REPCRT_FILE
printf "\n" >> $REPCRT_FILE
# Placeholder for command to list all installed packages
echo "Installed Packages:$(apt list --installed)" >> $REPCRT_FILE
printf "\n" >> $REPCRT_FILE
echo ^`^|Printing out logging configuration data ^`^}
# Placeholder for command to display logging data
echo "journald.conf file data: $(cat /etc/systemd/journald.conf)" >> $REPCRT_FILE
printf "\n" >> $REPCRT_FILE
# Placeholder for command to display logrotate data
echo "logrotate.conf file data:$(cat /etc/logrotate.conf)" >> $REPCRT_FILE
printf "\n" >> $REPCRT_FILE
echo "Script execution completed. Check $REPCRT_FILE for details."
scr
ima
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo

```

```
root@Baker_Street_Linux_Server:/home# ./hardening_script1.sh
bash: ./hardening_script1.sh: /bin/bash: bad interpreter: No such file or directory
root@Baker_Street_Linux_Server:/home# nano hardening_script1.sh
root@Baker_Street_Linux_Server:/home# ./hardening_script1.sh
Gathering hostname...
./hardening_script1.sh: line 7: $REPORT_FILE: ambiguous redirect
./hardening_script1.sh: line 8: $REPORT_FILE: ambiguous redirect
Gathering OS version...
./hardening_script1.sh: line 12: $REPORT_FILE: ambiguous redirect
./hardening_script1.sh: line 13: $REPORT_FILE: ambiguous redirect
Gathering memory information...
./hardening_script1.sh: line 17: $REPORT_FILE: ambiguous redirect
./hardening_script1.sh: line 18: $REPORT_FILE: ambiguous redirect
Gathering uptime information...
./hardening_script1.sh: line 22: $REPORT_FILE: ambiguous redirect
./hardening_script1.sh: line 23: $REPORT_FILE: ambiguous redirect
Backing up the OS...
```

I then ran the command `./hardening_script1.sh` and then I ran `./hardening_script2.sh` to make sure the scripts ran properly.

```
root@Baker_Street_Linux_Server:/home# nano hardening_script2.sh
root@Baker_Street_Linux_Server:/home# ./hardening_script2.sh
Gathering details from sshd configuration file
```

I ran the `crontab -e` command to schedule the scripts 1 and 2. Script 1 is going to run once a month for the first month. Script 2 is going to run once a week every Monday

```
root@Baker_Street_Linux_Server:/home# nano hardening_script1.sh
root@Baker_Street_Linux_Server:/home# nano hardening_script2.sh
root@Baker_Street_Linux_Server:/home# crontab -e
No modification made
root@Baker_Street_Linux_Server:/home#
```

```
root@Baker_Street_Linux_Server: /home
File Edit View Search Terminal Help
GNU nano 6.2          /tmp/crontab.UEY0gu/crontab
# Edit this file to introduce tasks to be run by cron.

# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

0 0 1 * * /home/hardening_script1.sh
0 0 * * 1 /home/hardening_script2.sh
```

## Final Project Summary

This project focused on hardening a Linux server to ensure the confidentiality, integrity, and availability of The Baker Street Corporation’s (BSC) sensitive data. Several key mitigation strategies were implemented to reduce the system’s attack surface and improve overall security.

One of the primary tasks involved auditing users, groups, and their permissions. Ensuring that users only have access to the files and directories necessary for their roles is critical—especially in dynamic environments where team members change roles or leave the organization. During this process, I reviewed user home directories and made appropriate changes to limit access based on job function and group membership.

To enhance authentication security, password policies were enforced across all accounts. Weak or default passwords, such as "Spring2021," were identified as unacceptable. Instead, strong password complexity and expiration rules were implemented to help prevent unauthorized access.

Controlling administrative access was another key objective. Sudo privileges were reviewed and restricted to only those who required them. For example, Sherlock was granted full sudo access, while Watson, Mycroft, and members of the research group were limited to executing a specific script. Restricting sudo access reduces the risk of privilege escalation and limits the potential for misuse or system compromise.

Additionally, I ensured that the system's software packages were up to date using apt update and apt upgrade. Keeping packages updated is essential for patching known vulnerabilities and defending against malware and exploits.

These hardening steps collectively strengthen the server's security posture and create a solid foundation for ongoing system monitoring and protection.