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.

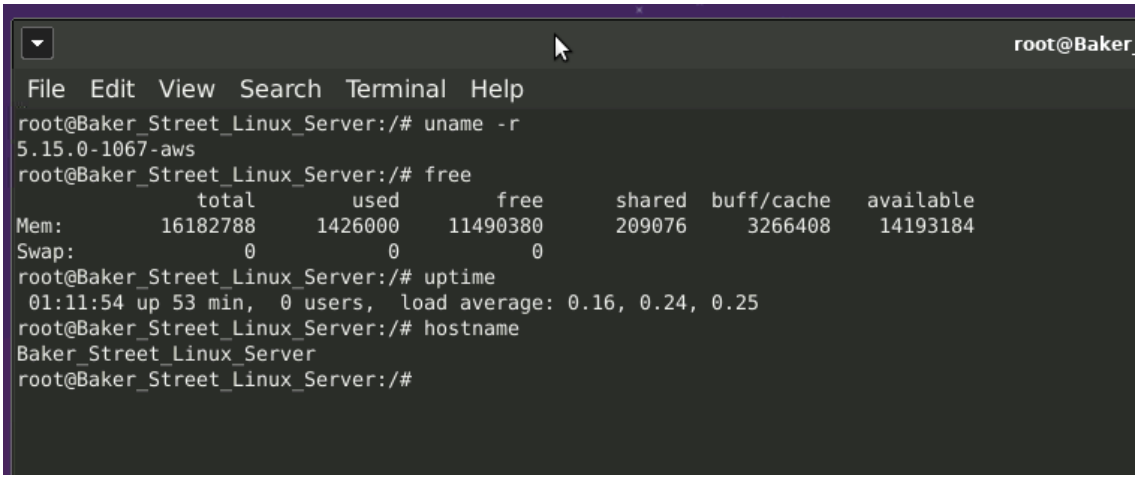
⚙️ Day 2: System Configuration Hardening

- 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

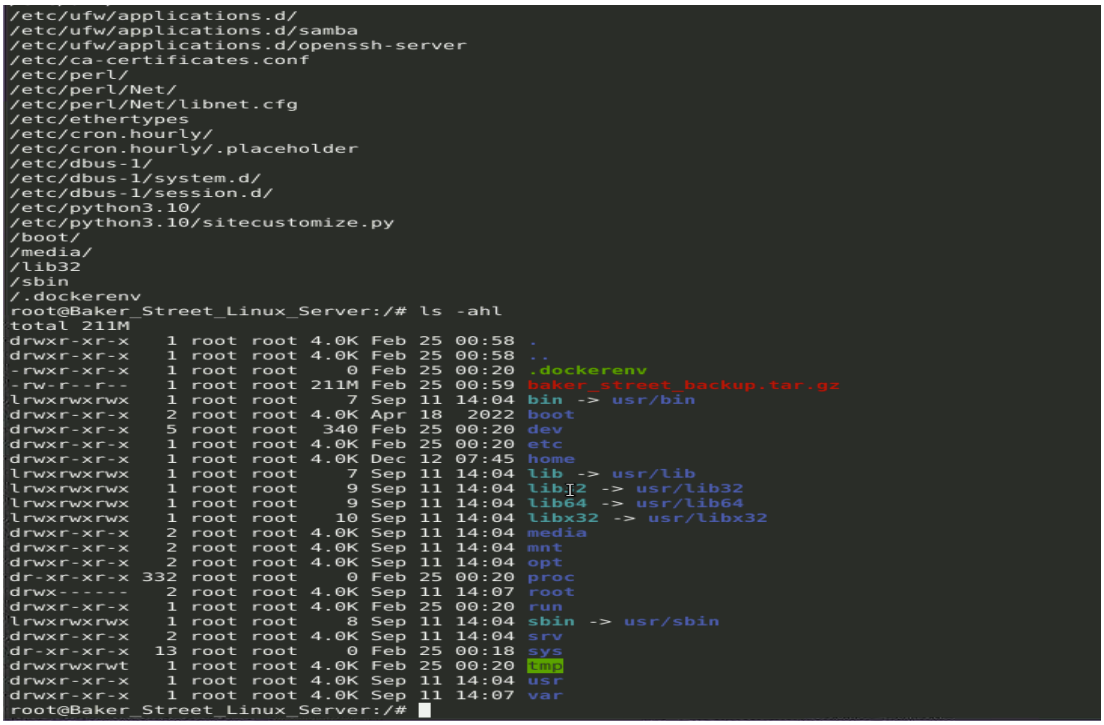
- Developed scripts to automate the security tasks completed in Days 1 and 2.
- Scheduled the scripts to ensure ongoing enforcement of hardening policies.
- Compiled a final summary report detailing all hardening actions taken.

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.



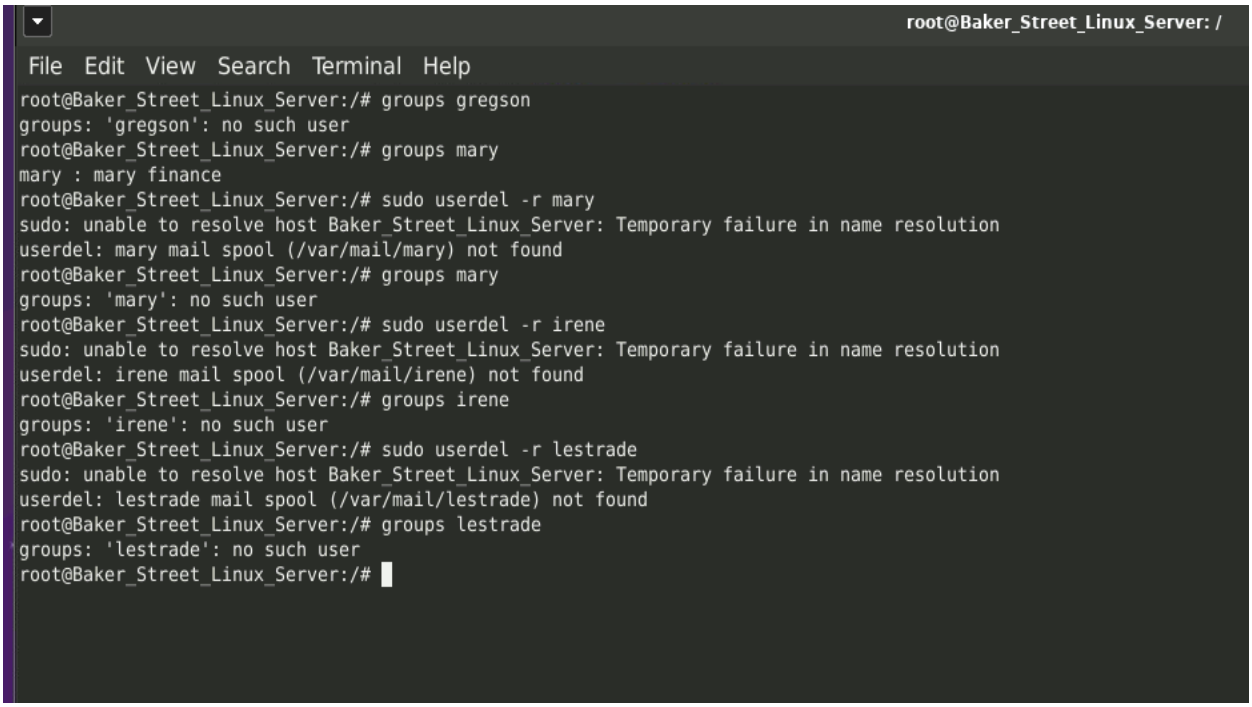
In Part 1, I ran the backup command given in the activity file to back up all files and directories. When it finished, I used the command ls—ahl to make sure the backup was completed. baker_street_backup. Tar.gz was listed in red, which let me know that the backup was completed.

- 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 /

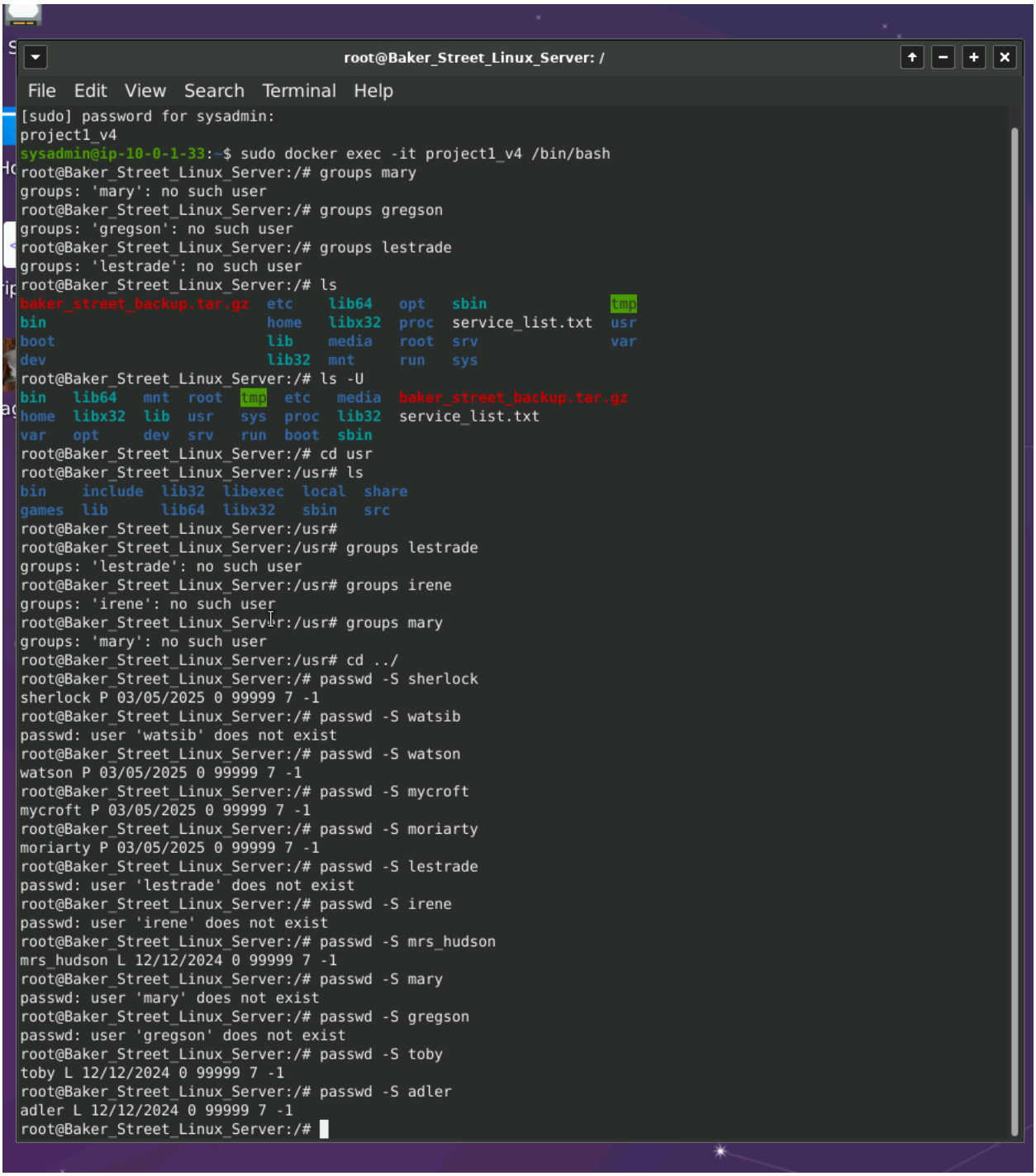


Part 2: I removed all the files and directories of all the employees who have been terminated, which are Mary, Gregson, Irene, and Lestrade. Then, I locked the employees' accounts on temporary leave, which were mrs_hudson and Moriarty. I then checked all the groups to make sure none of the employees 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.

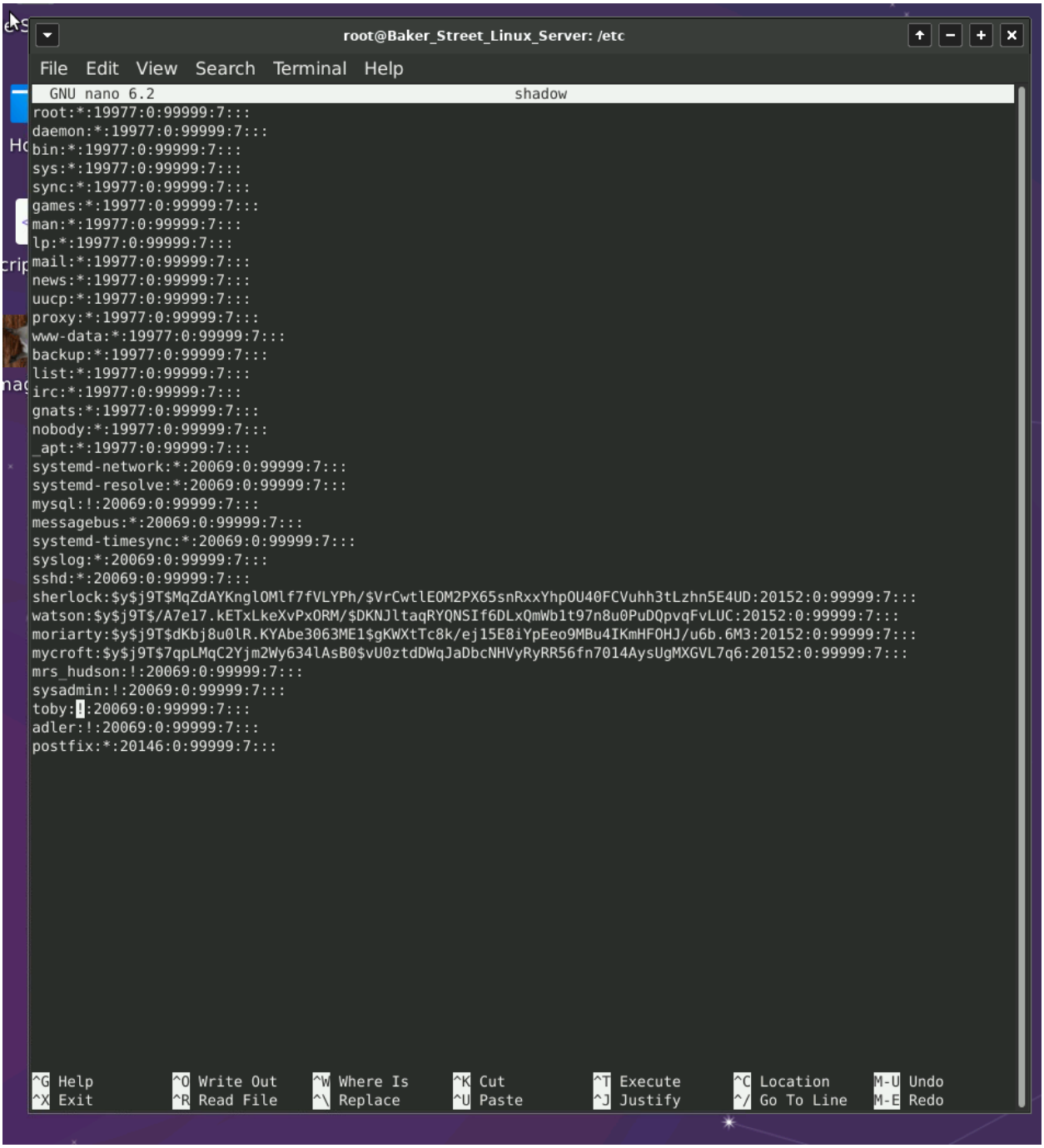
- When I deleted the user who had been terminated, I used the userdel -r (username) to make sure all the files and directories were deleted as well. To make sure the users were deleted, I ran the command groups with the names of terminated users Mary, Gregson, and Lestrade. (no need for me to use sudo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

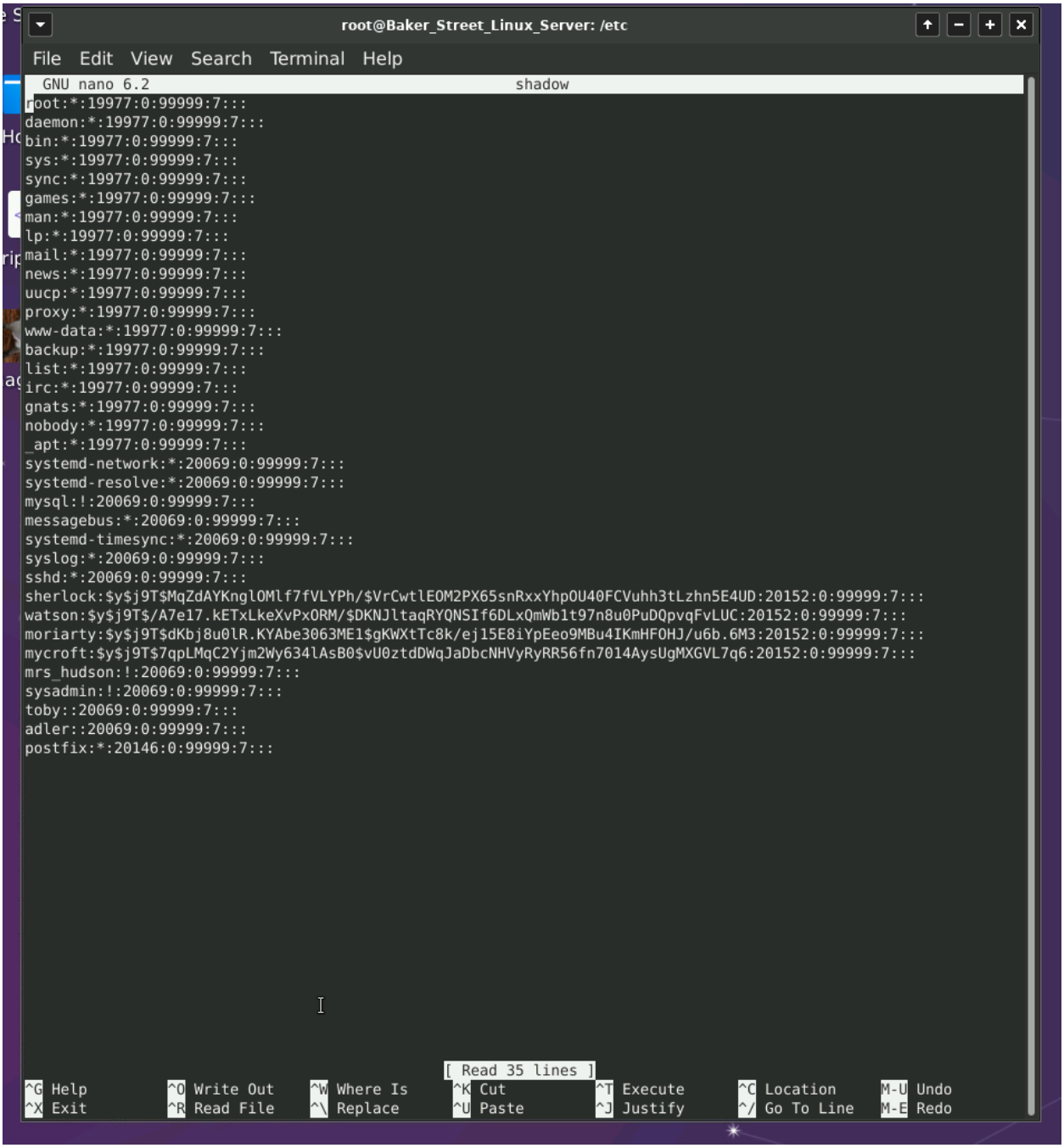


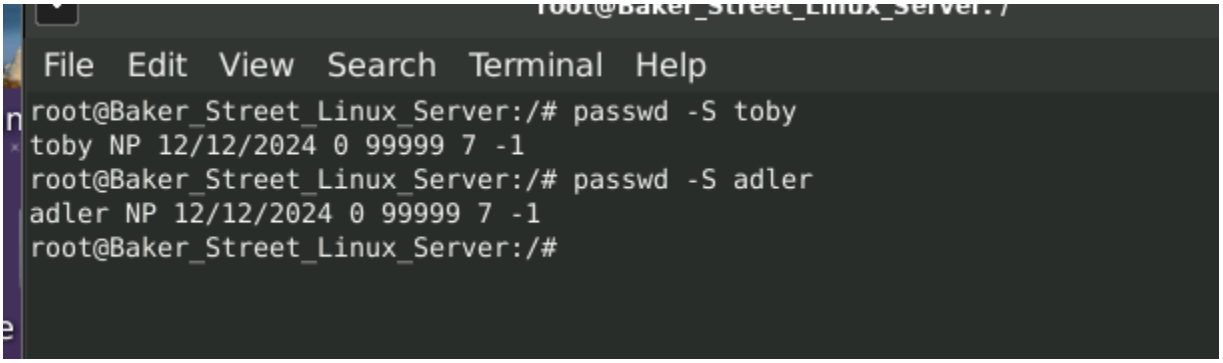
- I ran the command passwd -S (username) to check all the status of all employees listed. The terminated employees are shown below with “user doesn't exist” and we have two employees on the bottom that are locked out, which are Toby and Adler.
- The employees on temporary leave were Moriarty and mrs_hudson. 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.

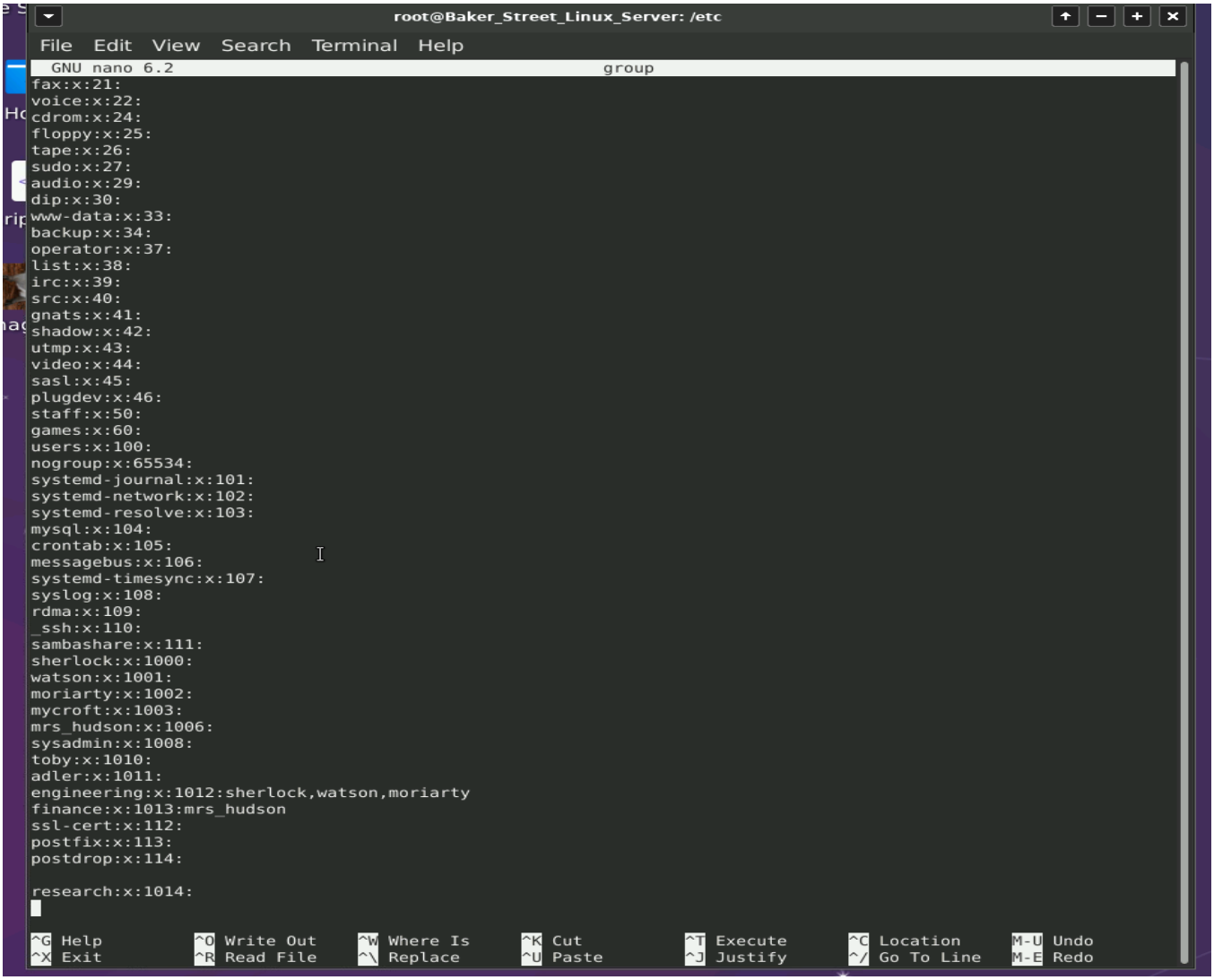


- I nano into /etc/shadow to remove the “!” by their hash to unlock the user account. With me deleting the “!” in their hash, they don't have a password. I ran the command passwd -S (toby and adler) and they have NP which means no password.


root@Baker_Street_Linux_Server: /etc
File Edit View Search Terminal Help
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:yj9T$MqZdAYKnglOMlf7fVLYPh/$VrCwtlEOM2PX65snRxxYhpOU40FCVuhh3tLzhn5E4UD:20152:0:99999:7:::
watson:yj9T$/A7e17.kETxLkeXvPxORM/$DKNJltaqRYQNSIf6DLxQmWb1t97n8u0PuDQpvqFvLUC:20152:0:99999:7:::
moriarty:yj9T$dKbj8u0lR.KYAbe3063ME1$gKWXtTc8k/ej15E8iYpEeo9MBu4IKmHFOHJ/u6b.6M3:20152:0:99999:7:::
mycroft:yj9T$7qpLMqC2Yjm2Wy634lAsB0$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:::
^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: /etc
File Edit View Search Terminal Help
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:yj9T$MqZdAYKnglOMlf7fVLYPh/$VrCwtlEOM2PX65snRxxYhpOU40FCVuhh3tLzhn5E4UD:20152:0:99999:7:::
watson:yj9T$/A7e17.kETxLkeXvPxORM/$DKNJltaqRYQNSIf6DLxQmWb1t97n8u0PuDQpvqFvLUC:20152:0:99999:7:::
moriarty:yj9T$dKbj8u0lR.KYAbe3063ME1$gKWXtTc8k/ej15E8iYpEeo9MBu4IKmHFOHJ/u6b.6M3:20152:0:99999:7:::
mycroft:yj9T$7qpLMqC2Yjm2Wy634lAsB0$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

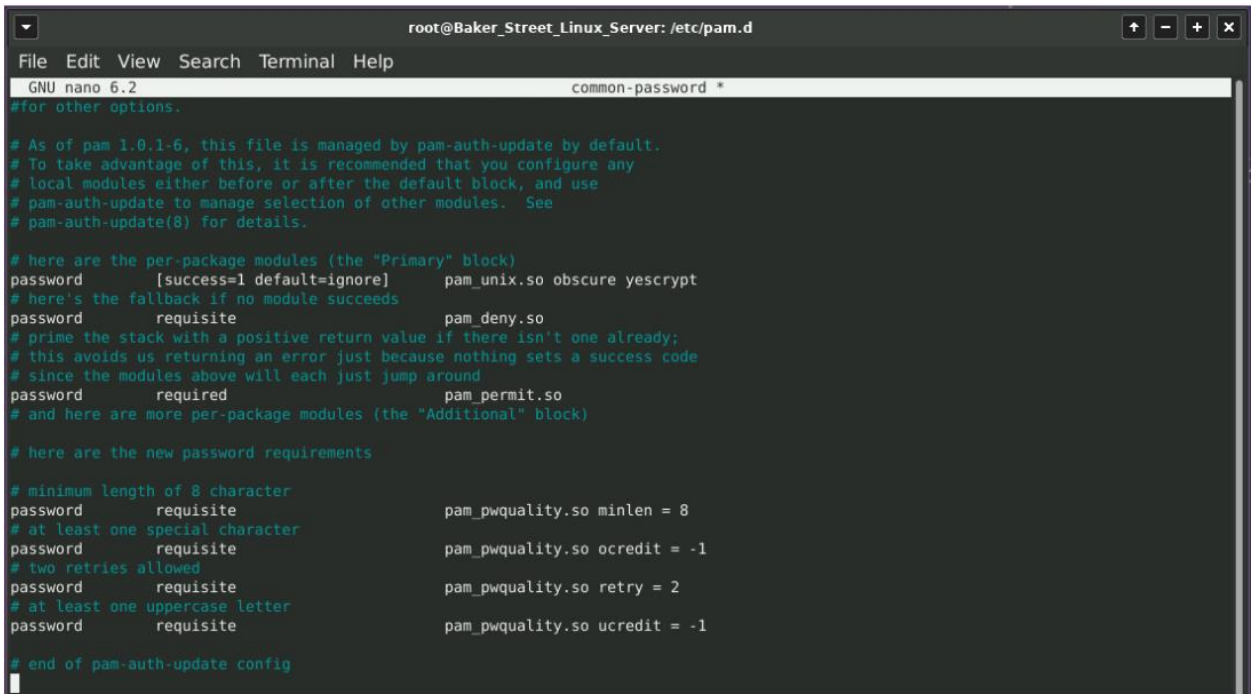

root@Baker_Street_Linux_Server:/
File Edit View Search Terminal Help
root@Baker_Street_Linux_Server:/# passwd -S toby
toby NP 12/12/2024 0 99999 7 -1
root@Baker_Street_Linux_Server:/# passwd -S adler
adler NP 12/12/2024 0 99999 7 -1
root@Baker_Street_Linux_Server:/#



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

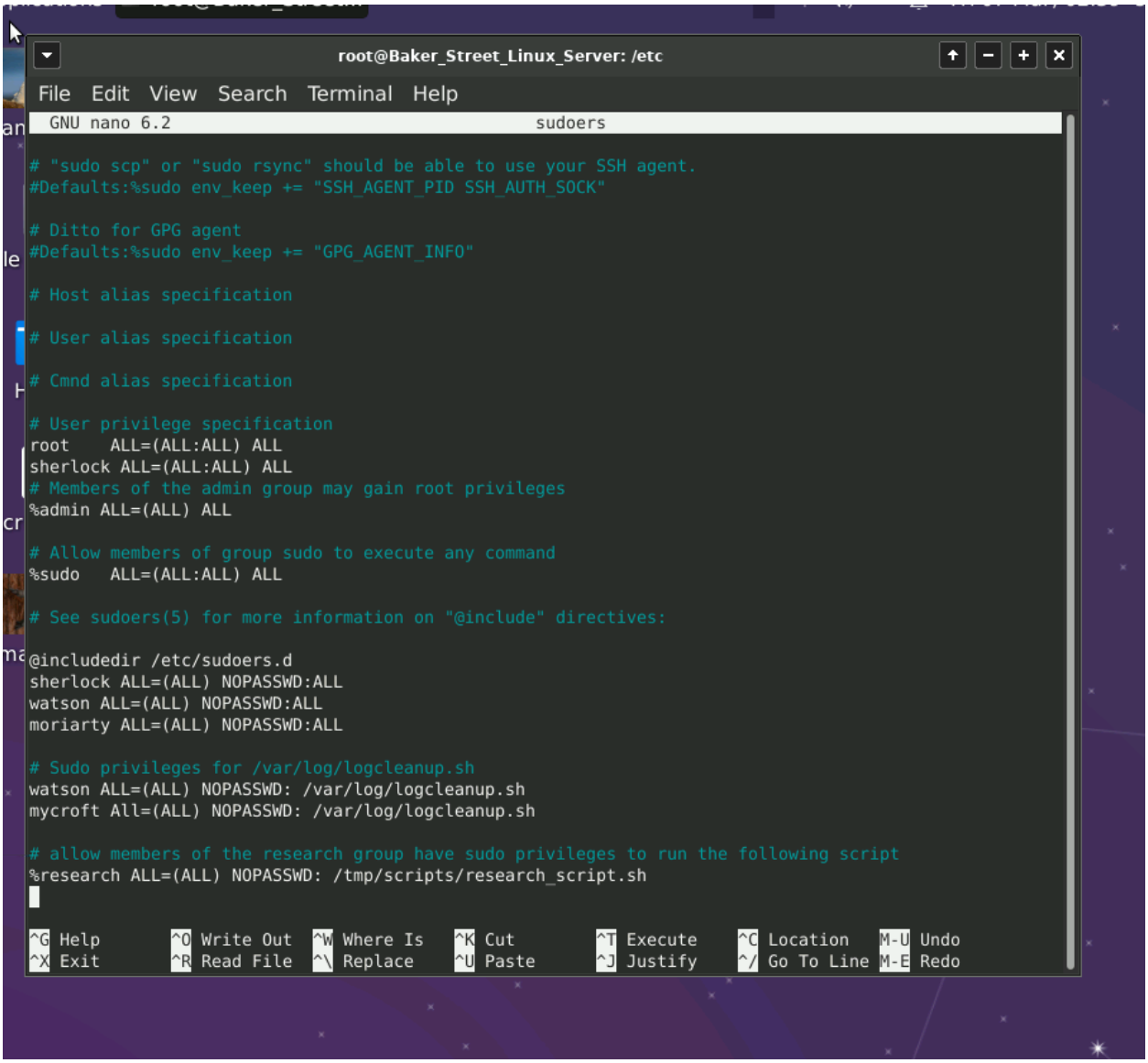
Part 3: I ran nano /etc/pam.d/common-password to edit this file. While in the file, I added a comment saying here are the new password requirements.

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



Part 4: all evidence is provided below in the screenshot.

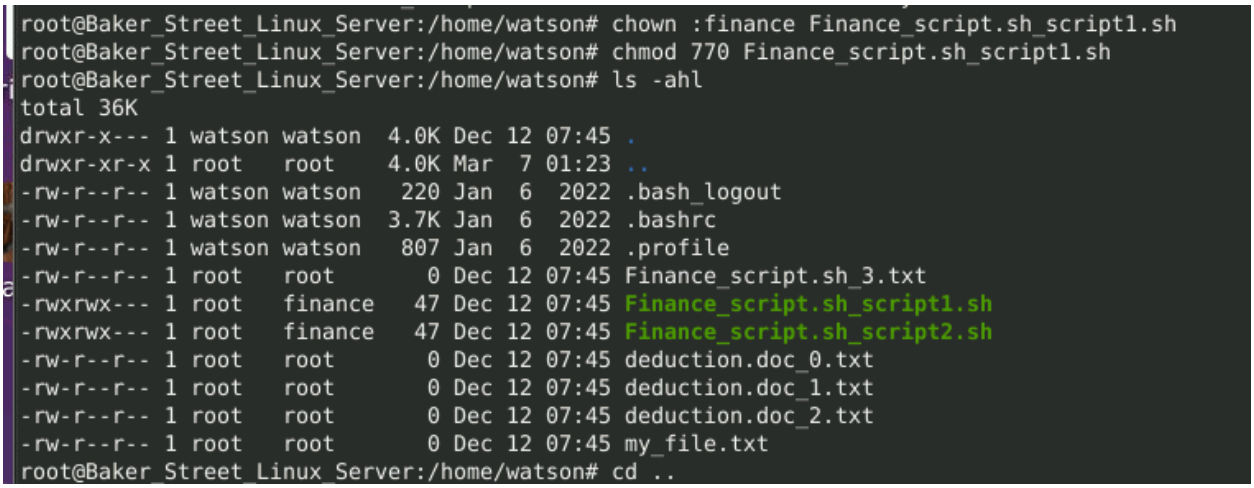
- I nano into /etc/sudoers to make changes to this file. The first change was giving Sherlock full sudo permissions.
- I then gave Watson and Mycroft sudo privileges to run the following script /var/log/logcleanup.sh
- I gave all the employees in the research group sudo privileges to run /tmp/scripts/research_script.sh.

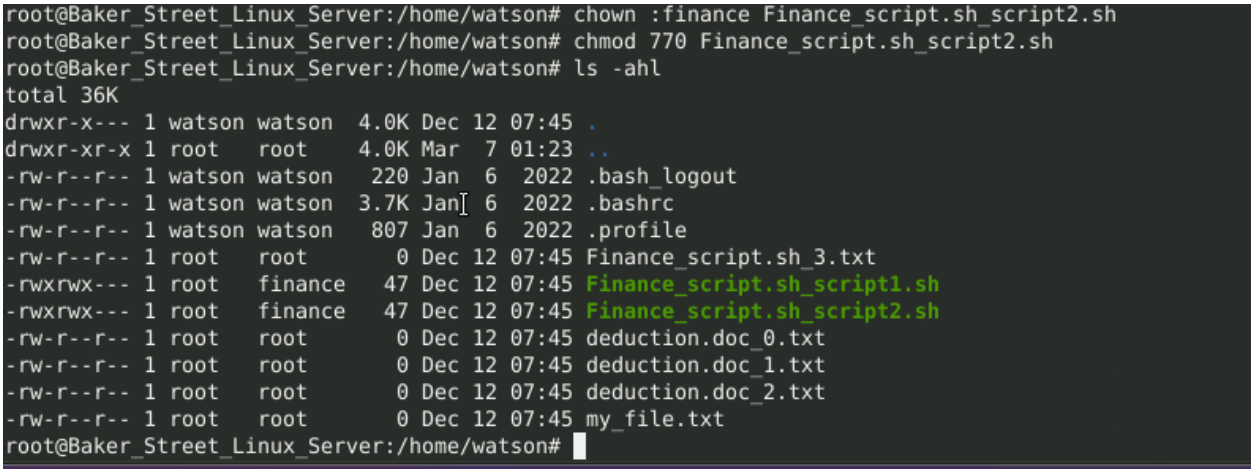


Part 5:

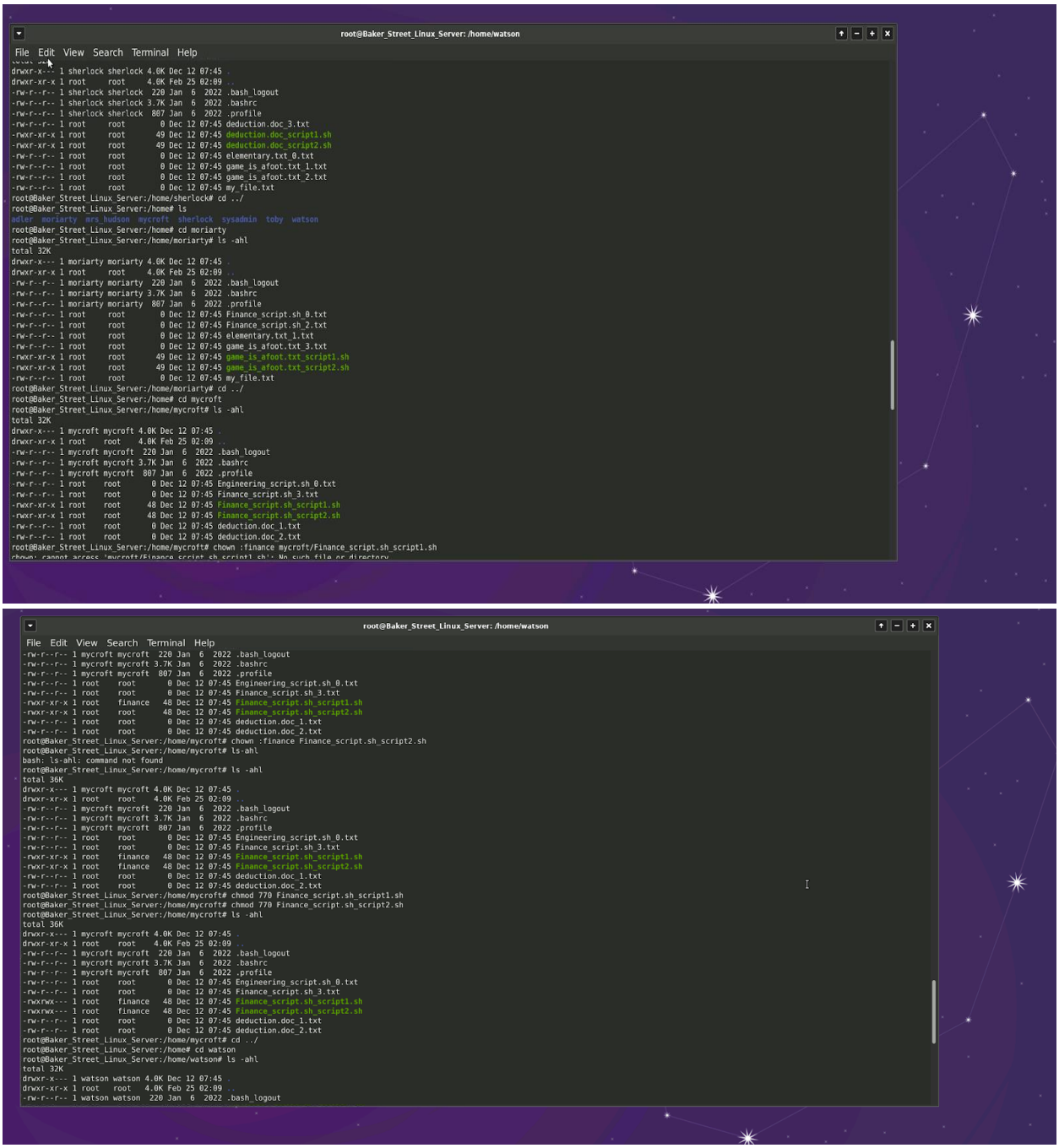
- I went to the home directory, ran the ls command to see all the users/employees. I would change the directory into one of the user/employee and use the ls -ahl command to get a long listing of everything in their directory.
    - I found certain scripts in certain user groups that they were not a part 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.



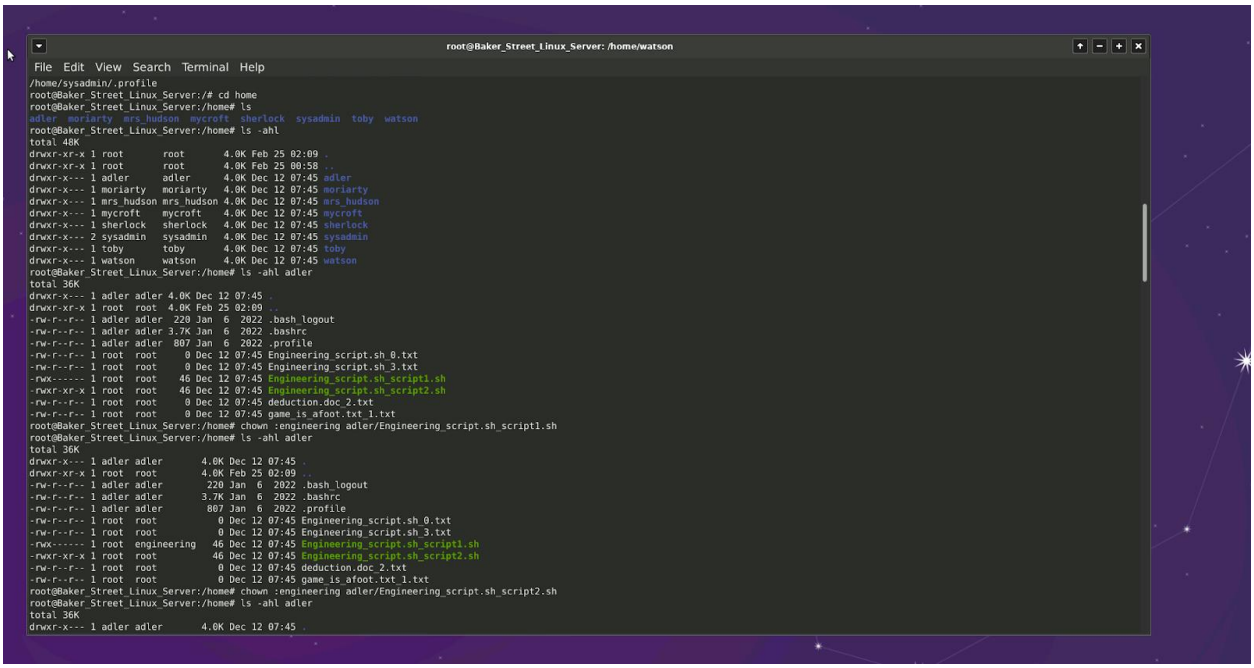


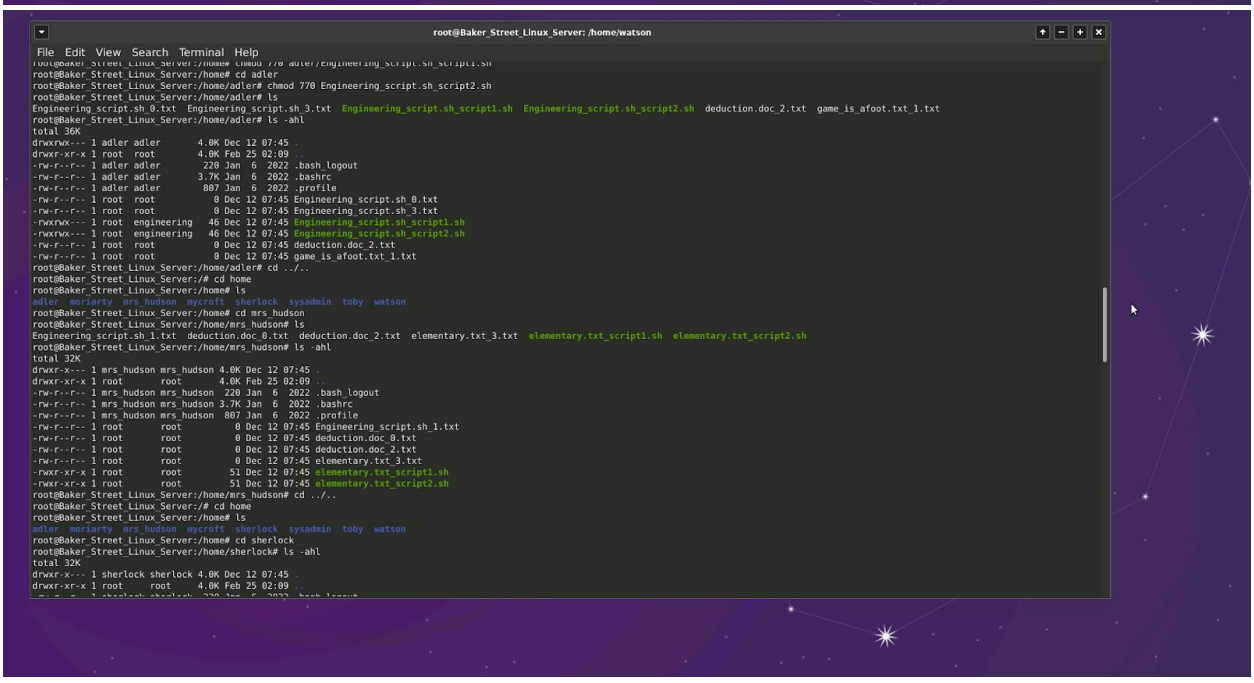
- I changed ownership (chown) to the finance group (I used the command chown :finance Finance_script.sh_script1.sh). Used the same command for the second script.
- I then had to change the permissions (chmod) to 770 to read, write, execute so all members of the finance group can read, write, execute. Command used (chmod 770 Finance_script.sh_script1.sh. I did the same thing for the second script as well.



- 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.





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

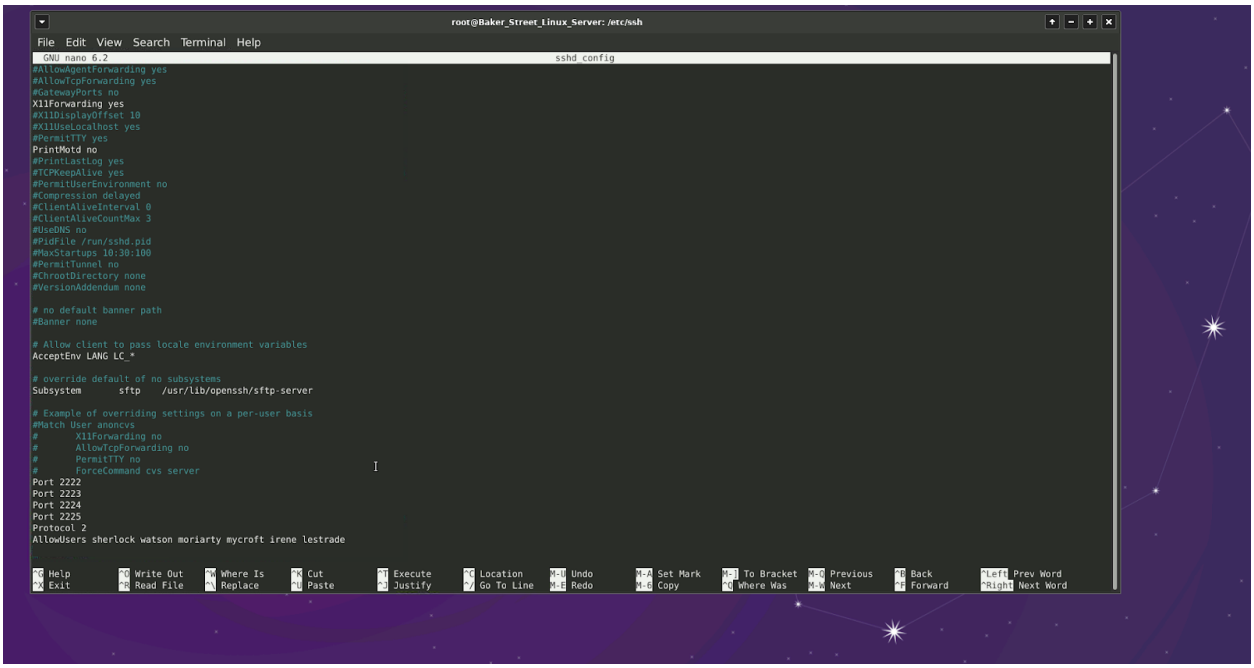
Auditing and Securing SSH:

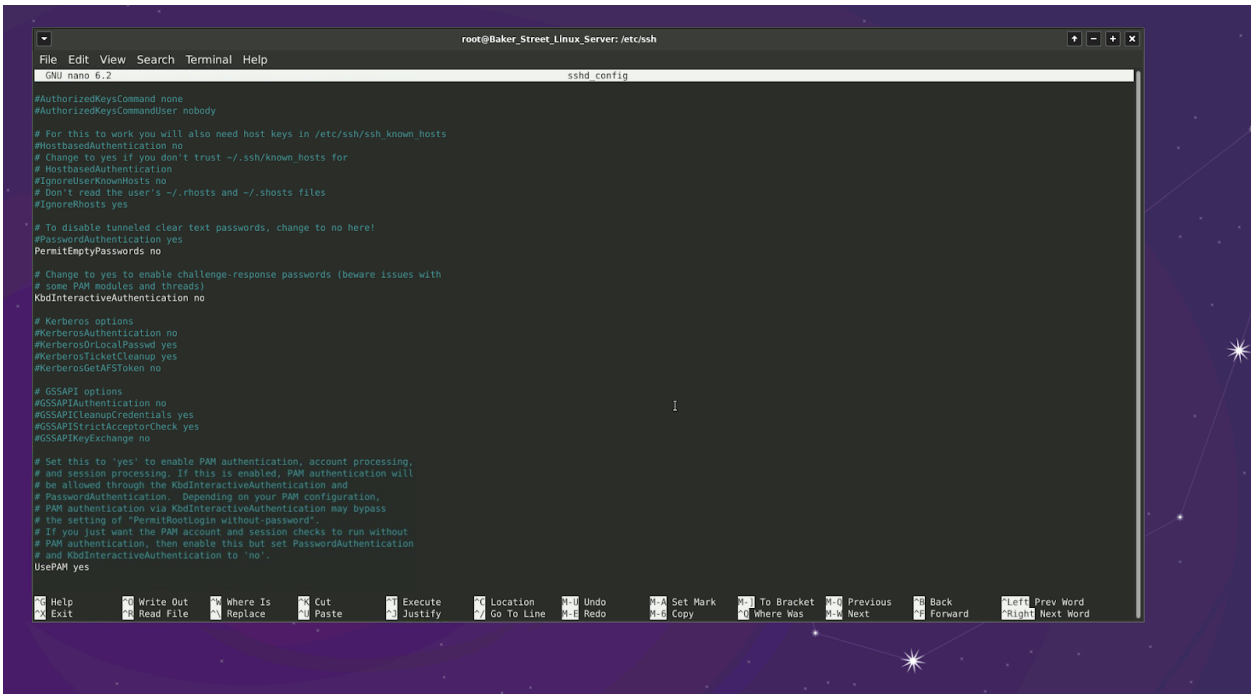
I ran the command nano /etc/ssh/sshd_config. While in this file I made the changes needed.

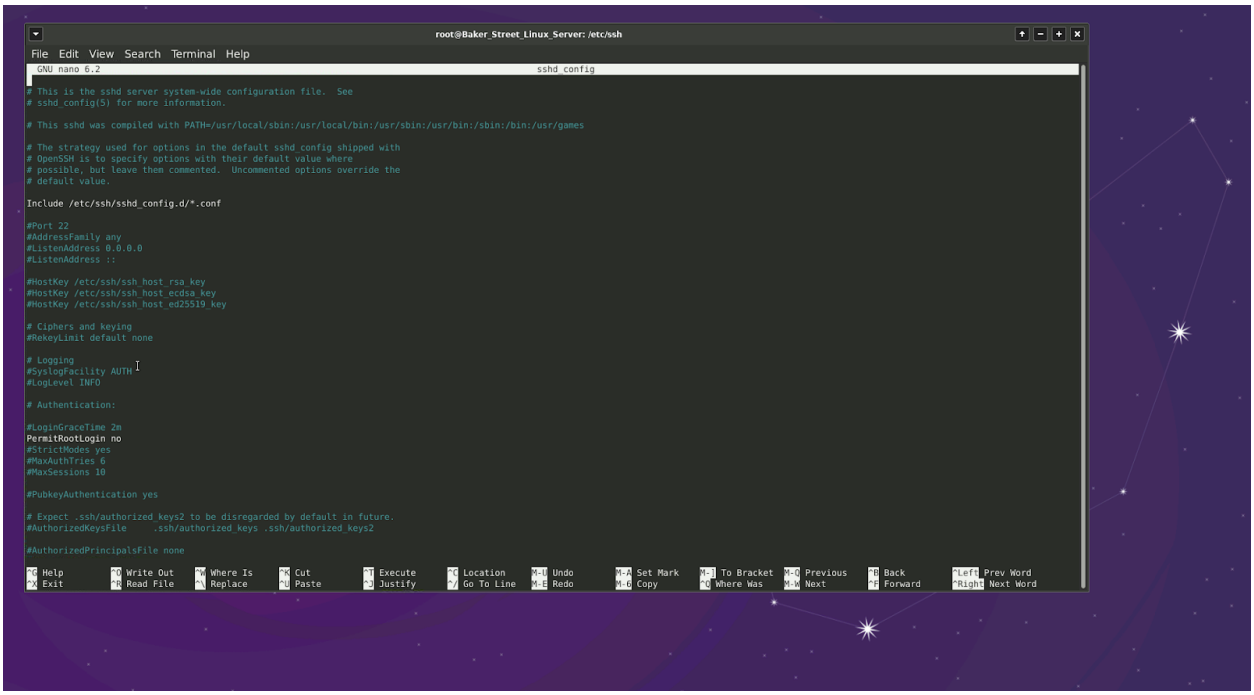
- I disable empty password,
- disable root login,
- enable ssh protocol 2.

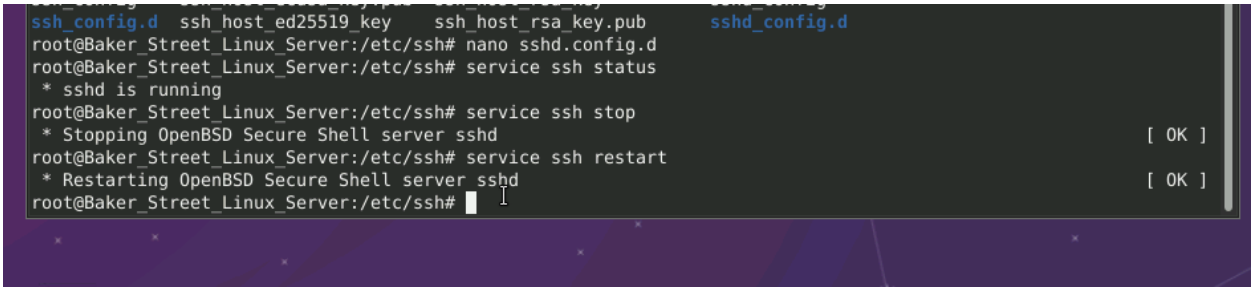
The screenshots will show the work that I did. Once I was done with that, I made sure I saved everything. Then I ran the command service ssh status to restart ssh (last

screenshot).

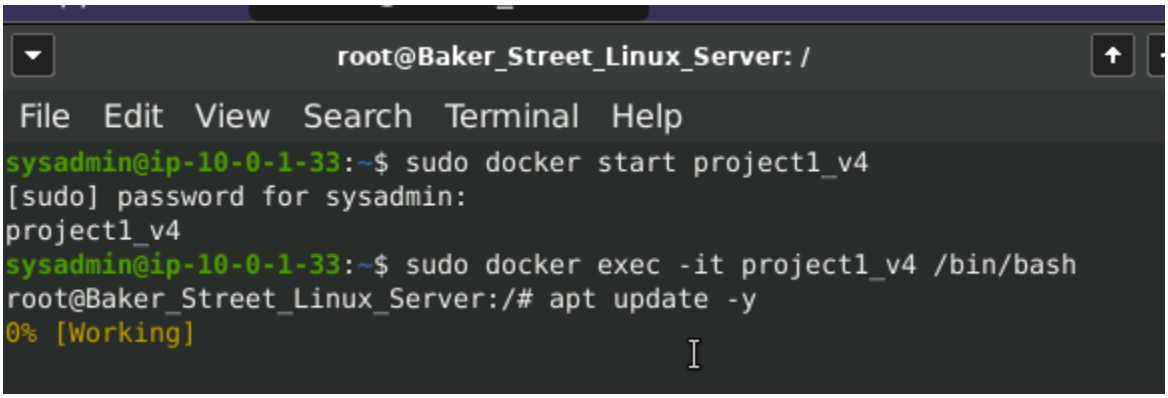

root@Baker_Street_Linux_Server: /etc/ssh
File Edit View Search Terminal Help
GNU nano 6.2 sshd_config
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment 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 anoncvs
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
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back ^Left Prev Word
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^O Where Was M-W Next ^F Forward ^Right Next Word


root@Baker_Street_Linux_Server: /etc/ssh
File Edit View Search Terminal Help
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
#IgnoreUserKnownHosts 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
#KerberosGetAFSToken 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 just want the PAM account and session checks to run without
PAM authentication, then enable this but set PasswordAuthentication
and KbdInteractiveAuthentication to 'no'.
UsePAM yes
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back ^Left Prev Word
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^O Where Was M-W Next ^F Forward ^Right Next Word

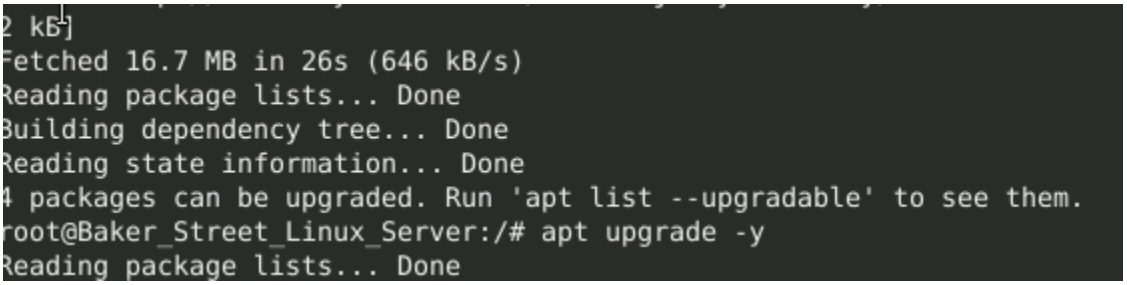




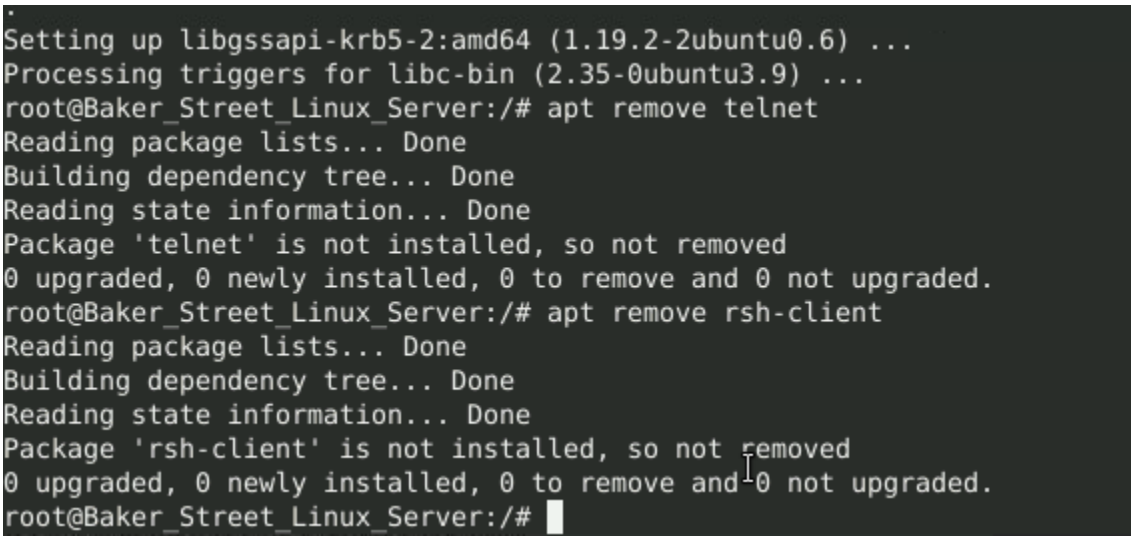
- I ran the command apt update to make sure it has the version of all packages.



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

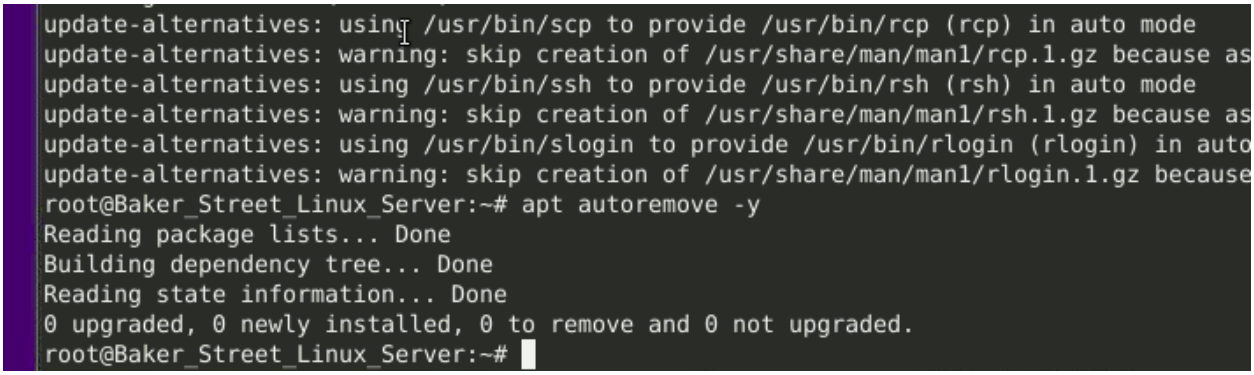


- I then removed the telnet package and rsh-client package.

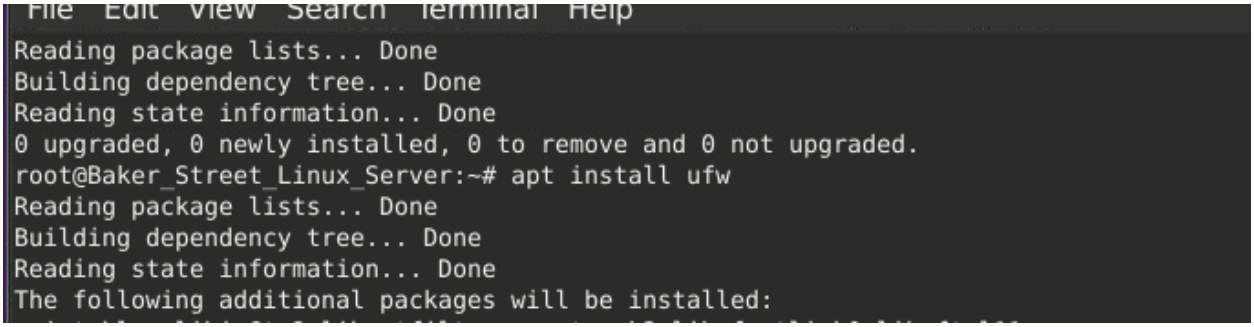


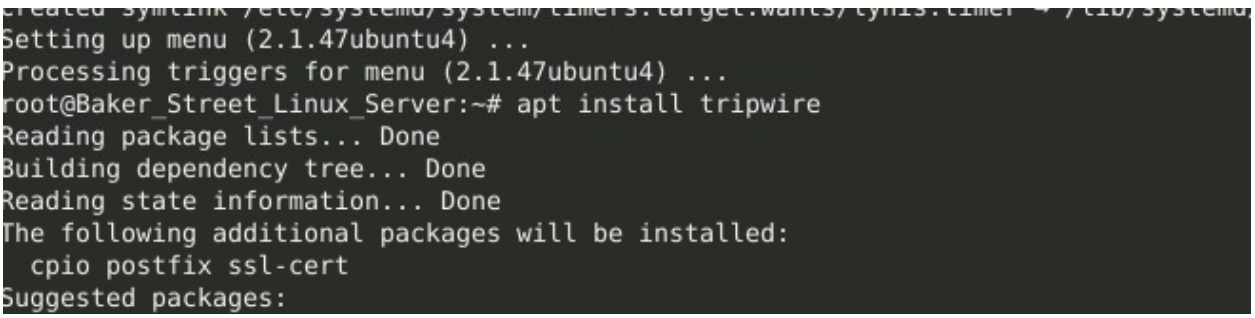
- I did some online research to find out why telnet and rsh-client needed to be removed. The reason why we removed telnet was because any username and password can be easily intercepted by hackers or attacks. Rsh-client was removed because of unencrypted information over the network, which makes us vulnerable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

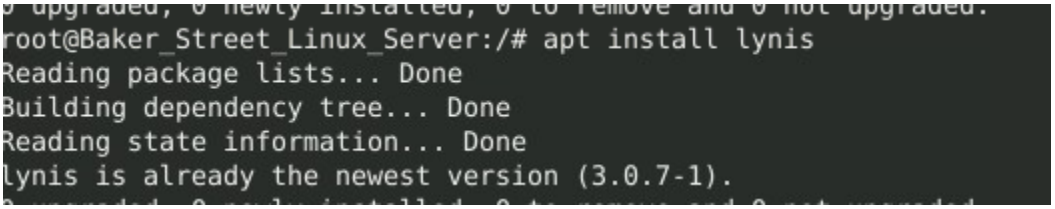
- I ran the command apt autoremove –y to clean up dependencies, remove disk space, and remove old files that are no longer needed to be on the system.



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

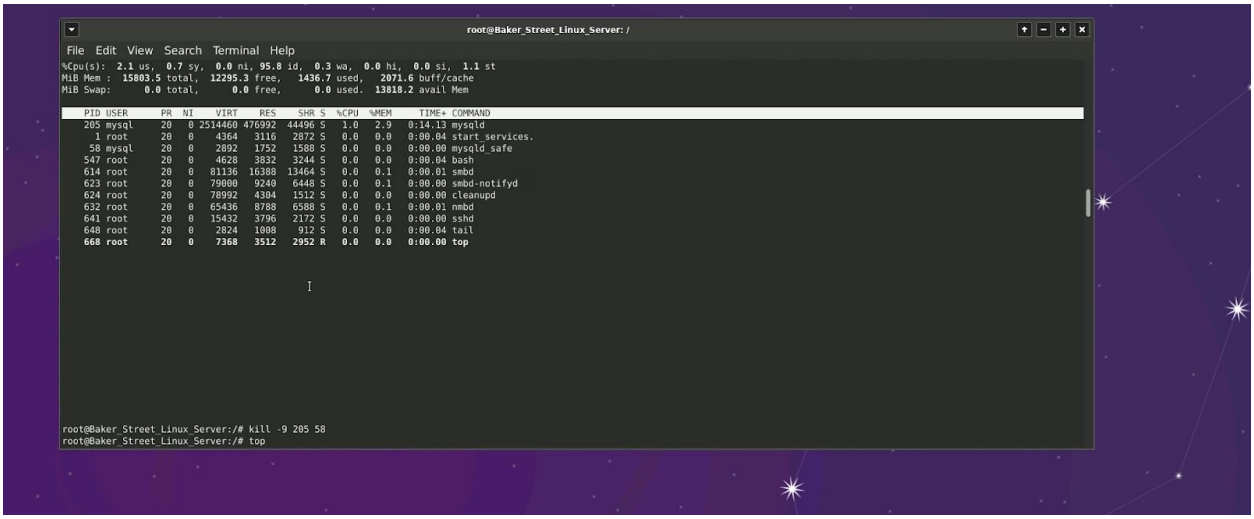




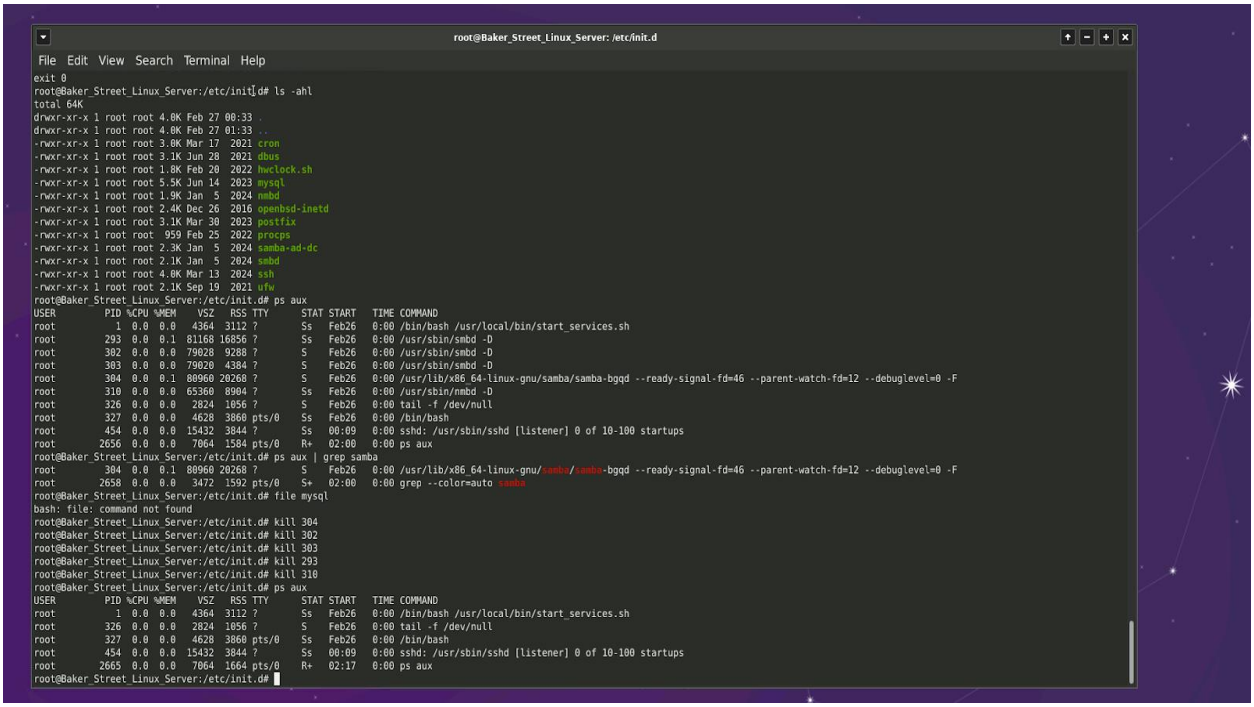


- UFW can block incoming traffic, deny and limit traffic for firewall rules and can log network traffic to help detect and analyze attacks. Lynis scans the system and checks for vulnerabilities within the system. Tripwire has multiple functions of monitoring and unauthorized changes.

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.

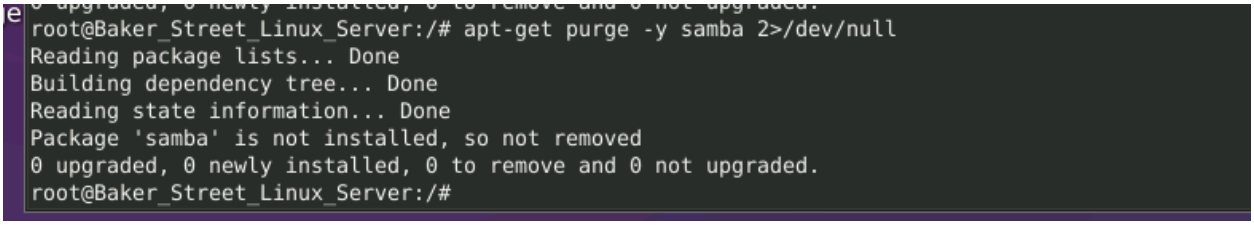


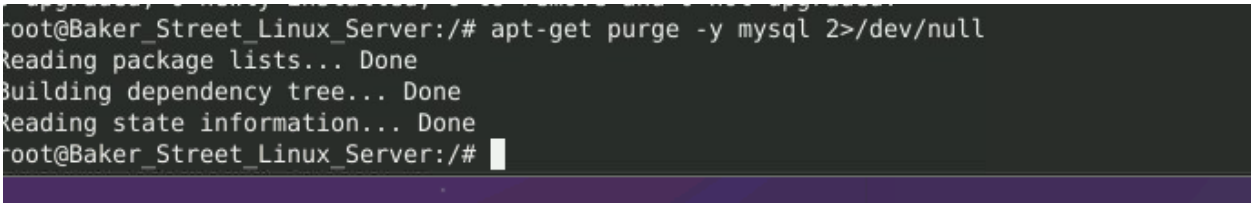
- I also ran the ps aux command and killed PID numbers 304,302,303,293,310.

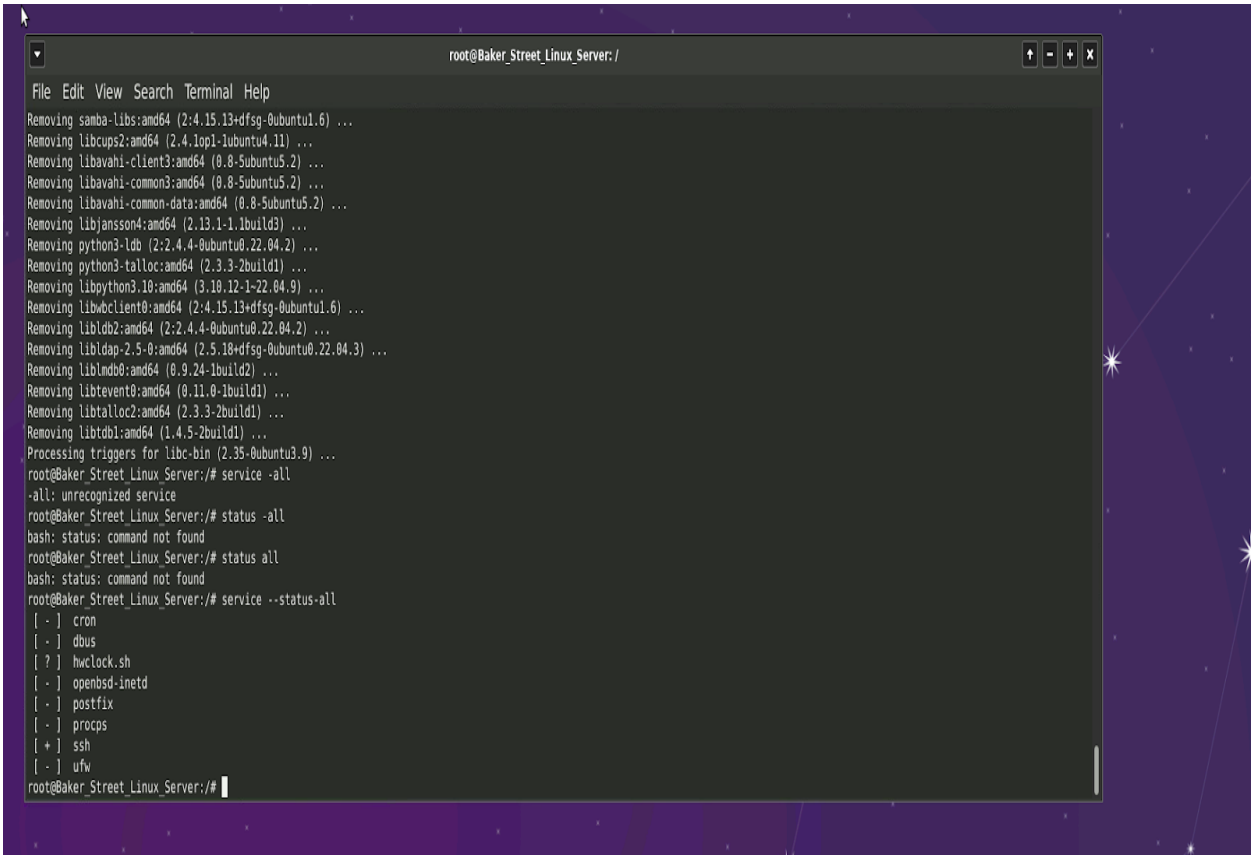


I then did some research because we couldn't use the systemctl to remove mysql and samba, so we needed to use the service command to remove it.

- 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.

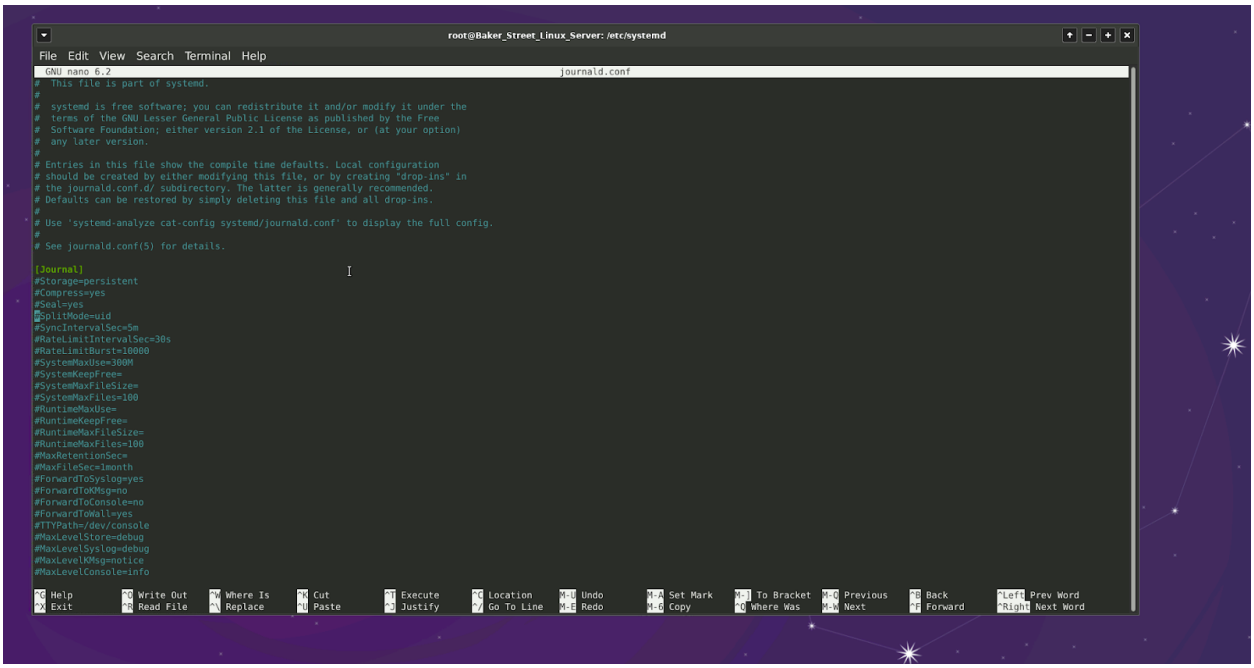




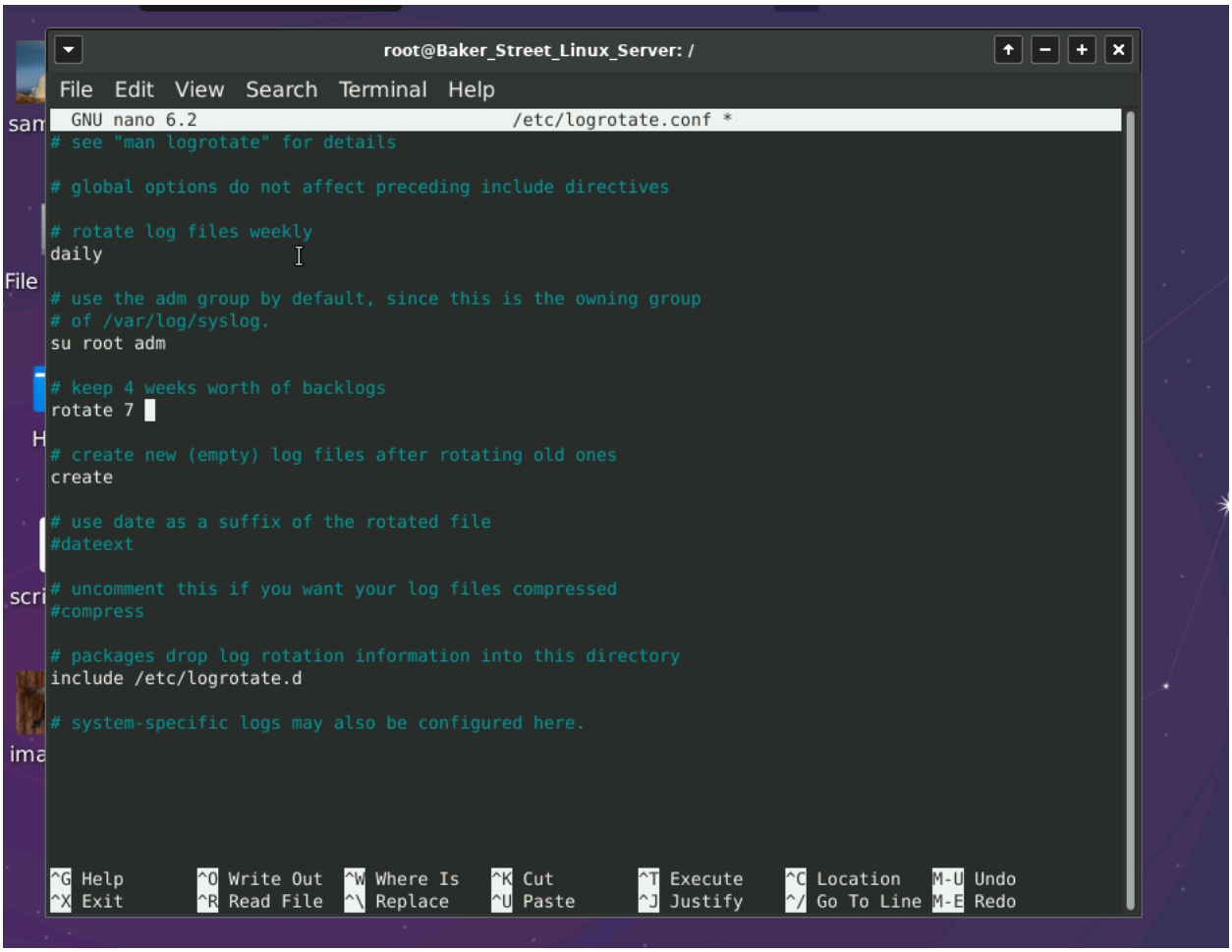


I ran the command nano /etc/systemd/journald.conf to make changes.

- 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.



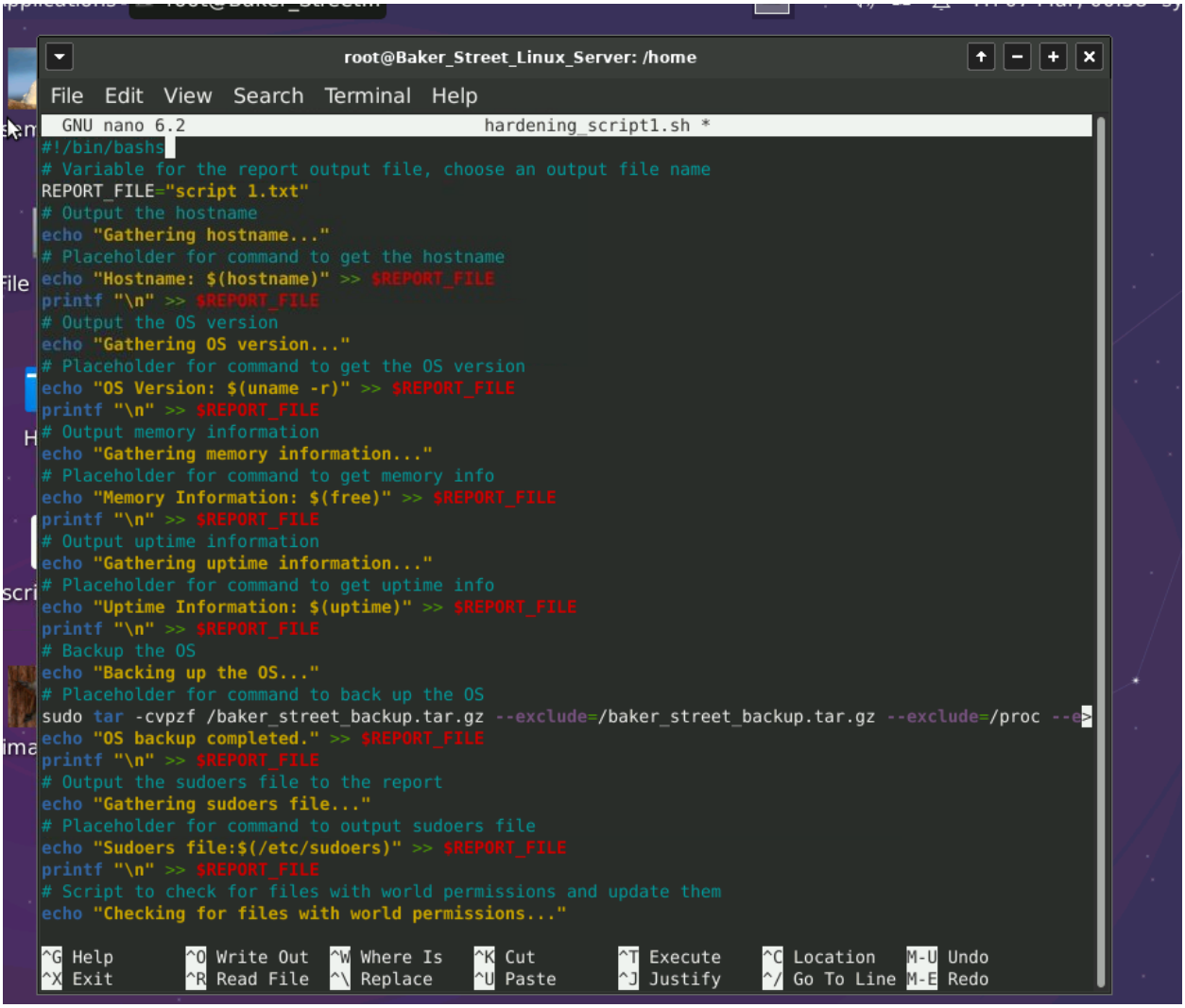
- Next, I ran the command nano /etc/logrotate.conf. The changes I made were changing the log rotation from weekly **to daily** and rotating out the **logs after 7 days.**

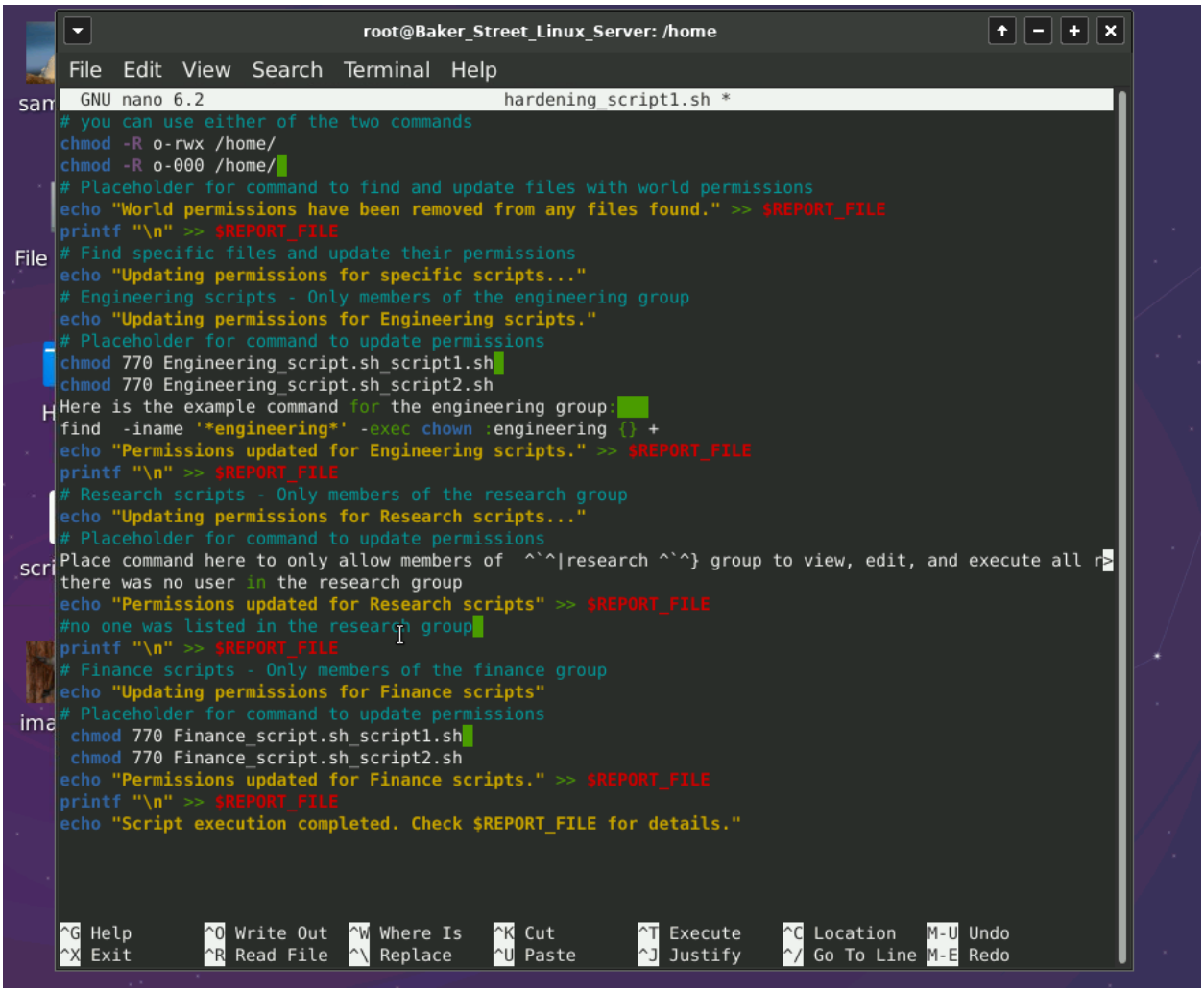


I created a script located in the home directory that was named hardening_script1.sh into the home directory. The changes I made are listed below.

- 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.

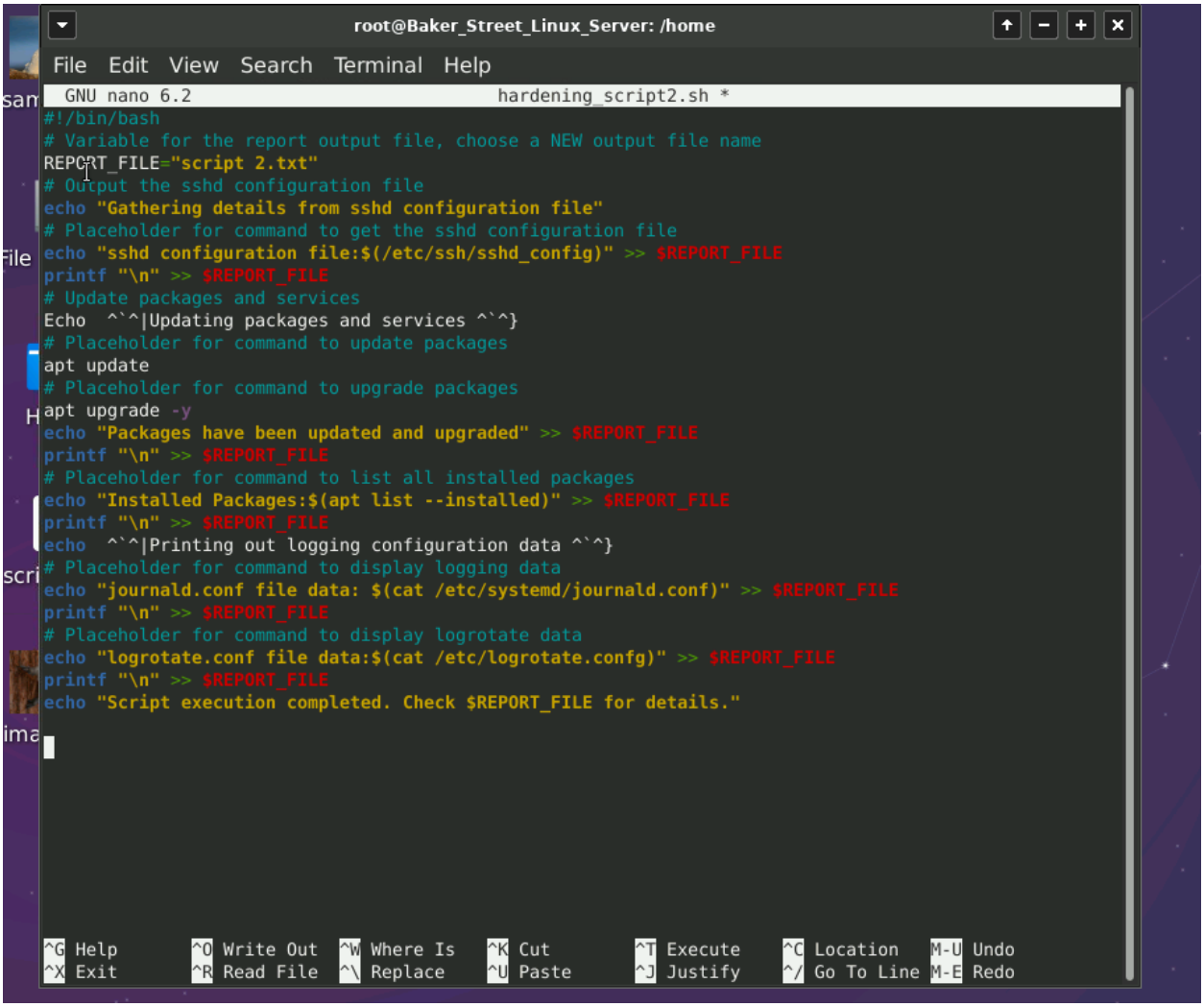


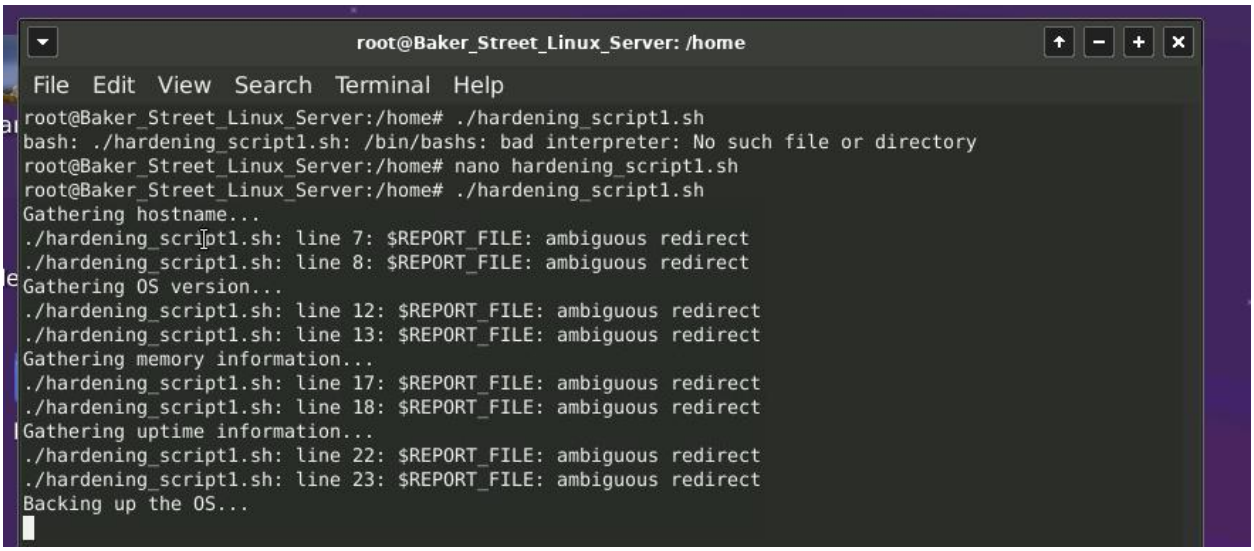


For the second script, the changes I made were

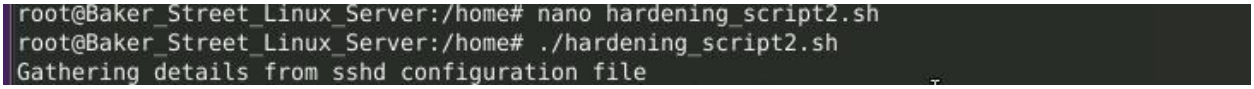
- Report_ file, which I named it as script2.txt
- I showed the command I used for the sshd configuration files, which was
    - /etc/ssh/sshd_config
- I showed the command I used to update packages, which was
    - Apt update
- Showed the command to upgrade packages which was
    - Apt upgrade -y
- The 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.confg
        - cat /etc/systemd/journal.conf

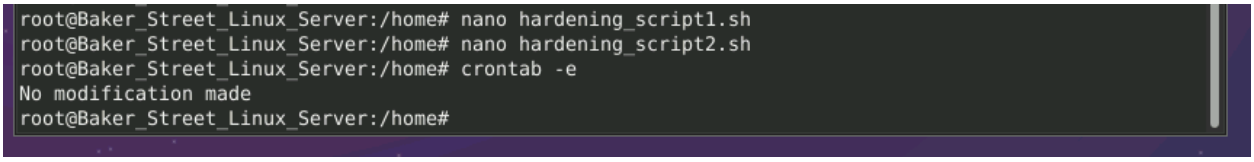


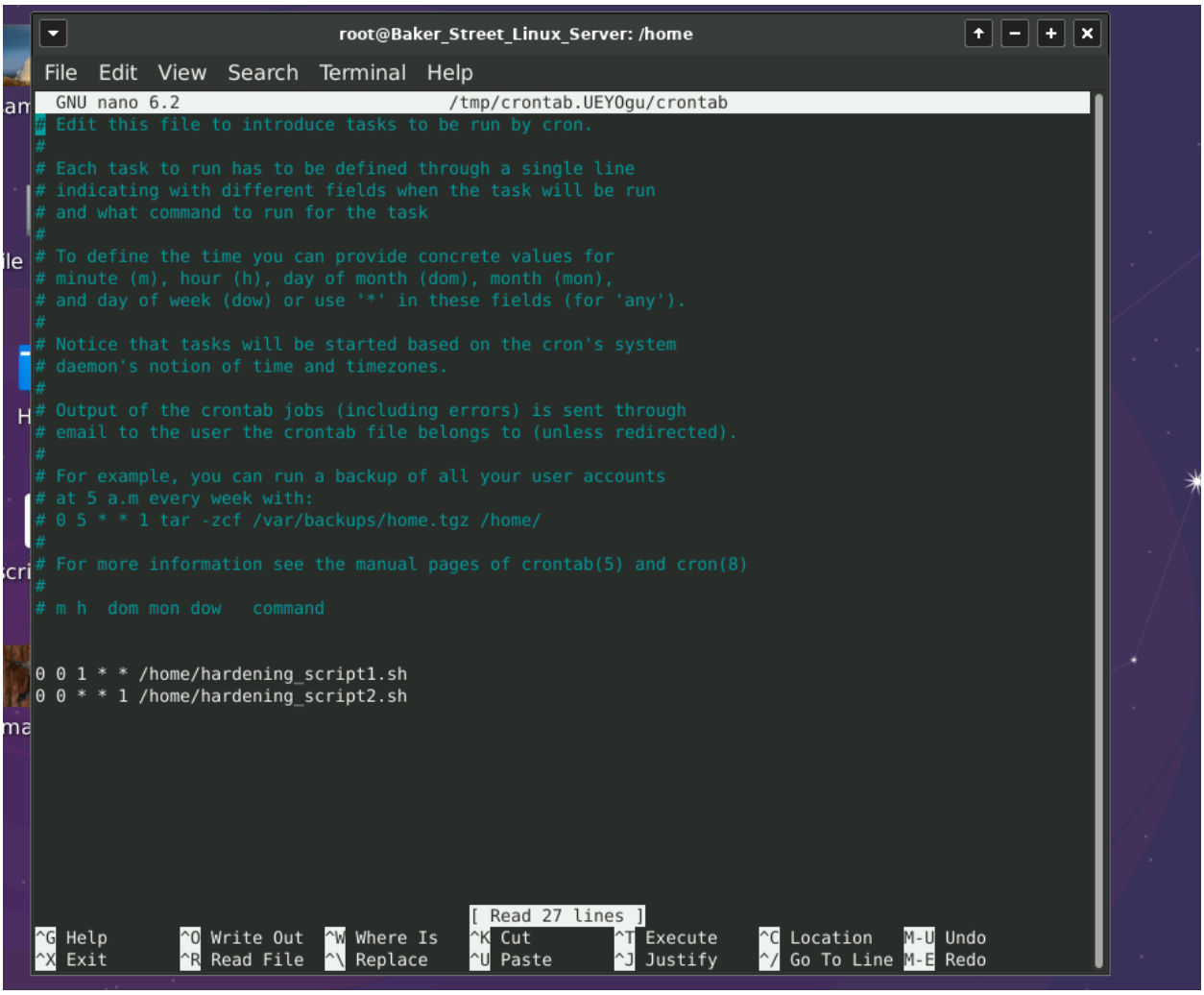


I then ran the command ./hardening_script1.sh and then I ran ./hardening_script2.sh to make sure the scripts ran properly.



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





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.