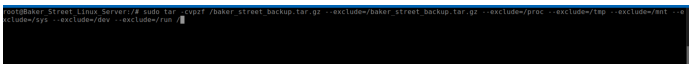
Cybersecurity

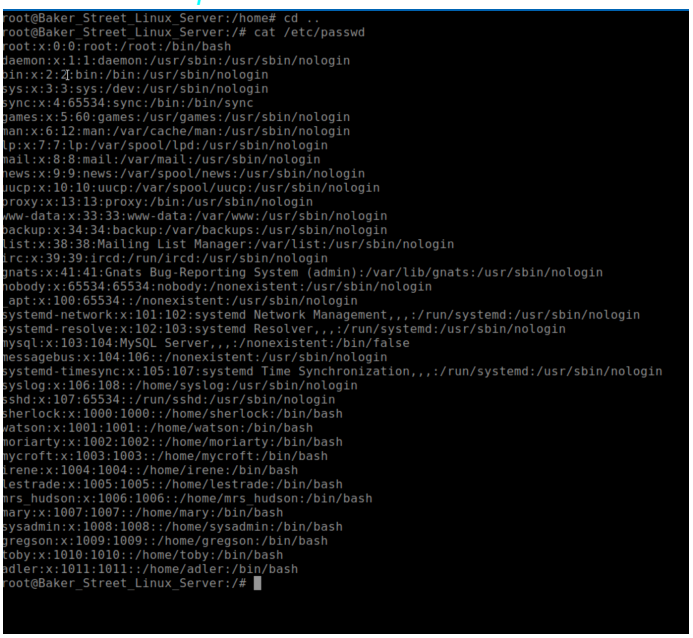
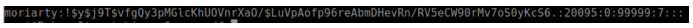
Project 1 Hardening Summary and Checklist

OS Information

Customer	Baker Street Corporation
Hostname	**Baker_Street_Linux_Server**
OS Version	**Ubuntu 22.04 22.04.5 LTS (Jammy Jellyfish) debian**
Memory information	**3.7 Gi total, 744 mb used, 421 mb free, 27Mi shared**
Uptime information	**03:02:36 up 43 min, 0 users**

Checklist

Completed	Activity	Script(s) used / Tasks completed / Screenshots
☑	OS backup	○ *sudo tar -cvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run /* Easy enough, made a .tar with no complications. 

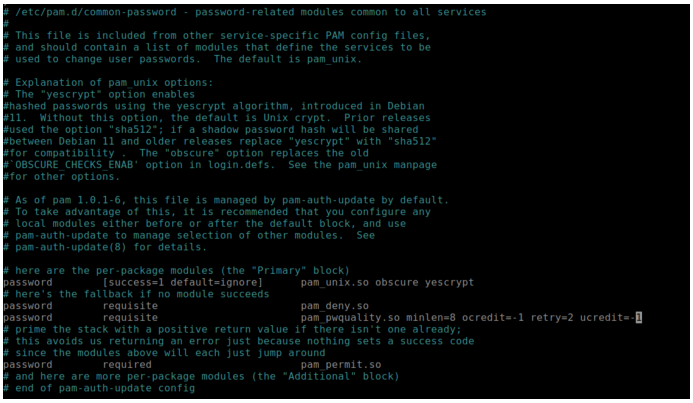
☑	Auditing users and groups	• *cat /etc/passwd*  • *sudo userdel -r (lestrade. Irene, mary, gregson)* This permanently deletes the user AND their home directories for all terminated employees. • *sudo usermod -L (moriarty and mrs hudson)* This locks user accounts, you can go back and verify if a user has been locked or not by using *sudo grep (username) /etc/shadow* If they are locked it will show an ! before their name in the shadow file.  • Employees Toby and Adler were locked so I used *sudo usermod -p LOGIN (toby, adler)* to unlock them and give them the temp password until they would set a new one. 

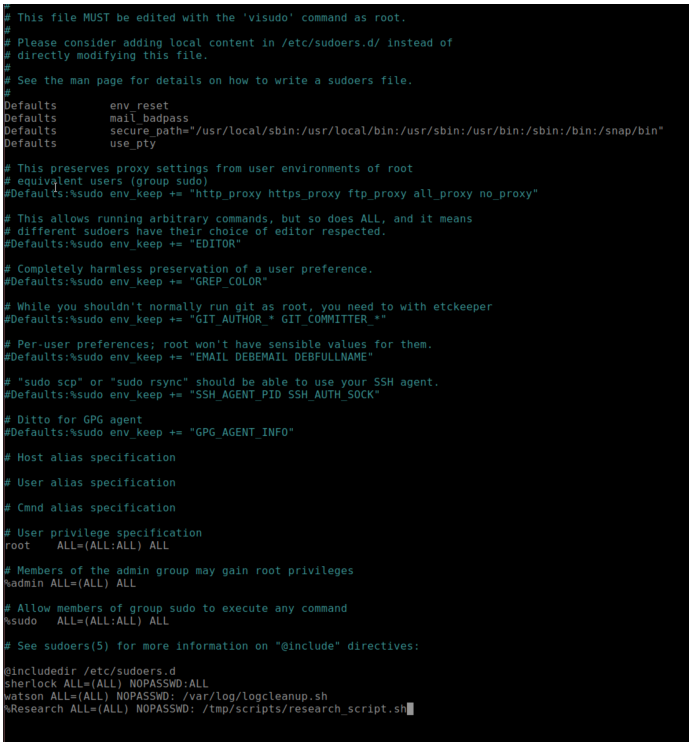
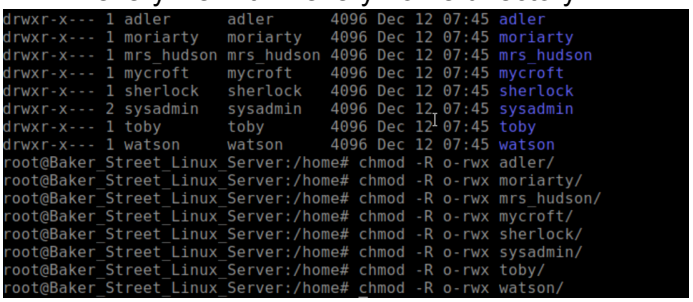
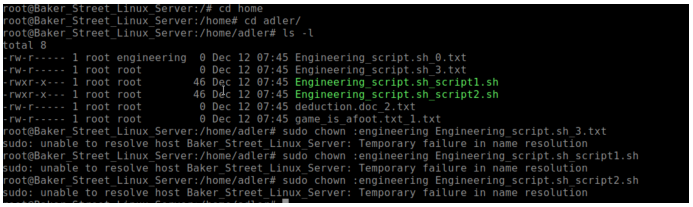
- I used *getent group marketing* to view everyone in the marketing group, however there were already no employees within that group, so to demonstrate knowledge I added a handful of employees to a new group called research and removed the old marketing group.

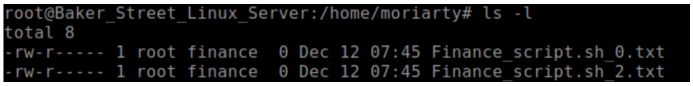
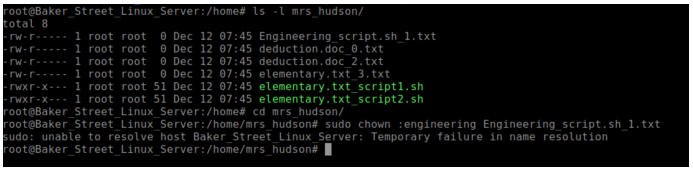
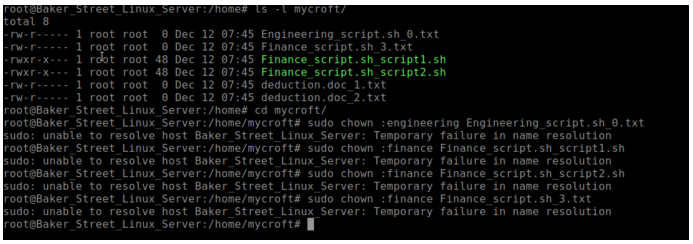
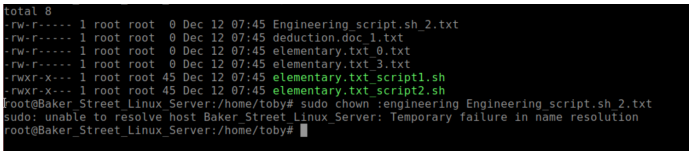
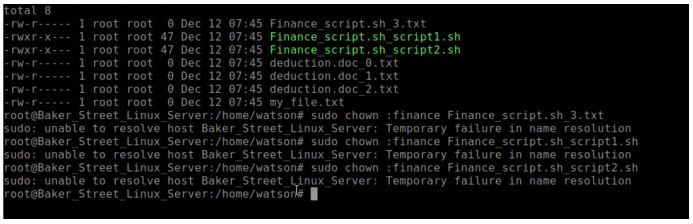
```
root@Baker_Street_Linux_Server:/home# getent group marketing
marketing:x:1014:
```

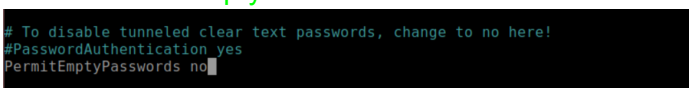
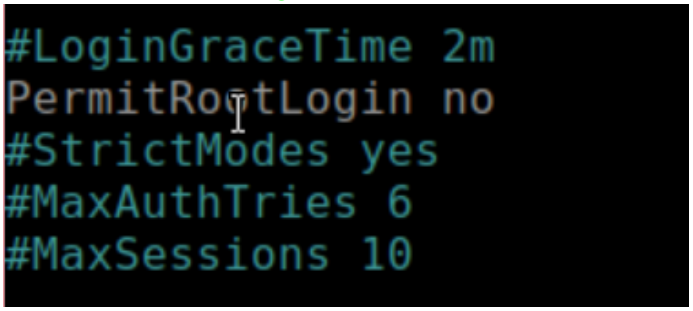
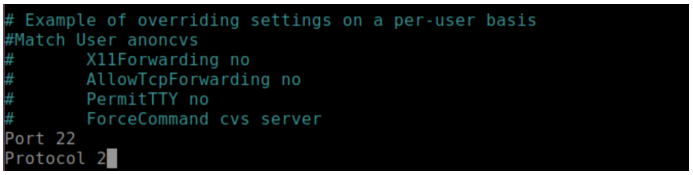
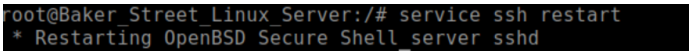
- For deletion of marketing I used *sudo delgroup marketing*, and for adding a new group called Research I used *sudo groupadd Research*, and to move users into it I used *sudo usermod -aG (username) Research*

```
root@Baker_Street_Linux_Server:/# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
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
Research:x:1014:toby,watson,mycroft
```

☑	Updating and enforcing password policies	• Fairly simple step, first just have to use *nano /etc/pam.d/common-password* followed by adding the *password requisite pam_pwquality.so* to the common-password file, followed by *minlen=8 ocredit=-1 retry=2 ucredit=-1* • Finish it off by saving with *CTRL+O* then closing with *CTRL+X* 
☑	Updating and enforcing sudo permissions	• To access the sudoers file all I had to do was run the *visudo* and once in I noticed some things out of place. Sherlock was correct and had full privilege, but Watson is not supposed to have full access, simply sudo to run the script /var/log/logcleanup.sh, so I changed the *N0PASSWD=ALL* to *N0PASSWD=* /var/log/logcleanup.sh. • Moriarty was full privilege, so for him it was just a matter of removing him off the sudoers using *CTRL+K* to remove the full line efficiently. • Finally I added the Research Group to be able to use sudo for /tmp/scripts/research_script.sh by simply adding them to the sudo file using the % to signify it as a group and using the same idea as before, screenshots as followed.

		
☑	Validating and updating permissions on files and directories	• First thing I did was *cd* into */home/* and using *ls -l* list all directories within */home/* , after that I used *chmod -R o-rwx (directory)* to recursively remove all world permissions on every file within every home directory.  • Next I went from directory to directory changing ownership of scripts and files to their respective groups using *sudo chown :(group) (file)* 

		     • One thing I and my group noticed was the lack of any Research files or scripts, plenty of Engineering and Finance but Research. Upon a thorough examination of all the home directories I was not able to find any passwords floating around, even with the *ls -a* and checking within each file, they were mostly empty and if not empty it simply says “this is ____ scripts”
☐	Optional: Updating password hashing configuration	

☑	Auditing and securing SSH	• First to configure the SSH file we need to *sudo nano /etc/ssh/sshd_config* to open the sshd_config file for editing. First thing to harden is the ability to disable empty passwords by changing PermitEmptyPasswords yes to PermitEmptyPasswords no.  • Next I disabled root login, very important of course, by changing PermitRootLogin to PermitRootLogin no.  • I also removed all ports that were NOT port 22, and while I was at it I changed the ssh protocol 1 to protocol 2.  • To fully cement the updates I applied to the SSH file, I ran *service ssh restart*. 
☑	Reviewing and updating system packages	• This was actually the first step I did after getting access to the lab, updating all the packages at least. First I ran *sudo apt update* followed by *apt upgrade -y*.

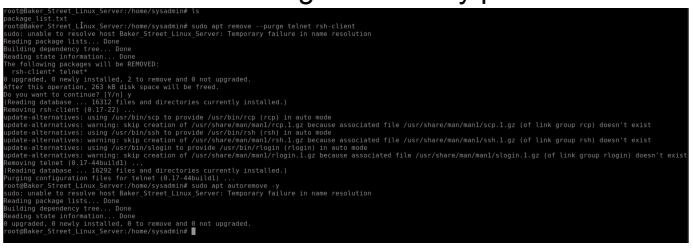
```
root@Baker_Street_Linux_Server:/# sudo apt update
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@Baker_Street_Linux_Server:/# sudo apt upgrade -y
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  libcephfs2 librados2
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 4342 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 librados2 amd64 17.2.7-0ubuntu0.22.04.2 [3594 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcephfs2 amd64 17.2.7-0ubuntu0.22.04.2 [748 kB]
Fetched 4342 kB in 1s (4422 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
(Reading database ... 16312 files and directories currently installed.)
Preparing to unpack .../librados2_17.2.7-0ubuntu0.22.04.2_amd64.deb ...
Unpacking librados2 (17.2.7-0ubuntu0.22.04.2) over (17.2.7-0ubuntu0.22.04.1) ...
Preparing to unpack .../libcephfs2_17.2.7-0ubuntu0.22.04.2_amd64.deb ...
Unpacking libcephfs2 (17.2.7-0ubuntu0.22.04.2) over (17.2.7-0ubuntu0.22.04.1) ...
Setting up librados2 (17.2.7-0ubuntu0.22.04.2) ...
Setting up libcephfs2 (17.2.7-0ubuntu0.22.04.2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
root@Baker_Street_Linux_Server:/#
```

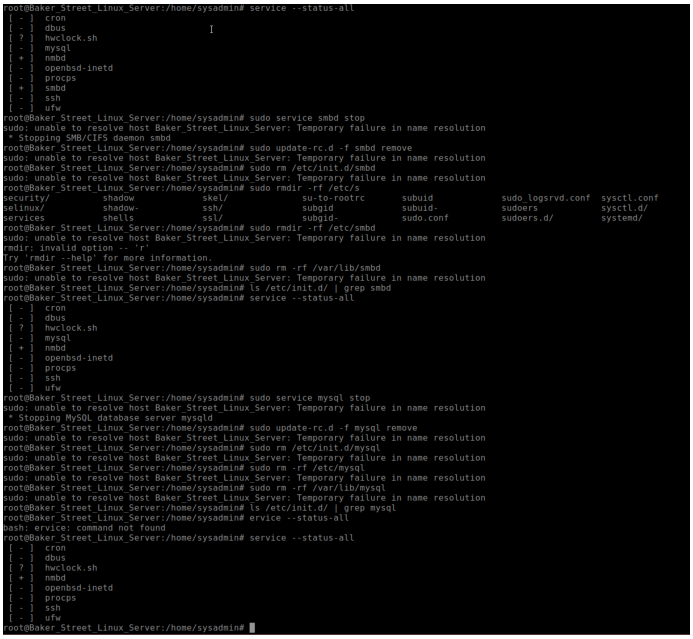
- To keep everything neat and organized within the client's server I used *cd home/sysadmin/* to keep the text file within the sysadmin home directory, again to keep things organized and reduce clutter. I used *apt list --installed > package_list.txt* to list all the installed packages into one text file.

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

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

root@Baker_Street_Linux_Server:/home/sysadmin# ls -l
total 24
-rw-r--r-- 1 root root 22280 Jan  8 03:19 package_list.txt
root@Baker_Street_Linux_Server:/home/sysadmin#
```

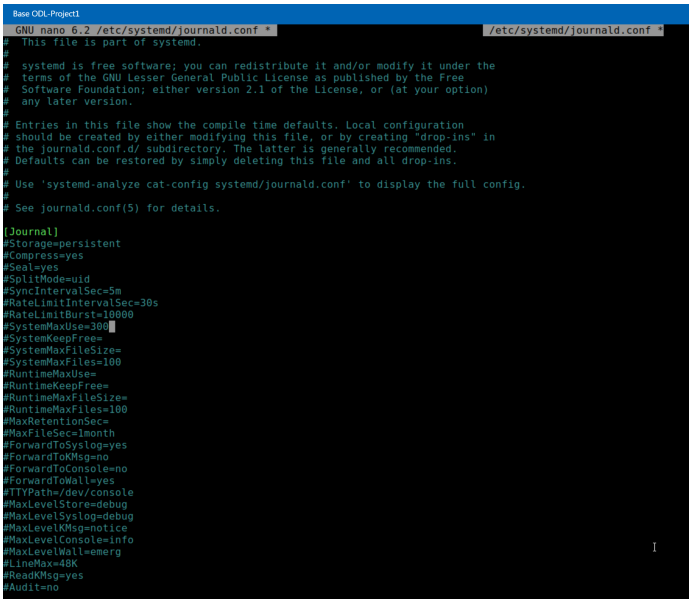
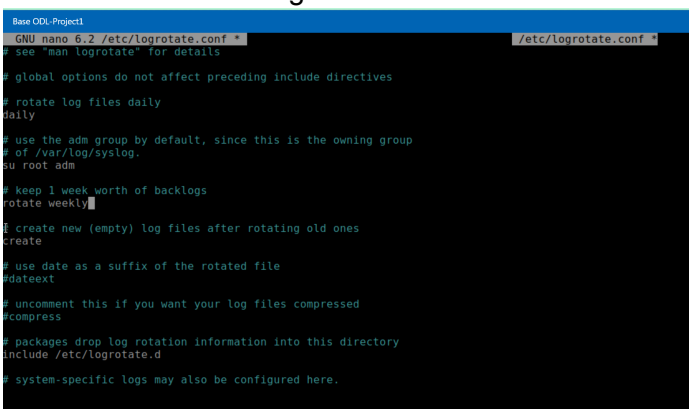
```
rsh-client/jammy,now 0.17-22 amd64 [installed]
rsh-server/jammy,now 0.17-22 amd64 [installed]
rsyslog/jammy-updates,jammy-security,now 8.2112.0-2ubuntu2.2 amd64 [installed]
samba-common-bin/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
samba-common/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 all [installed,automatic]
samba-dsdb-modules/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
samba-libs/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
samba-vfs-modules/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
samba/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed]
sed/jammy,now 4.8-1ubuntu2 amd64 [installed]
sensible-utils/jammy,now 0.0.17 all [installed]
shared-mime-info/jammy,now 2.1-2 amd64 [installed,automatic]
ssh-import-id/jammy,now 5.11-0ubuntu1 all [installed,automatic]
sudo/jammy-updates,jammy-security,now 1.9.9-1ubuntu2.4 amd64 [installed]
systemd-sysv/jammy-updates,now 249.11-0ubuntu3.12 amd64 [installed,automatic]
systemd-timesyncd/jammy-updates,now 249.11-0ubuntu3.12 amd64 [installed,automatic]
systemd/jammy-updates,now 249.11-0ubuntu3.12 amd64 [installed,automatic]
sysvinit-utils/jammy,now 3.01-1ubuntu1 amd64 [installed]
tar/jammy-updates,jammy-security,now 1.34+dfsg-1ubuntu0.1.22.04.2 amd64 [installed]
tcpd/jammy,now 7.6.q-31build2 amd64 [installed,automatic]
tdb-tools/jammy,now 1.4.5-2build1 amd64 [installed,automatic]
telnet/jammy,now 0.17-44build1 amd64 [installed]
tree/jammy,now 2.0.2-1 amd64 [installed]
ubuntu-keyring/jammy,now 2021.03.26 all [installed]
ucf/jammy,now 3.0043 all [installed,automatic]
update-inetd/jammy,now 4.51 all [installed,automatic]
usrmerge/jammy,now 25ubuntu2 all [installed]
util-linux/jammy-updates,jammy-security,now 2.37.2-4ubuntu3.4 amd64 [installed]
wget/jammy-updates,jammy-security,now 1.21.2-2ubuntu1.1 amd64 [installed,automatic]
xauth/jammy,now 1:1.1-1build2 amd64 [installed,automatic]
xdg-user-dirs/jammy,now 0.17-2ubuntu4 amd64 [installed,automatic]
zlib1g/jammy-updates,jammy-security,now 1:1.2.11.dfsg-2ubuntu9.2 amd64 [installed]
root@Baker_Street_Linux_Server:/home/sysadmin#
```

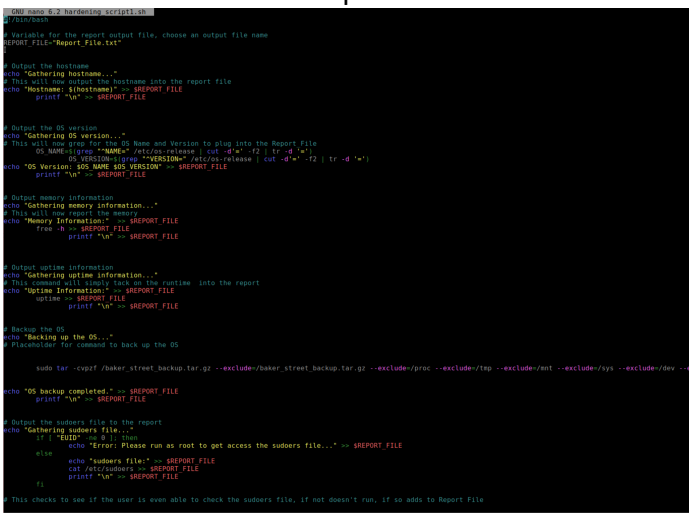
- As you can see, from reading the created text file both telnet and rsh-client were installed within the server, both of which can provide security issues.The reason why telnet can be a security liability is first and foremost it is outdated and obsolete, it is vulnerable to Man-in-the-middle attacks, and has a fundamental lack of encryption. The reason why rsh-client is unsecure is all the aforementioned reasons with telnet, but

		along with those it is also vulnerable to IP spoofing because it uses a trust based connection, which assumes they cannot be spoofed, which is incorrect. ○ These needed to be removed ASAP so I used *sudo apt remove --purge telnet rsh-client* to fully remove those packages from the system, filled by *apt autoremove -y* to remove all dependencies from those packages for good. I updated the *package_list.txt* to reflect the changes, and while I was at it I changed world rwx perms to none, as I don't feel like having that open to the world is good security practice.  ● I then went in and used *sudo apt install ufw*, *sudo apt install lynis*, and *sudo apt install tripwire*. I added those packages to the system, I updated the .txt file once more and researched what those add to the system security wise, which you can read down below.
☑	Disabling unnecessary services	● First I have to determine what services are running, and if any have to be removed. What I ended up using (since systemctl isn't installed) is *service --status-all > service_list.txt* to put all services into a file to view, again doing this in the sysadmin directory. 

		• To stop a service is easy all you have to do is update first using *sudo update-rc.d -f <service-name> remove* but again, since there is no systemctl it is much more of a process to remove a service. What I did is as follows. 1. *sudo service (service name) stop* This is to simply stop the process. 2. *sudo update-rc.d -f (service-name) remove* This is a basic removal, but if it is just this process still lingering on the system. 3. *sudo rm /etc/init.d/(service-name)* Steps 3-5 simply removes the rest of the service off the system permanently. 4. *sudo rm -rf /etc/(service-name)* See step 3 5. *sudo rm -rf /var/lib/(service-name)* See step 3 6. *ls /etc/init.d/ | grep (service-name)* . This is to verify the services have been purged off the system. • I then did what I did before, updated the .txt file, and removed the rwx for world. 

```
root@Baker_Street_Linux_Server:/home/sysadmin# service --status-all
 [ - ]  cron
 [ - ]  dbus
 [ ? ]  hwclock.sh
 [ - ]  mysql
 [ + ]  nmbd
 [ - ]  openbsd-inetd
 [ - ]  procps
 [ + ]  smbd
 [ - ]  ssh
 [ - ]  ufw
root@Baker_Street_Linux_Server:/home/sysadmin# sudo service smbd stop
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
 * Stopping SMB/CIFS daemon smbd
root@Baker_Street_Linux_Server:/home/sysadmin# sudo update-rc.d -f smbd remove
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
root@Baker_Street_Linux_Server:/home/sysadmin# sudo rm /etc/init.d/smbd
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
root@Baker_Street_Linux_Server:/home/sysadmin# sudo rmdir -rf /etc/s
security/          shadow             skel/              su-to-rootrc       subuid             sudo_logsrvd.conf  sysctl.conf
selinux/           shadow-            ssh/               subgid             subuid-            sudoers            sysctl.d/
services           shells             ssl/               subgid-            sudo.conf          sudoers.d/         systemd/
root@Baker_Street_Linux_Server:/home/sysadmin# sudo rmdir -rf /etc/smbd
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
rmdir: invalid option -- 'r'
Try 'rmdir --help' for more information.
root@Baker_Street_Linux_Server:/home/sysadmin# sudo rm -rf /var/lib/smbd
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
root@Baker_Street_Linux_Server:/home/sysadmin# ls /etc/init.d/ | grep smbd
root@Baker_Street_Linux_Server:/home/sysadmin# service --status-all
 [ - ]  cron
 [ - ]  dbus
 [ ? ]  hwclock.sh
 [ - ]  mysql
 [ + ]  nmbd
 [ - ]  openbsd-inetd
 [ - ]  procps
 [ - ]  ssh
 [ - ]  ufw
root@Baker_Street_Linux_Server:/home/sysadmin# sudo service mysql stop
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
 * Stopping MySQL database server mysqld
root@Baker_Street_Linux_Server:/home/sysadmin# sudo update-rc.d -f mysql remove
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
root@Baker_Street_Linux_Server:/home/sysadmin# sudo rm /etc/init.d/mysql
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
root@Baker_Street_Linux_Server:/home/sysadmin# sudo rm -rf /etc/mysql
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
root@Baker_Street_Linux_Server:/home/sysadmin# sudo rm -rf /var/lib/mysql
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
root@Baker_Street_Linux_Server:/home/sysadmin# ls /etc/init.d/ | grep mysql
root@Baker_Street_Linux_Server:/home/sysadmin# ervice --status-all
bash: ervice: command not found
root@Baker_Street_Linux_Server:/home/sysadmin# service --status-all
 [ - ]  cron
 [ - ]  dbus
 [ ? ]  hwclock.sh
 [ + ]  nmbd
 [ - ]  openbsd-inetd
 [ - ]  procps
 [ - ]  ssh
 [ - ]  ufw
root@Baker_Street_Linux_Server:/home/sysadmin#
```

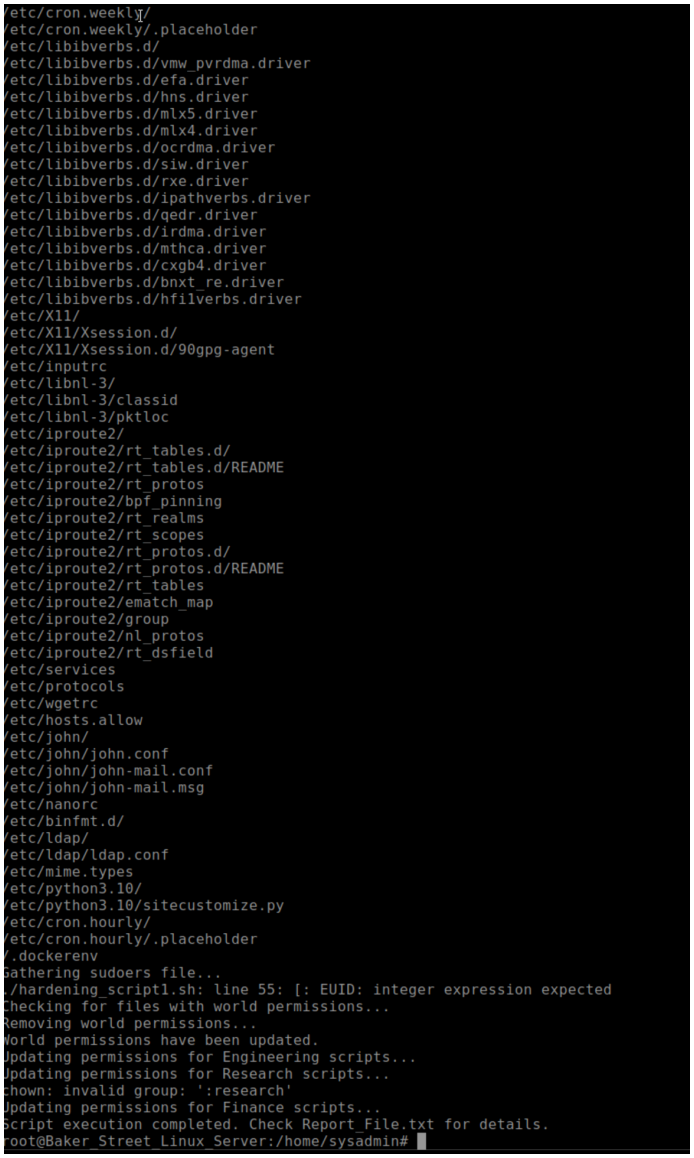
☑	Enabling and configuring logging	• To harden this part of the linux server, you have to use *sudo nano /etc/systemd/jounald.conf* • This brings you inside the file to edit, where you have to find where it says storage and change it to storage=persistent and do the same to systemMaxUse= and change it to systemMaxUse=300  • Next do the same to access logrotate.conf ○ Use *sudo nano /etc/logrotate.conf* ○ Once in, change the log rotation from weekly to daily and rotate out the logs after 7 days. This can be done in the following screenshot. 

☑	Scripts created	Firstly I will be posting the full scripts underneath this checklist, but will supply the screenshots within the VM. Not a lot to say here despite being the longest part, essentially just a recap of everything above combined with trial and error until my script came out flawlessly. Script 1   Script 2

```
  GNU nano 6.2 hardening_script2.sh
#!/bin/bash


# Variable for the report output file, choose a NEW output file name
REPORT_FILE="Report_File_2.txt"



# Ensure the output file is writable or create it
    if ! touch "$REPORT_FILE" 2>/dev/null; then
            echo "Error: Cannot write to $REPORT_FILE. Check permissions."
        exit 1
 fi

# Output the sshd configuration file
echo "Gathering details from sshd configuration file..."
        if [ -f /etc/ssh/sshd_config ]; then
                echo "sshd configuration file:" >> $REPORT_FILE
          cat /etc/ssh/sshd_config >> $REPORT_FILE
                printf "\n" >> $REPORT_FILE
              else
echo "Error: sshd configuration file not found at /etc/ssh/sshd_config." >> $REPORT_FILE
        printf "\n" >> $REPORT_FILE
fi

echo "Report saved to $REPORT_FILE"




# Update packages and services
echo "Updating packages and services..."
 echo "Updating package lists..." >> $REPORT_FILE
# Update package lists
if sudo apt update >> "$REPORT_FILE" 2>&1; then
        echo "Package lists updated successfully." >> $REPORT_FILE
         else
        echo "Failed to update package lists. Check the log above." >> $REPORT_FILE
 fi

        printf "\n" >> "$REPORT_FILE"
# Upgrade packages
echo "Upgrading packages..." >> $REPORT_FILE
 if sudo apt upgrade -y >> "$REPORT_FILE" 2>&1; then
        echo "Packages upgraded successfully." >> $REPORT_FILE
         else
    echo "Failed to upgrade packages. Check the log above." >> $REPORT_FILE
fi
        printf "\n" >> $REPORT_FILE
echo "Packages have been updated and upgraded."

# List all installed packages
echo "Listing all installed packages..." >> $REPORT_FILE
        dpkg --get-selections >> $REPORT_FILE
echo "Installed packages listed successfully." >> $REPORT_FILE

#Print out config data
echo "Printing out logging configuration data"

echo "journald.conf file data" >> $REPORT_FILE
        cat /etc/systemd/journald.conf >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Print out logrotate
echo "Printing out logrotate data"
```

```
#Print out config data
echo "Printing out logging configuration data"

echo "journald.conf file data" >> $REPORT_FILE
        cat /etc/systemd/journald.conf >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Print out logrotate
echo "Printing out logrotate data"
        cat /etc/logrotate.conf >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Last echo if all went through
echo "Script execution completed. Check $REPORT_FILE for details."
```

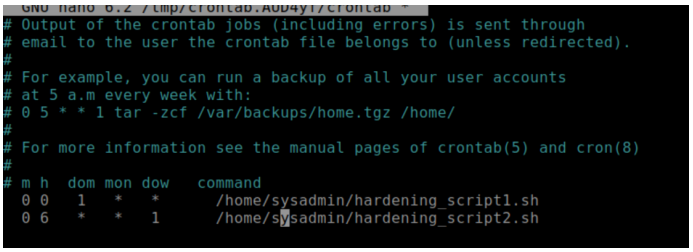
Results


/etc/cron.weekly/
/etc/cron.weekly/.placeholder
/etc/libibverbs.d/
/etc/libibverbs.d/vmw_pvrdma.driver
/etc/libibverbs.d/efa.driver
/etc/libibverbs.d/hns.driver
/etc/libibverbs.d/mlx5.driver
/etc/libibverbs.d/mlx4.driver
/etc/libibverbs.d/ocrdma.driver
/etc/libibverbs.d/siw.driver
/etc/libibverbs.d/rxe.driver
/etc/libibverbs.d/ipathverbs.driver
/etc/libibverbs.d/qedr.driver
/etc/libibverbs.d/irdma.driver
/etc/libibverbs.d/mthca.driver
/etc/libibverbs.d/cxgb4.driver
/etc/libibverbs.d/bnxt_re.driver
/etc/libibverbs.d/hfi1verbs.driver
/etc/X11/
/etc/X11/Xsession.d/
/etc/X11/Xsession.d/90gpg-agent
/etc/inputrc
/etc/libnl-3/
/etc/libnl-3/classid
/etc/libnl-3/pktloc
/etc/iproute2/
/etc/iproute2/rt_tables.d/
/etc/iproute2/rt_tables.d/README
/etc/iproute2/rt_protos
/etc/iproute2/bpf_pinning
/etc/iproute2/rt_realms
/etc/iproute2/rt_scopes
/etc/iproute2/rt_protos.d/
/etc/iproute2/rt_protos.d/README
/etc/iproute2/rt_tables
/etc/iproute2/ematch_map
/etc/iproute2/group
/etc/iproute2/nl_protos
/etc/iproute2/rt_dsfield
/etc/services
/etc/protocols
/etc/wgetrc
/etc/hosts.allow
/etc/john/
/etc/john/john.conf
/etc/john/john-mail.conf
/etc/john/john-mail.msg
/etc/nanorc
/etc/binfmt.d/
/etc/ldap/
/etc/ldap/ldap.conf
/etc/mime.types
/etc/python3.10/
/etc/python3.10/sitecustomize.py
/etc/cron.hourly/
/etc/cron.hourly/.placeholder
/.dockerenv
Gathering sudoers file...
./hardening_script1.sh: line 55: [: EUID: integer expression expected
Checking for files with world permissions...
Removing world permissions...
World permissions have been updated.
Updating permissions for Engineering scripts...
Updating permissions for Research scripts...
chown: invalid group: ':research'
Updating permissions for Finance scripts...
Script execution completed. Check Report_File.txt for details.
root@Baker_Street_Linux_Server:/home/sysadmin#

```
root@Baker_Street_Linux_Server:/home/sysadmin# cat Report_File.txt
Hostname: Baker_Street_Linux_Server

OS Version: "Ubuntu" "22.04.5 LTS (Jammy Jellyfish)"

Memory Information:
               total        used        free      shared  buff/cache   available
Mem:           3.7Gi       643Mi       2.4Gi        26Mi       693Mi       2.9Gi
Swap:             0B          0B          0B

Uptime Information:
 21:39:21 up 9 min,  0 users,  load average: 0.02, 0.07, 0.07

OS backup completed.

sudoers file:
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/sn
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "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

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

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

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

```
@includedir /etc/sudoers.d
sherlock ALL=(ALL) NOPASSWD:ALL
watson ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh
%Research ALL=(ALL) NOPASSWD: /tmp/scripts/research_script.sh

Files with world permissions in /home before changes:
   315527      4 -rwxr-xr-x   1 root     root         2040 Jan 10 05:43 /home/sysadmin/hardening_script2.sh
   262150      4 -rw-r--r--   1 root     root         2308 Jan 16 21:40 /home/sysadmin/Report_File.txt
Files with world permissions in /home after changes:

Updating permissions for specific scripts...
Permissions updated for Engineering scripts.

Permissions updated for Research scripts.

Permissions updated for Finance scripts.

root@Baker_Street_Linux_Server:/home/sysadmin#
```

```
root@Baker_Street_Linux_Server:/home/sysadmin# ./hardening_script2.sh
Gathering details from sshd configuration file...
Report saved to Report_File_2.txt
Updating packages and services...
Packages have been updated and upgraded.
Printing out logging configuration data
Printing out logrotate data
Script execution completed. Check Report_File_2.txt for details.
root@Baker_Street_Linux_Server:/home/sysadmin#
```

☑	Scripts scheduled with cron	Another very simple one to close out the project. Simply crontab -e, and set it up so the scripts are run when they need to be run. Screenshot included below. 

SCRIPT 1

```
#!/bin/bash

# Variable for the report output file, choose an output file name
REPORT_FILE="Report_File.txt"


# Output the hostname
echo "Gathering hostname..."
# This will now output the hostname into the report file
echo "Hostname: $(hostname)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE



# Output the OS version
echo "Gathering OS version..."
# This will now grep for the OS Name and Version to plug into the Report_File
OS_NAME=$(grep "^NAME=" /etc/os-release | cut -d'=' -f2 | tr -d '=')
OS_VERSION=$(grep "^VERSION=" /etc/os-release | cut -d'=' -f2 | tr -d '=')
echo "OS Version: $OS_NAME $OS_VERSION" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE


# Output memory information
echo "Gathering memory information..."
# This will now report the memory
echo "Memory Information:"  >> $REPORT_FILE
free -h >> $REPORT_FILE
printf "\n" >> $REPORT_FILE


# Output uptime information
echo "Gathering uptime information..."
# This command will simply tack on the runtime  into the report
echo "Uptime Information:" >> $REPORT_FILE
uptime >> $REPORT_FILE
```

```
printf "\n" >> $REPORT_FILE


# Backup the OS
echo "Backing up the OS..."
# Placeholder for command to back up the OS


tar -cvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc
--exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run /


echo "OS backup completed." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE


# Output the sudoers file to the report
echo "Gathering sudoers file..."
if [ "EUID" -ne 0 ]; then
echo "Error: Please run as root to get access the sudoers file..." >> $REPORT_FILE
else
echo "sudoers file:" >> $REPORT_FILE
cat /etc/sudoers >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
fi

# This checks to see if the user is even able to check the sudoers file, if not doesn't run, if so
adds to Report File


# Script to check for files with world permissions and update them
echo "Checking for files with world permissions..."
echo "Files with world permissions in /home before changes:" >> $REPORT_FILE
find /home -perm /o+rwx -ls >> $REPORT_FILE
# Remove world permissions
echo "Removing world permissions..."
find /home -perm /o+rwx -exec chmod o-rwx {} \;
# Verify changes
echo "Files with world permissions in /home after changes:" >> $REPORT_FILE
find /home -perm /o+rwx -ls >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
echo "World permissions have been updated."
```

```
# Find specific files and update their permissions
echo "Updating permissions for specific scripts..." >> $REPORT_FILE
# Engineering scripts - Only members of the engineering group
echo "Updating permissions for Engineering scripts..."
find / -iname '*engineering*' -exec chown :engineering {} + -exec chmod 770 {} +
echo "Permissions updated for Engineering scripts." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Research scripts - Only members of the research group
echo "Updating permissions for Research scripts..."
find / -iname '*research*' -exec chown :research {} + -exec chmod 770 {} +
echo "Permissions updated for Research scripts." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Finance scripts - Only members of the finance group
echo "Updating permissions for Finance scripts..."
find / -iname '*finance*' -exec chown :finance {} + -exec chmod 770 {} +
echo "Permissions updated for Finance scripts." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
echo "Script execution completed. Check $REPORT_FILE for details."
```

SCRIPT 2

```
#!/bin/bash

# Variable for the report output file, choose a NEW output file name
REPORT_FILE="Report_File_2.txt"

# Ensure the output file is writable or create it
if ! touch "$REPORT_FILE" 2>/dev/null; then
echo "Error: Cannot write to $REPORT_FILE. Check permissions."
exit 1
```

```
fi

# Output the sshd configuration file
echo "Gathering details from sshd configuration file..."
if [ -f /etc/ssh/sshd_config ]; then
echo "sshd configuration file:" >> $REPORT_FILE
cat /etc/ssh/sshd_config >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
else
echo "Error: sshd configuration file not found at /etc/ssh/sshd_config." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
fi

echo "Report saved to $REPORT_FILE"



# Update packages and services
echo "Updating packages and services..."
echo "Updating package lists..." >> $REPORT_FILE
# Update package lists
if sudo apt update >> "$REPORT_FILE" 2>&1; then
echo "Package lists updated successfully." >> $REPORT_FILE
else
echo "Failed to update package lists. Check the log above." >> $REPORT_FILE
fi

printf "\n" >> "$REPORT_FILE"
# Upgrade packages
echo "Upgrading packages..." >> $REPORT_FILE
if sudo apt upgrade -y >> "$REPORT_FILE" 2>&1; then
echo "Packages upgraded successfully." >> $REPORT_FILE
else
echo "Failed to upgrade packages. Check the log above." >> $REPORT_FILE
fi
printf "\n" >> $REPORT_FILE
echo "Packages have been updated and upgraded."

# List all installed packages
echo "Listing all installed packages..." >> $REPORT_FILE
dpkg --get-selections >> $REPORT_FILE
echo "Installed packages listed successfully." >> $REPORT_FILE
```

```
#Print out config data
echo "Printing out logging configuration data"

echo "journald.conf file data" >> $REPORT_FILE
cat /etc/systemd/journald.conf >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Print out logrotate
echo "Printing out logrotate data"
cat /etc/logrotate.conf >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Last echo if all went through
echo "Script execution completed. Check $REPORT_FILE for details."
```