Cybersecurity

Project 1 Hardening Summary and Checklist

Group: Matteo Verzi, Ido Edery, James Hollingsworth

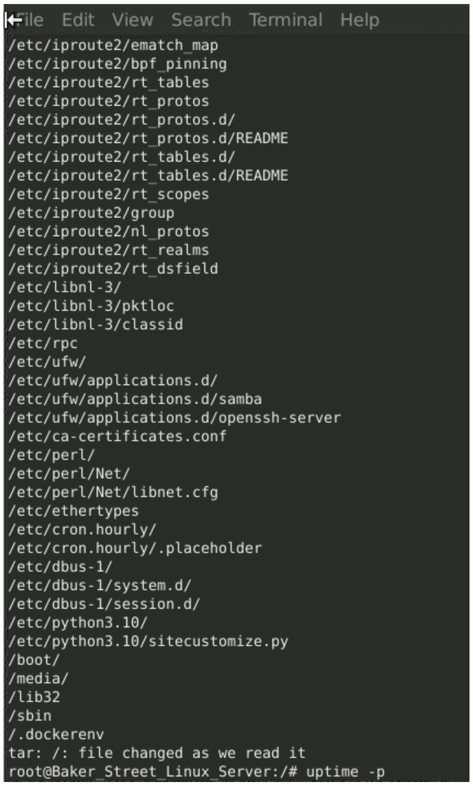
OS Information

Customer	Baker Street Corporation
Hostname	***Baker_Street_Linux_Server***
OS Version	**Ubuntu 22.04.5 LTS**
Memory information	**15Gi / 16.1Gb**
Uptime information	**1 hour 58 minutes**

Checklist

Completed	**Activity**	**Script(s) used / Tasks completed / Screenshots**

<table>
<tr>
<td>☑</td>
<td>OS backup</td>
<td>

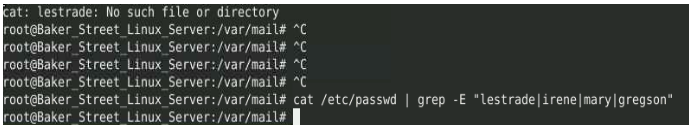


```
/etc/iproute2/ematch_map
/etc/iproute2/bpf_pinning
/etc/iproute2/rt_tables
/etc/iproute2/rt_protos
/etc/iproute2/rt_protos.d/
/etc/iproute2/rt_protos.d/README
/etc/iproute2/rt_tables.d/
/etc/iproute2/rt_tables.d/README
/etc/iproute2/rt_scopes
/etc/iproute2/group
/etc/iproute2/nl_protos
/etc/iproute2/rt_realms
/etc/iproute2/rt_dsfield
/etc/libnl-3/
/etc/libnl-3/pktloc
/etc/libnl-3/classid
/etc/rpc
/etc/ufw/
/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/ethertypes
/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
tar: /: file changed as we read it
root@Baker_Street_Linux_Server:/# uptime -p
```

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 /***

</td>
</tr>
<tr>
<td>☑</td>
<td>Auditing users and groups</td>
<td>

Removing terminated user: **userdel -r lestrade && userdel -r irene && userdel -r**



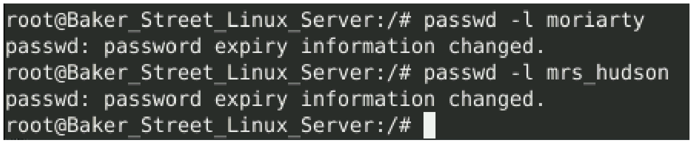
```
cat: lestrade: No such file or directory
root@Baker_Street_Linux_Server:/var/mail# ^C
root@Baker_Street_Linux_Server:/var/mail# ^C
root@Baker_Street_Linux_Server:/var/mail# ^C
root@Baker_Street_Linux_Server:/var/mail# ^C
root@Baker_Street_Linux_Server:/var/mail# cat /etc/passwd | grep -E "lestrade|irene|mary|gregson"
root@Baker_Street_Linux_Server:/var/mail#
```

(cat shows that the users no longer exist)

Locking Accounts for Temp Leave: **passwd -l moriarty && passwd -l mrs_hudson**

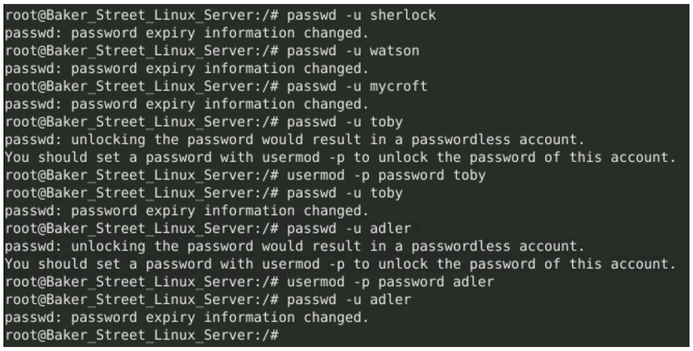
</td>
</tr>
</table>

mary && userdel -r gregson



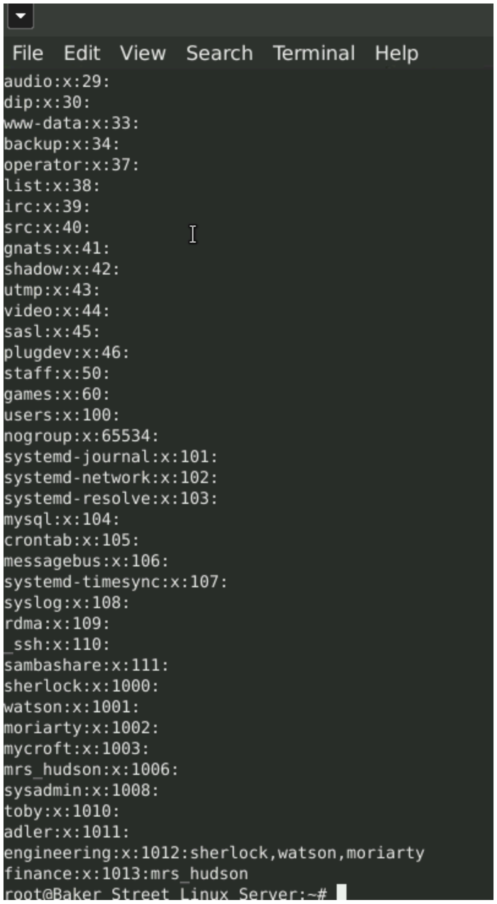
```
root@Baker_Street_Linux_Server:/# passwd -l moriarty
passwd: password expiry information changed.
root@Baker_Street_Linux_Server:/# passwd -l mrs_hudson
passwd: password expiry information changed.
root@Baker_Street_Linux_Server:/#
```

Unlocking Accounts for Employees: **passwd -u sherlock && passwd -u watson && passwd -u mycroft && passwd -u toby && passwd -u adler**



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

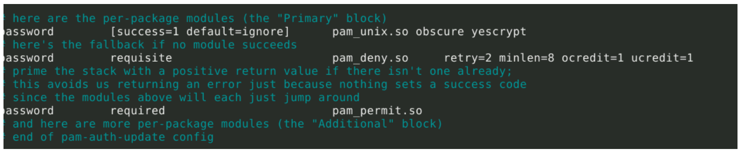
Deleting Marketing Group: **groupdel marketing**



```
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
root@Baker_Street_Linux_Server:~#
```

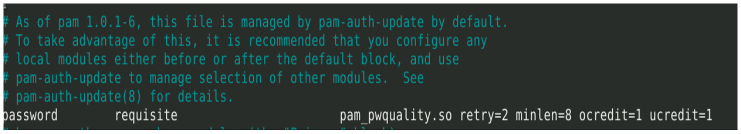
<table>
<tr>
<td>☑</td>
<td>Updating and enforcing password policies</td>
<td>

Setting password requirements: **retry=2 minlen=8 ocredit=1 ucredit=1**



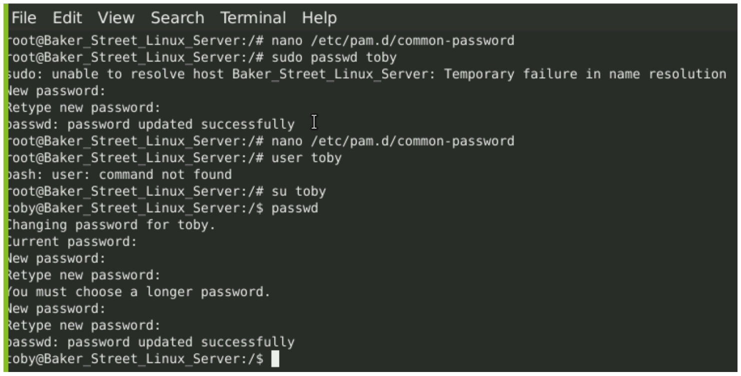
```
 here are the per-package modules (the "Primary" block)
assword        [success=1 default=ignore]      pam_unix.so obscure yescrypt
 here's the fallback if no module succeeds
assword        requisite                       pam_deny.so     retry=2 minlen=8 ocredit=1 ucredit=1
 prime the stack with 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
assword        required                        pam_permit.so
 and here are more per-package modules (the "Additional" block)
 end of pam-auth-update config
```

Added: retry=2 minlen=8 ocredit=1 ucredit=1

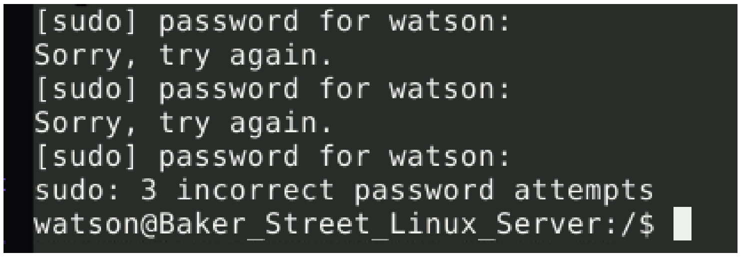


```
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.
password        requisite                      pam_pwquality.so retry=2 minlen=8 ocredit=1 ucredit=1
```

Proof of password policy:



```
oot@Baker_Street_Linux_Server:/# nano /etc/pam.d/common-password
oot@Baker_Street_Linux_Server:/# sudo passwd toby
udo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
ew password:
etype new password:
asswd: password updated successfully
oot@Baker_Street_Linux_Server:/# nano /etc/pam.d/common-password
oot@Baker_Street_Linux_Server:/# user toby
ash: user: command not found
oot@Baker_Street_Linux_Server:/# su toby
oby@Baker_Street_Linux_Server:/$ passwd
hanging password for toby.
urrent password:
ew password:
etype new password:
ou must choose a longer password.
ew password:
etype new password:
asswd: password updated successfully
oby@Baker_Street_Linux_Server:/$
```

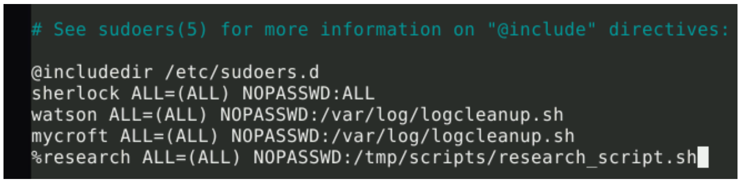


```
[sudo] password for watson:
Sorry, try again.
[sudo] password for watson:
Sorry, try again.
[sudo] password for watson:
sudo: 3 incorrect password attempts
watson@Baker_Street_Linux_Server:/$
```

(kicks you out after 3 attempts)

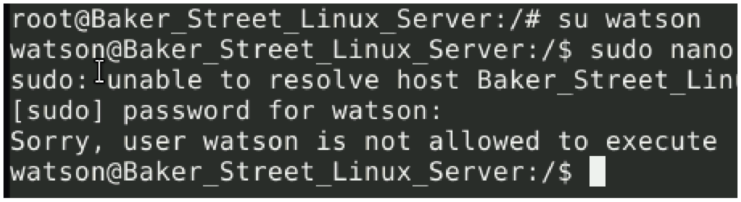
</td>
</tr>
<tr>
<td>☑</td>
<td>Updating and enforcing sudo permissions</td>
<td>

To edit the file: **nano /etc/sudoers**



```
# 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
mycroft ALL=(ALL) NOPASSWD:/var/log/logcleanup.sh
%research ALL=(ALL) NOPASSWD:/tmp/scripts/research_script.sh
```

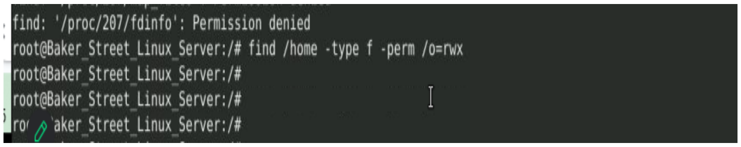
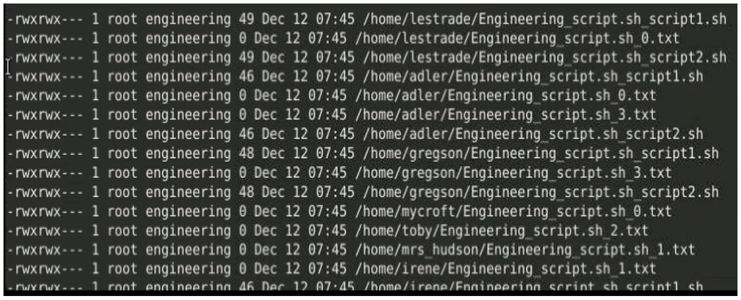
(set up the sudoers file like this)

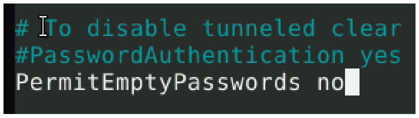
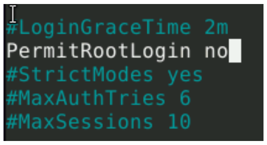
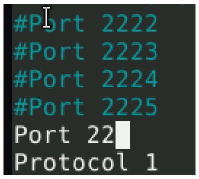
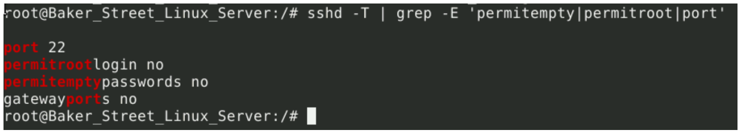
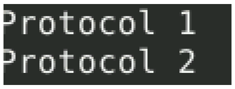


```
root@Baker_Street_Linux_Server:/# su watson
watson@Baker_Street_Linux_Server:/$ sudo nano
sudo: unable to resolve host Baker_Street_Lin
[sudo] password for watson:
Sorry, user watson is not allowed to execute
watson@Baker_Street_Linux_Server:/$
```

(watson tries to sudo nano the sudoers file and is unable)

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

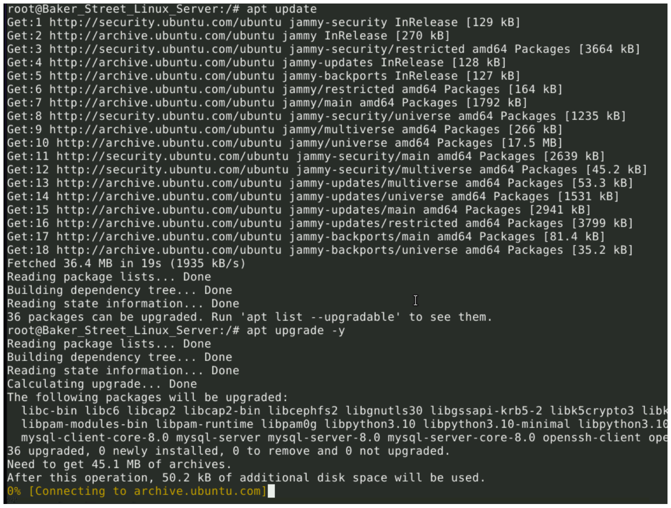
☑	Validating and updating permissions on files and directories	**To remove world permissions from home directories** chmod o-rwx $(sudo find /home -type f -perm /o=rwx) **Verified with**: find /home -type f -perm /o=rwx No output was received showing that there are no home directories with world permissions  find: '/proc/207/fdinfo': Permission denied root@Baker_Street_Linux_Server:/# find /home -type f -perm /o=rwx root@Baker_Street_Linux_Server:/# root@Baker_Street_Linux_Server:/# ro aker_Street_Linux_Server:/# **find /home -type f -iname “*engineering*” -exec chown root:engineering {} \; -exec chmod 770 {} \;** **find /home -type f -iname “*engineering*” -exec ls -l {} \;**  -rwxrwx--- 1 root engineering 49 Dec 12 07:45 /home/lestrade/Engineering_script.sh_script1.sh -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/lestrade/Engineering_script.sh_0.txt -rwxrwx--- 1 root engineering 49 Dec 12 07:45 /home/lestrade/Engineering_script.sh_script2.sh -rwxrwx--- 1 root engineering 46 Dec 12 07:45 /home/adler/Engineering_script.sh_script1.sh -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/adler/Engineering_script.sh_0.txt -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/adler/Engineering_script.sh_3.txt -rwxrwx--- 1 root engineering 46 Dec 12 07:45 /home/adler/Engineering_script.sh_script2.sh -rwxrwx--- 1 root engineering 48 Dec 12 07:45 /home/gregson/Engineering_script.sh_script1.sh -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/gregson/Engineering_script.sh_3.txt -rwxrwx--- 1 root engineering 48 Dec 12 07:45 /home/gregson/Engineering_script.sh_script2.sh -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/mycroft/Engineering_script.sh_0.txt -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/toby/Engineering_script.sh_2.txt -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/mrs_hudson/Engineering_script.sh_1.txt -rwxrwx--- 1 root engineering 0 Dec 12 07:45 /home/irene/Engineering_script.sh_1.txt -rwxrwx--- 1 root engineering 46 Dec 12 07:45 /home/irene/Engineering_script.sh_script1.sh Done for research: **find /home -type f -iname “*research*” -exec chown root:research {} \; -exec chmod 770 {} \;** **find /home -type f -iname “*research*” -exec ls -l {} \;** Done for finance group: **find /home -type f -iname “*finance*” -exec chown root:finance {} \; -exec chmod 770 {} \;** **find /home -type f -iname “*finance*” -exec ls -l {} \;**
☑	Optional: Updating password hashing configuration	

☑	Auditing and securing SSH	1.**nano /etc/ssh/sshd_config** (this is the file we must edit)
		
		`# To disable tunneled clear`
		`#PasswordAuthentication yes`
		`PermitEmptyPasswords no`
		
		`#LoginGraceTime 2m`
		`PermitRootLogin no`
		`#StrictModes yes`
		`#MaxAuthTries 6`
		`#MaxSessions 10`
		
		`#Port 2222`
		`#Port 2223`
		`#Port 2224`
		`#Port 2225`
		`Port 22`
		`Protocol 1`
		Confirm that it works: **sshd -T | grep -E 'permitempty|permitroot|port'**
		
		`root@Baker_Street_Linux_Server:/# sshd -T | grep -E 'permitempty|permitroot|port'`
		`port 22`
		`permitrootlogin no`
		`permitemptypasswords no`
		`gatewayports no`
		`root@Baker_Street_Linux_Server:/#`
		Enable protocol 2 by adding it to the script.
		
		`Protocol 1`
		`Protocol 2`

☑

Reviewing and updating system packages

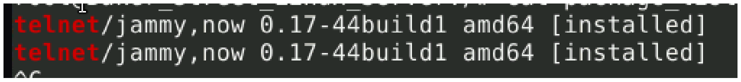
Updating and upgrading:



```
root@Baker_Street_Linux_Server:/# apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3664 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1235 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2639 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [53.3 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1531 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2941 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [3799 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [81.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [35.2 kB]
Fetched 36.4 MB in 19s (1935 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
36 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@Baker_Street_Linux_Server:/# apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  libc-bin libc6 libcap2 libcap2-bin libcephfs2 libgnutls30 libgssapi-krb5-2 libk5crypto3 libk
  libpam-modules-bin libpam-runtime libpam0g libpython3.10 libpython3.10-minimal libpython3.10
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0 openssh-client ope
36 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 45.1 MB of archives.
After this operation, 50.2 kB of additional disk space will be used.
0% [Connecting to archive.ubuntu.com]
```

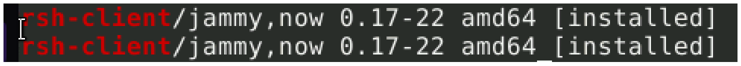
Next use: **apt list --installed >> package_list.txt** (puts our installed packages in a .txt file.

cat package_list.txt | grep telnet



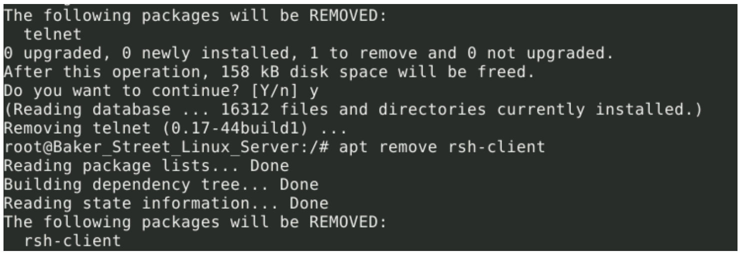
```
telnet/jammy,now 0.17-44build1 amd64 [installed]
telnet/jammy,now 0.17-44build1 amd64 [installed]
```

cat package_list.txt | grep rsh-client



```
rsh-client/jammy,now 0.17-22 amd64 [installed]
rsh-client/jammy,now 0.17-22 amd64 [installed]
```

Remove these unwanted applications: **apt remove telnet** and **apt remove rsh-client**



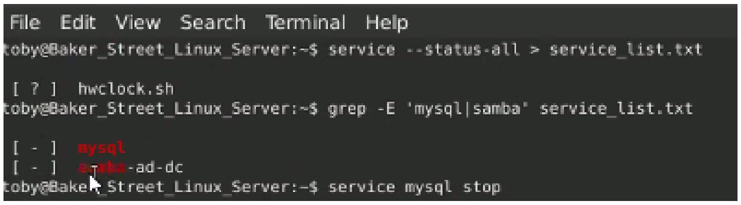
```
The following packages will be REMOVED:
  telnet
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 158 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 16312 files and directories currently installed.)
Removing telnet (0.17-44build1) ...
root@Baker_Street_Linux_Server:/# apt remove rsh-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  rsh-client
```

These applications are not needed because they are unencrypted communications and display user credentials in plain text over the network.

Then we use **apt autoremove -y** which scans and removes automatically installed applications that are not needed.

Install the next packages: **apt install ufw && apt install lynis && apt install tripwire**

*__UFW__ is a user-friendly front-end for managing linux firewalls. **Lynis** is a security auditing tool for linux

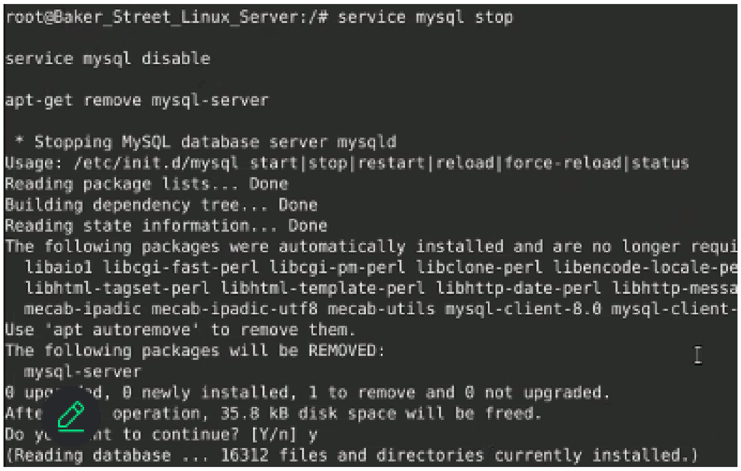
<table>
<tr><td></td><td></td><td>systems that can help generate system reports. <b>Tripwire</b> is an intrusion detection system that detects unauthorized changes to system files.*</td></tr>
<tr><td>☑</td><td>Disabling unnecessary services</td><td>
tobya@Baker_Street_Linux_Server:~$ service --status-all > service_list.txt

 [?] hwclock.sh
toby@Baker_Street_Linux_Server:~$ grep -E 'mysql|samba' service_list.txt

 [-] mysql
 [-] samba-ad-dc
toby@Baker_Street_Linux_Server:~$ service mysql stop

To list all services: <b>service --status-all > service_list.txt</b>

To check if mysql and samba services are running: <b>grep -E 'mysql|samba' service_list.txt</b>


root@Baker_Street_Linux_Server:/# service mysql stop

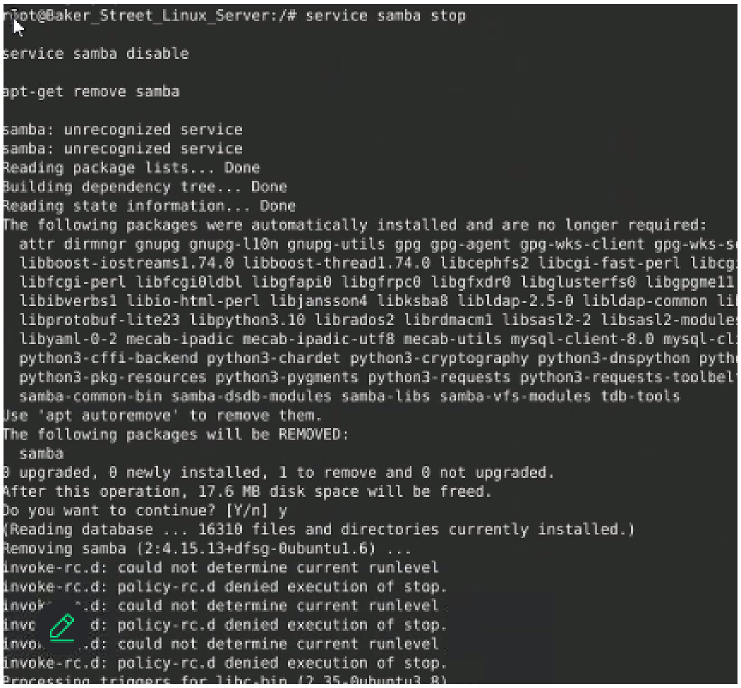
service mysql disable

apt-get remove mysql-server

 * Stopping MySQL database server mysqld
Usage: /etc/init.d/mysql start|stop|restart|reload|force-reload|status
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer requi
  libaio1 libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-pe
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-messa
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  mysql-server
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 35.8 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 16312 files and directories currently installed.)

To stop, disable and remove mysql: <b>service mysql stop</b>
<b>service mysql disable</b>
<b>apt-get remove mysql-server</b></td></tr>
</table>

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



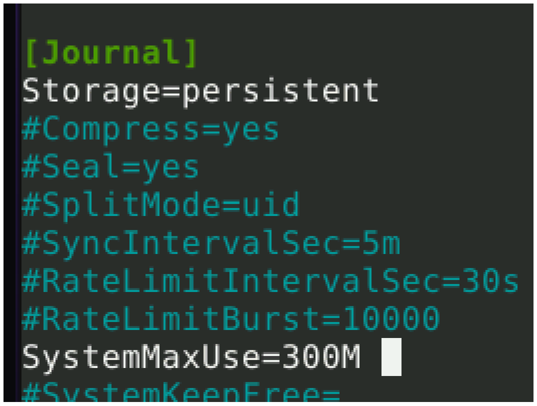
```
root@Baker_Street_Linux_Server:/# service samba stop

service samba disable

apt-get remove samba

samba: unrecognized service
samba: unrecognized service
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  attr dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-s
  libboost-iostreams1.74.0 libboost-thread1.74.0 libcephfs2 libcgi-fast-perl libcg
  libfcgi-perl libfcgi0ldbl libgfapi0 libgfrpc0 libgfxdr0 libglusterfs0 libgpgme11
  libibverbs1 libio-html-perl libjansson4 libksba8 libldap-2.5-0 libldap-common li
  libprotobuf-lite23 libpython3.10 librados2 librdmacm1 libsasl2-2 libsasl2-module
  libyaml-0-2 mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-cl
  python3-cffi-backend python3-chardet python3-cryptography python3-dnspython pyth
  python3-pkg-resources python3-pygments python3-requests python3-requests-toolbel
  samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  samba
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 17.6 MB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 16310 files and directories currently installed.)
Removing samba (2:4.15.13+dfsg-0ubuntu1.6) ...
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of stop.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of stop.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of stop.
Processing triggers for libc-bin (2.35-0ubuntu3.8)
```

To stop, disable and remove samba: **service samba stop**
service samba disable
apt-get remove samba

</td>
</tr>
<tr>
<td>☑</td>
<td>Enabling and configuring logging</td>
<td>

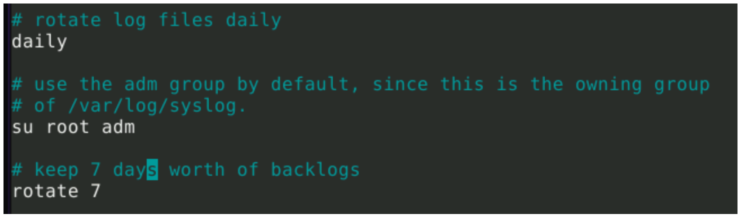
Access the file we will edit: **nano /etc/systemd/journald.conf**

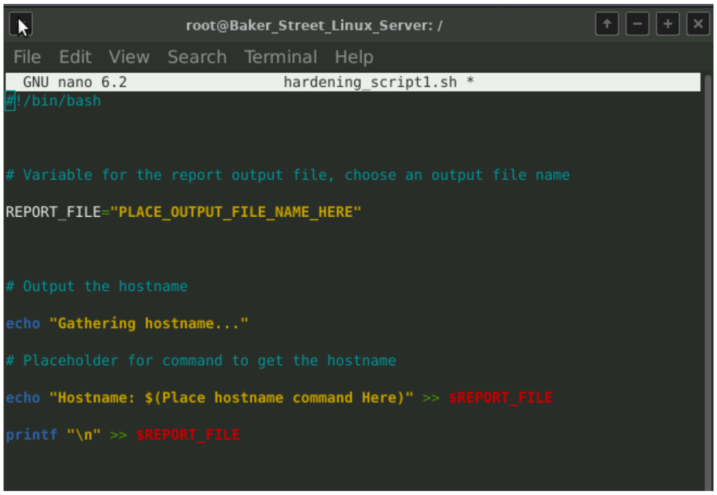
Set the following: **Storage=persistent**
SystemMaxUse=300M



```
[Journal]
Storage=persistent
#Compress=yes
#Seal=yes
#SplitMode=uid
#SyncIntervalSec=5m
#RateLimitIntervalSec=30s
#RateLimitBurst=10000
SystemMaxUse=300M
#SystemKeepFree=
```

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

		Next file: **nano /etc/logrotate.conf**
		
		`# rotate log files daily`
		`daily`
		`# use the adm group by default, since this is the owning group`
		`# of /var/log/syslog.`
		`su root adm`
		`# keep 7 days worth of backlogs`
		`rotate 7`
		Change it so that instead of weekly it says **daily**. Then change the rotation from 4 to 7 so that it keeps **7 days worth** of logs.

☑	Scripts created	To create script 1: **nano hardening_script1.sh** (copy and paste the contents of the doc file into your script file). (you are going to want to add to the script section by section testing the script as you go.)
		 root@Baker_Street_Linux_Server: /
		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="PLACE_OUTPUT_FILE_NAME_HERE"`
		`# Output the hostname`
		`echo "Gathering hostname..."`
		`# Placeholder for command to get the hostname`
		`echo "Hostname: $(Place hostname command Here)" >> $REPORT_FILE`
		`printf "\n" >> $REPORT_FILE`
		First for getting hostname use command: **hostname**
		
		For OS version use command: **uname -r**
		
		For memory information use: **free -h**
		
		For Uptime Information use: **uptime -p**
		
		To create the tar file we use the command: ***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 /***
		
		For viewing the information in the Suoders file use the command: **cat /etc/sudoers**
		
		For deleting files with world permissions we use the command: **chmod o-rwx $(find /home -type f -perm**

/o=rwx

For the group specific permissions use the command: **find -iname ‘*GroupName*’ -exec chown :GroupName {} +** where (GroupName) is the name of the group in which the permissions you are setting.

Script 1 runs with no errors. (too much to screenshot)

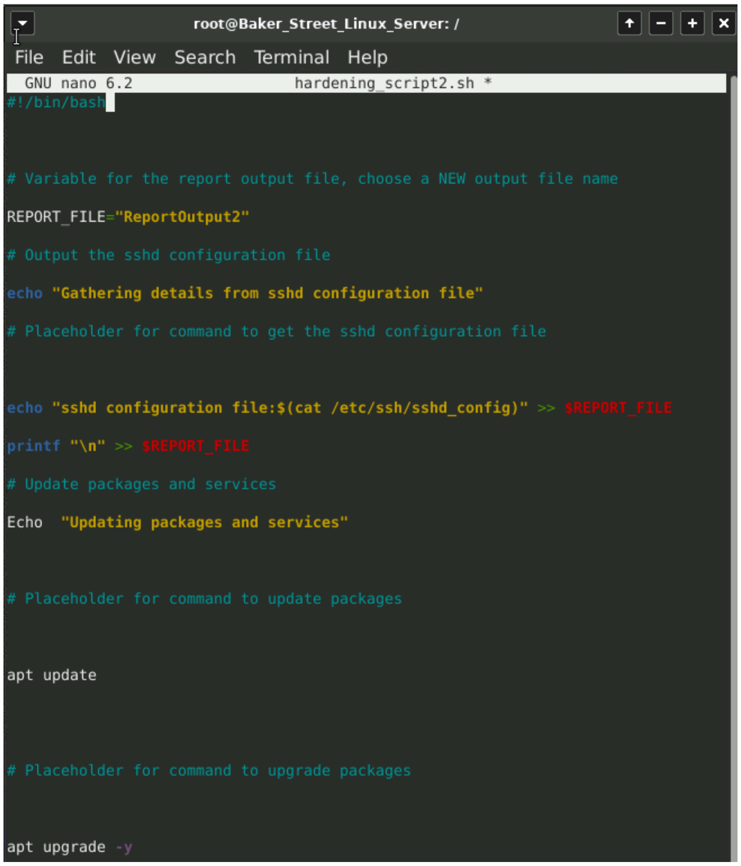
Here is the contents of the script:

```
#!/bin/bash



# Variable for the report output file, choose an output file name

REPORT_FILE="ReportOutput.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 -h)" >> $REPORT_FILE

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

		# Output uptime information # Placeholder for command to get uptime info echo "Uptime Information: $(uptime -p)" >> $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 --> echo "OS backup completed." >> $REPORT_FILE printf "\n" >> $REPORT_FILE # Output the sudoers file to the report echo "Gathering sudoers file..." # Placeholder for command to output sudoers file echo "Sudoers file:$(cat /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..." chmod o-rwx $(find /home -type f -perm /o=rwx) #Placeholder for command to find and update files with world permissions echo "World permissions have been removed from any files found." >> $REPORT_FILE

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

		# Finance scripts - Only members of the finance group echo "Updating permissions for Finance scripts" # Placeholder for command to update permissions find -iname '*finance*' -exec chown :finance {} + echo "Permissions updated for Finance scripts." >> $REPORT_FILE printf "\n" >> $REPORT_FILE echo "Script execution completed. Check $REPORT_FILE for details." To create script 2: **nano hardening_script2.sh**


root@Baker_Street_Linux_Server: /
File Edit View Search Terminal Help
GNU nano 6.2
hardening_script2.sh *

```
#!/bin/bash


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

REPORT_FILE="ReportOutput2"

# Output the sshd configuration file

echo "Gathering details from sshd configuration file"

# Placeholder for command to get the sshd configuration file


echo "sshd configuration file:$(cat /etc/ssh/sshd_config)" >> $REPORT_FILE

printf "\n" >> $REPORT_FILE

# Update packages and services

Echo  "Updating packages and services"


# Placeholder for command to update packages


apt update



# Placeholder for command to upgrade packages


apt upgrade -y
```

First set the Report_File variable as: **ReportOutput2** (this will be our output file for the results of the script)

To check sshd configurations use the command: **cat /etc/ssh/sshd_config**

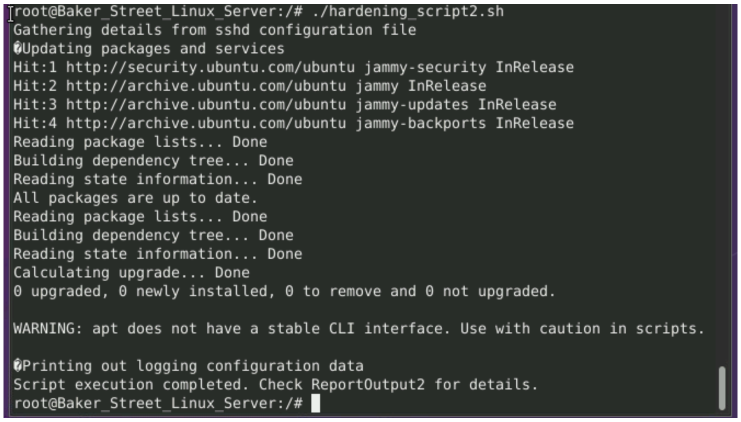
To update use the command: **apt update**

To upgrade use the command: **apt upgrade -y** (make sure to use the -y option to say yes to all the prompting in the upgrade process)

To check for installed packages use the command: **apt list --installed**

To display logging data use the command: **cat /etc/systemd/journald.conf**

To display logrotate data use the command: **cat /etc/logrotate.conf**



```
root@Baker_Street_Linux_Server:/# ./hardening_script2.sh
Gathering details from sshd configuration file
�Updating packages and services
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

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

�Printing out logging configuration data
Script execution completed. Check ReportOutput2 for details.
root@Baker_Street_Linux_Server:/# 
```

Runs with no errors!

```
#!/bin/bash



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

REPORT_FILE="ReportOutput2"

# Output the sshd configuration file

echo "Gathering details from sshd configuration file"

# Placeholder for command to get the sshd configuration file



echo "sshd configuration file:$(cat /etc/ssh/sshd_config)" >> $REPORT_FILE

printf "\n" >> $REPORT_FILE

# Update packages and services

echo  "Updating packages and services"



# Placeholder for command to update packages



apt update
```

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

☑	Scripts scheduled with cron	To schedule a cron job for script 1 do the following: Use the command **crontab -e**. For the first script use the command: **0 0 1 * * /bin/script_hardening1.sh** (this will schedule script 1 to run once a month on the first of the month. To schedule a cron job for script 2 do the following: Use the command **crontab -e.** For the second script use the command: **0 12 * * 1 /bin/script_hardening2.sh (**this will schedule script 2 to run once a week every Monday. For the purpose of testing I used the script (***/5 * * * * /bin/script_hardening1.sh**) to make the script run every 5 minutes.

Summary Report

Completed Tasks Checklist

SSH Hardening:

- Disabled root login via SSH.
- Prevented SSH access with empty passwords.
- Changed SSH port to 2222 since port 22 is in use by another service.
- Ensured SSH is using Protocol 2 for security.

Package Management:

- Updated the package manager and upgraded all installed packages.
- Removed insecure applications (telnet and rsh-client).
- Installed essential security tools: ufw, lynis, and tripwire.

Logging Configuration:

- Updated journald.conf to enable persistent log storage.
- Limited log size to 300MB to prevent excessive storage use.
- Configured log rotation in /etc/logrotate.conf:
  - Changed log rotation to daily.
  - Retained logs for seven days.

File and Directory Permissions:

- Removed world permissions (read, write, execute) from all files in /home.
- Set script access permissions:
  - Engineering scripts: Restricted to the engineering team.
  - Research scripts: Limited to research staff.
  - Finance scripts: Accessible only to the finance department.

User Management:

- Removed terminated employees' accounts, including their home directories and files.
- Locked accounts for staff on temporary leave.
- Unlocked accounts for active employees.
- Reorganized groups:
  - Moved all marketing staff to the newly created research group.
  - Deleted the marketing group since it was no longer needed.

Password Security:

- Updated password policies in /etc/pam.d/common-password to enforce security:
  - Minimum 8 characters.

    - Must include at least one uppercase letter and one special character.
    - Limited retries to two attempts.
- Enabled SHA-512 hashing for stronger password security.

Sudo Privileges:

- Sherlock is now the only user with full sudo access.
- Removed full sudo privileges from all other users.
- Restricted sudo permissions:
    - Watson & Mycroft can only execute /var/log/logcleanup.sh.
    - Research Group can run /tmp/scripts/research_script.sh with sudo.

Automation and Monitoring:

- Created and tested automated security scripts:
    - hardening_script1.sh: Handles system checks, backups, and permissions updates.
    - hardening_script2.sh: Automates updates, log monitoring, and security reporting.
- Scheduled these scripts using cron:
    - Script 1 runs monthly on the 1st.
    - Script 2 runs weekly on Mondays.

Security Issues Identified and Resolved

SSH Vulnerabilities:

- Problem: Root login and empty password access allowed unauthorized entry.

- Fix: Restricted SSH to non-root users and enforced strong password requirements.

Insecure Services:

- Problem: Outdated and insecure services (telnet, rsh-client) were active.
- Fix: Removed these services to prevent unencrypted data transmission.

File Permissions Risks:

- Problem: Files with world permissions were accessible to unauthorized users.
- Fix: Removed world read/write/execute permissions and applied group-specific restrictions.

User Management Risks:

- Problem: Terminated staff and users on leave still had active accounts.
- Fix:
  - Deleted terminated users' accounts and home directories.
  - Locked accounts of users on temporary leave.
  - Ensured active employee accounts were accessible as needed.

Sudo Privilege Risks:

- Problem: Too many users had unrestricted sudo access, increasing the risk of privilege abuse.
- Fix:
  - Sherlock remains the only user with full sudo rights.
  - Watson & Mycroft now have limited sudo access for a specific script.
  - Research group granted sudo access only for research_script.sh.

Weak Password Policies:

- Problem: Weak passwords made the system vulnerable to brute-force attacks.
- Fix: Strengthened password security with an 8-character minimum, uppercase, special character, and retry limits.

Automation & Monitoring:

- Problem: Lack of automated security checks and updates.
- Fix:
    - Implemented automated scripts for system hardening and audits.
    - Scheduled periodic execution using cron to maintain security.

Conclusion

By implementing these security improvements, we have:

- Eliminated unnecessary services to reduce attack surfaces.
- Enforced strong password policies and secure password hashing.
- Restricted sudo privileges to follow the principle of least privilege.
- Properly managed user accounts, removing inactive or unauthorized users.
- Secured file permissions, preventing unauthorized script access.
- Enabled logging and automation, ensuring continuous monitoring and system integrity.

These enhancements have significantly strengthened the security posture of our Linux environment, reducing vulnerabilities and improving overall system protection.