Carson Neidert
01/09/2024

Cybersecurity

Project 1 Hardening Summary and Checklist

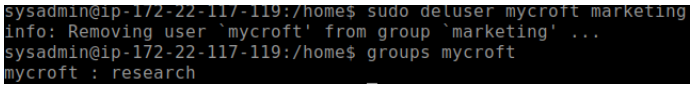
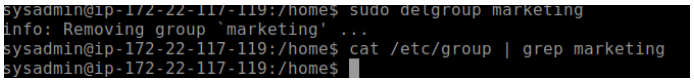
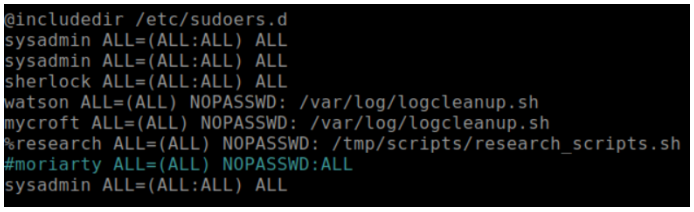
OS Information

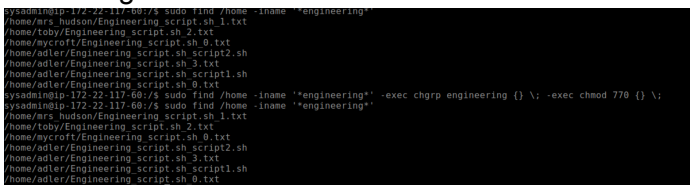
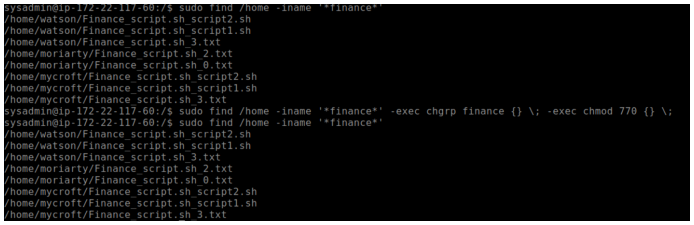
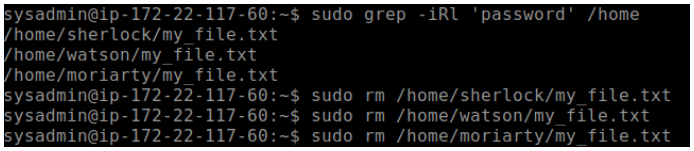
Customer	Baker Street Corporation
Hostname	**ip-172-22-117-119**
OS Version	**Linux ip-172-22-117-119 6.8.0-1016-aws #17-Ubuntu SMP Mon Sep 2 13:48:07 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux**
Memory information	**total used free shared buff/cache available Mem: 3.7Gi 953Mi 1.3Gi 27Mi 1.8Gi 2.8Gi Swap: 0B 0B 0B**
Uptime information	**23:56:24 up 42 min, 1 user, load average: 0.00, 0.00, 0.00**

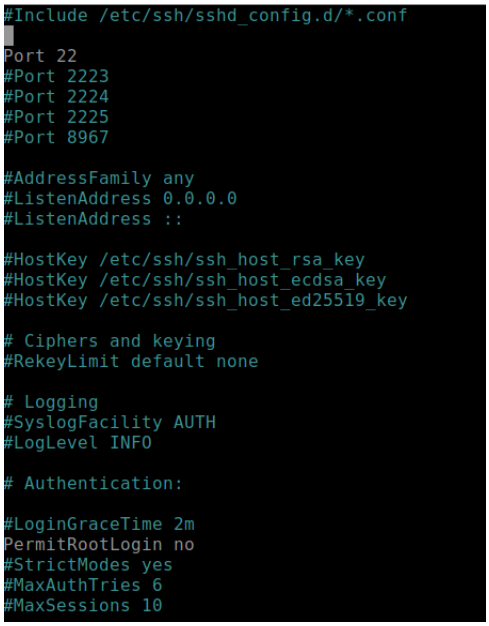
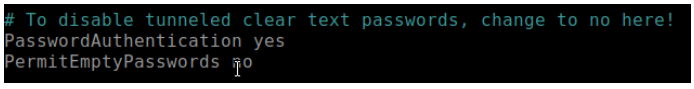
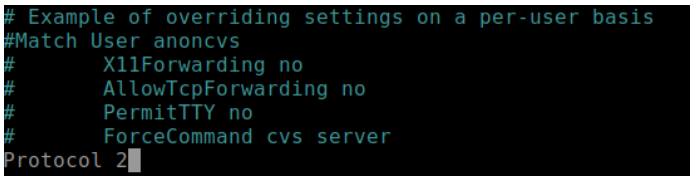
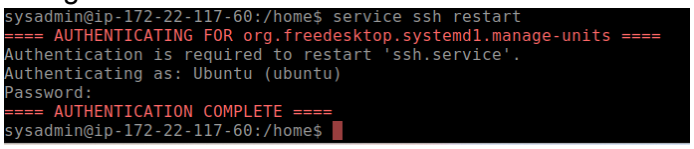
Checklist

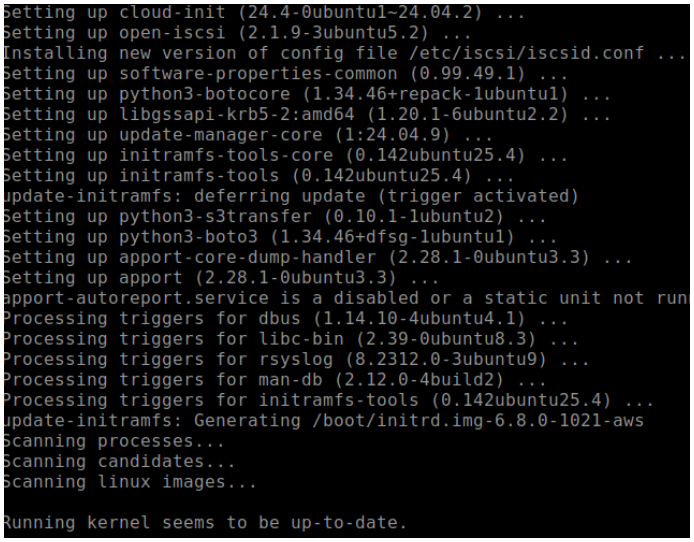
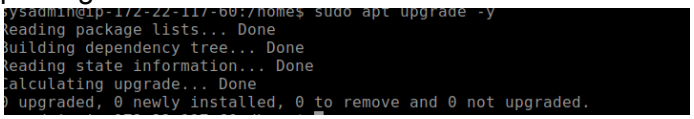
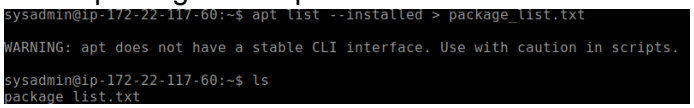
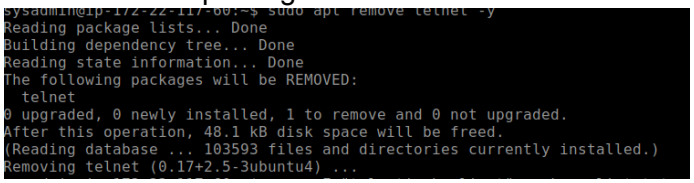
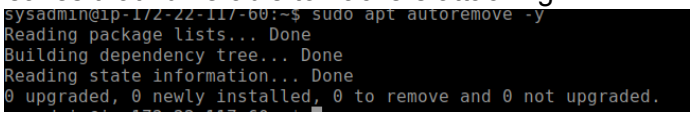
Completed	Activity	Script(s) used / Tasks completed / Screenshots
☑	OS backup	sudo tar -cvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run /

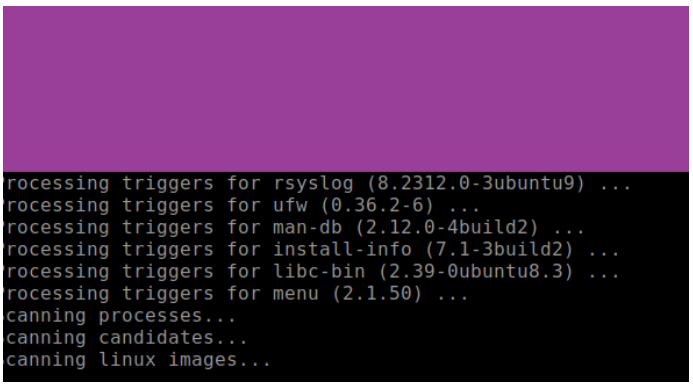
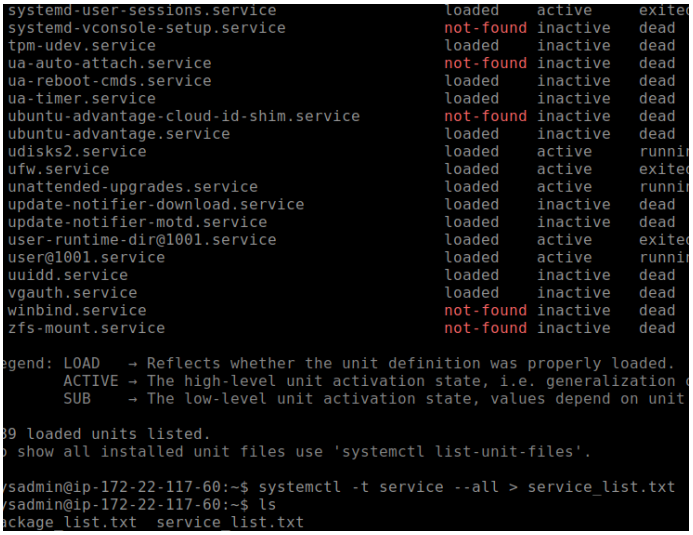
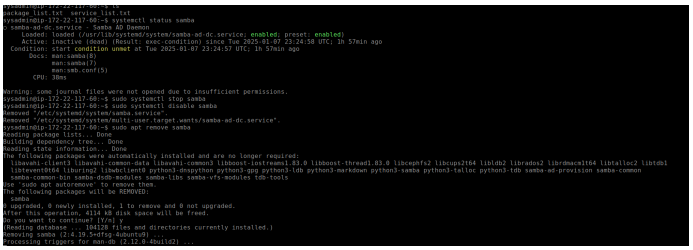
		/etc/mysql/mysql.conf.d/mysqld.cnf /etc/skel/ /etc/skel/.bashrc /etc/skel/.bash_logout /etc/skel/.profile /etc/rc0.d/ /etc/rc0.d/K01uuidd /etc/rc0.d/K01open-iscsi /etc/rc0.d/K01unattended-upgrades /etc/rc0.d/K01open-vm-tools /etc/rc0.d/K01nmbd /etc/rc0.d/K01mysql /etc/rc0.d/K01cryptdisks /etc/rc0.d/K01plymouth /etc/rc0.d/K01openbsd-inetd /etc/rc0.d/K01cryptdisks-early /etc/rc0.d/K01smbd /etc/rc0.d/K01irqbalance /etc/rc0.d/K01chrony /etc/rc0.d/K01samba-ad-dc /etc/rc0.d/K01iscsid /etc/vtrgb /etc/subuid Some of the items that were being backed up during the process
☑	Auditing users and groups	sysadmin@ip-172-22-117-119:/home$ sudo userdel -r irene userdel: irene mail spool (/var/mail/irene) not found sysadmin@ip-172-22-117-119:/home$ sudo userdel -r mary userdel: mary mail spool (/var/mail/mary) not found sysadmin@ip-172-22-117-119:/home$ sudo userdel -r gregson userdel: gregson mail spool (/var/mail/gregson) not found sysadmin@ip-172-22-117-119:/home$ ls adler moriarty mrs_hudson mycroft sherlock sysadmin toby ubuntu watson Deleted users lestrade,irene,mary,and gregson sysadmin@ip-172-22-117-119:/home$ sudo usermod -L moriarty sysadmin@ip-172-22-117-119:/home$ sudo usermod -L mrs_hudson sysadmin@ip-172-22-117-119:/home$ ls adler moriarty mrs_hudson mycroft sherlock sysadmin toby ubuntu watson Locked users moriarty and mrs_hudson as they are on temporary leave sysadmin@ip-172-22-117-119:/home$ sudo usermod -U sherlock sysadmin@ip-172-22-117-119:/home$ sudo usermod -U watson sysadmin@ip-172-22-117-119:/home$ sudo usermod -U mycroft sysadmin@ip-172-22-117-119:/home$ sudo usermod -p toby123 toby sysadmin@ip-172-22-117-119:/home$ sudo usermod -U toby sysadmin@ip-172-22-117-119:/home$ sudo usermod -U adler usermod: unlocking the user's password would result in a passwordless account. You should set a password with usermod -p to unlock this user's password. sysadmin@ip-172-22-117-119:/home$ sudo usermod -p adlerpwd adler sysadmin@ip-172-22-117-119:/home$ sudo usermod -U adler sysadmin@ip-172-22-117-119:/home$ ls adler moriarty mrs_hudson mycroft sherlock sysadmin toby ubuntu watson Unlocked users sherlock,watson,mycroft,toby, and adler. Had to create a password for users toby and adler to unlock. sysadmin@ip-172-22-117-119:/home$ sudo usermod -g research mycroft sysadmin@ip-172-22-117-119:/home$ groups mycroft mycroft : research marketing After discovering that mycroft was part of the marketing team using cat /etc/groups, we changed

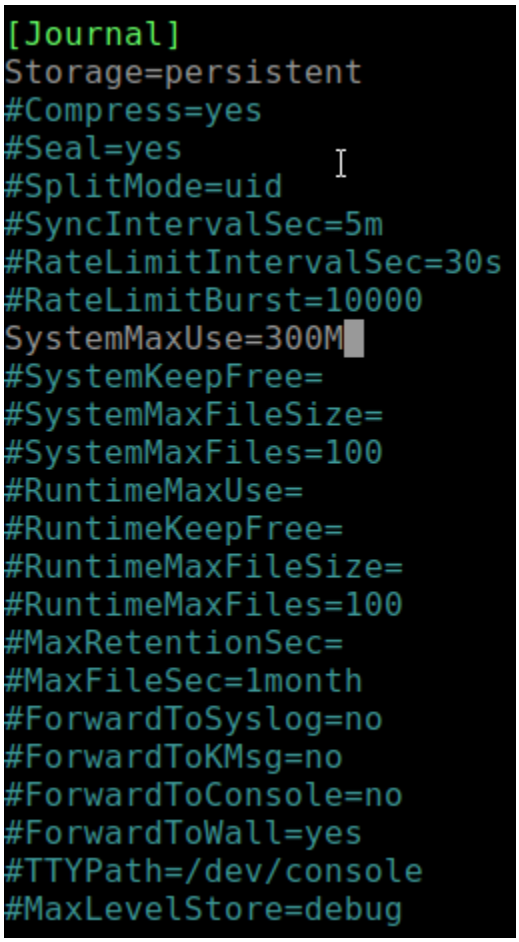
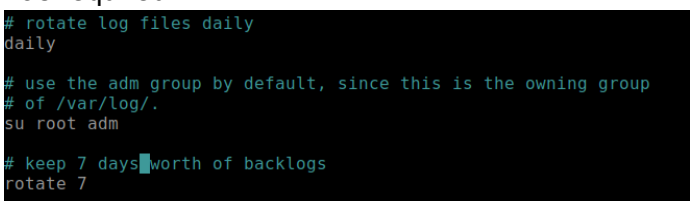
		the user to the research team  We then remove mycroft from the marketing group as it does not exist anymore  As there is no more marketing group, we delete it and then check to see if it is there, and it no longer exists.
☑	Updating and enforcing password policies	 Nanoed into the file and added the updates to the password: minlen=8,retry=2,ucredit=-1,ocredit=-1. The negative values are because it required at least one uppercase letter and special character. Had to add libpam-pwquality to the library as it was not present originally when nanoing into the file.
☑	Updating and enforcing sudo permissions	 In the sudoers file, changed sudo permissions to allow only sherlock all sudo permissions, watson and mycroft sudo permission to /var/loglogcleanup.sh, and research group sudo permission to run the script /tmp/script/research_script.sh.

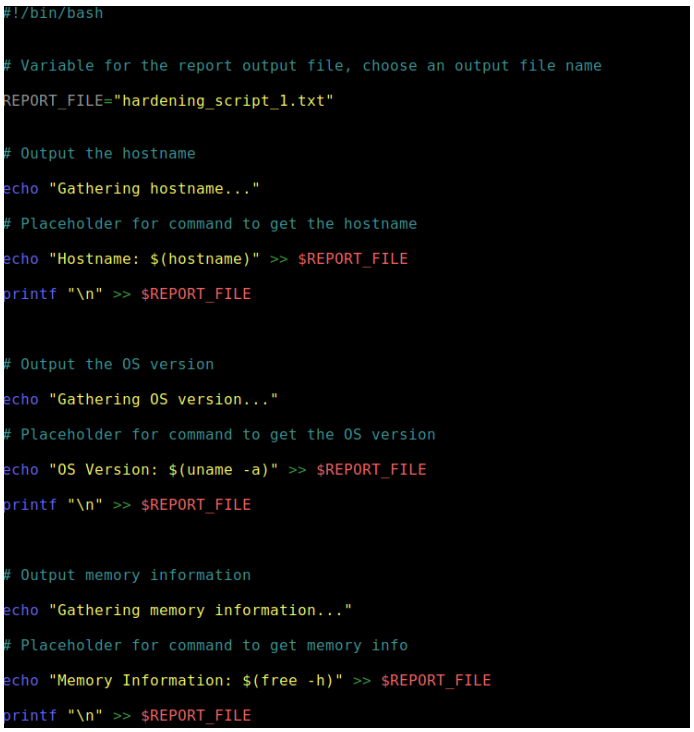
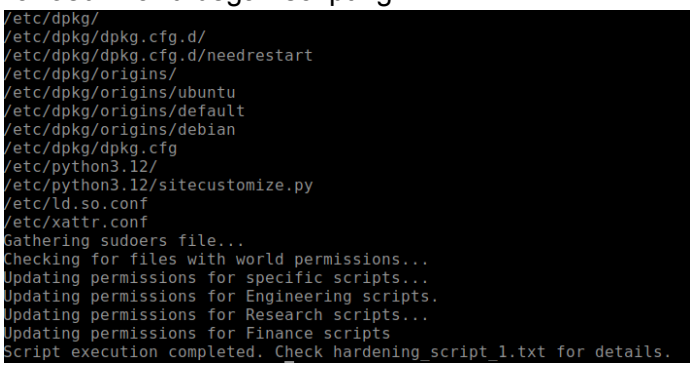
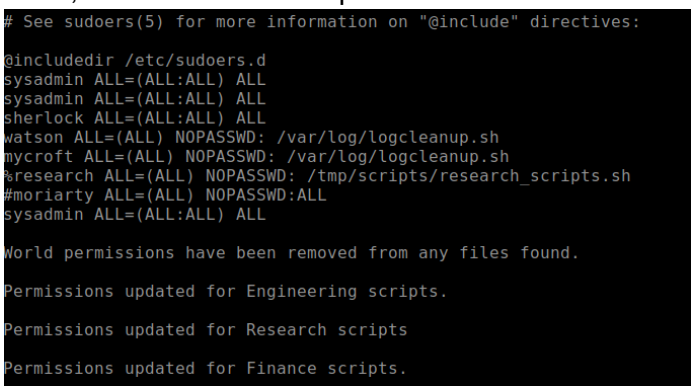
☑	Validating and updating permissions on files and directories	 After finding the world permission files, we changed them so none were world permission files, and confirming that none were there.  Next was to find the scripts with engineering in it. After finding them, changed it to wear only the owner and engineering group could read, write and execute.  Then did the same thing for finance and changed all the scripts to be read, written,and executed by the owner and finance group. Tried for research, but found no research.sh files in the home directory.  Then we were tasked with finding any files that employees may have left a password in. With the find, we found sherlock, watson, and mycroft had files where the word password was used, so we deleted those files from the system.
	Optional: Updating password hashing configuration	

☑	Auditing and securing SSH	  We were asked to ensure that SSH was not allowed to have the ability to have empty passwords, root users, and to have any ports besides 22. I changed the yes to no for empty passwords and root users, and then commented out all the ports and left Port 22 uncommented.  We were then asked to enable Protocol 2, so I changed Protocol 1 to Protocol 2.  I then updated the SSH service with the new updates using the given command *service ssh restart.*

☑	Reviewing and updating system packages	 After running apt update, we can see how the package manager is making sure it has all the latest packages.  It shows now that if we run apt upgrade -y, we see that all packages are updated and installed.  I created a text file called package_list.txt thats lists all the installed packages.  The telnet package was installed so I removed it. Telnet can lead to security issues as they leave passwords and data in plain text unencrypted, which leaves that vulnerable to hackers attacking.  Then I tried to remove any dependencies that may have been present, but there were none to be removed.

		 We were then tasked with installing,ufw,lynis, and tripwire. The image shows it being installed. At one point I was asked to create a sitekey and localkey for the configuration.UFW creates an easy interface for configuring common firewall cases. Lynis can perform an extensive health scan of your systems to check for security risks and creates a report of that. Tripwire can monitor for unauthorized file and directory manipulation on local systems.
☑	Disabling unnecessary services	 After listing out all the services, I added them to the text file service_list.txt.  This shows me stopping, disabling, and removing the service samba. This is one of the two services

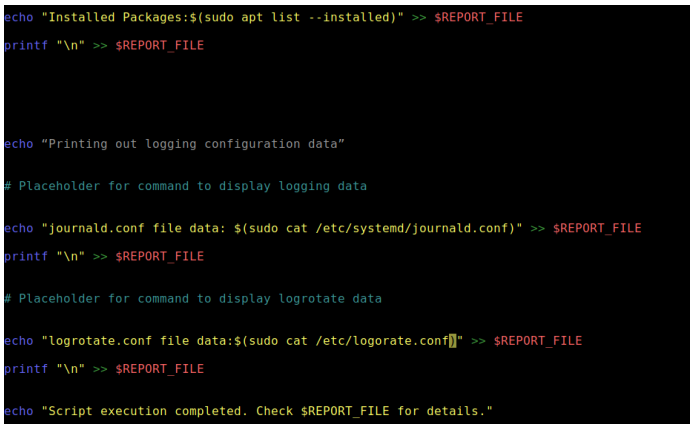
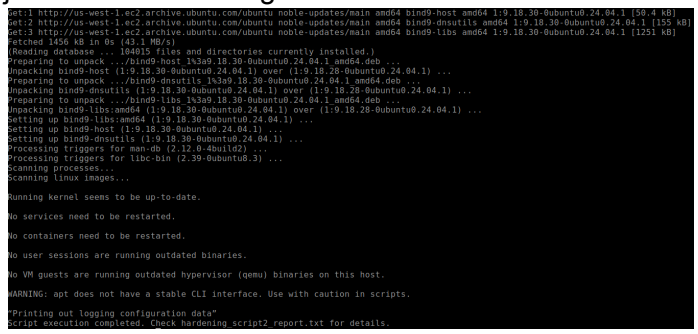
		we were asked to check for and remove from the system. The other service was mysql which looked similar to this when doing the process of stopping, disabling, and removing the services.
☑	Enabling and configuring logging	 I moved to the /etc/systemd/ directory and nanoed into the journald.conf file. In it, I changed storage=persistent, and SystemMaxUse=300M as was required.  We were then tasked to help prevent logs from taking up too much space. I changed the rotation of logs from weekly to daily and then changed the rotate of 4 weeks worth of logs to 7 days worth of logs. All the changes were saved.

☑	Scripts created	 Here is a little chunk out of hardening_script1.sh as I nanoed in and began scripting.  After fixing errors and running the script a couple times, we see that the script ran as written.  As we cat the report file, the permissions have been updated and the list of users with sudo access is

```
#!/bin/bash

# Variable for the report output file, choose an output file name
REPORT_FILE="hardening_script_1.txt"

# Output the hostname
echo "Gathering hostname..."
# Placeholder for command to get the hostname
echo "Hostname: $(hostname)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE


# Output the OS version
echo "Gathering OS version..."
# Placeholder for command to get the OS version
echo "OS Version: $(uname -a)" >> $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
```

```
/etc/dpkg/
/etc/dpkg/dpkg.cfg.d/
/etc/dpkg/dpkg.cfg.d/needrestart
/etc/dpkg/origins/
/etc/dpkg/origins/ubuntu
/etc/dpkg/origins/default
/etc/dpkg/origins/debian
/etc/dpkg/dpkg.cfg
/etc/python3.12/
/etc/python3.12/sitecustomize.py
/etc/ld.so.conf
/etc/xattr.conf
Gathering sudoers file...
Checking for files with world permissions...
Updating permissions for specific scripts...
Updating permissions for Engineering scripts.
Updating permissions for Research scripts...
Updating permissions for Finance scripts
Script execution completed. Check hardening_script_1.txt for details.
```

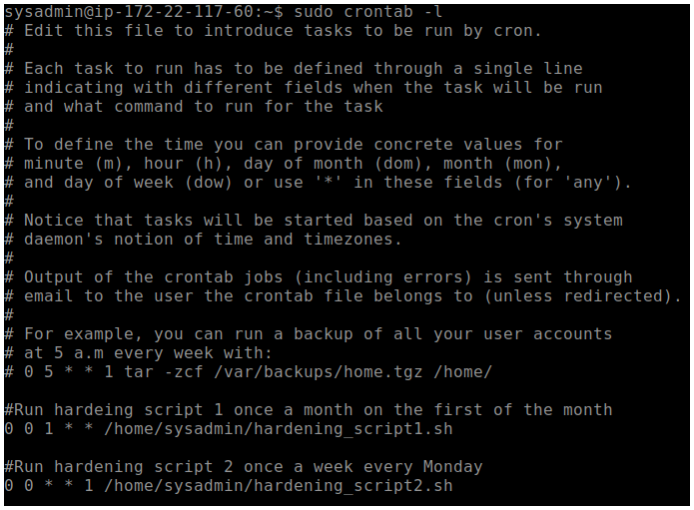
```
# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
sysadmin ALL=(ALL:ALL) ALL
sysadmin ALL=(ALL:ALL) ALL
sherlock ALL=(ALL:ALL) ALL
watson ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh
mycroft ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh
%research ALL=(ALL) NOPASSWD: /tmp/scripts/research_scripts.sh
#moriarty ALL=(ALL) NOPASSWD:ALL
sysadmin ALL=(ALL:ALL) ALL

World permissions have been removed from any files found.

Permissions updated for Engineering scripts.

Permissions updated for Research scripts

Permissions updated for Finance scripts.
```

		there. At the top it lists the hostname, os version, memory information and uptime information.  Here is a chunk of the second hardening script. In this example, we see the scripts for listing the installed packages, and showing the data from journald.conf and logrotate.conf.  After running the script, it was completed so I can go look at the report file.

```
echo "Installed Packages:$(sudo 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: $(sudo cat /etc/systemd/journald.conf)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

# Placeholder for command to display logrotate data

echo "logrotate.conf file data:$(sudo cat /etc/logorate.conf)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

echo "Script execution completed. Check $REPORT_FILE for details."
```

```
Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 bind9-host amd64 1:9.18.30-0ubuntu0.24.04.1 [50.4 kB]
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 bind9-dnsutils amd64 1:9.18.30-0ubuntu0.24.04.1 [155 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 bind9-libs amd64 1:9.18.30-0ubuntu0.24.04.1 [1251 kB]
Fetched 1456 kB in 0s (43.1 MB/s)
(Reading database ... 104015 files and directories currently installed.)
Preparing to unpack .../bind9-host 1%3a9.18.30-0ubuntu0.24.04.1_amd64.deb ...
Unpacking bind9-host (1:9.18.30-0ubuntu0.24.04.1) over (1:9.18.28-0ubuntu0.24.04.1) ...
Preparing to unpack .../bind9-dnsutils_1%3a9.18.30-0ubuntu0.24.04.1_amd64.deb ...
Unpacking bind9-dnsutils (1:9.18.30-0ubuntu0.24.04.1) over (1:9.18.28-0ubuntu0.24.04.1) ...
Preparing to unpack .../bind9-libs_1%3a9.18.30-0ubuntu0.24.04.1_amd64.deb ...
Unpacking bind9-libs:amd64 (1:9.18.30-0ubuntu0.24.04.1) over (1:9.18.28-0ubuntu0.24.04.1) ...
Setting up bind9-libs:amd64 (1:9.18.30-0ubuntu0.24.04.1) ...
Setting up bind9-host (1:9.18.30-0ubuntu0.24.04.1) ...
Setting up bind9-dnsutils (1:9.18.30-0ubuntu0.24.04.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

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

“Printing out logging configuration data”
Script execution completed. Check hardening_script2_report.txt for details.
```

		sshd configuration file: # This is the ssh client system-wide configuration file. See # ssh_config(5) for more information. This file provides defaults for # users, and the values can be changed in per-user configuration files # or on the command line. # Configuration data is parsed as follows: # 1. command line options # 2. user-specific file # 3. system-wide file # Any configuration value is only changed the first time it is set. # Thus, host-specific definitions should be at the beginning of the # configuration file, and defaults at the end. # Site-wide defaults for some commonly used options. For a comprehensive # list of available options, their meanings and defaults, please see the # ssh_config(5) man page. Include /etc/ssh/ssh_config.d/*.conf Host * # ForwardAgent no # ForwardX11 no # ForwardX11Trusted yes # PasswordAuthentication yes # HostbasedAuthentication no # GSSAPIAuthentication no # GSSAPIDelegateCredentials no # GSSAPIKeyExchange no # GSSAPITrustDNS no # BatchMode no # CheckHostIP no # AddressFamily any # ConnectTimeout 0 # StrictHostKeyChecking ask # IdentityFile ~/.ssh/id_rsa # IdentityFile ~/.ssh/id_dsa # IdentityFile ~/.ssh/id_ecdsa # IdentityFile ~/.ssh/id_ed25519 # Port 22 When opening hardening_script2_report.txt, I immediately see the sshd configuration file. Scrolling through it using the more command, I saw the list of installed packages as well as the data contents of the journald.conf and logrotate.conf files as well.
☑	Scripts scheduled with cron	# For example, you can run a backup of all your user accounts # at 5 a.m every week with: # 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/ #Run hardeing script 1 once a month on the first of the month 0 0 1 * * /home/sysadmin/hardening_script1.sh #Run hardening script 2 once a week every Monday 0 0 * * 1 /home/sysadmin/hardening_script2.sh The cron jobs for the hardening scripts were asked for script 1 on the first day of every month and for script 2 once a week every monday. So I used a cron job calculator to get the correct format for each specific job.

		 The cron jobs are now listed when asking the system for the list of cron jobs.

The only concern I had was the telnet package being installed on the system. Telnet being installed can lead to security issues as they leave passwords and data in plain text unencrypted, which leaves that vulnerable to hackers attacking. I removed this when looking for it so that this would not be a problem for the system and its users.