Cybersecurity

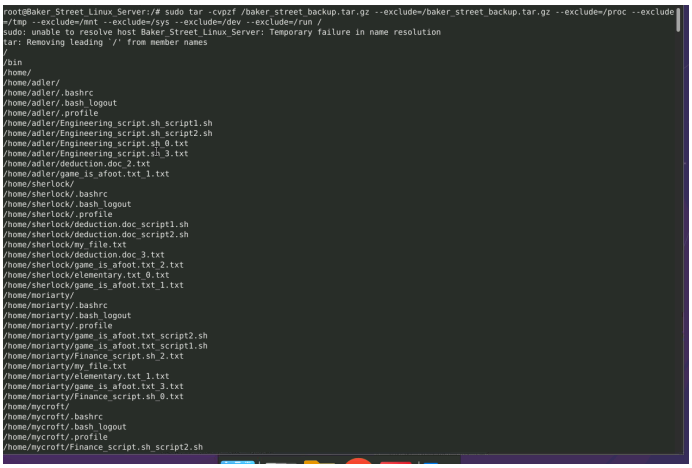
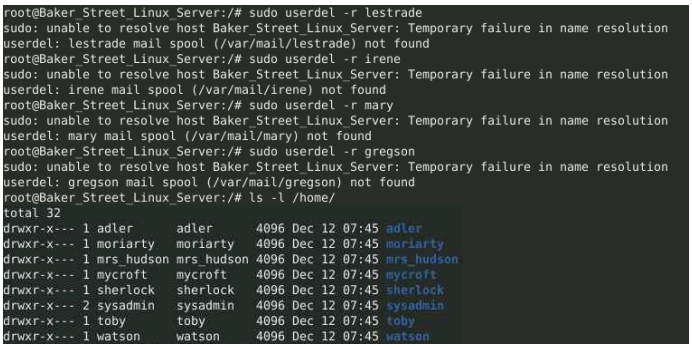
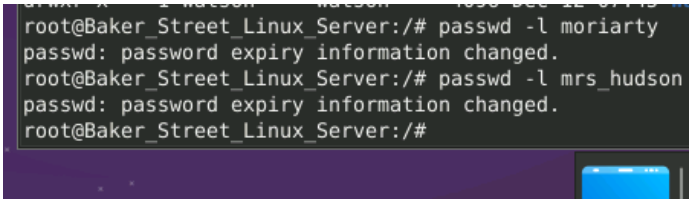
Project 1 Hardening Summary and Checklist

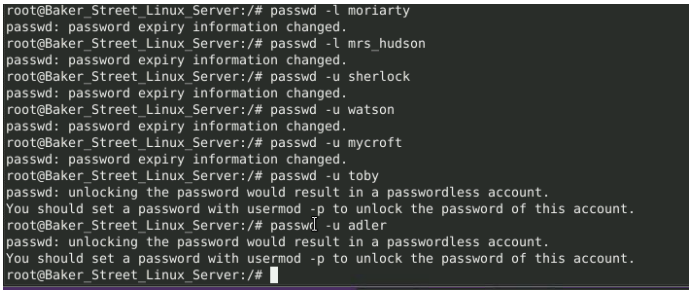
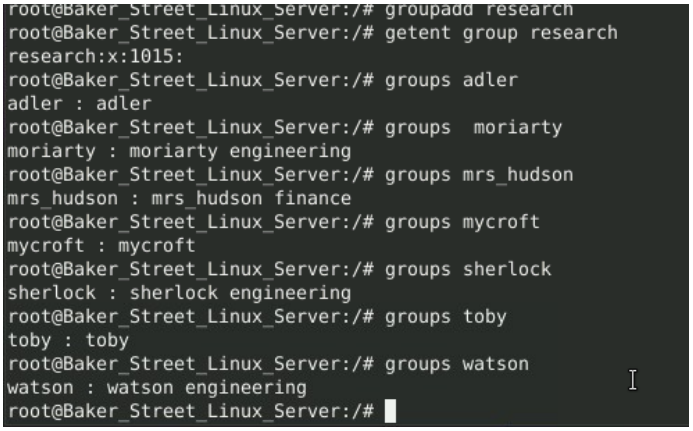
OS Information

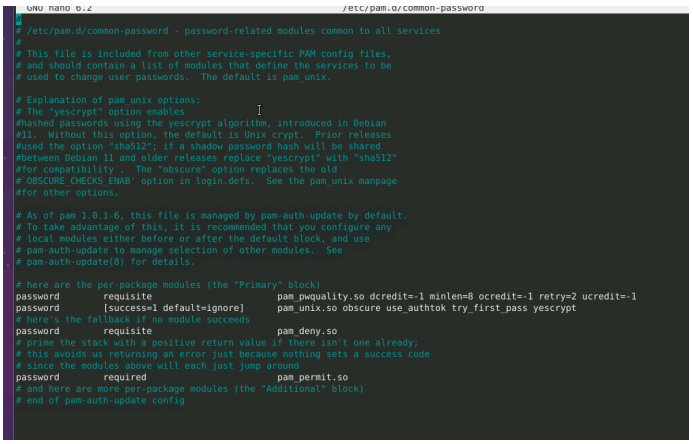
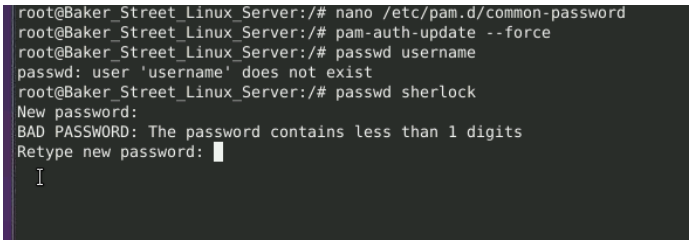
Customer	Baker Street Corporation
Hostname	hostname
OS Version	uname -a
Memory information	free -h
Uptime information	uptime

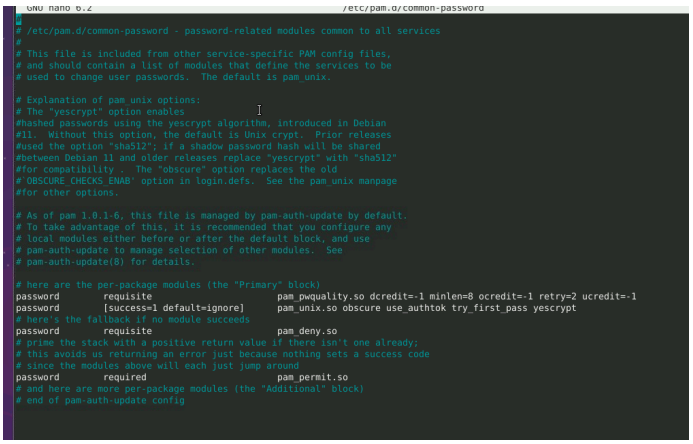
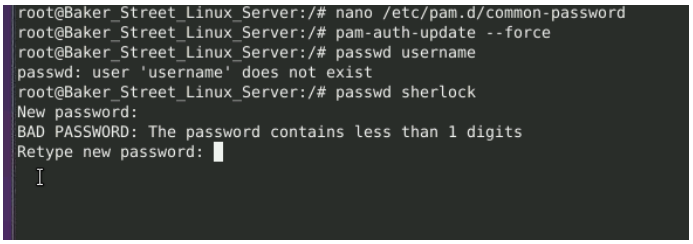
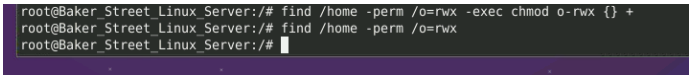
Checklist

Completed	Activity	Script(s) used / Tasks completed / Screenshots
☐	OS backup	-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 /

		
☐	Auditing users and groups	In the next step, I needed to remove users that have been terminated along with their home directories, with removing terminated users and their home directory I used the command userdel -r (username). This removed the user and the home directory.  In the next step locking all users that are on leave was simply using the command passwd -l (username). This locked the account so it cannot be used while they're away.  Then unlocking users that are employed used the command passwd -u (username).

		 In the next step I was tasked to move users that are in the marketing department, but after searching up the users in their group using the command groups (username) they're not anyone associated with the marketing group. 
☐	Updating and enforcing password policies	In this step I performed a nano command to set password requirements.First i used apt update && apt install libpam-pwquality -y, then using nano /etc/pam.d/common-password to input the password requirements for all users. The next command I used was pam-auth-update –force to require the users to reset their password that will have the correct password requirements then I tried to create a new password for a user to ensure the password requirement went through.

		 
☐	Updating and enforcing sudo permissions	In this step I performed a nano command to set password requirements.First i used apt update && apt install libpam-pwquality -y, then using nano /etc/pam.d/common-password to input the password requirements for all users. The next command I used was pam-auth-update –force to require the users to reset their password that will have the correct password requirements then I tried to create a new password for a user to ensure the password requirement went through.

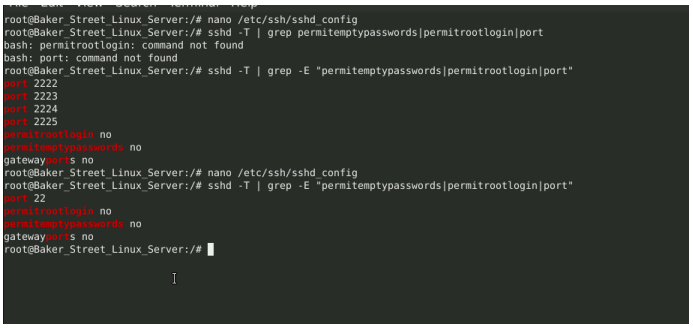
		 
☐	Validating and updating permissions on files and directories	In this part of the project my first task is making sure no files have and world permissions to rwx. With this I ran the command find /home -perm /o=rwx -exec chmod o-rwx {} + to remove the permissions and ran find /home -perm /o=rwx to ensure they were removed. 
☐	Auditing and securing SSH	In the first part I am to configure SSH to not use all the ability to SSH with empty password, root user, and other ports besides 22 which was done by using the command nano /etc/ssh/sshd_config. From there I looked for the text empty passwords and root login to make sure they says “no” and to make sure that the only port accessible is port 22 with no other text followed behind it.

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

```
# Authentication:

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

```
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

To check my file that was saved i ran a command to check my work using sshd -T | grep -E “permitemptypasswords|permitrootlogin|port”.

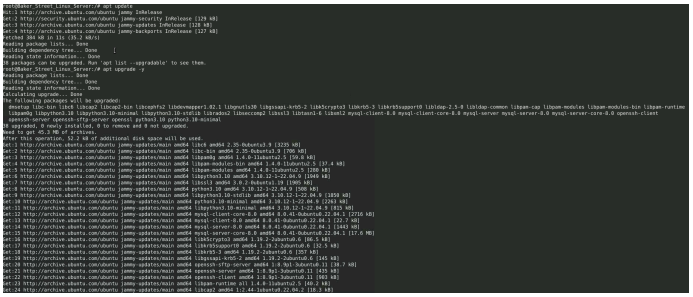
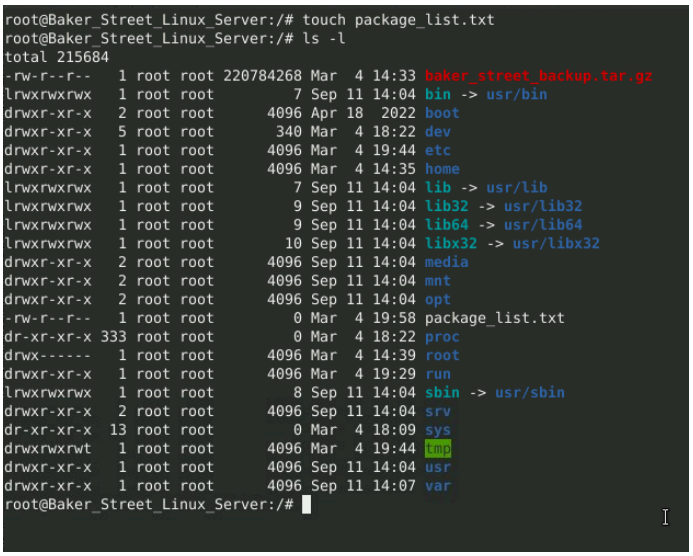


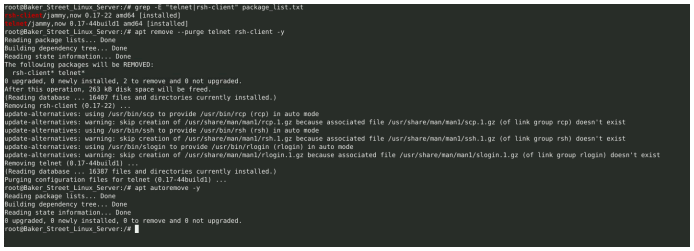
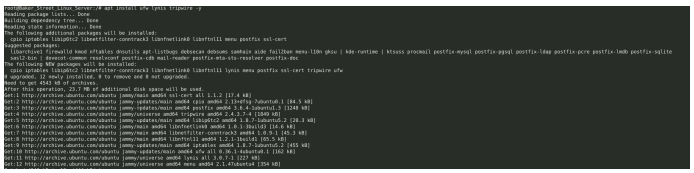
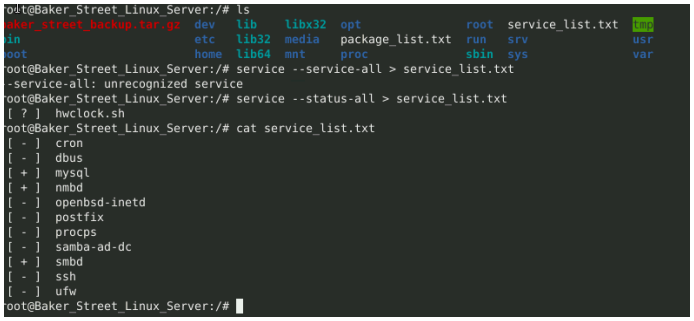
Next I enable the SSHProtocol 2 by configuring the file using command nano /etc/ssh/sshd_config. Protocol 1 was enabled so I changed it to Protocol 2 and saved the file.

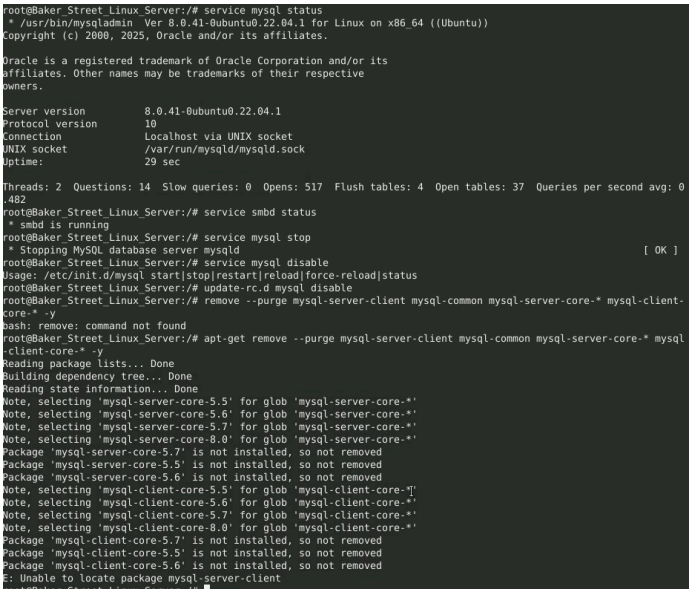
```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
Protocol 2
AllowUsers sherlock watson moriarty mycroft irene lestrade
```

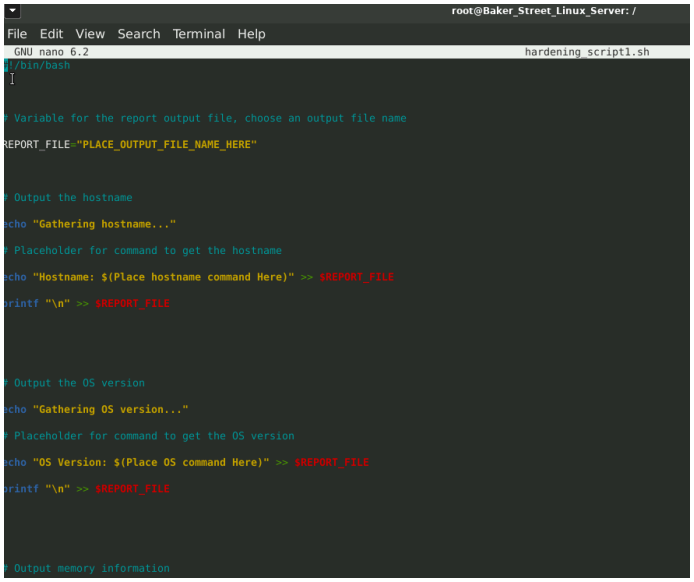
Lastly, I ran the command service ssh restart to set the updates that were recently done.

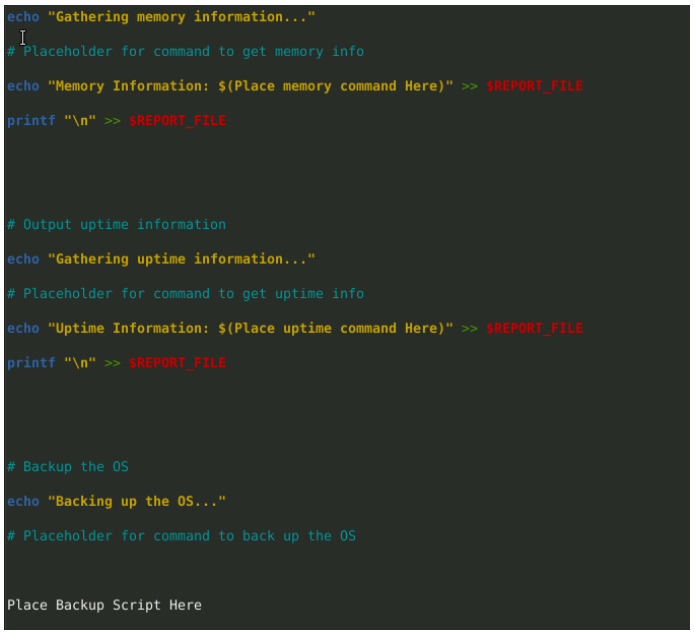
```
root@Baker_Street_Linux_Server:/# service ssh restart
 * Restarting OpenBSD Secure Shell server sshd
root@Baker_Street_Linux_Server:/#
```

☐	Reviewing and updating system packages	To start I ran commands apt update and apt upgrade -y for the latest version of all packages.  Next I created a file for installed packages using the command touch package_list.txt and ls -l to confirm it was successfully created and ran command cat package_list.txt to the view and make sure the installed packages were inside. 
☐	Disabling unnecessary services	Inside of the installed packages there were 2 packages called telnet and rsh-client that could potentially be security issues. Per research I found, they transmit user credentials like username and passwords in plain text over the network. So to uninstall those packages from the file I used the command apt remove –purge telnet rsh-client -y and

		to make sure any dependencies were installed I used command apt autoremove -y. Next I ran apt update followed by apt install ufw lynis tripwire -y to install new packages nfw which is a firewall tool, lynis which is a security auditing tool, and tripwire which is a intrusion detection system into package_list.txt.  
☐	Enabling and configuring logging	My next step is to run command service –status-all to see the list of all services. I then ran touch service_list.txt to create a file to later output all services in using command service –status-all > service_list.txt then ran command cat service_list.txt to ensure the services appeared in that file.   My next action of identifying if any services were running mysql and samba i used command service

		(service name) status and if they are running which they both were i ran a couple commands. To stop i ran service mysql stop, to disable i ran update-rc.d mysql disable, and to remove i ran apt-get remove --purge mysql-server-client mysql-common mysql-server-core-* mysql-client-core-* -y. For Samba I ran the same commands with smbd. 
☐	Scripts created	In my first action, I will be completing some tasks that make a script for my actions taken on day 1. First I ran nano into a file named hardening_script1.sh and copied the appropriate information into the file.


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
Output the OS version
echo "Gathering OS version..."
Placeholder for command to get the OS version
echo "OS Version: $(Place OS command Here)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
Output memory information


echo "Gathering memory information..."
Placeholder for command to get memory info
echo "Memory Information: $(Place memory command Here)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
Output uptime information
echo "Gathering uptime information..."
Placeholder for command to get uptime info
echo "Uptime Information: $(Place uptime command Here)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
Backup the OS
echo "Backing up the OS..."
Placeholder for command to back up the OS
Place Backup Script Here


Placeholder for command to update permissions
Place command here to only allow members of ^`^|engineering ^`^} group to view, edit, and execute all engineering scripts
Here is the example command for the engineering group
find -iname '*engineering*' -exec chown :engineering {} +
echo "Permissions updated for Engineering scripts." >> $REPORT_FILE
printf "\n" >> $REPORT_FILE
Research scripts - Only members of the research group
echo "Updating permissions for Research scripts..."
Placeholder for command to update permissions
Place command here to only allow members of ^`^|research ^`^} group to view, edit, and execute all research scripts. See above script for syntax.
echo "Permissions updated for Research scripts" >> $REPORT_FILE

```
# Output the sudoers file to the report
echo "Gathering sudoers file..."
# Placeholder for command to output sudoers file
echo "Sudoers file:$(Place sudoers display command Here)" >> $REPORT_FILE
printf "\n" >> $REPORT_FILE



# Script to check for files with world permissions and update them
echo "Checking for files with world permissions..."

Place command here to remove all world permissions, starting at the /home directory

# 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..."
```

Next I changed the information that was highlighted in RED to validate the file. In doing so changed the “Report File” to show /var/log/system_hardening_report.txt.

```
REPORT_FILE="/var/log/system_hardening_report.txt"
```

Next I moved to Hostname, OS version, Memory Information, and Uptime. In these options I changed the Report File to reflect the command that was ran as followed; hostname, uname -a, free -h, and uptime. This ensures that when the script runs it will run the correct commands line for those specific commands.

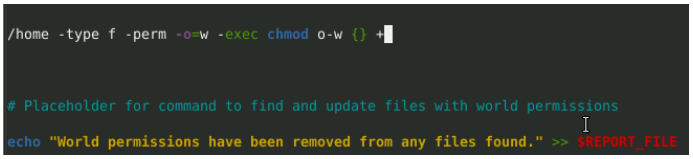
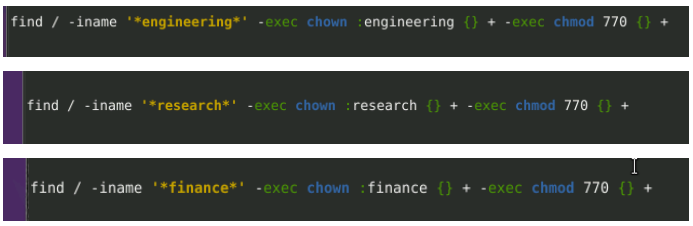
```
echo "Hostname: $(hostname)" >> $REPORT_FILE
```

```
echo "OS Version: $(uname -a)" >> $REPORT_FILE
```

```
echo "Memory Information: $(free -h)" >> $REPORT_FILE
```

```
echo "Uptime Information: $(uptime)" >> $REPORT_FILE
```

Next action is to input the back up for the OS which is ran as -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 /. This is the backup command that will restore the current

		OS that was backed up if anything were to happen to the system.  Next action is to input the command to view the sudoers file and the command is cat /etc/sudoers. This command opens the file to view sudoers permission.  Next action is checking for world permissions and removing them. This command was a combination to find the world permissions and to remove the files. The command to run is /home -type f -perm -o=w exec chmod o-w {} +.  My last action I updated permissions of the engineering, research, and finance scripts to have read, write, and executable permissions. The command for these actions is find / -iname '*engineering*' -exec chown :engineering {} + -exec chmod 770 {} +, find / -iname '*research*' -exec chown :research {} + -exec chmod 770 {} +, find / -iname '*finance*' -exec chown :finance {} + -exec chmod 770 {} +.  After completing the nano file I chmod +x hardening_script1.sh to ensure the program can run as a executable and sudo ./hardening_script1.sh to ensure the script ran properly.

```
root@Baker_Street_Linux_Server:/# chmod +x hardening_script1.sh
root@Baker_Street_Linux_Server:/# nano hardening_script1.sh
root@Baker_Street_Linux_Server:/# nano hardening_script1.sh
root@Baker_Street_Linux_Server:/# sudo ./hardening_script1.sh
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
Gathering hostname...
Gathering OS version...
Gathering memory information...
Gathering uptime information...
Backing up the OS...
./hardening_script1.sh: line 76: -tar: command not found
Gathering sudoers file...
Checking for files with world permissions...
./hardening_script1.sh: line 109: /home: Is a directory
Updating permissions for specific scripts...
Updating permissions for Engineering scripts.
find: '/proc/57/task/57/fdinfo': Permission denied
find: '/proc/57/map_files': Permission denied
find: '/proc/57/fdinfo': Permission denied
find: '/proc/208/task/208/fdinfo': Permission denied
find: '/proc/208/task/211/fdinfo': Permission denied
find: '/proc/208/task/212/fdinfo': Permission denied
find: '/proc/208/task/213/fdinfo': Permission denied
find: '/proc/208/task/214/fdinfo': Permission denied
find: '/proc/208/task/215/fdinfo': Permission denied
find: '/proc/208/task/216/fdinfo': Permission denied
find: '/proc/208/task/217/fdinfo': Permission denied
find: '/proc/208/task/218/fdinfo': Permission denied
find: '/proc/208/task/219/fdinfo': Permission denied
find: '/proc/208/task/220/fdinfo': Permission denied
find: '/proc/208/task/222/fdinfo': Permission denied
find: '/proc/208/task/223/fdinfo': Permission denied
find: '/proc/208/task/224/fdinfo': Permission denied
find: '/proc/208/task/225/fdinfo': Permission denied
find: '/proc/208/task/226/fdinfo': Permission denied
find: '/proc/208/task/227/fdinfo': Permission denied
find: '/proc/208/task/241/fdinfo': Permission denied
find: '/proc/208/task/242/fdinfo': Permission denied
find: '/proc/208/task/243/fdinfo': Permission denied
find: '/proc/208/task/244/fdinfo': Permission denied
find: '/proc/208/task/245/fdinfo': Permission denied
find: '/proc/208/task/246/fdinfo': Permission denied
find: '/proc/208/task/247/fdinfo': Permission denied
```

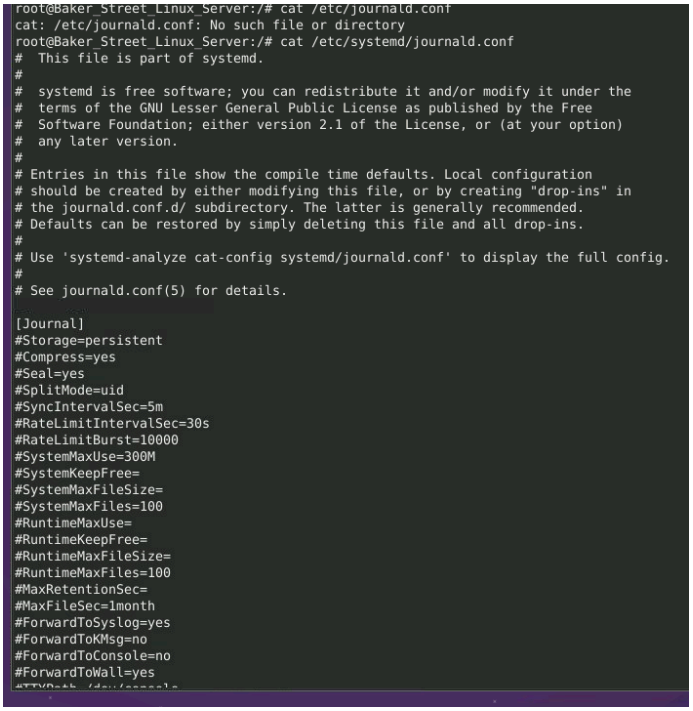
The next part of day 3 is running a second nano script. For this I ran nano hardening_script2.sh. I copied the template into this hardening file and edited the file as needed. First action was changing the Report File to /var/log/system_hardening_report2.txt.

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

REPORT_FILE="/var/log/system_hardening_report2.txt"
```

In my next action for the sshd configuration file, the command i input was cat /etc/ssh/sshd_config. This is to read and display the contents inside of this file.

```
# Output the sshd configuration file

echo "Gathering details from sshd configuration file"

cat /etc/ssh/sshd_config
```

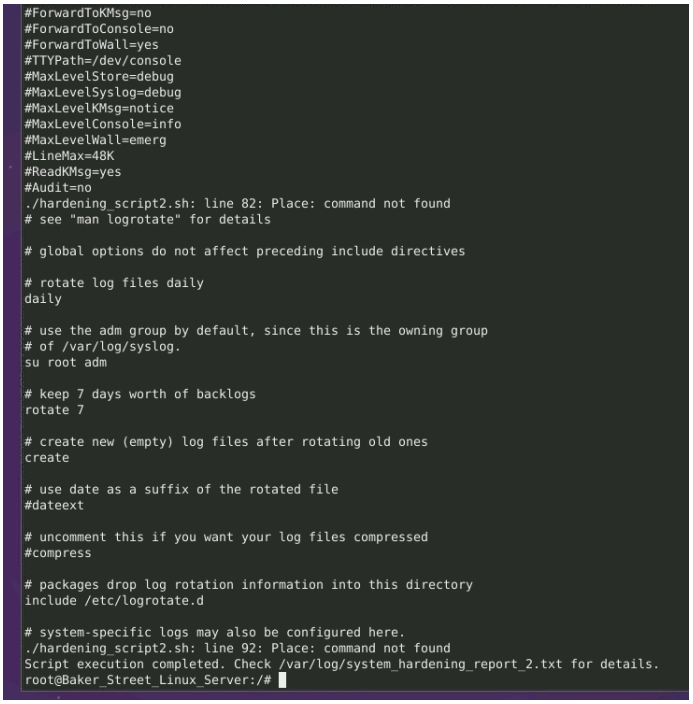
My next action is to input update and upgrade commands. The commands that are to be ran is apt update -y and apt upgrade -y. The command apt update -y is to update the list of packages and their versions from configured repositories and -y is to automatically confirm the update/ And the command apt upgrade -y is to install all the latest upgradable packages on the system and -y is to continue without requiring confirmation.

		`` apt update -y Place Update Packages Command Here ``` ``` apt upgrade -y Place Upgrade Packages Command Here ``` My next action is to input the command to view the journald.conf and logrotate.conf file. To view file journald.conf i used command cat /etc/systemd/journald.conf. After some research after running cat /etc/journald.conf and no file coming up I used the first command because this configuration file was found in the systemd file.  root@Baker_Street_Linux_Server:/# cat /etc/journald.conf cat: /etc/journald.conf: No such file or directory root@Baker_Street_Linux_Server:/# cat /etc/systemd/journald.conf # This file is part of systemd. # # systemd is free software; you can redistribute it and/or modify it under the # terms of the GNU Lesser General Public License as published by the Free # Software Foundation; either version 2.1 of the License, or (at your option) # any later version. # # Entries in this file show the compile time defaults. Local configuration # should be created by either modifying this file, or by creating "drop-ins" in # the journald.conf.d/ subdirectory. The latter is generally recommended. # Defaults can be restored by simply deleting this file and all drop-ins. # # Use 'systemd-analyze cat-config systemd/journald.conf' to display the full config. # # See journald.conf(5) for details. [Journal] #Storage=persistent #Compress=yes #Seal=yes #SplitMode=uid #SyncIntervalSec=5m #RateLimitIntervalSec=30s #RateLimitBurst=10000 #SystemMaxUse=300M #SystemKeepFree= #SystemMaxFileSize= #SystemMaxFiles=100 #RuntimeMaxUse= #RuntimeKeepFree= #RuntimeMaxFileSize= #RuntimeMaxFiles=100 #MaxRetentionSec= #MaxFileSec=1month #ForwardToSyslog=yes #ForwardToKMsg=no #ForwardToConsole=no #ForwardToWall=yes The second command was cat /etc/logrotate.conf.

```
root@Baker_Street_Linux_Server:/# cat /etc/logrotate.conf
# see "man logrotate" for details

# global options do not affect preceding include directives

# 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

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
#dateext

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# system-specific logs may also be configured here.
```

In my final action I ran the command chmod +x hardening_script2.sh so that the program is able to run as an executable and sudo ./hardening_script1.sh to ensure the script ran properly.

```
root@Baker_Street_Linux_Server:/# chmod +x hardening_script2.sh
root@Baker_Street_Linux_Server:/# sudo ./hardening_script2.sh
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
Gathering details from sshd configuration file

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

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

#PubkeyAuthentication yes
```

		
☐	Scripts scheduled with cron	