



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
<input type="checkbox"/>	OS backup	<pre>-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 /</pre>

		<pre> root@Baker Street Linux Server:/# sudo tar -xvpzf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc --exclude -/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run / sudo: unable to resolve host Baker Street Linux Server: Temporary failure in name resolution tar: Removing leading '/' from member names / ./bin ./home/ ./home/adler/ ./home/adler/.bashrc ./home/adler/.bash_logout ./home/adler/.profile ./home/adler/Engineering_script.sh script1.sh ./home/adler/Engineering_script.sh script2.sh ./home/adler/Engineering_script.sh 0.txt ./home/adler/Engineering_script.sh 3.txt ./home/adler/deduction.doc 2.txt ./home/adler/game_is_afoot.txt 1.txt ./home/sherlock/ ./home/sherlock/.bashrc ./home/sherlock/.bash_logout ./home/sherlock/.profile ./home/sherlock/deduction.doc script1.sh ./home/sherlock/deduction.doc script2.sh ./home/sherlock/my_file.txt ./home/sherlock/deduction.doc 3.txt ./home/sherlock/game_is_afoot.txt 2.txt ./home/sherlock/elementary.txt 0.txt ./home/sherlock/game_is_afoot.txt 1.txt ./home/moriarty/ ./home/moriarty/.bashrc ./home/moriarty/.bash_logout ./home/moriarty/.profile ./home/moriarty/game_is_afoot.txt script2.sh ./home/moriarty/game_is_afoot.txt script1.sh ./home/moriarty/Finance_script.sh 2.txt ./home/moriarty/my_file.txt ./home/moriarty/elementary.txt 1.txt ./home/moriarty/game_is_afoot.txt 3.txt ./home/moriarty/Finance_script.sh 0.txt ./home/mycroft/ ./home/mycroft/.bashrc ./home/mycroft/.bash_logout ./home/mycroft/.profile ./home/mycroft/Finance_script.sh script2.sh </pre>
<input type="checkbox"/>	<p>Auditing users and groups</p>	<p>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 <code>userdel -r (username)</code>. This removed the user and the home directory.</p> <pre> root@Baker Street Linux Server:/# sudo userdel -r lestrade sudo: unable to resolve host Baker Street Linux Server: Temporary failure in name resolution userdel: lestrade mail spool (/var/mail/lestrade) not found root@Baker Street Linux Server:/# sudo userdel -r irene sudo: unable to resolve host Baker Street Linux Server: Temporary failure in name resolution userdel: irene mail spool (/var/mail/irene) not found root@Baker Street Linux Server:/# sudo userdel -r mary sudo: unable to resolve host Baker Street Linux Server: Temporary failure in name resolution userdel: mary mail spool (/var/mail/mary) not found root@Baker Street Linux Server:/# sudo userdel -r gregson sudo: unable to resolve host Baker Street Linux Server: Temporary failure in name resolution userdel: gregson mail spool (/var/mail/gregson) not found root@Baker Street Linux Server:/# ls -l /home/ total 32 drwxr-x--- 1 adler adler 4096 Dec 12 07:45 adler drwxr-x--- 1 moriarty moriarty 4096 Dec 12 07:45 moriarty drwxr-x--- 1 mrs_hudson mrs_hudson 4096 Dec 12 07:45 mrs_hudson drwxr-x--- 1 mycroft mycroft 4096 Dec 12 07:45 mycroft drwxr-x--- 1 sherlock sherlock 4096 Dec 12 07:45 sherlock drwxr-x--- 2 sysadmin sysadmin 4096 Dec 12 07:45 sysadmin drwxr-x--- 1 toby toby 4096 Dec 12 07:45 toby drwxr-x--- 1 watson watson 4096 Dec 12 07:45 watson </pre> <p>In the next step locking all users that are on leave was simply using the command <code>passwd -l (username)</code>. This locked the account so it cannot be used while they're away.</p> <pre> 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:/# </pre> <p>Then unlocking users that are employed used the command <code>passwd -u (username)</code>.</p>

		<pre> 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:/# 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:/# 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:/# </pre> <p>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.</p> <pre> root@Baker_Street_Linux_Server:/# groupadd research root@Baker_Street_Linux_Server:/# getent group research research:x:1015: root@Baker_Street_Linux_Server:/# groups adler adler : adler root@Baker_Street_Linux_Server:/# groups moriarty moriarty : moriarty engineering root@Baker_Street_Linux_Server:/# groups mrs_hudson mrs_hudson : mrs_hudson finance root@Baker_Street_Linux_Server:/# groups mycroft mycroft : mycroft root@Baker_Street_Linux_Server:/# groups sherlock sherlock : sherlock engineering root@Baker_Street_Linux_Server:/# groups toby toby : toby root@Baker_Street_Linux_Server:/# groups watson watson : watson engineering root@Baker_Street_Linux_Server:/# </pre>
<input type="checkbox"/>	Updating and enforcing password policies	<p>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.</p>

		<pre> root@Baker_Street_Linux_Server:~# nano /etc/pam.d/common-password /etc/pam.d/common-password - password-related modules common to all services # # This file is included from other service-specific PAM config files, # and should contain a list of modules that define the services to be # used to change user passwords. The default is pam_unix. # Explanation of pam_unix options: # The "yescrypt" option enables # hashed passwords using the yescrypt algorithm, introduced in Debian # 11. Without this option, the default is Unix crypt. Prior releases # added the option "sha512"; if a shadow password hash will be shared # between Debian 11 and older releases replace "yescrypt" with "sha512" # for compatibility. The "obscure" option replaces the old # "OBSOLETE_CHECKS_ENABLE" option in login.defs. See the pam_unix manpage # for other options. # 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. # here are the per-package modules (the "Primary" block) password requisite pam_pwquality.so dcredit=1 minlen=8 ocredit=1 retry=2 ucredit=1 password [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass yescrypt # here - the fallback if no module succeeds password requisite pam_deny.so # 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 password required pam_permit.so # and here are more per-package modules (the "Additional" block) # end of pam-auth-update config </pre> <pre> root@Baker_Street_Linux_Server:~# nano /etc/pam.d/common-password root@Baker_Street_Linux_Server:~# pam-auth-update --force root@Baker_Street_Linux_Server:~# passwd username passwd: user 'username' does not exist root@Baker_Street_Linux_Server:~# passwd sherlock New password: BAD PASSWORD: The password contains less than 1 digits Retype new password: </pre>
<input type="checkbox"/>	Updating and enforcing sudo permissions	<p>In this step I performed a nano command to set password requirements. First I used <code>apt update</code> & <code>apt install libpam-pwquality -y</code>, then using nano <code>/etc/pam.d/common-password</code> to input the password requirements for all users. The next command I used was <code>pam-auth-update --force</code> 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.</p>

		<pre> GNU nano 0.2 /etc/pam.d/common-password # # /etc/pam.d/common-password - password-related modules common to all services # # This file is included from other service-specific PAM config files, # and should contain a list of modules that define the services to be # used to change user passwords. The default is pam_unix. # Explanation of pam_unix options: # The "yescrypt" option enables # hashed passwords using the yescrypt algorithm, introduced in Debian # 11. Without this option, the default is Unix crypt. Prior releases # added the option "sha512" if a shadow password hash will be shared # between Debian 11 and older releases replace "yescrypt" with "sha512" # for compatibility. The "obscure" option replaces the old # "OBSOLETE_CHECKS_ENABLE" option in login.defs. See the pam_unix manpage # for other options. # 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. # here are the per-package modules (the "Primary" block) password requisite pam_pwquality.so dcredit=1 minlen=8 ocredit=1 retry=2 ucredit=1 password [success=1 default=ignore] pam_unix.so obscure use_authok try_first_pass yescrypt # here - the fallback if no module succeeds password requisite pam.deny.so # 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 password required pam.permit.so # and here are more per-package modules (the "Additional" block) # end of pam-auth-update config </pre> <pre> root@Baker_Street_Linux_Server:/# nano /etc/pam.d/common-password root@Baker_Street_Linux_Server:/# pam-auth-update --force root@Baker_Street_Linux_Server:/# passwd username passwd: user 'username' does not exist root@Baker_Street_Linux_Server:/# passwd sherlock New password: BAD PASSWORD: The password contains less than 1 digits Retype new password: </pre>
<input type="checkbox"/>	Validating and updating permissions on files and directories	<p>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 <code>find /home -perm /o=rwx -exec chmod o-rwx {} +</code> to remove the permissions and ran <code>find /home -perm /o=rwx</code> to ensure they were removed.</p> <pre> root@Baker_Street_Linux_Server:/# find /home -perm /o=rwx -exec chmod o-rwx {} + root@Baker_Street_Linux_Server:/# find /home -perm /o=rwx root@Baker_Street_Linux_Server:/# </pre>
<input type="checkbox"/>	Auditing and securing SSH	<p>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 <code>nano /etc/ssh/sshd_config</code>. 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.</p>

```
# 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 "peremptypasswords|permitrootlogin|port"`.

```
root@Baker_Street_Linux_Server:~# nano /etc/ssh/sshd_config
root@Baker_Street_Linux_Server:~# sshd -T | grep peremptypasswords|permitrootlogin|port
bash: permitrootlogin: command not found
bash: port: command not found
root@Baker_Street_Linux_Server:~# sshd -T | grep -E "peremptypasswords|permitrootlogin|port"
port 2222
port 2223
port 2224
port 2225
permitrootlogin no
peremptypasswords no
gatewayports no
root@Baker_Street_Linux_Server:~# nano /etc/ssh/sshd_config
root@Baker_Street_Linux_Server:~# sshd -T | grep -E "peremptypasswords|permitrootlogin|port"
port 22
permitrootlogin no
peremptypasswords no
gatewayports no
root@Baker_Street_Linux_Server:~#
```

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:~#
```


		<p>to make sure any dependencies were installed I used command <code>apt autoremove -y</code>. Next I ran <code>apt update</code> followed by <code>apt install ufw lynis tripwire -y</code> to install new packages <code>nfw</code> which is a firewall tool, <code>lynis</code> which is a security auditing tool, and <code>tripwire</code> which is an intrusion detection system into <code>package_list.txt</code>.</p> <pre> root@Baker_Street_Linux_Server:/# grep -i -n install{rah-client} package_list.txt 11 /usr/share/doc/0.17-22 ufwdb [installed] 12 /usr/share/doc/0.17-44 ufwdb [installed] root@Baker_Street_Linux_Server:/# apt remove --purge telnet rah-client -y Reading package lists... Done Building dependency tree... Done Reading state information... Done The following packages will be REMOVED: rah-client telnet 0 upgraded, 0 newly installed, 2 to remove and 0 not upgraded. After this operation, 263 kB disk space will be freed. Removing telnet, 263 kB of disk space will be freed. Removing rah-client, 0.17-22. Removing files and directories currently installed. Removing rah-client (0.17-22) ... update-alternatives: warning: skip creation of /usr/share/man/man1/rp.1.gz because associated file /usr/share/man/man1/rp.1.gz (of link group rp) doesn't exist update-alternatives: warning: /usr/share/man to provide /usr/share/man/rp.1.gz in auto mode update-alternatives: warning: skip creation of /usr/share/man/man1/rh.1.gz because associated file /usr/share/man/man1/rh.1.gz (of link group rh) doesn't exist update-alternatives: warning: /usr/share/man to provide /usr/share/man/rh.1.gz in auto mode update-alternatives: warning: skip creation of /usr/share/man/man1/rhgs.1.gz because associated file /usr/share/man/man1/rhgs.1.gz (of link group rhgs) doesn't exist update-alternatives: warning: /usr/share/man to provide /usr/share/man/rhgs.1.gz in auto mode Removing telnet (0.17-44build1) ... Purging configuration files for telnet (0.17-44build1) ... root@Baker_Street_Linux_Server:/# apt autoremove -y Reading package lists... Done Building dependency tree... Done Reading state information... Done 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded. root@Baker_Street_Linux_Server:/# </pre> <pre> root@Baker_Street_Linux_Server:/# touch service_list.txt touch: cannot touch 'service_list.txt': No such file or directory root@Baker_Street_Linux_Server:/# </pre> <pre> root@Baker_Street_Linux_Server:/# service --status-all [-] cron [-] dbus [?] hwclock.sh [+] mysql [+] nmbd [-] openbsd-inetd [-] postfix [-] procpss [-] samba-ad-dc [+] smbd [-] ssh [-] ufw root@Baker_Street_Linux_Server:/# touch service_list.txt touch: cannot touch 'service_list.txt': No such file or directory root@Baker_Street_Linux_Server:/# </pre> <pre> root@Baker_Street_Linux_Server:/# ls dev lib libx32 apt root service_list.txt etc lib32 media package_list.txt run srv usr home lib64 mnt proc sbinsrv var root@Baker_Street_Linux_Server:/# service --service-all > service_list.txt service-all: unrecognized service root@Baker_Street_Linux_Server:/# service --status-all > service_list.txt [?] hwclock.sh root@Baker_Street_Linux_Server:/# cat service_list.txt [-] cron [-] dbus [+] mysql [+] nmbd [-] openbsd-inetd [-] postfix [-] procpss [-] samba-ad-dc [+] smbd [-] ssh [-] ufw root@Baker_Street_Linux_Server:/# </pre>
<input type="checkbox"/>	<p>Enabling and configuring logging</p>	<p>My next step is to run command <code>service --status-all</code> to see the list of all services. I then ran <code>touch service_list.txt</code> to create a file to later output all services in using command <code>service --status-all > service_list.txt</code> then ran command <code>cat service_list.txt</code> to ensure the services appeared in that file.</p> <pre> root@Baker_Street_Linux_Server:/# service --status-all [-] cron [-] dbus [?] hwclock.sh [+] mysql [+] nmbd [-] openbsd-inetd [-] postfix [-] procpss [-] samba-ad-dc [+] smbd [-] ssh [-] ufw root@Baker_Street_Linux_Server:/# touch service_list.txt touch: cannot touch 'service_list.txt': No such file or directory root@Baker_Street_Linux_Server:/# </pre> <pre> root@Baker_Street_Linux_Server:/# ls dev lib libx32 apt root service_list.txt etc lib32 media package_list.txt run srv usr home lib64 mnt proc sbinsrv var root@Baker_Street_Linux_Server:/# service --service-all > service_list.txt service-all: unrecognized service root@Baker_Street_Linux_Server:/# service --status-all > service_list.txt [?] hwclock.sh root@Baker_Street_Linux_Server:/# cat service_list.txt [-] cron [-] dbus [+] mysql [+] nmbd [-] openbsd-inetd [-] postfix [-] procpss [-] samba-ad-dc [+] smbd [-] ssh [-] ufw root@Baker_Street_Linux_Server:/# </pre> <p>My next action of identifying if any services were running <code>mysql</code> and <code>samba</code> i used command <code>service</code></p>

		<p>(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 smb.d.</p> <pre>root@Baker-Street-Linux-Server:/# service mysql status * /usr/bin/mysqldadmin Ver 8.0.41-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu)) Copyright (c) 2000, 2025, Oracle and/or its affiliates. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. Server version 8.0.41-0ubuntu0.22.04.1 Protocol version 10 Connection Localhost via UNIX socket UNIX socket /var/run/mysqld/mysqld.sock Uptime: 29 sec Threads: 2 Questions: 14 Slow queries: 0 Opens: 517 Flush tables: 4 Open tables: 37 Queries per second avg: 0.482 root@Baker-Street-Linux-Server:/# service smbd status * smbd is running root@Baker-Street-Linux-Server:/# service mysql stop * Stopping MySQL database server mysqlql [OK] root@Baker-Street-Linux-Server:/# service mysql disable Usage: /etc/init.d/mysql start stop restart reload force-reload status root@Baker-Street-Linux-Server:/# update-rc.d mysql disable root@Baker-Street-Linux-Server:/# remove --purge mysql-server-client mysql-common mysql-server-core-* mysql-client-core-* -y bash: remove: command not found root@Baker-Street-Linux-Server:/# apt-get remove --purge mysql-server-client mysql-common mysql-server-core-* mysql-client-core-* -y Reading package lists... Done Building dependency tree... Done Reading state information... Done Note, selecting 'mysql-server-core-5.5' for glob 'mysql-server-core-*' Note, selecting 'mysql-server-core-5.6' for glob 'mysql-server-core-*' Note, selecting 'mysql-server-core-5.7' for glob 'mysql-server-core-*' Note, selecting 'mysql-server-core-8.0' for glob 'mysql-server-core-*' Package 'mysql-server-core-5.7' is not installed, so not removed Package 'mysql-server-core-5.5' is not installed, so not removed Package 'mysql-server-core-5.6' is not installed, so not removed Note, selecting 'mysql-client-core-5.5' for glob 'mysql-client-core-?' Note, selecting 'mysql-client-core-5.6' for glob 'mysql-client-core-*' Note, selecting 'mysql-client-core-5.7' for glob 'mysql-client-core-*' Note, selecting 'mysql-client-core-8.0' for glob 'mysql-client-core-*' Package 'mysql-client-core-5.7' is not installed, so not removed Package 'mysql-client-core-5.5' is not installed, so not removed Package 'mysql-client-core-5.6' is not installed, so not removed E: Unable to locate package mysql-server-client E: Couldn't find any package by regex 'mysql-server-client' root@Baker-Street-Linux-Server:/# root@Baker-Street-Linux-Server:/# service smbd stop * Stopping SMB/CIFS daemon smbd [OK] root@Baker-Street-Linux-Server:/# update-rc.d smbd disable root@Baker-Street-Linux-Server:/# apt-get remove --purge smbd-server-client smbd-common smbd-server-core-* smbd-client-core-* -y Reading package lists... Done Building dependency tree... Done Reading state information... Done E: Unable to locate package smbd-server-client E: Unable to locate package smbd-common E: Unable to locate package smbd-server-core-* E: Couldn't find any package by glob 'smbd-server-core-*' E: Couldn't find any package by regex 'smbd-server-core-*' E: Unable to locate package smbd-client-core-* E: Couldn't find any package by glob 'smbd-client-core-*' E: Couldn't find any package by regex 'smbd-client-core-*' root@Baker-Street-Linux-Server:/#</pre>
<input type="checkbox"/>	Scripts created	<p>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.</p>

		<div><div>root@Baker_Street_Linux_Server: /</div><div>File Edit View Search Terminal Help</div><div>GNU nano 6.2hardening script1.sh</div><div><pre>#!/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 █ ind -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</pre></div></div>
--	--	---

```

# 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

		<p>OS that was backed up if anything were to happen to the system.</p> <pre>hardening_tar_command=\$(cat /dev/urandom tr -dc 'a-z0-9' fold -w 64 xargs sha256sum sed 's/^[a-f0-9]* //')</pre> <pre>tar -czpf /baker_street_backup.tar.gz --exclude=/baker_street_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run /</pre> <pre>echo "OS backup completed." >> \$REPORT_FILE</pre> <p>Next action is to input the command to view the sudoers file and the command is <code>cat /etc/sudoers</code>. This command opens the file to view sudoers permission.</p> <pre>echo "Sudoers file:\$(cat /etc/sudoers)" >> \$REPORT_FILE</pre> <pre>printf "\n" >> \$REPORT_FILE</pre> <p>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 <code>/home -type f -perm -o=w exec chmod o-w {} +</code>.</p> <pre>/home -type f -perm -o=w -exec chmod o-w {} +</pre> <pre># Placeholder for command to find and update files with world permissions</pre> <pre>echo "World permissions have been removed from any files found." >> \$REPORT_FILE</pre> <p>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 <code>find / -iname '*engineering*' -exec chown :engineering {} + -exec chmod 770 {} +</code>, <code>find / -iname '*research*' -exec chown :research {} + -exec chmod 770 {} +</code>, <code>find / -iname '*finance*' -exec chown :finance {} + -exec chmod 770 {} +</code>.</p> <pre>find / -iname '*engineering*' -exec chown :engineering {} + -exec chmod 770 {} +</pre> <pre>find / -iname '*research*' -exec chown :research {} + -exec chmod 770 {} +</pre> <pre>find / -iname '*finance*' -exec chown :finance {} + -exec chmod 770 {} +</pre> <p>After completing the nano file I <code>chmod +x hardening_script1.sh</code> to ensure the program can run as a executable and <code>sudo ./hardening_script1.sh</code> to ensure the script ran properly.</p>
--	--	--

```

root@Baker-Street Linux Server:~# chmod +x 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
#TTYBack=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

```

		<pre>#ForwardToKMsg=no #ForwardToConsole=no #ForwardToWall=yes #TTYPath=/dev/console #MaxLevelStore=debug #MaxLevelSyslog=debug #MaxLevelKMsg=notice #MaxLevelConsole=info #MaxLevelWall=emerg #LineMax=48K #ReadKMsg=yes #Audit=no ./hardening_script2.sh: line 82: Place: command not found # 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. ./hardening_script2.sh: line 92: Place: command not found Script execution completed. Check /var/log/system_hardening_report_2.txt for details. root@Baker_Street_Linux_Server:/#</pre>
<input type="checkbox"/>	Scripts scheduled with cron	