



Cybersecurity

Project 1 Hardening Summary and Checklist

OS Information

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

Checklist

Completed	Activity	Script(s) used / Tasks completed / Screenshots
<input checked="" type="checkbox"/>	OS backup	<ul style="list-style-type: none">○ <code>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</code> / Easy enough, made a .tar with no complications. <pre>b0t@Baker_Street_Linux_Server:/# 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</pre>



Auditing users and groups

- `cat /etc/passwd`

```
root@Baker_Street_Linux_Server:/home# cd ..
root@Baker_Street_Linux_Server:# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:1:bin:/bin:/usr/sbin/nologin
sync:x:3:1:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:103:104:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:104:106::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:105:107:Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:106:108::/home/syslog:/usr/sbin/nologin
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
sherlock:x:1000::/home/sherlock:/bin/bash
watson:x:1001:1001::/home/watson:/bin/bash
moriarty:x:1002:1002::/home/moriarty:/bin/bash
mycroft:x:1003:1003::/home/mycroft:/bin/bash
irene:x:1004:1004::/home/irene:/bin/bash
testrade:x:1005:1005::/home/testrade:/bin/bash
mrs_hudson:x:1006:1006::/home/mrs_hudson:/bin/bash
mary:x:1007:1007::/home/mary:/bin/bash
sysadmin:x:1008:1008::/home/sysadmin:/bin/bash
gregson:x:1009:1009::/home/gregson:/bin/bash
toby:x:1010:1010::/home/toby:/bin/bash
adler:x:1011:1011::/home/adler:/bin/bash
root@Baker_Street_Linux_Server:/#
```

- `sudo userdel -r (testrade, Irene, mary, gregson)` This permanently deletes the user AND their home directories for all terminated employees.
- `sudo usermod -L (moriarty and mrs hudson)` This locks user accounts, you can go back and verify if a user has been locked or not by using `sudo grep (username) /etc/shadow` If they are locked it will show an ! before their name in the shadow file.

```
moriarty:!$y$J9T$vfq0y3pMGlckNuU0VnrXa0/$LuVpA0fp96reAbmDHevRn/RV5eCw90rMv7oS0yKc56.:20095:0:99999:7::
```

- Employees Toby and Adler were locked so I used `sudo usermod -p LOGIN (toby, adler)` to unlock them and give them the temp password until they would set a new one.

```
root@Baker_Street_Linux_Server:# cat /etc/shadow
root::19977:0:99999:7:::
daemon::19977:0:99999:7:::
bin::19977:0:99999:7:::
sys::19977:0:99999:7:::
sync::19977:0:99999:7:::
games::19977:0:99999:7:::
man::19977:0:99999:7:::
lp::19977:0:99999:7:::
mail::19977:0:99999:7:::
news::19977:0:99999:7:::
uucp::19977:0:99999:7:::
proxy::19977:0:99999:7:::
www-data::19977:0:99999:7:::
backup::19977:0:99999:7:::
list::19977:0:99999:7:::
irc::19977:0:99999:7:::
gnats::19977:0:99999:7:::
nobody::19977:0:99999:7:::
root@Baker_Street_Linux_Server:# systemd-network::20069:0:99999:7:::
systemd-resolve::20069:0:99999:7:::
mysql::20069:0:99999:7:::
messagebus::20069:0:99999:7:::
systemd-timesync::20069:0:99999:7:::
syslog::20069:0:99999:7:::
sshd::20069:0:99999:7:::
root@Baker_Street_Linux_Server:# moriarty:sys$9T$openovok4U.Sel9KMr52I:$GnQ5dExyA0j0wZn1FtAoAsTjFNu68d4b9wPjbsvr2P.:20095:0:99999:7:::
watson:sys$9T$Typh0u9rTouWgSD8goz/$NTN1rlw/kaP5kongye.iafkZ5v1mJlIzzwlkt/Ibt5:20095:0:99999:7:::
moriarty:sys$9T$openovok4U.Sel9KMr52I:$GnQ5dExyA0j0wZn1FtAoAsTjFNu68d4b9wPjbsvr2P.:20095:0:99999:7:::
mycroft:sys$9T$7f13NnP9RqRp3H4wyGfd/$WusfJKNS1BlwmqoDXGybkmNhNuotYLyxdLGFE38BRD6:20095:0:99999:7:::
mrs_hudson::20069:0:99999:7:::
sysadmin::20069:0:99999:7:::
toby:LOGIN:20095:0:99999:7:::
adler:LOGIN:20095:0:99999:7:::
```

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

```
root@Baker Street Linux_Server:/home# getent group marketing
marketing:x:1014:
```

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

```
root@Baker_Street_Linux_Server:# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:
audio:x:29:
dip:x:30:
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
mysql:x:104:
crontab:x:105:
messagebus:x:106:
systemd-timesync:x:107:
syslog:x:108:
rdma:x:109:
ssh:x:110:
sambashare:x:111:
sherlock:x:1000:
watson:x:1001:
moriarty:x:1002:
mycroft:x:1003:
mrs_hudson:x:1006:
sysadmin:x:1008:
toby:x:1010:
adler:x:1011:
engineering:x:1012:sherlock,watson,moriarty
finance:x:1013:mrs_hudson
Research:x:1014:toby,watson,mycroft
```

<input checked="" type="checkbox"/>	<p>Updating and enforcing password policies</p>	<ul style="list-style-type: none"> Fairly simple step, first just have to use <code>nano /etc/pam.d/common-password</code> followed by adding the <code>password requisite pam_pwquality.so</code> to the common-password file, followed by <code>minlen=8 ocredit=-1 retry=2 ucredit=-1</code> Finish it off by saving with <code>CTRL+O</code> then closing with <code>CTRL+X</code> <pre> # /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 "vescrypt" option enables # hashed passwords using the vescrypt algorithm, introduced in Debian # #11. Without this option, the default is Unix crypt. Prior releases # used the option "sha512"; if a shadow password hash will be shared # between Debian #11 and older releases replace "vescrypt" with "sha512". # For details see /usr/share/doc/pam-auth-update/NEWS.Debian.1 # The "obscure" option replaces the old # "OBSCURE CHECKS ENAB" 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 [success=1 default=ignore] pam_unix.so obscure vescrypt password requisite pam_deny.so password requisite pam_pwquality.so minlen=8 ocredit=-1 retry=2 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 password required pam_permit.so # and here are more per-package modules (the "Additional" block) # end of pam-auth-update config </pre>
<input checked="" type="checkbox"/>	<p>Updating and enforcing sudo permissions</p>	<ul style="list-style-type: none"> To access the sudoers file all I had to do was run the <code>visudo</code> and once in I noticed some things out of place. Sherlock was correct and had full privilege, but Watson is not supposed to have full access, simply sudo to run the script <code>/var/log/logcleanup.sh</code>, so I changed the <code>NOPASSWD=ALL</code> to <code>NOPASSWD= /var/log/logcleanup.sh</code>. Moriarty was full privilege, so for him it was just a matter of removing him off the sudoers using <code>CTRL+K</code> to remove the full line efficiently. Finally I added the Research Group to be able to use sudo for <code>/tmp/scripts/research_script.sh</code> by simply adding them to the sudo file using the % to signify it as a group and using the same idea as before, screenshots as followed.

```

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

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:sudo env_keep += "EDITOR"
# Completely harmless preservation of a user preference.
#Defaults:sudo env_keep += "GREP_COLOR"
# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_"
# Per-user preferences; root won't have sensible values for them.
#Defaults:sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"
# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"
# Ditto for GPG agent
#Defaults:sudo env_keep += "GPG_AGENT_INFO"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "@include" directives:
@includedir /etc/sudoers.d
sherlock ALL=(ALL) NOPASSWD:ALL
watson ALL=(ALL) NOPASSWD: /var/log/logcleanup.sh
%Research ALL=(ALL) NOPASSWD: /tmp/scripts/research_script.sh

```

<input checked="" type="checkbox"/>	<p>Validating and updating permissions on files and directories</p>	<ul style="list-style-type: none"> First thing I did was <code>cd</code> into <code>/home/</code> and using <code>ls -l</code> list all directories within <code>/home/</code>, after that I used <code>chmod -R o-rwx (directory)</code> to recursively remove all world permissions on every file within every home directory. <pre> 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 root@Baker_Street_Linux_Server:/home# chmod -R o-rwx adler/ root@Baker_Street_Linux_Server:/home# chmod -R o-rwx moriarty/ root@Baker_Street_Linux_Server:/home# chmod -R o-rwx mrs_hudson/ root@Baker_Street_Linux_Server:/home# chmod -R o-rwx mycroft/ root@Baker_Street_Linux_Server:/home# chmod -R o-rwx sherlock/ root@Baker_Street_Linux_Server:/home# chmod -R o-rwx sysadmin/ root@Baker_Street_Linux_Server:/home# chmod -R o-rwx toby/ root@Baker_Street_Linux_Server:/home# chmod -R o-rwx watson/ </pre> Next I went from directory to directory changing ownership of scripts and files to their respective groups using <code>sudo chown :(group) (file)</code> <pre> root@Baker_Street_Linux_Server:# cd home root@Baker_Street_Linux_Server:/home# cd adler/ root@Baker_Street_Linux_Server:/home/adler# ls -l total 8 -rw-r----- 1 root engineering 0 Dec 12 07:45 Engineering script.sh 0.txt -rw-r----- 1 root root 0 Dec 12 07:45 Engineering script.sh 3.txt -rw-r----- 1 root root 46 Dec 12 07:45 Engineering script.sh script1.sh -rw-r----- 1 root root 46 Dec 12 07:45 Engineering script.sh script2.sh -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc 2.txt -rw-r----- 1 root root 0 Dec 12 07:45 game_is_afoot.txt 1.txt root@Baker_Street_Linux_Server:/home/adler# sudo chown :engineering Engineering script.sh 3.txt sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/adler# sudo chown :engineering Engineering script.sh script1.sh sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/adler# sudo chown :engineering Engineering script.sh script2.sh sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution </pre>

	<pre>root@Baker_Street_Linux_Server:/home/moriarty# ls -l total 8 -rw-r----- 1 root finance 0 Dec 12 07:45 Finance_script.sh_0.txt -rw-r----- 1 root finance 0 Dec 12 07:45 Finance_script.sh_2.txt</pre> <pre>root@Baker_Street_Linux_Server:/home# ls -l mrs_hudson/ total 8 -rw-r----- 1 root root 0 Dec 12 07:45 Engineering_script.sh_1.txt -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_0.txt -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_2.txt -rw-r----- 1 root root 0 Dec 12 07:45 elementary.txt_3.txt -rwxr-x--- 1 root root 0 Dec 12 07:45 elementary.txt_script1.sh -rwxr-x--- 1 root root 51 Dec 12 07:45 elementary.txt_script2.sh root@Baker_Street_Linux_Server:/home# cd mrs_hudson/ root@Baker_Street_Linux_Server:/home/mrs_hudson# sudo chown :engineering Engineering_script.sh_1.txt sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/mrs_hudson#</pre>
	<pre>root@Baker_Street_Linux_Server:/home# ls -l mycroft/ total 8 -rw-r----- 1 root root 0 Dec 12 07:45 Engineering_script.sh_0.txt -rw-r----- 1 root root 0 Dec 12 07:45 Finance_script.sh_3.txt -rwxr-x--- 1 root root 48 Dec 12 07:45 Finance_script.sh_script1.sh -rwxr-x--- 1 root root 48 Dec 12 07:45 Finance_script.sh_script2.sh -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_1.txt -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_2.txt root@Baker_Street_Linux_Server:/home# cd mycroft/ root@Baker_Street_Linux_Server:/home/mycroft# sudo chown :engineering Engineering_script.sh_0.txt sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/mycroft# sudo chown :finance Finance_script.sh_3.txt sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/mycroft# sudo chown :finance Finance_script.sh_script1.sh sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/mycroft# sudo chown :finance Finance_script.sh_script2.sh sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/mycroft#</pre>
	<pre>total 8 -rw-r----- 1 root root 0 Dec 12 07:45 Engineering_script.sh_2.txt -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_1.txt -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_2.txt -rw-r----- 1 root root 0 Dec 12 07:45 elementary.txt_3.txt -rwxr-x--- 1 root root 45 Dec 12 07:45 elementary.txt_script1.sh -rwxr-x--- 1 root root 45 Dec 12 07:45 elementary.txt_script2.sh root@Baker_Street_Linux_Server:/home/toby# sudo chown :engineering Engineering_script.sh_2.txt sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/toby#</pre>
	<pre>total 8 -rw-r----- 1 root root 0 Dec 12 07:45 Finance_script.sh_3.txt -rwxr-x--- 1 root root 47 Dec 12 07:45 Finance_script.sh_script1.sh -rwxr-x--- 1 root root 47 Dec 12 07:45 Finance_script.sh_script2.sh -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_0.txt -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_1.txt -rw-r----- 1 root root 0 Dec 12 07:45 deduction.doc_2.txt -rw-r----- 1 root root 0 Dec 12 07:45 my_file.txt root@Baker_Street_Linux_Server:/home/watson# sudo chown :finance Finance_script.sh_3.txt sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/watson# sudo chown :finance Finance_script.sh_script1.sh sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/watson# sudo chown :finance Finance_script.sh_script2.sh sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution root@Baker_Street_Linux_Server:/home/watson#</pre>

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

<input checked="" type="checkbox"/>	Auditing and securing SSH	<ul style="list-style-type: none"> First to configure the SSH file we need to <code>sudo nano /etc/ssh/sshd_config</code> to open the <code>sshd_config</code> file for editing. First thing to harden is the ability to disable empty passwords by changing <code>PermitEmptyPasswords yes</code> to <code>PermitEmptyPasswords no</code>. <pre># To disable tunneled clear text passwords, change to no here! #PasswordAuthentication yes PermitEmptyPasswords no</pre> <ul style="list-style-type: none"> Next I disabled root login, very important of course, by changing <code>PermitRootLogin</code> to <code>PermitRootLogin no</code>. <pre>#LoginGraceTime 2m PermitRootLogin no #StrictModes yes #MaxAuthTries 6 #MaxSessions 10</pre> <ul style="list-style-type: none"> I also removed all ports that were NOT <code>port 22</code>, and while I was at it I changed the ssh <code>protocol 1</code> to <code>protocol 2</code>. <pre># Example of overriding settings on a per-user basis #Match User anoncvs # X11Forwarding no # AllowTcpForwarding no # PermitTTY no # ForceCommand cvs server Port 22 Protocol 2</pre> <ul style="list-style-type: none"> To fully cement the updates I applied to the SSH file, I ran <code>service ssh restart</code>. <pre>root@Baker_Street_Linux_Server:/# service ssh restart * Restarting OpenBSD Secure Shell server sshd</pre>
<input checked="" type="checkbox"/>	Reviewing and updating system packages	<ul style="list-style-type: none"> This was actually the first step I did after getting access to the lab, updating all the packages at least. First I ran <code>sudo apt update</code> followed by <code>apt upgrade -y</code>.

```

root@Baker_Street_Linux_Server:/# sudo apt update
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@Baker_Street_Linux_Server:/# sudo apt upgrade -y
sudo: unable to resolve host Baker_Street_Linux_Server: Temporary failure in name resolution
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
libcephfs2 libcephfs2-amd64
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 4342 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get: http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcephfs2 amd64 17.2.7~Ubuntu0.22.04.2 [3594 kB]
Get: http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcephfs2 amd64 17.2.7~Ubuntu0.22.04.2 [748 kB]
Fetched 4342 kB in 1s (4422 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
(Reading database ... 16312 files and directories currently installed.)
Preparing to unpack .../libcephfs2_17.2.7~Ubuntu0.22.04.2_amd64.deb ...
Unpacking libcephfs2 (17.2.7~Ubuntu0.22.04.2) over (17.2.7~Ubuntu0.22.04.1) ...
Preparing to unpack .../libcephfs2_17.2.7~Ubuntu0.22.04.2_amd64.deb ...
Unpacking libcephfs2 (17.2.7~Ubuntu0.22.04.2) over (17.2.7~Ubuntu0.22.04.1) ...
Setting up libcephfs2 (17.2.7~Ubuntu0.22.04.2) ...
Setting up libcephfs2 (17.2.7~Ubuntu0.22.04.2) ...
Processing triggers for libc-bin (2.35~Ubuntu3.8) ...
root@Baker_Street_Linux_Server:/# █

```

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

```

root@Baker_Street_Linux_Server:/home/sysadmin# apt list --installed > package_list.txt
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
root@Baker_Street_Linux_Server:/home/sysadmin# ls -l
total 24
-rw-r--r-- 1 root root 22280 Jan  8 03:19 package_list.txt
root@Baker_Street_Linux_Server:/home/sysadmin# █

rsh-client/jammy,now 0.17-22 amd64 [installed]
rsh-server/jammy,now 0.17-22 amd64 [installed]
rsyslog/jammy-updates,jammy-security,now 8.2112.0-2ubuntu2.2 amd64 [installed]
samba-common-bin/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
samba-common/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 all [installed,automatic]
samba-dsdb-modules/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
samba-vfs-modules/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
samba/jammy-updates,now 2:4.15.13+dfsg-0ubuntu1.6 amd64 [installed,automatic]
sed/jammy,now 4.8-lubuntu2 amd64 [installed]
sensible-utils/jammy,now 0.0.17 all [installed]
shared-mime-info/jammy,now 2.1-2 amd64 [installed,automatic]
ssh-import-id/jammy,now 5.11-0ubuntu1 all [installed,automatic]
sudo/jammy-updates,jammy-security,now 1.9.9-lubuntu2.4 amd64 [installed]
systemd-sysv/jammy-updates,now 249.11-0ubuntu3.12 amd64 [installed,automatic]
systemd-timesyncd/jammy-updates,now 249.11-0ubuntu3.12 amd64 [installed,automatic]
systemd/jammy-updates,now 249.11-0ubuntu3.12 amd64 [installed,automatic]
sysvinit-utils/jammy,now 3.0.1-0ubuntu1 amd64 [installed]
tar/jammy-updates,jammy-security,now 1.34+dfsg-0ubuntu0.1.22.04.2 amd64 [installed]
cpd/jammy,now 7.6.q3lbuild2 amd64 [installed,automatic]
tftp-tools/jammy,now 1.4.5-2build1 amd64 [installed,automatic]
telnet/jammy,now 0.17-44build1 amd64 [installed]
tree/jammy,now 2.0.2-1 amd64 [installed]
ubuntu-keyring/jammy,now 2021.03.26.all [installed]
ucf/jammy,now 3.0043 all [installed,automatic]
update-inetd/jammy,now 4.51 all [installed,automatic]
usmerge/jammy,now 25ubuntu2 all [installed]
util-linux/jammy-updates,jammy-security,now 2.37.2-4ubuntu3.4 amd64 [installed]
wget/jammy-updates,jammy-security,now 1.21.2-2ubuntu1.1 amd64 [installed,automatic]
xauth/jammy,now 1:1.1-lbuild2 amd64 [installed,automatic]
xdg-user-dirs/jammy,now 0.17-2ubuntu4 amd64 [installed,automatic]
xlibig/jammy-updates,jammy-security,now 1:1.2.11.dfsg-2ubuntu9.2 amd64 [installed]
root@Baker_Street_Linux_Server:/home/sysadmin# █

```

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

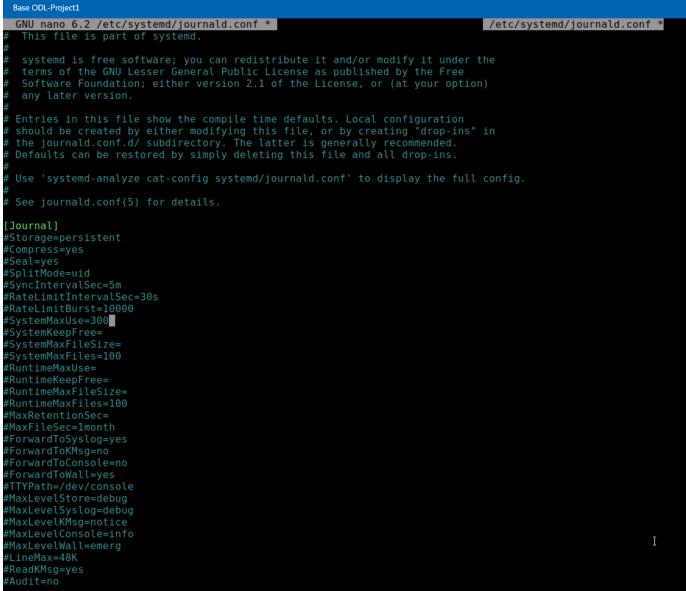
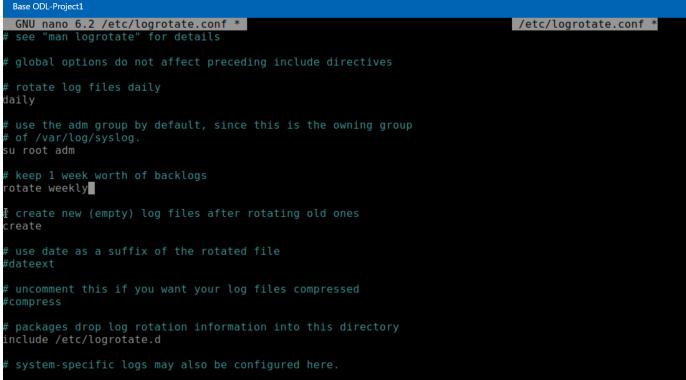
		<p>along with those it is also vulnerable to IP spoofing because it uses a trust based connection, which assumes they cannot be spoofed, which is incorrect.</p> <ul style="list-style-type: none"> These needed to be removed ASAP so I used <code>sudo apt remove --purge telnet rsh-client</code> to fully remove those packages from the system, filled by <code>apt autoremove -y</code> to remove all dependencies from those packages for good. I updated the <code>package_list.txt</code> to reflect the changes, and while I was at it I changed world rwx perms to none, as I don't feel like having that open to the world is good security practice. <pre>root@Baker_Street_Linux_Server:/home/sysadmin# ls root@Baker_Street_Linux_Server:/home/sysadmin# sudo apt remove --purge telnet rsh-client [sudo] password for root: Reading package lists... Done Building dependency tree... Done The following packages will be REMOVED: rsh-client 0.17-4ubuntu1 telnet 0.17-4ubuntu1 0 upgraded, 0 newly installed, 2 to remove and 0 not upgraded. [...] Do you want to continue [Y/n]? y [...] Removing rsh-client (0.17-4ubuntu1) ... [...] Purging configuration files for telnet (0.17-4ubuntu1) [...] Reading package lists... Done Building dependency tree... Done The following packages will be REMOVED: rsh-client 0.17-4ubuntu1 telnet 0.17-4ubuntu1 0 upgraded, 0 newly installed, 2 to remove and 0 not upgraded. [...] root@Baker_Street_Linux_Server:/home/sysadmin#</pre> <pre>root@Baker_Street_Linux_Server:/home/sysadmin# chmod o-rwx package_list.txt root@Baker_Street_Linux_Server:/home/sysadmin# ls -l total 24 -rw-r----- 1 root root 22184 Jan 8 03:43 package_list.txt root@Baker_Street_Linux_Server:/home/sysadmin#</pre>
<input checked="" type="checkbox"/>	Disabling unnecessary services	<ul style="list-style-type: none"> I then went in and used <code>sudo apt install ufw</code>, <code>sudo apt install lynis</code>, and <code>sudo apt install tripwire</code>. I added those packages to the system, I updated the .txt file once more and researched what those add to the system security wise, which you can read down below.

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

```

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

```

<input checked="" type="checkbox"/>	<p>Enabling and configuring logging</p>	<ul style="list-style-type: none"> • To harden this part of the linux server, you have to use <code>sudo nano /etc/systemd/journald.conf</code> • This brings you inside the file to edit, where you have to find where it says <code>storage</code> and change it to <code>storage=persistent</code> and do the same to <code>systemMaxUse=</code> and change it to <code>systemMaxUse=300</code> 
		<ul style="list-style-type: none"> • Next do the same to access <code>logrotate.conf</code> <ul style="list-style-type: none"> ○ Use <code>sudo nano /etc/logrotate.conf</code> ○ Once in, change the log rotation from weekly to daily and rotate out the logs after 7 days. This can be done in the following screenshot. 

<input checked="" type="checkbox"/>	<p>Scripts created</p>	<p>Firstly I will be posting the full scripts underneath this checklist, but will supply the screenshots within the VM. Not a lot to say here despite being the longest part, essentially just a recap of everything above combined with trial and error until my script came out flawlessly.</p>
		<h3>Script 1</h3> <pre>#!/bin/bash # Variables for the report output file, choose an output file name REPORT_FILE="Report_File.txt" # Output the Hostname echo "Gathering hostname..." # This will now output the hostnames into the report file echo Hostname:"`hostname`" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Output the OS version... echo "Gathering OS version..." # This will now output the OS Name and Version to plug into the Report File OS_NAME=\$(grep "NAME=" /etc/os-release cut -d'=' -f2 tr -d "'") OS_VERSION=\$(grep "VERSION_ID=" /etc/os-release cut -d'=' -f2 tr -d "'") echo "OS Version: \$OS_NAME \$OS_VERSION" >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Output memory information echo "Gathering memory information..." # This will now output the memory into the report file echo "Memory Information:" >> \$REPORT_FILE free -h >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Output uptime information echo "Gathering uptime information..." # This will now output the runtime into the report echo "Uptime Information:" >> \$REPORT_FILE uptime >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Backup the OS echo "Backing up the OS..." # Placeholder for command to back up the OS sudo tar -cvzf /baker_street_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/var --exclude=/sys --exclude=/dev -- echo "OS backup completed." >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Output the sudoers file to the report echo "Gathering sudoers file..." if [! `id -u` = 0]; then echo "Error: Please run as root to get access the sudoers file..." >> \$REPORT_FILE else echo "sudoers file:" >> \$REPORT_FILE cat /etc/sudoers >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE fi # This checks to see if the user is even able to check the sudoers file, if not doesn't run, if so adds to Report File</pre>
		<h3>Script 2</h3> <pre># Script to check for files with world permissions and update them echo "Checking for files with world permissions" echo "Files with world permissions in /home before changes:" >> \$REPORT_FILE find /home -perm /o+rwx -ls >> \$REPORT_FILE # Remove world permissions echo "Removing world permissions..." find /home -perm /o+rwx -exec chmod 0-rwx {} \; # Verify changes echo "Files with world permissions in /home after changes:" >> \$REPORT_FILE find /home -perm /o+rwx -ls >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE echo "World permissions have been updated." [# Find specific files and update their permissions echo "Updating permissions for specific scripts..." >> \$REPORT_FILE # Engineering scripts - Only members of the engineering group echo "Updating permissions for Engineering scripts..." find / -name "engineering" -exec chown :engineering:{} + -exec chmod 770 {} + echo "Permissions updated for Engineering scripts." >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Research scripts - Only members of the research group echo "Updating permissions for Research scripts..." find / -name "research" -exec chown :research:{} + -exec chmod 770 {} + echo "Permissions updated for Research scripts." >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE # Finance scripts - Only members of the finance group echo "Updating permissions for Finance scripts..." find / -name "finance" -exec chown :finance:{} + -exec chmod 770 {} + echo "Permissions updated for Finance scripts." >> \$REPORT_FILE printf "\n" >> \$REPORT_FILE echo "Script execution completed. Check \$REPORT_FILE for details."</pre>

```

GNU nano 6.2 hardening script2.sh
#!/bin/bash

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

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

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

echo "Report saved to $REPORT_FILE"


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

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

printf "\n" >> $REPORT_FILE
echo "Packages have been updated and upgraded."


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

# Print out config data
echo "Printing out logging configuration data"
echo "journald.conf file data" >> $REPORT_FILE
cat /etc/systemd/journald.conf >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

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

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

```

```

#Print out config data
echo "Printing out logging configuration data"
echo "journald.conf file data" >> $REPORT_FILE
cat /etc/systemd/journald.conf >> $REPORT_FILE
printf "\n" >> $REPORT_FILE

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

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

```

Results

```
/etc/cron.weekly/
/etc/cron.weekly/.placeholder
/etc/libibverbs.d/
/etc/libibverbs.d/vmw_pvrDMA.driver
/etc/libibverbs.d/efa.driver
/etc/libibverbs.d/hns.driver
/etc/libibverbs.d/mlx5.driver
/etc/libibverbs.d/mlx4.driver
/etc/libibverbs.d/ocrdma.driver
/etc/libibverbs.d/siw.driver
/etc/libibverbs.d/rxe.driver
/etc/libibverbs.d/ipathverbs.driver
/etc/libibverbs.d/qdr.driver
/etc/libibverbs.d/irdma.driver
/etc/libibverbs.d/mthca.driver
/etc/libibverbs.d/cxgb4.driver
/etc/libibverbs.d/bnxt_re.driver
/etc/libibverbs.d/hfiverbs.driver
/etc/X11/
/etc/X11/Xsession.d/
/etc/X11/Xsession.d/90gpg-agent
/etc/inputrc
/etc/libnl-3/
/etc/libnl-3/classid
/etc/libnl-3/pktloc
/etc/iprooute2/
/etc/iprooute2/rt_tables.d/
/etc/iprooute2/rt_tables.d/README
/etc/iprooute2/rt_protos
/etc/iprooute2/bpf_pinning
/etc/iprooute2/rt_realms
/etc/iprooute2/rt_scopes
/etc/iprooute2/rt_protos.d
/etc/iprooute2/rt_protos.d/README
/etc/iprooute2/rt_tables
/etc/iprooute2/ematch_map
/etc/iprooute2/group
/etc/iprooute2/nl_protos
/etc/iprooute2/rt_dspecfield
/etc/services
/etc/protocols
/etc/wgetrc
/etc/hosts.allow
/etc/john/
/etc/john/john.conf
/etc/john/john-mail.conf
/etc/john/john-mail.msg
/etc/nanorc
/etc/binfmt.d/
/etc/ldap/
/etc/ldap/ldap.conf
/etc/mime.types
/etc/python3.10/
/etc/python3.10/sitecustomize.py
/etc/cron.hourly/
/etc/cron.hourly/.placeholder
/.dockerenv
Gathering sudoers file...
./hardening_script1.sh: line 55: [: EUID: integer expression expected
Checking for files with world permissions...
Removing world permissions...
World permissions have been updated.
Updating permissions for Engineering scripts...
Updating permissions for Research scripts...
chown: invalid group ':research'
Updating permissions for Finance scripts...
Script execution completed. Check Report File.txt for details.
root@Baker_Street_Linux_Server:/home/sysadmin# █
```

```
root@Baker_Street_Linux_Server:/home/sysadmin# cat Report_File.txt
Hostname: Baker_Street_Linux_Server
OS Version: "Ubuntu" "22.04.5 LTS (Jammy Jellyfish)"

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

Uptime Information:
21:39:21 up 9 min,  0 users,  load average: 0.02, 0.07, 0.07
OS backup completed.

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

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

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

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

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

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

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults:$(sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
Defaults:$(sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL

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

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

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

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

Files with world permissions in /home before changes:
 315527  4 -rwxr-xr-X  1 root   root  2040 Jan 10 05:43 /home/sysadmin/hardening_script2.sh
 262150  4 -rw-r--r-  1 root   root  2388 Jan 16 21:40 /home/sysadmin/Report_File.txt

Files with world permissions in /home after changes:
Updating permissions for specific scripts...
Permissions updated for Engineering scripts.
Permissions updated for Research scripts.
Permissions updated for Finance scripts.
root@Baker_Street_Linux_Server:/home/sysadmin#
```

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

<input checked="" type="checkbox"/>	Scripts scheduled with cron	Another very simple one to close out the project. Simply crontab -e, and set it up so the scripts are run when they need to be run. Screenshot included below. <pre>GNU nano 0.2 /tmp/crontab.AU04yI/crontab ~ # Output of the crontab jobs (including errors) is sent through # email to the user the crontab file belongs to (unless redirected). # # 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/ # # For more information see the manual pages of crontab(5) and cron(8) # # m h dom mon dow command 0 0 1 * * /home/sysadmin/hardening_script1.sh 0 6 * * 1 /home/sysadmin/hardening_script2.sh</pre>
-------------------------------------	-----------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SCRIPT 1

```
#!/bin/bash
```

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

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

```

SCRIPT 2

```

#!/bin/bash

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

```

```

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

```

fi

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

echo "Report saved to $REPORT_FILE"
```

fi

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

fi

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

fi

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

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

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

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

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