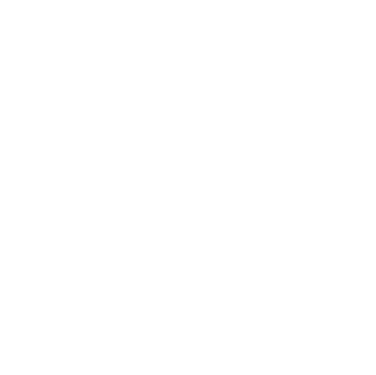
Cybersecurity

**Project 1 Hardening Summary and Checklist**



**OS Information**

|  |  |
| --- | --- |
| Customer | Baker Street Corporation |
| Hostname | **Baker\_Street\_Linux\_Server** |
| OS Version | **GNU/Linux** |
| Memory information | **total: 3.7Gi used: 739Mi free: 459Mi shared: 27Mi buff/cache: 2.6Gi available: 2.7Gi** |
| Uptime information | **03:11:56 up 41 min, 0 users, load average: 0.06, 0.03, 0.07**  **up 41 minutes**  **since: 2025-01-07 02:30:50** |

**Checklist**

| **Completed** | **Activity** | **Script(s) used / Tasks completed / Screenshots** |
| --- | --- | --- |
|  | OS backup | Script(s)  hostname  uname -o  free -h  uptime  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 /  Tasks   * Backing up system but excluding the following:   + The backup file itself   + /proc directory and contents (System and Process data)   + /tmp directory and contents (temporary files   + /mnt directory and contents (mounting external files systems)   + /sys directory and contents (system information)   + /dev directory and contents (device files)   Screenshots |
|  | Auditing users and groups | Script(s)  getent passwd | awk -F: ‘$3 >= 1000 {print $1}’  userdel -r lestrade  userdel -r irene  userdel -r mary  userdel -r gregson  userdel -r nobody  getent passwd | awk -F: ‘$3 >= 1000 {print $1}’  for user in $(awk -F: '$3 >= 1000 {print $1}' /etc/passwd); do passwd -S "$user"; done  usermod -L moriarty  usermod -L mrs\_hudson  usermod -U toby  usermod -U adler  passwd toby  passwd adler  for user in $(awk -F: '$3 >= 1000 {print $1}' /etc/passwd); do passwd -S "$user"; done  cat /etc/group  addgroup research  cat /etc/group  grep “marketing” /etc/group | cut -d: -f4  delgroup marketing  Tasks completed   * Listed all users * Removed all terminated staff user profiles and the nobody profile * Listed all users to confirm terminated staff profiles are removed * Locked all user profiles of staff on temporary leave * Unlocked all user profiles of staff that are active employees * Checked the status of all user profiles * Listed all groups * Added the research group * Listed all usernames in the marketing group (no one in marketing group to move) * Deleted marketing group   Screenshots |
|  | Updating and enforcing password policies | Script(s)  nano /etc/pam.d/common-password  password requisite pam\_pwquality.so minlen=8 ucredit=-1 ocredit=-1 retry=2  Tasks completed   * Opened the common-password file with nano * Modified the pam\_pwquality.so line   + Changed required to requisite and added minimum length of 8, at least 1 uppercase, and at least 1 special character, and retries to 2 * Saved the file and exited   Screenshots |
|  | Updating and enforcing sudo permissions | Script(s)  sudo visudo  The following was added below:  @includedir /etc/sudoers.d  sherlock ALL=(ALL:ALL) ALL  watson ALL=(ALL) /var/log/logcleanup.sh  mycroft ALL=(ALL) /var/log/logcleanup.sh  %research ALL=(ALL) /tmp/scripts/research\_script.sh  Tasks completed   * Opened the /etc/sudoers.tmp file using sudo visudo * Edited lines below @includedir /etc/sudoers.d * Added full sudo privileges to sherlock * Changed sudo privileges for watson and mycroft for only the logcleanup.sh script * Added sudo privileges for research\_script.sh script to all users in research group   Screenshots  Changed below    to |
|  | Validating and updating permissions on files and directories | Script(s)  cd home  chmod -R o-rwx ./home  find . -iname ‘\*engineering\*.sh’  find . -iname ‘\*research\*.sh’  find . -iname ‘\*finance\*.sh’  find . -iname '\*engineering\*.sh' -exec chmod g+rwx {} \; -exec chgrp engineering {} \;  find . -iname '\*research\*.sh' -exec chmod g+rwx {} \; -exec chgrp research {} \;  find . -iname '\*finance\*.sh' -exec chmod g+rwx {} \; -exec chgrp finance {} \;  Tasks completed   * Removed all world permissions from all files in each user home directory * Searched for any script files with the word engineering * Searched for any script files with the word research (none found) * Searched for any script files with the word finance * Changed permissions to all script files with the word engineering to only have read, write, and execute for the engineering group * Change permissions to all script files with the word research to only have read, write, and execute for the research group * Change permissions to all script files with the word finance to only have read, write, and execute for the finance group * Using nano, searched all files, including the hidden files, in all home directories for any hidden passwords. Founding nothing suspicious.   Screenshots      Repeated this for each identified group, i.e., engineering, research, and finance |
|  | Optional: Updating password hashing configuration | Script(s)  sudo nano /etc/login.defs  source: https://www.baeldung.com/linux/default-password-hashing-sha512  Tasks completed   * Searched for ENCRYPT\_METHOD and it was already set to SHA512.   Screenshots |
|  | Auditing and securing SSH | Script(s)  nano /etc/ssh/sshd\_config  service ssh restart  Tasks completed   * Opened the sshd\_config file with nano * Uncommented Port 22 * Changed PermitRootLogin from yes to no * Changed PermitEmptyPasswords from yes to no * Removed Port 2222, 2223, 2224, 2225 * Instructions did not stated to disable Protocol 1 so I left it there * Added Protocol 2 * Changed AllowUsers to only have sherlock watson mycroft since the others were terminated or locked. * Saved file * Exited file * Restarted ssh service   Screenshots |
|  | Reviewing and updating system packages | Script(s)  apt update  apt upgrade -y  apt list --installed > package\_list.txt  nano package\_list.txt  apt autoremove -y telnet rsh-client  apt list --installed > updated\_package\_list.txt  nano updated\_package\_list.txt  apt install ufw lynis tripwire -y  Tasks completed   * Initiated an update * Initiated an upgrade * Created a list of all installed packages to a file named package\_list.txt * Opened the file package\_list.txt * Identified one rsh-client and one telnet packages * Removed the unsecure packages * Created a new list of all installed packages to a file named updated\_package\_list.txt * Opened that file to confirm the removal of unsecure packages * Installed the following packages:   + ufw (Uncomplicated Firewall)   + lynis (Security Auditing Tool)   + tripwire (File Integrity Checking Tool)   Screenshots |
|  | Disabling unnecessary services | Script(s)  service --status-all  service --status-all > service\_list.txt  nano service\_list.txt  update-rc.d smbd stop  update-rc.d mysql disable  update-rc.d samba-ad-dc disable  update-rc.d smbd disable  update-rc.d -f mysql remove  rm /etc/init.d/mysql  rm -rf /etc/mysql  rm -rf /var/lib/mysql  update-rc.d -f samba-ad-dc remove  rm /etc/init.d/samba-ad-dc  rm -rf /etc/samba  rm -rf /var/lib/samba  update-rc.d -f smbd remove  rm /etc/init.d/smbd  rm -rf /etc/smbd  rm -rf /var/lib/smbd  service --status-all  service --status-all > updated\_service\_list.txt  nano updated\_service\_list.txt  Tasks completed   * Ran the service status command for all services * Ran the same command but had it write to a file named service\_list.txt * Opened the file using nano   + NOTE: It omitted the [ ? ] hwclock.sh service * The services mysql and samba-ad-dc are not running * Stopped the smbd service * Disabled the services mysql, samba-ad-dc, and smbd * Removed the services mysql, samba-ad-dc, and smbd * Removed dependencies for mysql, samba-ad-dc, and smbd * Ran the service status command for all services * Confirmed mysql, samba-ad-dc, and smbd were no longer there * Created an updated list to document the change, named updated\_service\_list.txt   NOTE: My screenshots show some mistakes along the way, but then show how I corrected them. My steps and commands above reflect only the corrected steps for what needed to take place.  Also, when running into issues, co-student Benjamin Chavez provided his assistance with providing commands he used to accomplish the task.  Screenshots |
|  | Enabling and configuring logging | Script(s)  nano /etc/systemd/journald.conf  nano /etc/logrotate.conf  Tasks completed   * In the journald.conf file:   + Changed #Storage=Auto to Storage=persistent   + Changed #SystemMaxUse= to SystemMaxUse=300M * In the logrotate.conf file: * Changed from:   # rotate log files weekly weekly  to  # rotate log files daily daily   * Changed from: # keep 4 weeks worth of backlogs rotate 4  to  # keep 7 days worth of backlogs rotate 7 * Saved file and exited   Screenshots |
|  | Scripts created | Script(s)  touch hardening\_script1.sh  nano hardening\_script1.sh  chmod +x hardening\_script1.sh  ./hardening\_script1.sh  touch hardening\_script2.sh  nano hardening\_script2.sh  chmod +x hardening\_script2.sh  ./hardening\_script2.sh  Tasks completed   * Created the hardening\_script1.sh shell file * Opened the script1\_template file and copied the contents. * In the VM, opened the hardening\_script1.sh shell file using nano * Pasted the contents of the script1\_template into the hardening\_script1.sh shell file. * Replaced all placeholders with corresponding commands and scripts. * Saved the shell file and closed it. * Changed the permissions of the file to be executable. * Ran the script to verify everything ran as expected and troubleshot some sudo issues. * Created the hardening\_script2.sh shell file * Opened the script2\_template file and copied the contents. * In the VM, opened the hardening\_script2.sh shell file using nano * Pasted the contents of the script2\_template into the hardening\_script2.sh shell file. * Replaced all placeholders with corresponding commands and scripts. * Saved the shell file and closed it. * Changed the permissions of the file to be executable. * Ran the script to verify everything ran as expected.   Screenshots |
|  | Scripts scheduled with cron | Script(s)  crontab -e -u root  crontab -l  Tasks completed   * Created a crontab for root since one did not exist. * Edited that crontab to run schedules for the following scripts   + hardening\_script1.sh     - Run once per month on the first of the month   + hardening\_script2.sh     - Run once per week every Monday * Saved and exited * Ran command to confirm edits were saved   Screenshots |