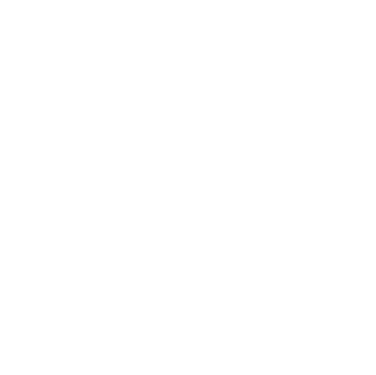
PROJECT 1: HARDENING A LINUX SERVER

Cybersecurity



**OS Information – Day 1 Part 1**

|  |  |
| --- | --- |
| Customer | Baker Street Corporation |
| Hostname | Baker Street Linux Server command: hostname |
| OS Version | Ubuntu 22.04.5 LTS (Jammy Jellyfish) command: uname -a |
| Memory information | 16182788 K total memory command: vmstat -s OR command: top |
| Uptime information | 8:04 up 55 min, 0 users, load average: 0.30, 0.29, 0.28 command: uptime |

**SCREENSHOTS FOR OS INFO:**

**Hostname; OS; Uptime**

A screenshot of a computer

Description automatically generated

**Memory**

A screenshot of a computer

Description automatically generated

**Memory: vmstat -s**

****

**Checklist**

|  |  |
| --- | --- |
| **Activity** | **Script(s) used / Tasks completed / Screenshots** |
|  |  |
| DAY 1 PART 1:  OS backup | sudo tar -cvpzf /baker\_street\_backup.tar.gz --exclude=/baker\_street\_backup.tar.gz --exclude=/proc --exclude=/tmp --exclude=/mnt --exclude=/sys --exclude=/dev --exclude=/run /  Backup:    Directories excluded:  - backup file for baker street – this is the backup.  - proc – virtual system that provides information about system processes, kernel and hardware; cleared on reboot so no need to backup.  - tmp – temporary storage of applications and processes; automatically created and deleted from the computer so no need to backup.  - mnt – mount point for temporarily attaching filesystems; used for temp mounting filesystems so no need to backup.  - sys – virtual filesystem providing information about hardware devices and drivers; cleared on reboot so no need to backup.  - dev – contains device representing hardware and virtual devices; represent hardware data and files are dynamically created by the system at boot, and recreated by OS when needed, no need to backup.  - run – runtime data for system and application state files; cleared on backup so no need to reboot.  Resources used:  Server Fault  Chatgpt  Google AI |
| DAY 1 PART 2:  Auditing users and groups | cat /etc/passwd – view users and id  id <username> – see id, group  userdel -r <username> -- delete user with recursive directory, deleting previous directories; terminated employees OR  sudo deluser --remove-home <username> -- delete user, deletes home directory; terminated employees  passwd -S <username> – to check status of a user  passwd -l <username> – to lock account; employees on leave  usermod -p <new\_password> <username\_to\_change\_password\_for> -- to unlock and assign password in one step  NOTE: you can use passwd -u <username> to unlock a user, you will be prompted to change the password;  chage -d 0 (username) – prompts user to make new password when logging in  user ID and their group ID:    UID history of command:    UID status:    cat /etc/group – view all groups  delgroup marketing – delete marketing group, no users were in this group  sudo addgroup research – add new group research  sudo usermod -aG research <username> -- add users who are not in a group to the research group OR  sudo usermod -G research <username> – removes a user from a group and specifies the only group the user should belong to; research  Groups: |
| DAY 1 PART 3:  Updating and enforcing password policies | sudo cp /etc/pam.d/common-password /etc/pam.d/common-password.bak – backup file just in case error made when doing nano  sudo nano /etc/pam.d/common-password – edit and add setting password requisite pam\_pwquality.so retry=2 minlen=8 ocredit=1 ucredit=1  Password quality: |
| DAY 1 PART 4:  Updating and enforcing sudo permissions | sudo -lU <username> -- check who has sudo privileges  sudo cat /etc/sudoers – view and list sudo privileges  sudo -l – check who has sudo privileges  sudo usermod -aG sudo <username> -- adds a user to the sudo group, giving full sudo access  sudo visudo – modify sudo privileges – see screenshot  NOTE: in sudo visudo: more efficient way to modify research group:  %research ALL=(ALL) NOPASSWD:  /tmp/scripts/research\_script.sh  sudo visudo: #1    sudo visudo: #2    sudo -lU for all users – 2 screenshots---status before sudo visudo completed: |
| DAY 1 PART 5:  Validating and updating permissions on files and directories | cd ~[username] – go into user home directory  ls -l – view permissions  sudo chmod o=--- <filename> -- removes world permissions; world relates to “others” not u or g; u and g permissions remain same  In one command (more efficient):  find /home/user -perm -0002 -exec chmod o-rwx {} \;  – to find and modify any files with world permissions in home directory of a user  World permissions – 4 screenshots:          Find files with “engineering” and “finance” and “research” text in them:  sudo find / -type f -iname '\*engineering\*'  sudo find / -type f -iname '\*finance\*'  sudo find / -type f -iname '\*research\*'  sudo chown root:engineering <filename> -- changes group to engineering  sudo chmod u=rw,g=rwx <filename> -- gives engineering group rwx permisssions; user has rw  sudo chown root:finance <filename> -- changes group to finance  sudo chmod u=rw,g=rwx <filename> -- gives finance group rwx permisssions; user has rw  sudo chown root:research <filename> – changes group to research  sudo chmod u=rw,g=rwx <filename> – gives research group rwx permissions; user has rw  In one command (more efficient):  find -iname '\*text\*' -exec chown :groupname {} +  -to find items with the text “engineering” and “research” and “finance” in it and change ownership to <groupname>  find / -type f -iname '\*text\*' -exec chmod u=rw,g=rwx,o=--- {} +  -to find files with the text “engineering” and “research” and “finance” in it and change permissions for group to read,write and execute; user to read,write; others have no permissions  cp <filename> /home/<employee in the group that owns filename>/  \*\*this step was done to place the .sh files in the proper employee group that owned the file  sudo rm <filename>  \*\*this step was done to remove .sh files in employee folders who should not have those files  \*\*\*when user is root, you do not need sudo\*\*\*  To verify files in group:  find /home -group <groupname>  Engineering owned files with permissions rwx:    Engineering group files:    Finance owned files with permissions rwx:    Finance group files:    Research owned files with permissions rwx:    Research group files: |
| DAY 1  Optional: Updating password hashing configuration | \*\*\*CRACKING HASHED PASSWORDS  I wanted to do this as part of verifying passwords and also to obtain passwords to try and find the employee’s file(s) with hidden password(s)  cd etc – move into etc directory  cat passwd – review username and UID for employees  cat shadow – review pw hashes for employees  Attempt #1:  sudo cp /etc/shadow shadow\_copy – create hashlist by copying shadow file  nano shadow\_copy – removed usernames:hashes that I did not want to search ( ctrl k )  sudo john shadow\_copy  Message: no password hashes loaded  Attempt #2:  sudo unshadow /etc/passwd /etc/shadow > hashes.txt  --to create hashlist, format username:hash  nano hashes.txt – removed users not needed  sudo john hashes.txt – Message: no password hashes loaded  checked format of hashes; seems like yescrypt hash not md5 ( thanks for the guidance Everton)….so had to use a different command:  sudo john --format=yescrypt hashes.txt  Message: unknown ciphertext format name requested  After further research, yescrypt format is available in “jumbo” version of john, which it seems we do not have  Looked at /etc/pam.d/common-password file and found a message about using “sha512” in place of “yescrypt” based on the release version; ran command:  sudo john --format=sha512crypt hashes.txt  Message: unknown ciphertext format name requested  Installed rockyou.txt wordlist in the directory. |
| DAY 2 PART 1  Auditing and securing SSH | nano /etc/ssh/sshd\_config – edit the script  PermitEmptyPasswords no  PermitRootLogin no  Allow Port 22 (removed “#”)  Don’t allow other Ports 2222,2223,2224,2225 (added “#”)  Remove Protocol 1, Add Protocol 2  \*\*\*\*Additional find – last line of script:  AllowUsers – irene, lestrade, moriarty were users allowed; deleted irene and lestrade due to termination; deleted moriarty as he is on temporary leave; added toby and adler  ssh config – 4 screenshots:        service ssh restart      sshd -T | grep -E "permitemptypasswords|permitrootlogin|port|protocol"  \*\*\*above command outputs the changes in the configuration |
| DAY 2 PART 2  Reviewing and updating system packages | apt update  apt upgrade -y --- “-y” gives a yes answer to all  touch package\_list.txt  apt list --installed > package\_list.txt  cat package\_list.txt – view installed package list  \*\*\*package\_list.txt screenshots not provided as the list was too long  apt update and upgrade -y    grep -E "telnet|rsh-client" package\_list.txt -- command outputs the package\_list.txt file that has telnet and rsh-client    sudo apt remove -y telnet rsh-client – remove telnet and rsh-client packages      apt autoremove -y – remove unnecessary dependencies on those packages    Potential security issues with telnet and rsh-client:  **TELNET:**  - data, including passwords, is sent in plaintext with no encryption thereby leaving it open for attackers to gain access and obtain information more easily; plaintext makes it more vulnerable to eavesdropping  - services can be exploited to obtain and share information about hostname, IP address and other sensitive data  - authentication is weak  - does not verify the integrity of the data; if information is compromised, it cannot detect it  **RSH-CLIENT:**  - uses DNS and IP addresses for authentication and can easily be spoofed  - unencrypted data is transmitted over the network which makes it easier to grab and use by attackers; plaintext makes it more vulnerable to eavesdropping  - if it is misconfigured, attackers can gain access to root and the entire system becomes compromised  Resources used:  Google AI  Server Fault  Chatgpt  sudo apt install -y ufw lynis tripwire – install packages    Research and document the hardening features of ufw, lynis, and tripwire:  **UFW** – user friendly front end system for managing IP table firewall rules:  - default deny – default behaviour to deny incoming connections thereby minimizing exposure to unauthorized access  - Ports and Ips – allows defined rules that restrict access to specific ports, ips and protocols which helps limit exposure to trusted sources only  - Rate limit – limits the number of connection attempts from a single IP thereby mitigating brute-force attacks  - Profiles can be preconfigured making it easier to manage firewall rules for services  - Monitoring & Logging – robust features to monitor and log traffic, detecting suspicious activity  **LYNIS** – security auditing tool:  - Comprehensive auditing of OS settings, software and network configuration and security controls; identify misconfigurations and areas to be improved  - Compliance testing – checks system configurations against industry standards to ensure compliance within regulatory framework  - Security control – after auditing, specific hardening recommendations are provided  - Plugin and customization – adminstrators can extend its functionality with plugs ins and customize its tests depending on the specific needs of the orgainzation  - Scoring and reporting – provides a hardening score based on the audit results and reports vulnerabilities while providing actionable steps to improve  **TRIPWIRE** – intrusion detection system and file integrity monitoring tool:  - Monitoring – file integrity is monitored and detects changes to critical system files and directories  - Policy customization – create or modify policies, define t, define types of changes that trigger alerts  - Reporting – comprehensive, detailed reports on detected changes  - Cryptographic verification – uses cryptographic hashes to verify file contents  - Email alerts – notifications when charges are detected allowing immediate action to be taken  Resource used: Chatgpt |
| DAY 2, PART 3  Disabling unnecessary services | man service – to get info on service commands, etC  service --status-all > service\_list.txt – lists all services and outputs to txt file  cat service\_list.txt – to verify what is in file  mysql is running (+); and samba is not running (-):    grep -E "mysql|samba" service\_list.txt -- command outputs the service\_list.txt file that shows mysql is running (+); and samba is not running (-):    man pages confirm commands “disable” and “remove” do not work with service:  Failed commands (various error messages):  sudo update-rc.d <service-name> disable  sudo update-rc.d <service-name> remove  sudo apt remove --purge -y <service-name>  sudo service <service> stop – stops the services, “+” means running, “-“ means not running  service --status-all – to list the services and status    Command to remove service:  sudo apt-get remove --purge \*mysql\\*  sudo apt-get remove --purge \*samba\*  service --status-all – to list the services and status    Attempted commands to disable and remove: |
| DAY 2 PART 4  Enabling and configuring logging | cd /etc/systemd  cat journald.conf – view the file  nano journald.conf – edit settings in the file  Set storage=persistent – remove # comment, change current setting from auto to persistent  Set systemMaxUse=300M – remove # comment, change current setting from nothing to 300M  cat journald.conf – view file    cat /etc/logrotate.conf – view the file  nano logrotate.conf – edit settings in the file:  change log rotation from weekly to daily; also modified the #comments to change from weekly to daily  change rotate logs from every 4 weeks to 7 days; also modified the #comments to change 4 weeks to 7 days |
| DAY 3 PART 1  Scripts created #1  Scripts created #2 | nano hardening\_script1.sh – copy file as per instruction and pasted into script 1 for modifications  Commands added:  $Report File: labelled the report: REPORT\_SUMMARY\_GROUP\_PROJECT\_1  All commands to output to $REPORT FILE  Hostname: hostname  OS version: uname -a  Memory: vmstat -s  Uptime: uptime  Backup 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 /  Sudoers: cat /etc/sudoers  World permissions:  find /home/user -perm -0002 -exec chmod o-rwx {} \;  Find files for Engineering, Research, Finance groups and update permissions:  find -iname '\*engineering\*' -exec chown :engineering {} +  find / -type f -iname '\*engineering\*' -exec chmod u=rw,g=rwx,o= {} +  find -iname '\*research\*' -exec chown :research {} +  find / -type f -iname '\*research\*' -exec chmod u=rw,g=rwx,o= {} +  find -iname '\*finance\*' -exec chown :finance {} +  find / -type f -iname '\*finance\*' -exec chmod u=rw,g=rwx,o= {} +  SCRIPT 1 – 9 screenshots:                    chmod u=rwx,g=rwx,o=rwx hardening\_script1.sh – to update permissions in order to run the script    to run the script: ./hardening\_script1.sh  to view the output:  cat REPORT\_SUMMARY\_GROUP\_PROJECT\_1  nano hardening\_script2.sh – copy file as per instruction and pasted into script 2 for modifications  Commands added:  $Report File: labelled the report: REPORT\_SUMMARY\_GROUP\_PROJECT\_2  All commands to output to $REPORT FILE  sshd config file: cat /etc/ssh/sshd\_config  Updating packages and services:  apt update  apt upgrade -y  Installed Packages: apt list --installed  Logging data config:  cat /etc/systemd/journald.conf  Logrotate data config:  cat /etc/logrotate.conf  SCRIPT 2 – 4 screenshots:        chmod u=rwx,g=rwx,o=rwx hardening\_script2.sh – to update permissions in order to run the script    to run the script: ./hardening\_script2.sh  to view the output:  cat REPORT\_SUMMARY\_GROUP\_PROJECT\_2 |
| DAY 3 PART 2  Scripts scheduled with cron | service cron start – to start the cron service as it was not running  crontab -e – go into cron file  schedule hardening\_script1.sh to run once a month on the first day of the month:  0 0 1 \* \* /home/hardening\_script1.sh  schedule hardening\_script2.sh to run once a week on every Monday:  0 0 \* \* 1 /home/hardening\_script2.sh  ls /var/spool/cron/crontabs/ -- list all schedule cron tasks for all users  tail /var/spool/cron/crontabs/root – displays the last lines of the cronjob for root; confirms our edit |