Week 5 Homework Submission File: Archiving and Logging Data

Please edit this file by adding the solution commands on the line below the prompt.

Save and submit the completed file for your homework submission.

Step 1: Create, Extract, Compress, and Manage tar Backup Archives

* Command to extract the TarDocs.tar archive to the current directory:
* **tar -xvf TarDocs.tar -C ~/Projects**
* Command to create the Javaless\_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Javadirectory:
* **tar cvf Javaless\_Docs.tar --exclude "Java" TarDocs/Documents**
* Command to ensure Java/ is not in the new Javaless\_Docs.tar archive:
* **tar -tvf Javaless\_Docs.tar | grep -c "Java"**

Bonus

* Command to create an incremental archive called logs\_backup\_tar.gz with only changed files to snapshot.file for the /var/logdirectory:
* **sudo tar --listed-incremental=snapshot.file -cvzf logs\_backup.tar.gz /var/log**

Critical Analysis Question

* Why wouldn't you use the options -x and -c at the same time with tar?
* **You wouldn’t use both -x and -c at the same time with tar because when you use -c you’re creating the archive and -x is extracting the archive meaning that it has to be created before it can be extracted. Using them at the same time means -x is extracting an archive that yet has to be created.**

Step 2: Create, Manage, and Automate Cron Jobs

* Cron job for backing up the /var/log/auth.log file:
* **0 6 \* \* 3 sudo tar czf /var/log/auth\_backup.tgz --absolute-names /var/log/auth.log**

Step 3: Write Basic Bash Scripts

* Brace expansion command to create the four subdirectories:
  + **mkdir -p ~/backups/{freemem,diskuse,openlist,freedisk}**
* Paste your system.sh script edits below:
* **#!/bin/bash**
* **echo $(free -h | awk 'FNR == 2 {print}'| awk '{print $4}') > ~/backups/freemem/free\_mem.txt**
* **echo $(du -h) > ~/backups/diskuse/disk\_usage.txt**
* **echo $(lsof) > ~/backups/openlist/open\_list.txt**
* **echo $(df -h) > ~/backups/freedisk/free\_disk.txt**
* Command to make the system.sh script executable:
* **chmod +x system.sh**
* **./system.sh**

Optional

* Commands to test the script and confirm its execution:

Bonus

* Command to copy system to system-wide cron directory:

Step 4. Manage Log File Sizes

* Run sudo nano /etc/logrotate.conf to edit the logrotate configuration file.
* Configure a log rotation scheme that backs up authentication messages to the /var/log/auth.log.
* Add your config file edits below:
* **/var/log/auth.log {**
* **weekly**
* **rotate 7**
* **notifempty**
* **delaycompress**
* **missingok**
* **endscript**
* **}**

Bonus: Check for Policy and File Violations

* Command to verify auditd is active:
* Command to set number of retained logs and maximum log file size:
* Add the edits made to the configuration file below:
* [Your solution edits here]
* Command using auditd to set rules for /etc/shadow, /etc/passwd and /var/log/auth.log:
* Add the edits made to the rules file below:
* [Your solution edits here]
* Command to restart auditd:
* Command to list all auditd rules:
* Command to produce an audit report:
* Create a user with sudo useradd attacker and produce an audit report that lists account modifications:
* Command to use auditd to watch /var/log/cron:
* Command to verify auditd rules:

Bonus (Research Activity): Perform Various Log Filtering Techniques

* Command to return journalctl messages with priorities from emergency to error:
* Command to check the disk usage of the system journal unit since the most recent boot:
* Comand to remove all archived journal files except the most recent two:
* Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority\_High.txt:
* Command to automate the last command in a daily cronjob. Add the edits made to the crontab file below:
* [Your solution cron edits here]