**## 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**

**1.** Command to \*\*extract\*\* the `TarDocs.tar` archive to the current directory:

tar -xvf TarDocs.tar

**2.** Command to \*\*create\*\* the `Javaless\_Doc.tar` archive from the `TarDocs/` directory, while excluding the `TarDocs/Documents/Java` directory:

tar -cvf Javaless\_Docs.tar --exclude-tag-under=Java ~/Projects/TarDocs

**3.** Command to ensure `Java/` is not in the new `Javaless\_Docs.tar` archive:

tar -tf Javaless\_Docs.tar | grep -rw Java\*

\*\*Bonus\*\*

**-** Command to create an incremental archive called `logs\_backup\_tar.gz` with only changed files to `snapshot.file` for the `/var/log` directory:

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`?

When you use -x, it extracts a file from an archive.

When you use -c, it creates a new archive.

When you are creating a new file, it's not possible to also extract it at the same time.

**---**

**### Step 2: Create, Manage, and Automate Cron Jobs**

**1.** Cron job for backing up the `/var/log/auth.log` file:

0 6 \* \* 3 tar -czf /auth\_backup.tgz /var/log/auth.log

**---**

**### Step 3: Write Basic Bash Scripts**

**1.** Brace expansion command to create the four subdirectories:

sudo mkdir -p ~/backups/{freemem,diskuse,openlist,freedisk}

**2.** Paste your `system.sh` script edits below:

```bash

#!/bin/bash

**# Free memory output to a free\_mem.txt file**

free -mh > awk 'NR==2{printf "Memory Usage: %s/%sMB (%.2f%%)\n", $3,$2,$3\*100/$2 }' > ~/backups/freemem/free\_mem.txt

**# Disk usage output to a disk\_usage.txt file**

df -h | awk '$NF=="/"{printf "Disk Usage: %d/%dGB (%s)\n", $3,$2,$5}' > ~/backups/diskuse/disk\_usage.txt

**# List open files to a open\_list.txt file**

lsof > ~/backups/openlist/open\_list.txt

**# Free disk space to a free\_disk.txt file**

df -h >> ~/backups/freedisk/free\_disk.txt

```

**3.** Command to make the `system.sh` script executable:

chmod +x system.sh

\*\*Optional\*\*

**-** Commands to test the script and confirm its execution:

sudo ./system.sh

cat ~/backups/diskuse/disk\_usage.txt

\*\*Bonus\*\*

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

sudo cp system.sh /etc/cron.weekly

**---**

**### Step 4. Manage Log File Sizes**

**1.** 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:

```bash

/var/log/auth.log

Weekly

rotate 7

Notifempty

Delaycompress

Missingok

endscript

```

**---**

**### Bonus: Check for Policy and File Violations**

**1.** Command to verify `auditd` is active:

sudo systemctl status auditd

\*\* shows "active (running)"

**2.** Command to set number of retained logs and maximum log file size:

sudo nano /etc/audit/auditd.conf

**-** Add the edits made to the configuration file below:

```bash

max\_log\_file = 35

num\_logs = 7

```

**3.** Command using `auditd` to set rules for `/etc/shadow`, `/etc/passwd` and `/var/log/auth.log`:

sudo nano /etc/audit/rules.d/audit.rules

**-** Add the edits made to the `rules` file below:

```bash

-w /etc/shadow -p wra -k hashpass\_audit

-w /etc/passwd -p wra -k userpass\_audit

-w /var/log/auth.log -p wra -k authlog\_audit

```

**4.** Command to restart `auditd`:

sudo systemctl restart auditd

**5.** Command to list all `auditd` rules:

sudo auditctl -l

**6.** Command to produce an audit report:

sudo aureport -au

**7.** Create a user with `sudo useradd attacker` and produce an audit report that lists account modifications:

sudo useradd attacker

sudo aureport -m

**8.** Command to use `auditd` to watch `/var/log/cron`:

sudo auditctl -w /var/log/cron

**9.** Command to verify `auditd` rules:

sudo auditctl -l

**---**

**### Bonus (Research Activity): Perform Various Log Filtering Techniques**

**1.** Command to return `journalctl` messages with priorities from emergency to error:

sudo journalctl -b -p emerg..err

**1.** Command to check the disk usage of the system journal unit since the most recent boot:

sudo journalctl -b -u systemd-journald | less