**## 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 xvvf Tardoc.tar

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

tar cvvf Javaless\_Docs.tar --exclude='TarDocs/Documents/Java' ./TarDocs/Documents

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

tar tvvf Javaless\_Docs.tar | less

**\*\*Bonus\*\***

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

**#### Critical Analysis Question**

- Why wouldn't you use the options `-x` and `-c` at the same time with `tar`?

Because they are two different modes for tar, -c Creates and -x Extract you dont want to extract something you are creating

**---**

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

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

0 6 \* \* 3 sudo tar cvvfz ./auth\_backup.gz /var/log/auth.log

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

1. Brace expansion command to create the four subdirectories:

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

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

```bash

#!/bin/bash

# INSTRUCTIONS: Edit the following placeholder command and output filepaths

# For example: cpu\_usage\_tool > ~/backups/cpuuse/cpu\_usage.txt

# The cpu\_usage\_tool is the command and ~/backups/cpuuse/cpu\_usage.txt is the filepath

# In the above example, the `cpu\_usage\_tool` command will output CPU usage information into a `cpu\_usage.txt` file.

# Do not forget to use the -h option for free memory, disk usage, and free disk space

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

free > backups/freemem/free\_mem.txt

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

du -h > backups/diskuse/disk\_usage.txt

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

lsof /dev/null > backups/openlist/open\_list.txt

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

df -h | awk -F' ' '{print $1,$4}' > 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:

**\*\*Bonus\*\***

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

---

**### 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{

rotate 7

weekly

notifempty

delaycompress

missingok

}

```

---

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

1. Command to verify `auditd` is active:

systemctl status auditd

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

sudo vim /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`:

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

```bash

[Your solution edits here]

```

4. Command to restart `auditd`:

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 adduser 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 cat /etc/audit/audit.rules

---

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

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

journalctl -p 0..3

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

journalctl --disk-usage -b -0

4. Comand to remove all archived journal files except the most recent two:

journalctl --vacuum-files=2

6. Command to filter all log messages with priority levels between zero and two, and save output to `/home/sysadmin/Priority\_High.txt`:

journalctl -p 0..2 >> /home/sysadmin/Priority\_High.txt

7. Command to automate the last command in a daily cronjob. Add the edits made to the crontab file below:

```bash

0 6 \* \* \* journalctl -p 0..2 >> /home/sysadmin/Priority\_High.txt

```