﻿**﻿Week 5 Homework Submission File: Archiving and Logging Data**

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

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

**Answer: -** tar xvvf TarDocs.tar

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

﻿ **Answer: -**

tar cvvf Javaless\_Docs.tar --exclude="TarDocs/Documents/Java" /home/sysadmin/Projects/

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

**Answer: -** ﻿tar -tvf Javaless\_Docs.tar | grep 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: -

**Answer: -** 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`?

**Answer: -** Well, -x & -c are opposite to each other. -x used to extract the tar file and -c will create a new tar file. Let see the example (tar -xc newfile.tar) and execute it, the terminal will get confused and wont be able to understand to extract the file or create the file and it will leads to an error and command will not execute.

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

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

﻿

**Answer: -** 0 6 \* \* 3 tar -zcf /var/backups/auth\_backup.tgz --absolute-names /var/log/auth.log

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

1. Brace expansion command to create the four subdirectories:

﻿

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

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

**Answer: -**

```bash

#!/bin/bash

[Your solution script contents here]

free -h > ~/backups/freemem/free\_mem.txt

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

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

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

```

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

**Answer: -** chmod +x system.sh

**\*\*Optional\*\***

- Commands to test the script and confirm its execution:

**Answer: -** sudo ./system.sh

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

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

**Answer: -** sudo cp system.sh /var/spool/cron

---

**### 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: -

**Answer: -**

```bash

[Your logrotate scheme edits here]

**/var/log/auth.log {**

**weekly**

**rotate 7**

**notifempty**

**compress**

**delaycompress**

**missingok**

**}**

```

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

1. Command to verify `auditd` is active:

**Answer: -** systemctl status auditd

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

- Add the edits made to the configuration file below:

**Answer: -**

```bash

[Your solution edits here]

**num\_logs = 7**

**max\_log\_file = 35**

```

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

**Answer: -** sudo nano /etc/audit/rules.d/audit.rules

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

**Answer: -**

```bash

[Your solution edits here]

**w /ect/shadow -p wra -k hashpass\_audit**

**-w /ect/passwd -p wra -k userpass\_audit**

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

```

4. Command to restart `auditd`:

**Answer: -** sudo service auditd restart

5. Command to list all `auditd` rules:

﻿ **Answer: -** sudo auditctl -l

6. Command to produce an audit report:

**Answer: -** sudo aureport -au

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

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

**Answer: -** sudo auditctl -w /var/log/cron -p rwa -k cron\_audit

9. Command to verify `auditd` rules:

**Answer: -** sudo auditctl -l

---

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

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

**Answer: -** journalctl -b -p emerg..err

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

**Answer: -** journalctl --disk-usage

3. Command to remove all archived journal files except the most recent two:

**Answer: -** journalctl --vacuum-time=2d

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

**Answer: -** sudo journalctl -p 0..2 > /home/sysadmin/Priority\_High.txt

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

**Answer: -**

```bash

[Your solution cron edits here]

**@daily sudo journalctl -p 0..2 > /home/sysadmin/Priority\_High.txt**

**Or**

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

```

**Thank you,**

Aziz Somani