**Gaven Glasco**

**Week 5 HW**

## 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: **sudo tar xvvf TarDocs.tar -C /home/sysadmin/Projects/TarDocs --wildcard**

2. Command to \*\*create\*\* the `Javaless\_Doc.tar` archive from the `TarDocs/` directory, while excluding the `TarDocs/Documents/Java` directory: **sudo tar cvvf Javaless\_Docs.tar –exclude=‘Java’ ~/Projects/TarDocs/Documents**

3. Command to ensure `Java/` is not in the new `Javaless\_Docs.tar` archive: **sudo tar tvvf 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: **sudo tar –listed-incremental=snapshot.file –level=0 -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`? **According to the man pages for tar. -c is for creating a new archive. -x is for extracting files from an existing archive. So it wouldn’t work trying to create and extract a file in the same command. You could create a new archive and pipe into an extraction command.**

---

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

1. Cron job for backing up the `/var/log/auth.log` file: **0 6 \* \* 3 tar -czvf auth\_backup.tgz /var/log/auth.log >> /home/sysadmin/auth\_backup.tgz**

---

### Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories: **mkdir -p ~/backups/{freeman,diskuse,openlist,freedisk}**

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

```bash

#!/bin/bash

[Your solution script contents here]

Graphical user interface, text

Description automatically generated

```

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

\*\*Optional\*\* **sudo chmod +x system.sh**

- Commands to test the script and confirm its execution:

\*\*Bonus\*\* **[ cd backups/ ]**

**[ cd freedisk/ && cat free\_disk.txt ] [cd openlist/ && cat open\_list.txt ]**

**[ cd diskuse/ && cat disk\_use.txt ] [ cd freeman/ && cat free\_man.txt ]**

- Command to copy `system` to system-wide cron directory: **sudo cp /home/sysadmin/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 [Your logrotate scheme edits here] Graphical user interface, text

Description automatically generated

```

---

### 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:

- Add the edits made to the configuration file below:

```bash

[Your solution edits here]

Graphical user interface, text

Description automatically generated

```

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]

Text

Description automatically generated

```

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

5. Command to list all `auditd` rules: **sudo auditctl -l**

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

7. Create a user with `sudo useradd attacker` and produce an audit report that lists account modifications: **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 -1 -p “emerg”..”err”**

1. Command to check the disk usage of the system journal unit since the most recent boot: **sudo journalctl -u system-journald**

1. Comand to remove all archived journal files except the most recent two: **sudo journalctl -–vacuum-files=2**

1. Command to filter all log messages with priority levels between zero and two, and save output to `/home/sysadmin/Priority\_High.txt`: **sudo journalctl -b -l -p “0”..”2” >> /home/sysadmin/Priority\_High.txt**

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

```bash

[Your solution cron edits here] Text

Description automatically generated

```

---

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.