A computer screen capture

Description automatically generated with medium confidence

A picture containing table

Description automatically generated

**Linux Scripts**

**Create, Extract, Compress, and Manage tar Backup Archives**

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

**tar -xvf TarDocs.tar**

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**

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

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

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`?

**We cannot create a tar file and extract at the same time.**

**-c Create a new archive.**

**-x Extract files from an archive.**

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

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

**# Disk usage**

**du -h ~/ >> ~/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**

 ```

1. 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**

\*\*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**

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

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

    bash

**sudo nano /etc/audit/auditd.conf**

**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

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

**## For the permissions to monitor**

**-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**

1. Command to restart `auditd`:

**sudo systemctl restart auditd**

1. Command to list all `auditd` rules:

**sudo auditctl -l**

1. Command to produce an audit report:

**sudo aureport -au**

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

**sudo aureport -m**

1. 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**

A picture containing text

Description automatically generated

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

**sudo journalctl -b -u systemd-journald | less**

A picture containing text

Description automatically generated

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

**sudo journalctl --vacuum-files=2**

Text

Description automatically generated

1. 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/student/Priority\_High.txt**

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

**sudo crontab -e**

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

© 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.