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 -xvf TarDocs.tar**

2. Command to **create** the `Javaless_Doc.tar` archive from the `TarDocs/` directory, while excluding the `TarDocs/Documents/Java` directory:
   sudo tar --exclude='Documents/Java' -cvf Javaless_Doc.tar TarDocs

3. Command to ensure `Java/` is not in the new `Javaless_Docs.tar` archive:
   tar -tvf Javaless_Doc.tar | grep -i 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`?

  The flag -x expands to --extract

  The flag -c expands to --create

--Extract requires a tar file that already exists

--Create creates a new file

As a result, passing both arguments in the same command results in the error "You may not specify more than one '-Acdtrux', '--delete' or '--test-label' option"

-

Step 2: Create, Manage, and Automate Cron Jobs

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

   sudo crontab -e

   0 0 * * * tar -cfv auth_backup.tar /var/log/auth.log > authcronjob.log

-

Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:

   mkdir ~/backups/ ; mkdir ~/backups/{freemem,diskuse,openlist,freedisk}

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

```
#!/bin/bash
#Free memory output to a free_mem.txt file
free --mega > ~/Homework/week5/free_mem.txt
#Disk usage output to a disk_usage.txt file
du -h ~/Homework/week5/free_mem.txt
#List open files to a open_list.txt file
```

```
lsof > ~/Homework/week5/free_mem.txt
#Free disk space to a free_disk.txt file
df -h > ~/Homework/week5/free_disk.txt
```

-

1. Command to make the system.sh script executable:

   sudo chmod +x system.sh

   Optional

- Commands to test the script and confirm its execution:

  ./system.sh

  find ~/Homework/week5/ -type f (-name "*free*.txt" -o -name "*disk*.txt" -o -name "*open*.txt")

```
sysadmin@UbuntuDesktop:~/Homework$ find ~/Homework/week5/ -type f \( -name "*free*.txt" -o -name "*disk*.txt" -o -name "*open*.txt" \)
/home/sysadmin/Homework/week5/open_list.txt
/home/sysadmin/Homework/week5/free_mem.txt
/home/sysadmin/Homework/week5/free_disk.txt
/home/sysadmin/Homework/week5/disk_usage.txt
```

Bonus

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

sudo cp system.sh /etc/cron.daily/

-

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:

/var/log/auth.log {
weekly
rotate 7
size 10M
compress
delaycompress
missingok
emptyok
}

```

-

## Bonus: Check for Policy and File Violations

1. Command to verify `auditd` is active:
   **systemctl is-enabled auditd**

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:

3. Command using `auditd` to set rules for `/etc/shadow`, `/etc/passwd` and `/var/log/auth.log`:
   sudo auditctl -w /etc/shadow -p rwa -k shadowfile
   sudo auditctl -w /etc/passwd -p rwa -k passwdfile
   sudo auditctl -w /var/log/auth.log -p rwa -k authlog

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

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 -t

7. Create a user with `sudo useradd attacker` and produce an audit report that lists account modifications:
   ausearch -f /etc/passwd -i

8. Command to use `auditd` to watch `/var/log/cron`:
   sudo auditctl -w /var/log/cron -p rwxa -k cronlogs

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 -p 0..2

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

   sudo journalctl -b -u systemd-journald

   results in (excerpt):

```
Jul 23 19:26:20 UbuntuDesktop systemd-journald[234]: Runtime journal (/run/log/journal/e5853fe375964d39b27025eb6608e969) is 4.9M, max 39.3M, 34.4M free.
```

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

2. Command to filter all log messages with priority levels between zero and two, and save output to `/home/sysadmin/Priority_High.txt`:
   sudo journalctl -p 0..2 > ~/home/sysadmin/Priority_High.txt

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

```
sudo nano HiPro.sh
#!/bin/bash
journalctl -p 0..2 > ~/home/sysadmin/Priority_High.txt
exit

sudo cp ./HiPro.sh /etc/cron.daily
sudo chmod +x /etc/cron.daily/HiPro.sh
```

```
sudo crontab -e
40 4 * * * bash /etc/cron.daily/HiPro.sh
```
