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'

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

   sudo tar -cvvf Javaless_Doc.tar --exclude="TarDocs/Documents/Java" TarDocs/

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

   sudo tar tvvf Javaless_Doc.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 -cvvzf logs_backup.tar.gz /var/log

Critical Analysis Question

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

  You wouldn't use the options -x and -c at the same time with tar because the -x option extracts files from a tar archive. The -c option is used to compress files and create a tar archive. It is not possible to compress a file and extract it at the same time.

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 -cvvzf /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:

```
#!/bin/bash

# Free memory output to a free_mem.txt file
free -h > ~/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 > ~/backups/openlist/open_list.txt

# Free disk space to a free_disk.txt file
df -h > ~/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:

```
sudo ./system.sh
cat ~/backups/freemem/free_mem.txt
cat ~/backups/diskuse/disk_usage.txt
cat ~/backups/openlist/open_list.txt
cat ~/backups/freedisk/free_disk.txt
```

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`.

```
/var/log/auth.log {
        weekly
        rotate 7
        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:
   - Add the edits made to the configuration file below:

```
num_logs = 7 (changed from num_logs = 5)
max_log_file = 35 (changed from max_log_file = 8)
```

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:

```
-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
```

4. Command to restart `auditd`:

```
sudo systemctl restart auditd
```

5. Command to list all `auditd` rules:

```
sudo auditctl -l
```

6. Command to produce an audit report:

```
sudo aureport -au -i
```

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

   sudo aureport -m -i

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

   sudo auditctl -w /var/log/cron -p wra -k cron_change

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:
2. Command to check the disk usage of the system journal unit since the most recent boot:
3. Comand to remove all archived journal files except the most recent two:
4. Command to filter all log messages with priority levels between zero and two, and save output to `/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:

   ```
   [Your solution cron edits here]
   ```