Cybersecurity

Module 5 Challenge Submission File

Archiving and Logging Data

Make a copy of this document to work in, and then for each step, add the solution command below the prompt. Save and submit this completed file as your Challenge deliverable.

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

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

```
Tar xvf ~/Projects/TarDocs.tar
```

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

```
Tar cvf Javaless-Doc.tar -exclude=TarDocs/Documents/Java TarDocs
```

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

```
Tar tvf Javaless_Doc.tar | grep Java
```

Optional

4. 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 cvvwf Javaless_docs.tar -listed-incremental=Javaless_Docs.snar
-level=0 Javaless_Docs.tar
```

Critical Analysis Question

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

-c creates the tarball, while -x extracts the tarball. Usually, you would do them in separate commands, because they have the opposite function of one another. Also, there are not many cases where you would archive something, just to immediately unarchive it.

Step 2: Create, Manage, and Automate Cron Jobs

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

```
0 06 * * 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:

```
#!/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 -ha > ~/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

4. Commands to test the script and confirm its execution:

```
./system.sh
```

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

```
Sudo cp ~/system.sh /etc/cron.d
```

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`.

   a. Add your config file edits:

```
/var/log/auth.log {
Rotate 7
Weekly
Notifempty
```

```
Delaycompress
Missingok
endscript
}
```

Optional Additional Challenge: 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:

```
[Enter answer here]
```

   Add the edits made to the configuration file:

```
[Enter answer here]
```

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

```
[Enter answer here]
```

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

```
[Enter answer here]
```

4. Command to restart `auditd`:

```
[Enter answer here]
```

5. Command to list all `auditd` rules:

```
[Enter answer here]
```

6. Command to produce an audit report:

```
[Enter answer here]
```

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

```
[Enter answer here]
```

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

```
[Enter answer here]
```

9. Command to verify `auditd` rules:

```
[Enter answer here]
```

Optional (Research Activity): Perform Various Log Filtering Techniques

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

```
[Enter answer here]
```

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

```
[Enter answer here]
```

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

```
[Enter answer here]
```

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

```
[Enter answer here]
```

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

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