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 xvvf TarDocs.tar
```

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

```
tar cvvWf Javaless_Docs.tar
--exclude=’/home/sysadmin/Projects/TarDocs/Documents/Java’
~/Projects/TarDocs/Documents
```

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

```
tar tvvf Javaless_Docs.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 logs_backup_01.tar.gz --listed-incremental=logs_backup.snar
--level=0 /var/log/
```

Critical Analysis Question

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

-x is to extract from and -c is to save to the tar file. It doesn't make sense to do both 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 tar -zcf /auth_backup.tgz /var/log/auth.log
```

Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:

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

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

```
#!/bin/bash

# Free Memory Output to text file free_mem.txt

free -h | awk '/Mem:/ {print $4}' >>
/home/sysadmin/backups/freemem/free_mem.txt

# Disk usage output to text file disk_usage.txt
```

```
du -h >> /home/sysadmin/backups/diskuse/disk_usuage.txt

# List of open files output to text file open_list.txt

lsof >> /home/sysadmin/backups/openlist/open_list.txt

# Free Disk space output to free_disk.txt

df -h  >> /home/sysadmin/backups/freedisk/free_disk.txt
```

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

```
sudo chmod +x system.sh
```

Optional

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

```
sudo ./system.sh to run the script and then we can navigate to ~/backups/
and cat any txt file in any of the subdirectories to check that the script
ran and logged the appropriate info (cat free_disk.txt).
```

5. 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`.

   a. Add your config file edits:

```
/var/log/auth.log {

          rotate 7
          weekly
          notifempty
          compress
          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:

```
Run sudo nano /etc/audit/auditd.conf
```

   Add the edits made to the configuration file:

```
Then set “num_logs = 7”
         “max_log_file = 35”
```

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

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

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
```

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

```
Run: sudo useradd attacker
Then run: sudo aureport -m (this lists user modifications)
```

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
```

Optional (Research Activity): Perform Various Log Filtering Techniques

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

```
sudo journalctl -p 0..3
```

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

```
sudo journalctl -b --disk-usage
```

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

```
sudo journalctl --vacuum-file=2
```

4. 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

```

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

```
0 0 * * * sudo journalctl -p 0..2 >> /home/sysadmin/Priority_High.txt
```