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 cvvf 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_Docs.tar | grep Java
```

Bonus

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 --listed -incremental=snapshot.file -cvzf logs_backup_tar.gz
/var/log
```

Critical Analysis Question

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

-c stands for “create” and -x stands for “extract” this would be ineffective because your work would be entirely invalid as these commands are exactly opposite.

source: http://linuxcommand.org/lc3_man_pages/tar1.html

Step 2: Create, Manage, and Automate Cron Jobs

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

```
0 6 * * 3 tar -zcvf auth_backup.tgz/var/log/auth
```

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 -m > backups/freemem/free_mem.txt

#For disk usage in human readable form:

df -BM -h > backups/diskuse/disk_usage.txt

#For all open files:

lsod > backups/openlist/open_list.txt
```

```
#For file system disk space and statistics:

df  -k -BM -h | awk ‘{print $1,$4}’ > backups/freedisk/free_disk.txt

#End of script]
```

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

```
chmod +x ./system.sh
```

Optional

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

```
sudo ./system.sh
```

Bonus

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 { Weekly rotate 7 Notifempty Delaycompress missingok
endscript
}
```

```

```

Bonus: Check for Policy and File Violations

1. Command to verify `auditd` is active:

```
systemctl status auditd
```

```
systemctl status auditd
```

Add the edits made to the configuration file:

```
max_log_file = 35
num_logs = 7
```

2. 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 rwa -k hashpass_audit
-w /etc/shadow -p rwa -k userpass_audit
-w /var/log/auth.log rwa -k authlog.audit
```

3. Command to restart `auditd`:

```
sudo systemctl restart auditd
```

4. Command to list all `auditd` rules:

```
sudo auditctl -l
```

5. Command to produce an audit report: aureport -au (not sure where the text box went!!)

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

```
sudo aureport --mods
```

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

```
 sudo auditctl -w /var/log/cron
```

8. 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:

```
journalctl -b -p 0..3
```

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

```
journal -b -u systemd-journald -- disk-usage
```

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

```
sudo journalctl --vacuum-files=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 jounalctl -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:

```
daily journalctl -p 0..2 >> /home/sysadmin/Priority_High.txt ‘’’
```

Sources: https://www.redhat.com/sysadmin/configure-linux-auditing-auditd
https://linuxhandbook.com/journalctl-command/
https://www.2daygeek.com/journalctl-read-linux-system-logs/#:~:text=5)%20Checking%20disk%20usage%20of%20all%20journal%20files&text=To%20see%20how%20much%20storage,M%20in%20the%20file%20system.

UOFM GitLab Archving and Logging data summary notes