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 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_Docs.tar --exclude="Documents/Java" TarDocs/

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

   tar -tf 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?

You wouldnt use the -x and -c option together because one will create the tar and the other will extract. You cannot create and extract at 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 -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/{freemen,diskuse,openlist,freedisk}
```

2. Paste your system.sh script edits:

Sudo nano system.sh

#! /bin/bash

free -h > ~/backups/freemem/free_mem.txt

du -h > ~/backups/diskuse/disk_usage.txt

lsof > ~/backups/openlist/open_list.txt

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:

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:

```
sudo systemctl status auditd
```

2. Command to set number of retained logs and maximum log file size:

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

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 shadow
-w /etc/shadow -p wra -k passwd
-w /var/log/auth.log -p wra -k auth.log
```

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:

```
sudo useradd attacker
sudo aureport -m
```

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
```

Bonus (Research Activity): Perform Various Log Filtering Techniques

1. Command to return journalctl messages with priorities from emergency to error:
    sudo journalctl -b -p 3

2. Command to check the disk usage of the system journal unit since the most recent boot:
3. sudo journalctl -b -u systemd-journald |less

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

```
sudo journalctl --vacuum-files=2
```

5. 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 2 >> /home/sysadmin/Priority_High.txt
```

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

```
sudo crontab -e

@daily journalctl -p 2 >> /home/sysadmin/Priority_High.txt
```

© 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.