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

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

tar xvvf ~/Downloads/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="Java" ~/Projects/TarDocs

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

tar tvvf Javaless\_Docs.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 cvvf logs\_backup.tar.gz --listed-incremental=snapshot.file /var/log

#### Critical Analysis Question

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

An -x is used to extract files. A -c is used to create files. You cannot extract and create 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 -czvf /var/auth\_backup.tgz /var/log/auth.log

### Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:

mkdir -p ~/backups/{freemem,diskuse,openlist,freedisk}

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

nano system.sh

#!/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:

:~/backups/freemem$ ls

free\_mem.txt

sysadmin@UbuntuDesktop:~/backups/freemem$ cat free\_mem.txt

total used free shared buff/cache available

Mem: 3.9G 1.4G 132M 38M 2.3G 2.1G

Swap: 1.9G 7.8M 1.9G

\*\*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`.

- Add your config file edits below:

```bash

/var/log/auth.log {

weekly

rotate 7

notifempty

compress

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:

- Add the edits made to the configuration file below:

```bash

[# This file controls the configuration of the audit daemon

local\_events = yes

write\_logs = yes

log\_file = /var/log/audit/audit.log

log\_group = adm

log\_format = RAW

flush = INCREMENTAL\_ASYNC

freq = 50

max\_log\_file = 35

num\_logs = 7

priority\_boost = 4

disp\_qos = lossy

dispatcher = /sbin/audispd

name\_format = NONE

]

```

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 -wra -k hashpass\_audit

-w /etc/passwd -wra -k userpass\_audit

-w /var/log/auth.log -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:

sudo aureport -m

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

auditctl -w /var/log/cron

9. Command to verify `auditd` rules:

auditclt -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 emerg..err

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

sudo journalctl --disk-usage | less

1. Comand to remove all archived journal files except the most recent two:

sudo journalctl --vacuum-file=2

1. 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/student/Priority\_High.txt

1. Command to automate the last command in a daily cronjob. Add the edits made to the crontab file below:

```bash

[0 0 \* \* \* journalctl -p 0..2 > /home/student/Priority\_High.txt >/dev/null 2>&1

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: Shadow People**

1. Create a secret user named sysd. Make sure this user doesn't have a home folder created:  
   * sudo useradd -M sysd
2. Give your secret user a password:  
   * hacker
3. Give your secret user a system UID < 1000:  
   * nano /etc/passwd
   * #Changed UID from 1007 to 876
4. Give your secret user the same GID:  
   * nano /etc/passwd
   * #Changed GID from 1008 to 876
   * Control X
5. Give your secret user full sudo access without the need for a password:  
   * sudo visudo
6. Test that sudo access works without your password:  
     
   sudo -l

**Step 2: Smooth Sailing**

1. Edit the sshd\_config file:  
     
    nano /etc/ssh/sshd\_config

Uncomment Port 22

Add Port 2222

Control O

Control X

**Step 3: Testing Your Configuration Update**

1. Restart the SSH service:  
   * systemctl restart sshd
2. Exit the root account:  
   * Exit
3. SSH to the target machine using your sysd account and port 2222:  
   * ssh sysd@192.168.6.105 -p 2222
4. Use sudo to switch to the root user:  
   * sudo su

**Step 4: Crack All the Passwords**

1. SSH back to the system using your sysd account and port 2222:  
   * ssh sysd@192.168.6.105 -p 2222
2. Escalate your privileges to the root user. Use John to crack the entire /etc/shadow file:  
   * john /etc/shadow









