



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:

```
sudo tar xvf TarDocs.tar
```

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

```
sudo tar --exclude=TarDocs/Documents/Java -cvf Javaless_Docs.tar TarDocs
```

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

```
sudo tar -tf 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.tar.gz --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`?

```
-c tells tar to create an archive
-x tells tar to extract from an archive
It cannot perform both simultaneously
```

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 /var/log/auth.log /auth_backup.tgz
```

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
echo -e "$(date): $(free | grep Mem | awk '{print $4}')" | >>
home/sysadmin/backups/freemem/free_mem.txt
echo -e "$(date): $(du -H)" >> /home/sysadmin/backups/diskuse/disk_usage.txt
echo -e "$(date): $(lsof)" >> /home/sysadmin/backups/openlist/open_list.txt
echo -e "$(date): $(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
```

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  
    compress  
    delaycompress  
    missingok  
    notifempty
```

```
endscript }
```

Optional Additional Challenge: 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:

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

Add the edits made to the configuration file:

```
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/rules
```

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

```
-w /etc/shadow -p wra hashpas_audit  
-w /etc/passwd -p wra userpass_audit  
-w /var/log/auth.log -p wra 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 aureport -m
```

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

```
-w /var/log/cron -p wra cron_watch
```

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 emerg..err
```

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

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

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

```
sudo journalctl -vacuum-time=2d
```

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/sudent/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 12 * * * sudo journalctl -p 0..2 >> /home/sudent/Priority_High.txt
```