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

**0 6 \* \* 3 tar -czf /auth\_backup.tgz  /var/log/auth.log**

**Sudo grep -i cron /var/log/auth.log**

**Step 3: Write Basic Bash Scripts**

1. Brace expansion command to create the four subdirectories:  
   Sudo mkdir -p  ~/{backups/freeman,diskuse,openlist,freedisk}

Paste your system.sh script edits below:  
  
 #!/bin/bash

# Free memory output to a free\_mem.txt file

free -h > ~/backups/freeman/free\_mem.txt

#Disk usage output to a disk\_usage.txt file

du -h > ~/backups/disk\_usage.txt

 # List open files to a open\_list.txt file

isof > ~/backup/openlist/open.lists.txt

#Free disk space to a free\_disk.txt file

df -h > ~/backups/freedisk/free\_disk.txt

1. Command to make the system.sh script executable:

**Chmod +x system.sh**

**Optional**

* Commands to test the script and confirm its execution:

./system.sh

Text

Description automatically generated with low confidence

**Bonus**

* Command to copy system to system-wide cron directory:

**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.

1. Text

   Description automatically generated

**Bonus: 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:  
     
   * Add the edits made to the configuration file below:

Text

Description automatically generated

1. Text

   Description automatically generated
2. 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:
3. [Your solution edits here]
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:
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:  
   **$journalctl -p warning**
2. Command to check the disk usage of the system journal unit since the most recent boot:  
   **$journalctl -b**
3. Comand to remove all archived journal files except the most recent two:  
   **$journalctl -n 2**
4. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority\_High.txt:
5. Command to automate the last command in a daily cronjob. Add the edits made to the crontab file below:  
     
    [Your solution cron edits here]