**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

1. Command to **create** the Javaless\_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:  
   tar cvf Javaless\_Doc.tar TarDocs --exclude=/Java\*

1. Command to ensure Java/ is not in the new Javaless\_Docs.tar archive:  
   tar tvvf Javaless\_Doc.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:

tar tvvf logs\_backup\_tar.snar  /var/log --incremental

**Critical Analysis Question**

* Why wouldn't you use the options -x and -c at the same with tar? -c is used to create a compressed file and -x is used to extract a zipped file.

**Step 2: Create, Manage, and Automate Cron Jobs**

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

0 18 \* \* 0-6 mv  /var/log/auth.log file /usr/backups >/dev/null 2>&1

**Step 3: Write Basic Bash Scripts**

1. Brace expansion command to create the four subdirectories:  
   Paste your system.sh script edits below:  
     
   #!/bin/bash

sudo mkdir {dir1,dir2,dir3,dir4}

1. Command to make the system.sh script executable:  
   chmod +x system.sh

**Optional**

* Commands to test the script and confirm its execution:

test system.sh

**Bonus**

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

rsync -avzh --stats --progress remoteuser@remoteip  localpath

**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:

/var/log/auth.log

{

# auth config for rotating logs

rotate 180

daily

notifempty

compress

delaycompress

endscript

}

0 18 \* \* 0-6 mv ~/Downloads/doctors\*.docx /usr/share/doctors >/dev/null 2>&1

**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:  sudo nano /etc/audit/auditd.conf
3. Command using auditd to set rules for /etc/shadow, /etc/passwd and /var/log/auth.log   sudo nano /etc/audit/rules.d/audit.rule
4. Command to restart auditd: systemctl restart auditd
5. Command to list all auditd rules:  sudo nano /etc/audit/rules.d/audit.rules
6. Command to produce an audit report:
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:

auditctl -w /var/log/cron

1. 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:
2. Command to check the disk usage of the system journal unit since the most recent boot:
3. Comand to remove all archived journal files except the most recent two:
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]