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

1. Command to **extract** the TarDocs.tar archive to the current directory:
   tar -xf TarDocs.tar
2. Command to **create** the Javaless_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:
   tar -cf Javaless_Doc.tar --exclude=Documents/Java Tardocs/
3. Command to ensure Java/ is not in the new Javaless_Docs.tar archive:
   tar -cf 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 -etvf --listed-incremental=snapshot.file -cvzf logs_backup.tar.gz /var/log/*

Critical Analysis Question

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

  -x extracts files from the archive whilst -c creates the archive they are conflicting process

---

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

Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:
   ~/backups/freemem ~/backups/diskuse

   ~/backups/openlist ~/backups/freedisk

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

Paste your system.sh script edits below:

```
 #!/bin/bash
free -h >>~/backups/freemem/free_mem.txt
df -h >> ~/backups/diskuse/disk_usage/txt
lsof >> ~/backups/openlist/open_list.txt
```

du -h >> ~/backups/freedisk/free_disk.txt

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

   chmod -x system.sh

Optional

- Commands to test the script and confirm its execution:

  cat ~/backups/freemem/free_mem.txt

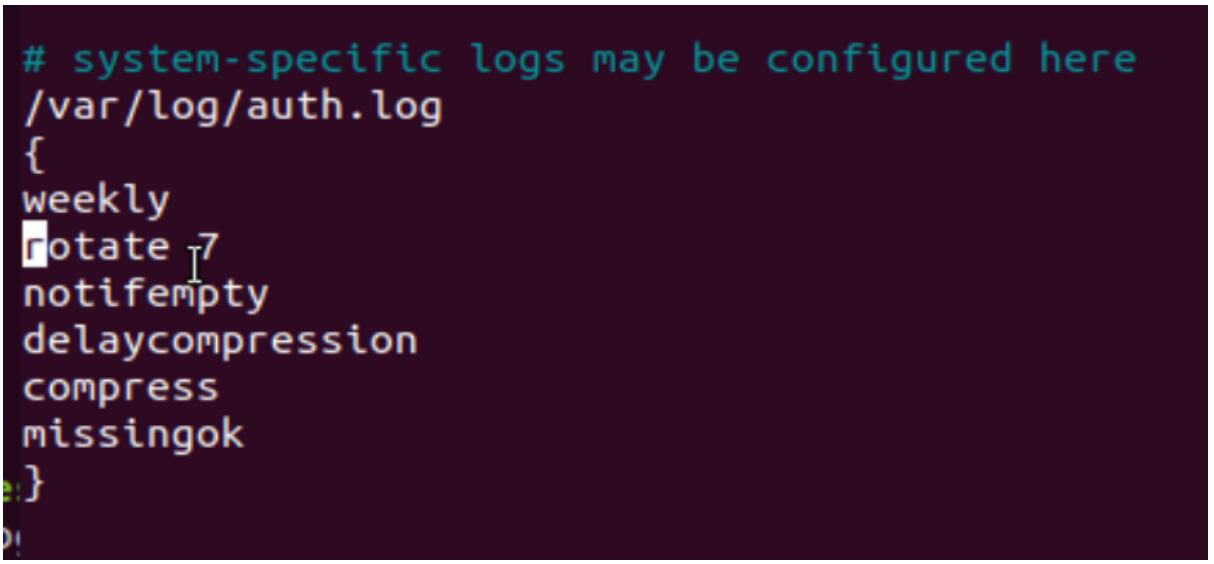
Bonus

- cp system.sh /etc/cron.weekly

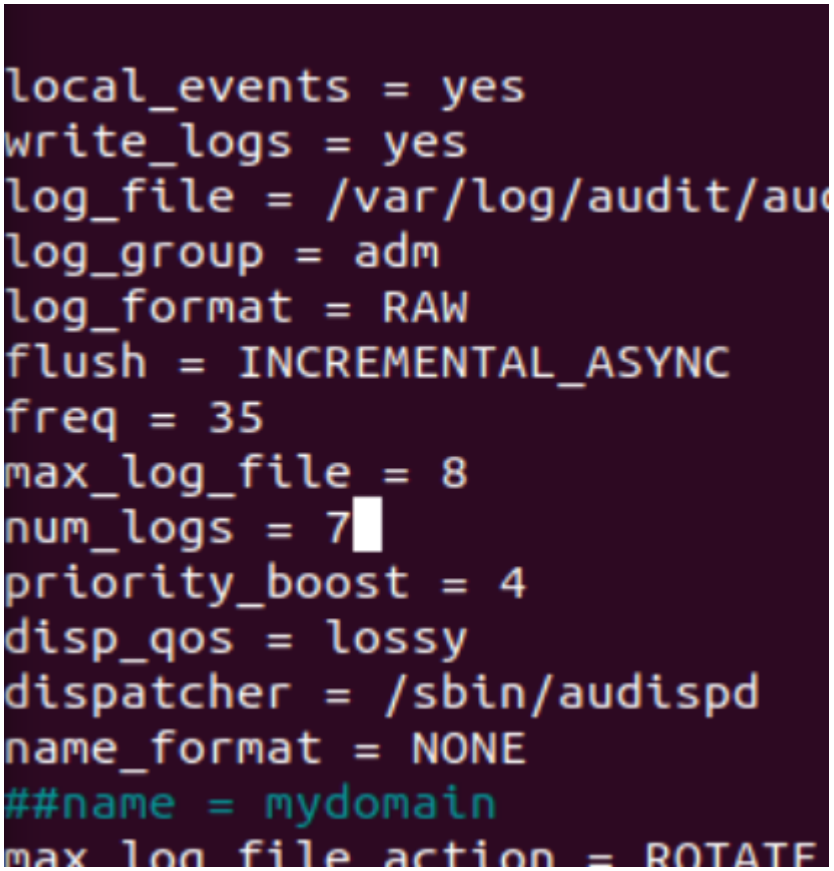
Step 4. Manage Log File Sizes

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 directory using the following settings:

- 

Bonus: Check for Policy and File Violations

1. Verify the auditd service is active using the systemctl command.
   Systemctl status audit
2. Run sudo nano /etc/audit/auditd.conf to edit the auditd config file using the following parameters. You can run this command from anywhere using the terminal.
   - 

3. Next, run sudo nano /etc/audit/rules.d/audit.rules to edit the rules for auditd. Create rules that watch the following paths:
   - For /etc/shadow, set wra for the permissions to monitor and set the keyname for this rule to hashpass_audit.
   - For /etc/passwd, set wra for the permissions to monitor and set the keyname for this rule to userpass_audit.
   - For /var/log/auth.log, set wra for the permissions to monitor and set the keyname for this rule to authlog_audit.

     -w /etc/shadow -p wra -k hashpass_audit

4. Restart the auditd daemon.
   sudo systemctl restart auditd
5. Perform a listing that reveals all existing auditd rules.

```
sysadmin@UbuntuDesktop:/etc$ sudo systemctl restart auditd
sysadmin@UbuntuDesktop:/etc$ sudo auditctl -l
-w /etc/shadow -p rwa -k hashpass_audit
-w /etc/passwd -p rwa -k userpass_audit
sysadmin@UbuntuDesktop:/etc$
```

6. Using sudo, produce an audit report that returns results for all user authentications.

```
sysadmin@UbuntuDesktop:~$ sudo aureport -au

Authentication Report
============================================
# date time acct host term exe success event
============================================
1. 09/28/2021 06:30:06 gdm UbuntuDesktop /dev/tty1 /usr/lib/gdm3/gdm-session-wo
rker yes 261
2. 09/28/2021 06:31:01 sysadmin UbuntuDesktop /dev/tty1 /usr/lib/gdm3/gdm-sessi
on-worker yes 291
3. 09/28/2021 06:31:55 sysadmin ? /dev/pts/0 /usr/bin/sudo no 307
4. 09/28/2021 06:31:59 sysadmin ? /dev/pts/0 /usr/bin/sudo yes 308
5. 09/28/2021 06:36:15 sysadmin ? ? /usr/lib/policykit-1/polkit-agent-helper-1
yes 375
6. 09/28/2021 06:40:56 sysadmin UbuntuDesktop /dev/tty2 /usr/lib/gdm3/gdm-sessi
on-worker yes 434
sysadmin@UbuntuDesktop:~$
```

7. Now you will shift into hacker mode. Create a user with sudo useradd attacker and produce an audit report that lists account modifications.

   Sudo aureport -m

8. Use auditctl to add another rule that watches the /var/log/cron directory.
   sudo auditctl -s /var/log/cron
9. Perform a listing that reveals changes to the auditd rules took affect.

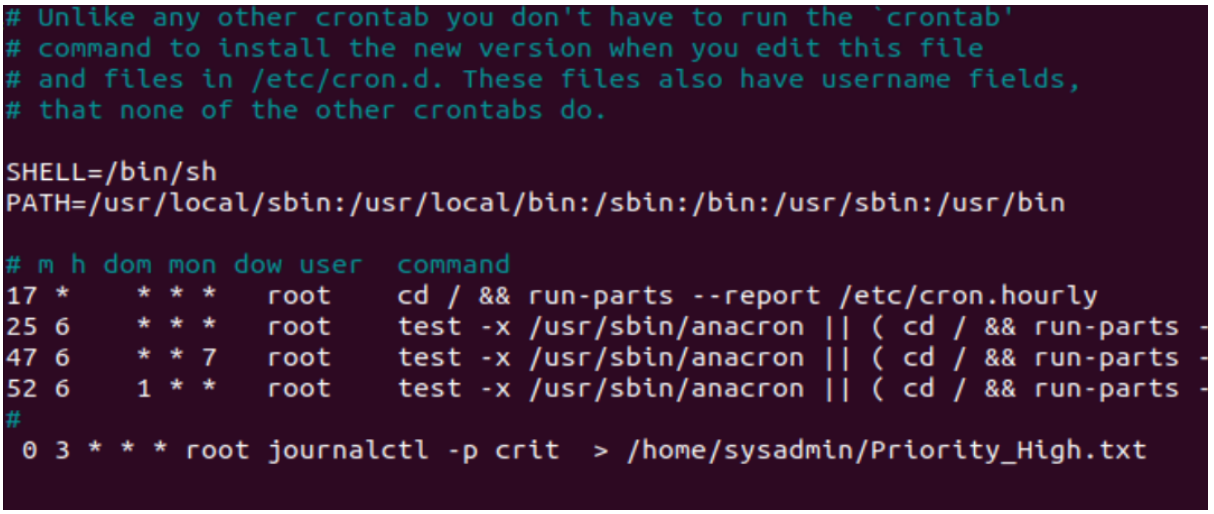
```
sysadmin@UbuntuDesktop:~$ sudo auditctl -l
[sudo] password for sysadmin:
-w /etc/shadow -p wa -k hashpass_audit
-w /etc/passwd -p wa -k userpass_audit
```

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
2. Command to check the disk usage of the system journal unit since the most recent boot:
   sudo journalctl -b -u systemd-journald | less
3. Comand to remove all archived journal files except the most recent two:
   sudo journalctl --vacuum-files=2

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 > /home/sysadmin/Priority_High.txt

   Automate the last task by creating a cron job that runs daily in the user crontab.



```
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts -
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts -
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts -
#
 0 3 * * * root journalctl -p crit  > /home/sysadmin/Priority_High.txt
```

0 0 * * * journalctl -p crit > /home/sysadmin/Priority_High.txt >/dev/null 2>&1