ANSWERS IN BLUE

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

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

   tar -vxf TarDocs/

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

   Javaless_Docs.tar --exclude="TarDocs/Documents/Java" TarDocs

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

   tar -tvf Javaless_Docs.tar | grep -I '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 -cvvzf logs_backup.tar.gz --listed-incremental=logarch.snar --level=0 log/

  Attempt to login to various users using incorrect passwords

  sudo tar -cvvzf logs_backup1.tar.gz --listed-incremental=logarch.snar --level=1 log/

  Verify :

  tar -tvvf log_backup1.tar.gz --incremental

Critical Analysis Question

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

Step 2: Create, Manage, and Automate Cron Jobs

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

   Navigate to /var and sudo mkdir ARCHIVE

   Crontab -e

   Cron line: (this will run on Wednesdays at 6AM, define a variable for that day, create an archive compressed file with the date in the title, then search for any records older than 5 years within the folder and remove them ((chosen because it's a financial institution standard…no particular reason)))

   0 6 * * 3 date_now=$(date "+%F") && tar -cvvzf /var/ARCHIVE/auth_backup$date_now.tar.gz /var/log/auth.log && find /var/ARCHIVE/ -type f -maxdepth 1 -exec rm {} \;

Step 3: Write Basic Bash Scripts

Brace expansion command to create the four subdirectories:

mkdir /home/sysadmin/Projects/backups/{freemem,diskuse,openlist,freedisk}

Paste your system.sh script edits below:

```
#!/bin/bash

#script for homework 5

#check for the existence of sub directories, if not there, create them:
# help with the below code came from reviewing this website:https://unix.stackexchange.com/questions/503830/checking-for-the-existence-of-multiple-directories


direct=("/home/sysadmin/Projects/backups/freemem/" "/home/sysadmin/Projects/backups/diskuse/" "/home/sysadmin/Projects/backups/openlist/" "/home/sysadmin/Projects/backups/freedisk/")
```

```
for dir in "${direct[@]}";
do
if [ -d "$dir" ]
then
echo " $dir directory already exists!"
else
mkdir $dir
fi
done

#define a variable for the date format, for reference throughout remaining script
d=$(date "+%F")

#script to write available free memory to freemem folder:
echo "Writing available free memory to
/home/sysadmin/Projects/backup/freemem/free_mem_$d.txt"
free -h | awk -F" " 'NR == 2 {print "Free Memory: "$4""}' >> free_mem_$d.txt && echo $(date
"+%F") >> free_mem_$d.txt && mv free_mem_$d.txt
/home/sysadmin/Projects/backups/freemem/

#command to print diskusage to disk use folder
echo "Writing current disk usage to /home/sysadmin/Projects/backup/diskuse/disk_usage$d.txt"
echo $(date "+%F") >> disk_usage$d.txt && df -h | awk -F" " '{print "Disk Usage: "$1" "$3""}' >>
disk_usage$d.txt && du -h >> disk_usage$d.txt && mv disk_usage$d.txt
/home/sysadmin/Projects/backups/diskuse/


# command to list all open files
echo "Writing all current open files to /home/sysadmin/Projects/backup/openlist/open_list$d.txt"
echo $(date "+%F") >> open_list$d.txt && lsof /dev/null >> open_list$d.txt && mv open_list$d.txt
/home/sysadmin/Projects/backups/openlist/

#create a script which shows available free space
echo "Writing current free disk space to
/home/sysadmin/Projects/backup/freedisk/free_disk$d.txt"
echo $(date "+%F") >> free_disk$d.txt && df -h | awk -F" " '{print "Free Disk Space: "$1" "$4""}'
>> free_disk$d.txt && mv free_disk$d.txt /home/sysadmin/Projects/backups/freedisk/
```

Command to make the system.sh script executable:

```
sudo chmod +x system.sh
```

Optional

- Commands to test the script and confirm its execution:

  bash system.sh

  Cd backups

  ls

Bonus

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

  sudo crontab -e

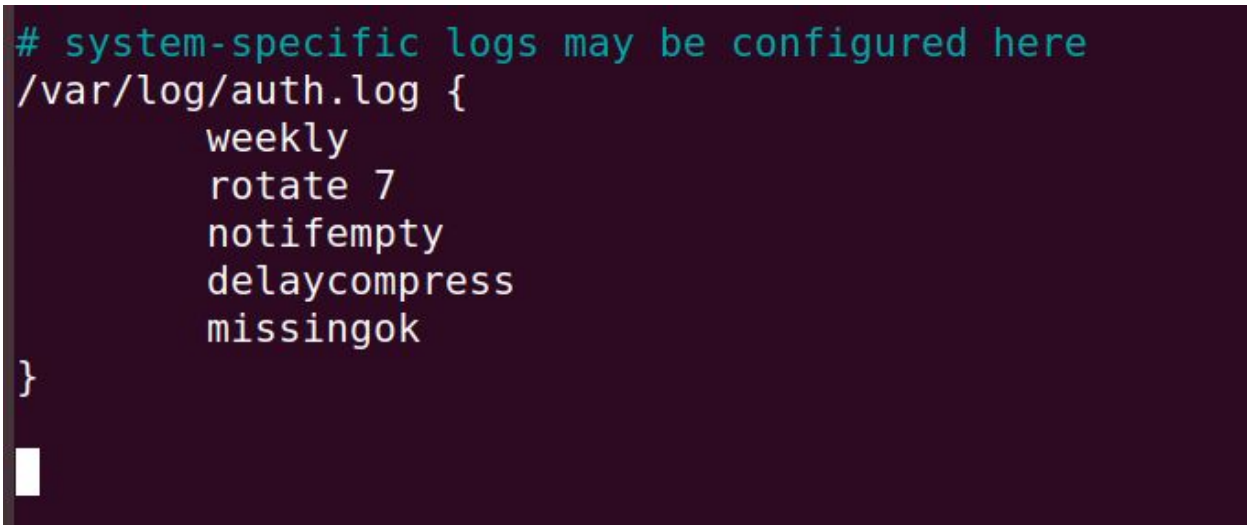
  0 6 * * 1 bash /home/sysadmin/Projects/system.sh

  ctrl + x

Step 4. Manage Log File Sizes

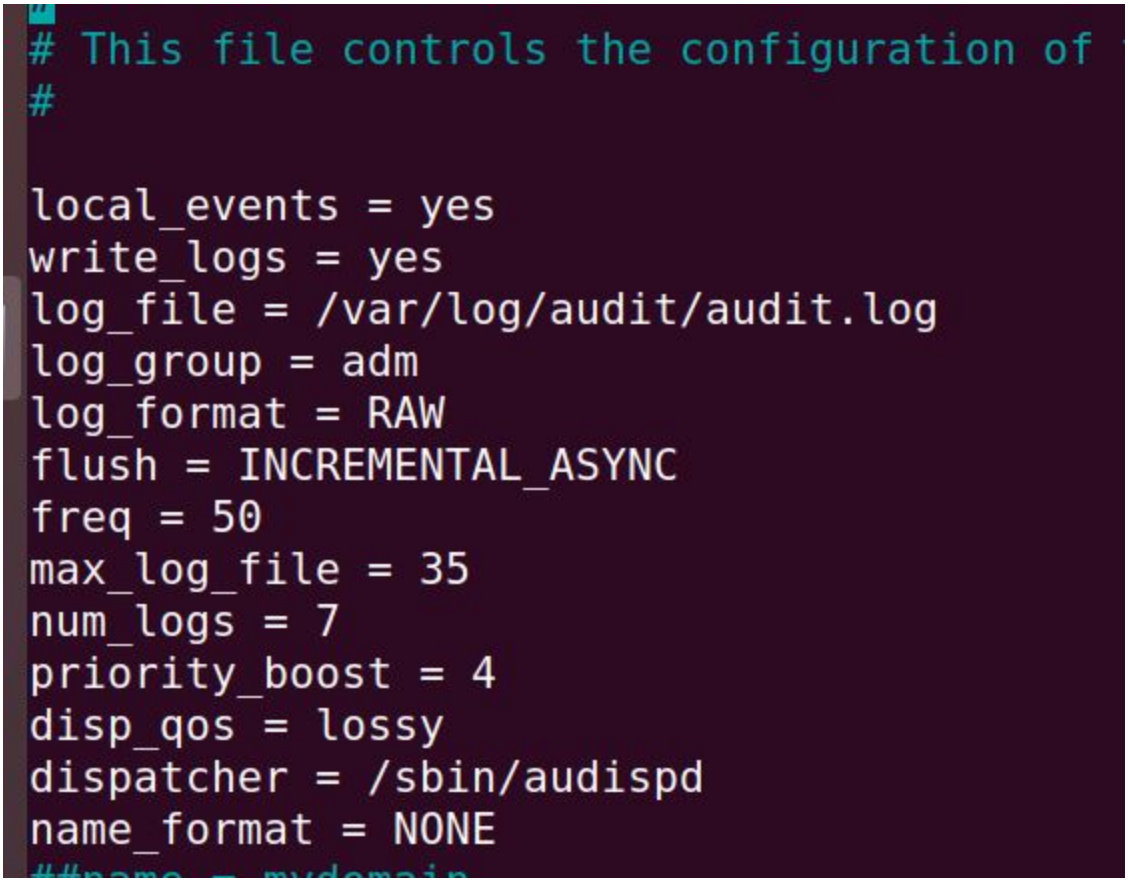
1. Run sudo nano /etc/logrotate.conf to edit the logrotate configuration file.

   Configure a log rotation scheme that backs up authentication messages to the /var/log/auth.log.

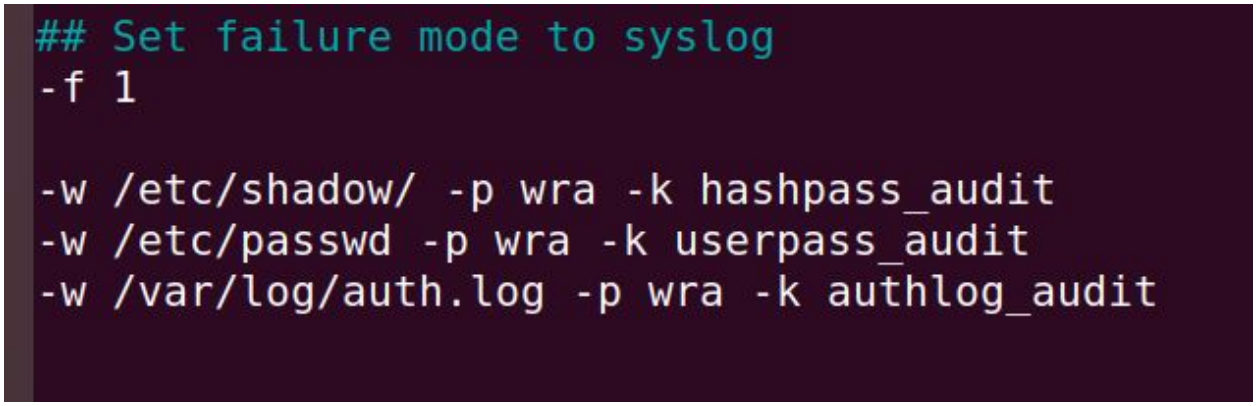
   - 

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:
   - 

3. Command using auditd to set rules for /etc/shadow, /etc/passwd and /var/log/auth.log:



4. Command to restart auditd:
   service auditd restart
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:
   sudo auditctl -w /var/log/cron/ -p wra -k cron_audit
9. Command to verify auditd rules:
   sudo auditctl -l

Bonus (Research Activity): Perform Various Log Filtering Techniques

Used : https://www.loggly.com/ultimate-guide/using-journalctl/

1. Command to return journalctl messages with priorities from emergency to error:
   `journalctl -b -1  -p "emerg".."error"`

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:

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]