CYSE 270: Linux System for Cybersecurity

Week 10: Lab 9 – Crontab & Scheduling Tasks

Date: July 20, 2025

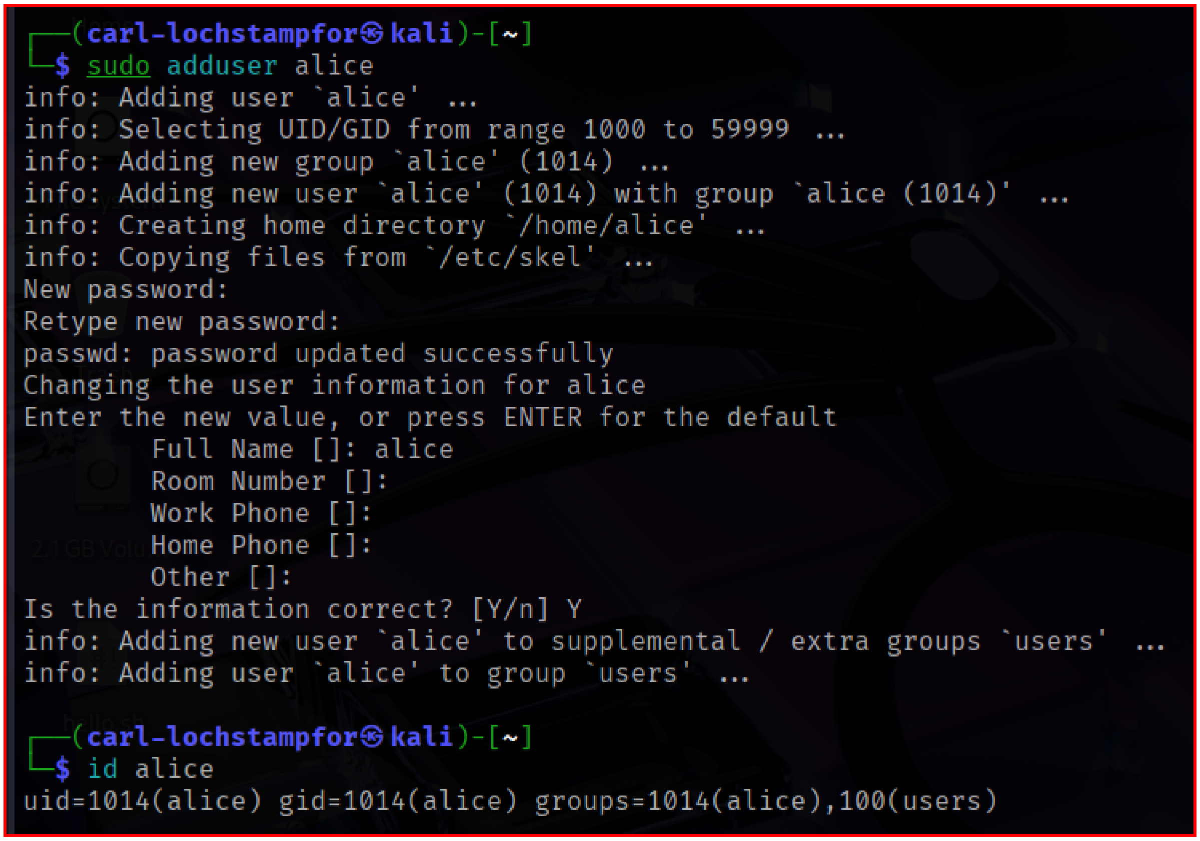
(Total 100 Points)

Task A - Backup your system (Using crontab) [100 points]

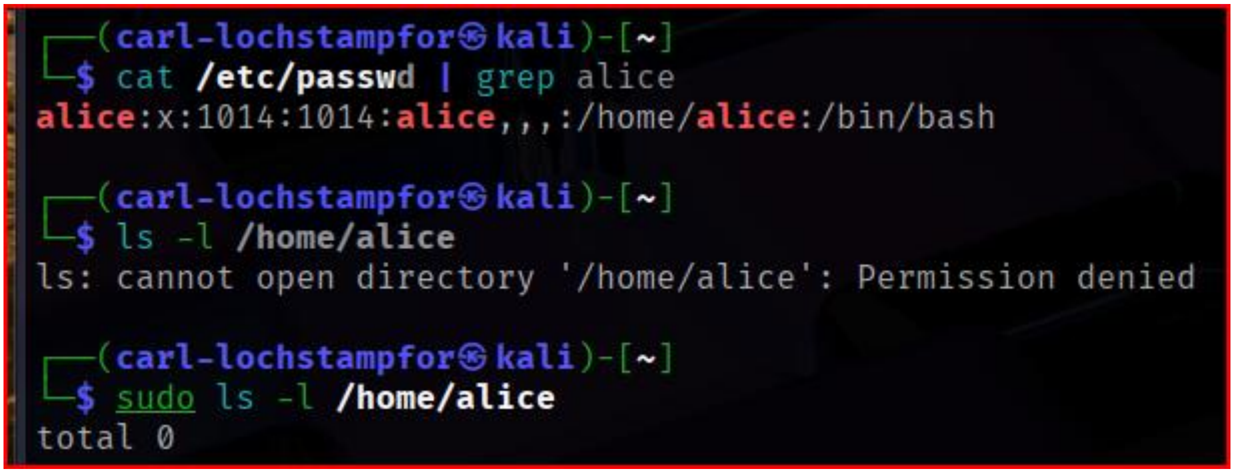
Scenario: Performing system backup can be time-consuming, and the process is often overlooked. For this scenario:

1. **(10 Points)** Create a new user **Alice (with home directory**).

Method 1

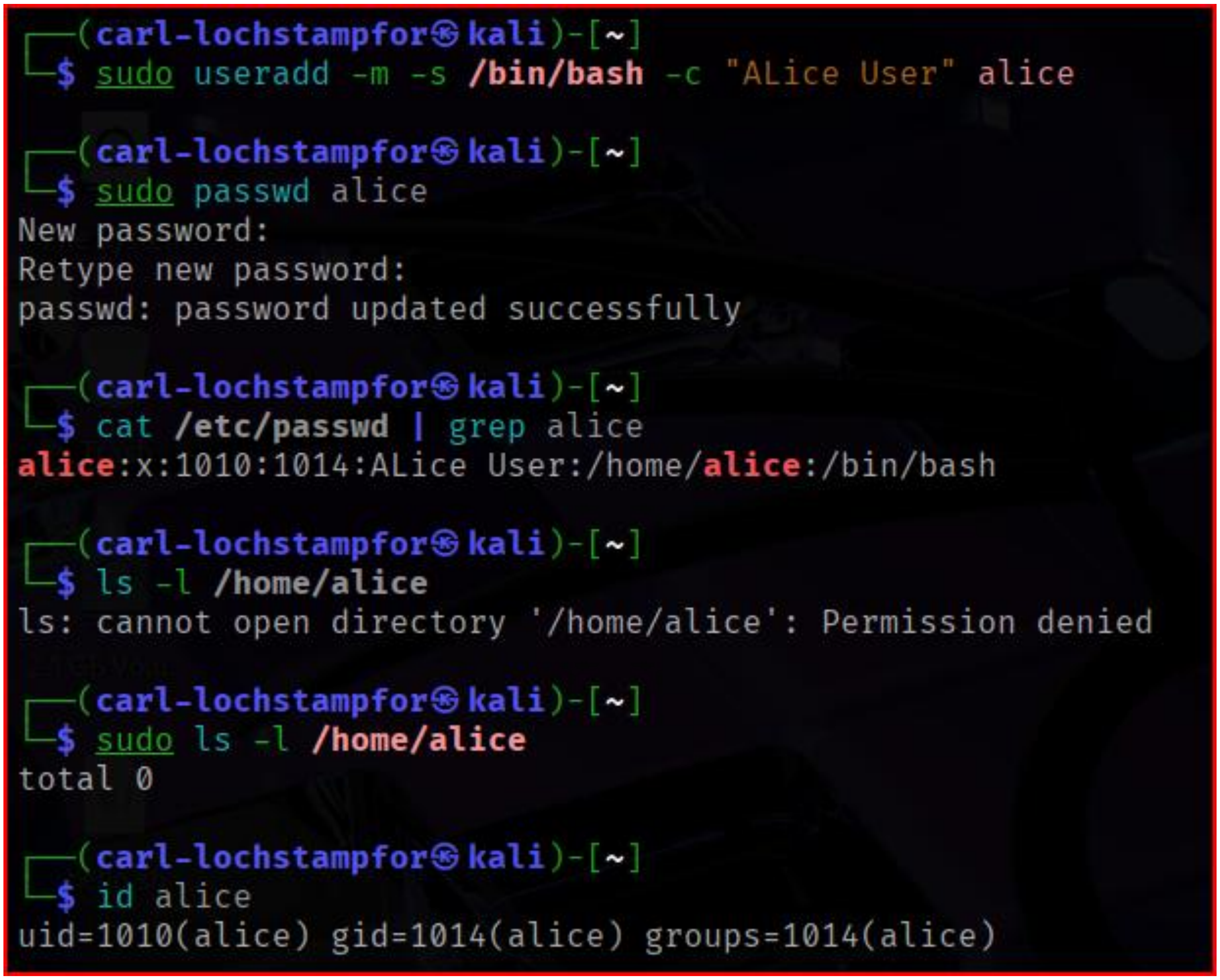


```
┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo adduser alice
info: Adding user `alice' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `alice' (1014) ...
info: Adding new user `alice' (1014) with group `alice (1014)' ...
info: Creating home directory `/home/alice' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alice
Enter the new value, or press ENTER for the default
        Full Name []: alice
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
info: Adding new user `alice' to supplemental / extra groups `users' ...
info: Adding user `alice' to group `users' ...

┌──(carl-lochstampfor㉿kali)-[~]
└─$ id alice
uid=1014(alice) gid=1014(alice) groups=1014(alice),100(users)
```



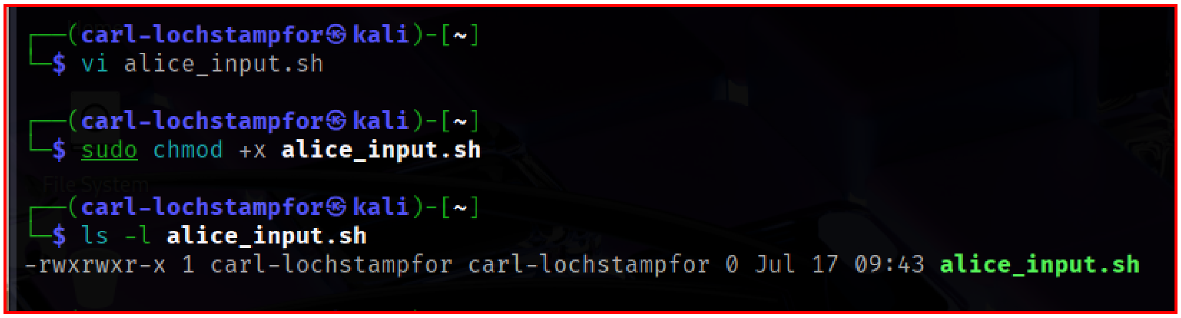
```
┌──(carl-lochstampfor㉿kali)-[~]
└─$ cat /etc/passwd | grep alice
alice:x:1014:1014:alice,,,:/home/alice:/bin/bash

┌──(carl-lochstampfor㉿kali)-[~]
└─$ ls -l /home/alice
ls: cannot open directory '/home/alice': Permission denied

┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo ls -l /home/alice
total 0
```

Method 2

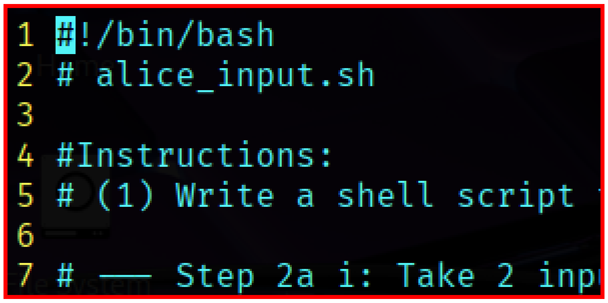


```
┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo useradd -m -s /bin/bash -c "ALice User" alice

┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo passwd alice
New password:
Retype new password:
passwd: password updated successfully

┌──(carl-lochstampfor㉿kali)-[~]
└─$ cat /etc/passwd | grep alice
alice:x:1010:1014:ALice User:/home/alice:/bin/bash

┌──(carl-lochstampfor㉿kali)-[~]
└─$ ls -l /home/alice
ls: cannot open directory '/home/alice': Permission denied

┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo ls -l /home/alice
total 0

┌──(carl-lochstampfor㉿kali)-[~]
└─$ id alice
uid=1010(alice) gid=1014(alice) groups=1014(alice)
```

2. **(50 Points) Write a shell script that backups Alice's home directory by creating a tar file (tape archive), using the following steps:**



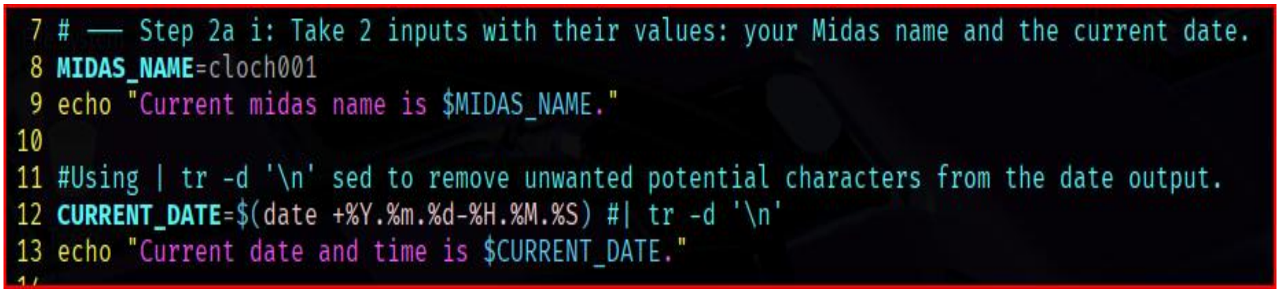
```
┌──(carl-lochstampfor㉿kali)-[~]
└─$ vi alice_input.sh

┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo chmod +x alice_input.sh

┌──(carl-lochstampfor㉿kali)-[~]
└─$ ls -l alice_input.sh
-rwxrwxr-x 1 carl-lochstampfor carl-lochstampfor 0 Jul 17 09:43 alice_input.sh
```



```
#!/bin/bash
# alice_input.sh

#Instructions:
# (1) Write a shell script

# —— Step 2a i: Take 2 inp
```

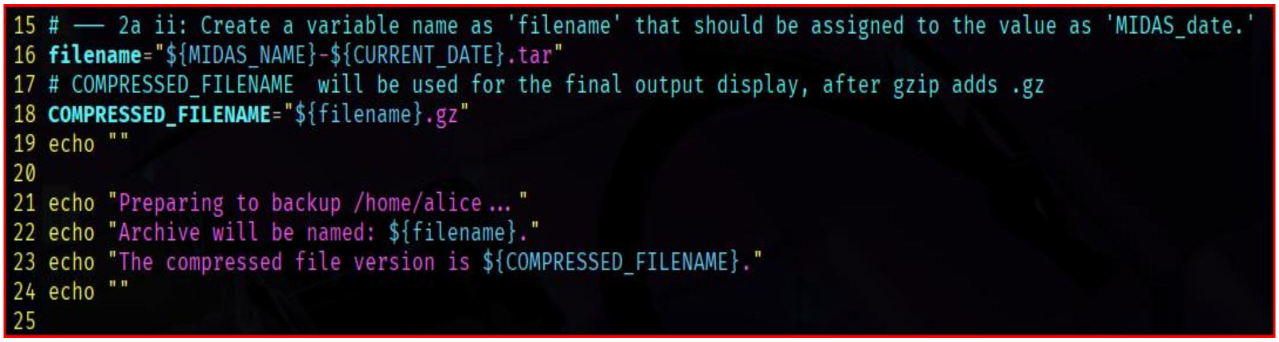
a. Do the following:

- Take **2 inputs** with their values- your **MIDAS (cloch001)** name and **current date.**



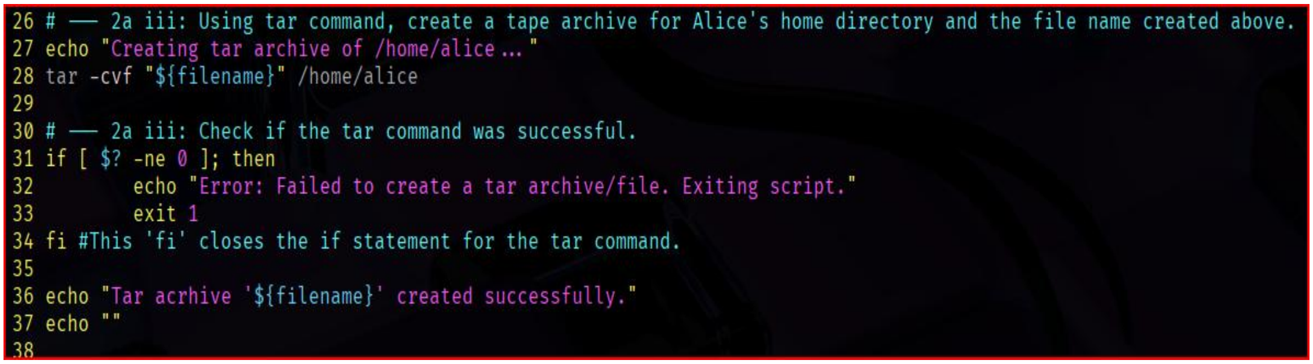
```
# —— Step 2a i: Take 2 inputs with their values: your Midas name and the current date.
MIDAS_NAME=cloch001
echo "Current midas name is $MIDAS_NAME."

#Using | tr -d '\n' sed to remove unwanted potential characters from the date output.
CURRENT_DATE=$(date +%Y.%m.%d-%H.%M.%S) #| tr -d '\n'
echo "Current date and time is $CURRENT_DATE."
```

- Create a variable named as **filename** that should be assigned the value as **MIDAS-date** (example output after executing the script would be like, **Mohammed-2024.11.04-22.08.01.tar.gz**).



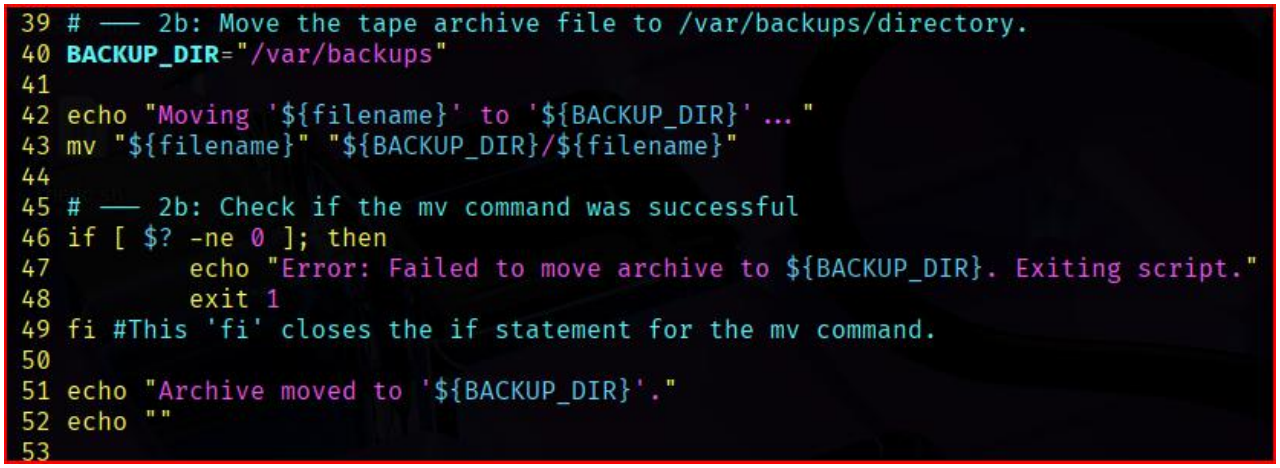
```
# —— 2a ii: Create a variable name as 'filename' that should be assigned to the value as 'MIDAS_date.'
filename="${MIDAS_NAME}-${CURRENT_DATE}.tar"
# COMPRESSED_FILENAME  will be used for the final output display, after gzip adds .gz
COMPRESSED_FILENAME="${filename}.gz"
echo ""

echo "Preparing to backup /home/alice..."
echo "Archive will be named: ${filename}."
echo "The compressed file version is ${COMPRESSED_FILENAME}."
echo ""

```

- Using **tar** command, create a tape archive for Alice's home directory (/home/Alice) and the **filename** created above (in step-2-ii). (Please learn about tar command in Linux for its usage).



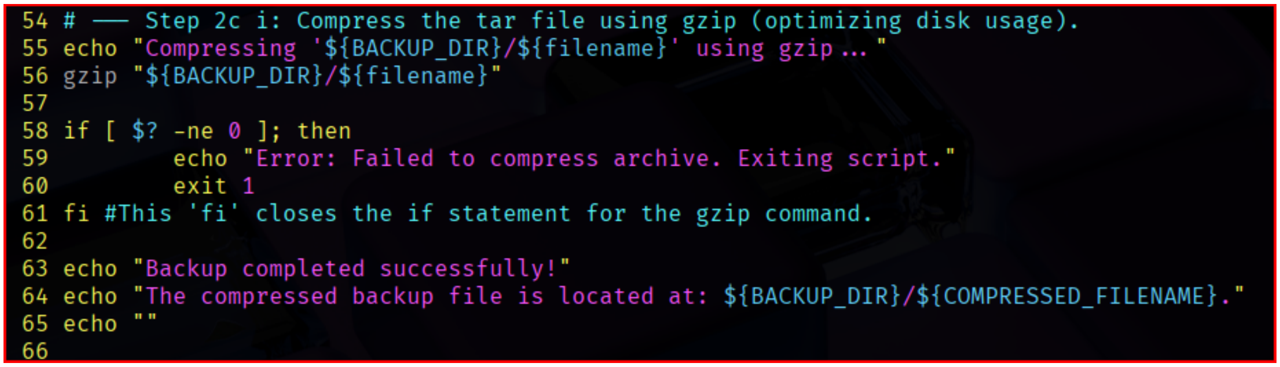
```
# —— 2a iii: Using tar command, create a tape archive for Alice's home directory and the file name created above.
echo "Creating tar archive of /home/alice ... "
tar -cvf "${filename}" /home/alice

# —— 2a iii: Check if the tar command was successful.
if [ $? -ne 0 ]; then
        echo "Error: Failed to create a tar archive/file. Exiting script."
        exit 1
fi #This 'fi' closes the if statement for the tar command.

echo "Tar acrhive '${filename}' created successfully."
echo ""
```

b. Move the tape archive file/tar file (created in step 2-iii) to /var/backups/ directory using correct command in linux.



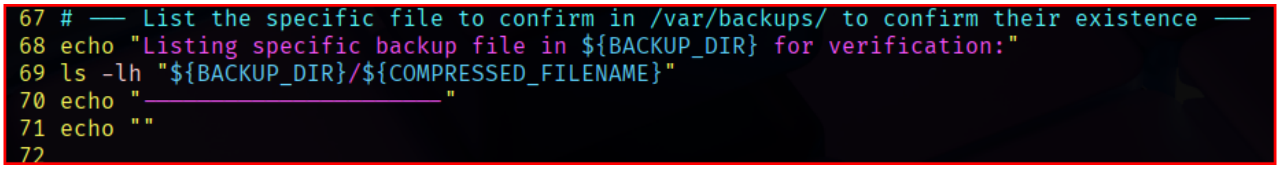
```
# —— 2b: Move the tape archive file to /var/backups/directory.
BACKUP_DIR="/var/backups"

echo "Moving '${filename}' to '${BACKUP_DIR}' ... "
mv "${filename}" "${BACKUP_DIR}/${filename}"

# —— 2b: Check if the mv command was successful
if [ $? -ne 0 ]; then
        echo "Error: Failed to move archive to ${BACKUP_DIR}. Exiting script."
        exit 1
fi #This 'fi' closes the if statement for the mv command.

echo "Archive moved to '${BACKUP_DIR}'."
echo ""
```

c. To optimize the disk usage, pick a compression algorithm (bz2, gzip, or xv) to compress the tar file you created in /var/backups/ in the previous step-2b.



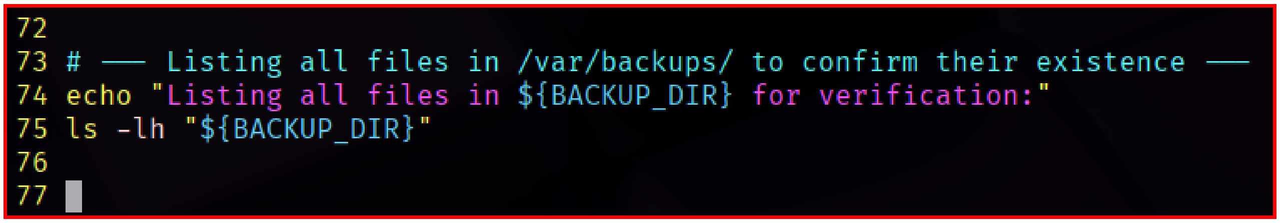
```
# —— Step 2c i: Compress the tar file using gzip (optimizing disk usage).
echo "Compressing '${BACKUP_DIR}/${filename}' using gzip ... "
gzip "${BACKUP_DIR}/${filename}"

if [ $? -ne 0 ]; then
        echo "Error: Failed to compress archive. Exiting script."
        exit 1
fi #This 'fi' closes the if statement for the gzip command.

echo "Backup completed successfully!"
echo "The compressed backup file is located at: ${BACKUP_DIR}/${COMPRESSED_FILENAME}."
echo ""
```

Optional: Listing the specific file and all of the files in /var/backups to confirm they exist.



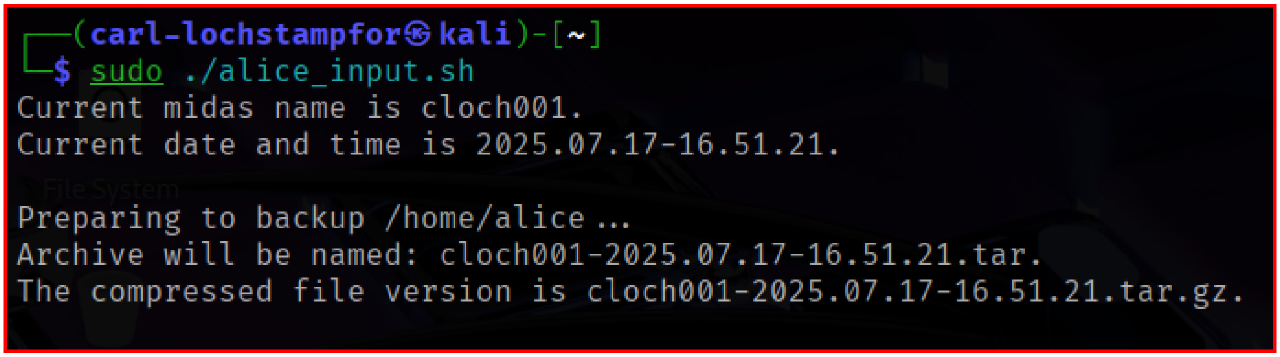
```
# —— List the specific file to confirm in /var/backups/ to confirm their existence ——
echo "Listing specific backup file in ${BACKUP_DIR} for verification:"
ls -lh "${BACKUP_DIR}/${COMPRESSED_FILENAME}"
echo "————————————————————"
echo ""
```

Output of the code

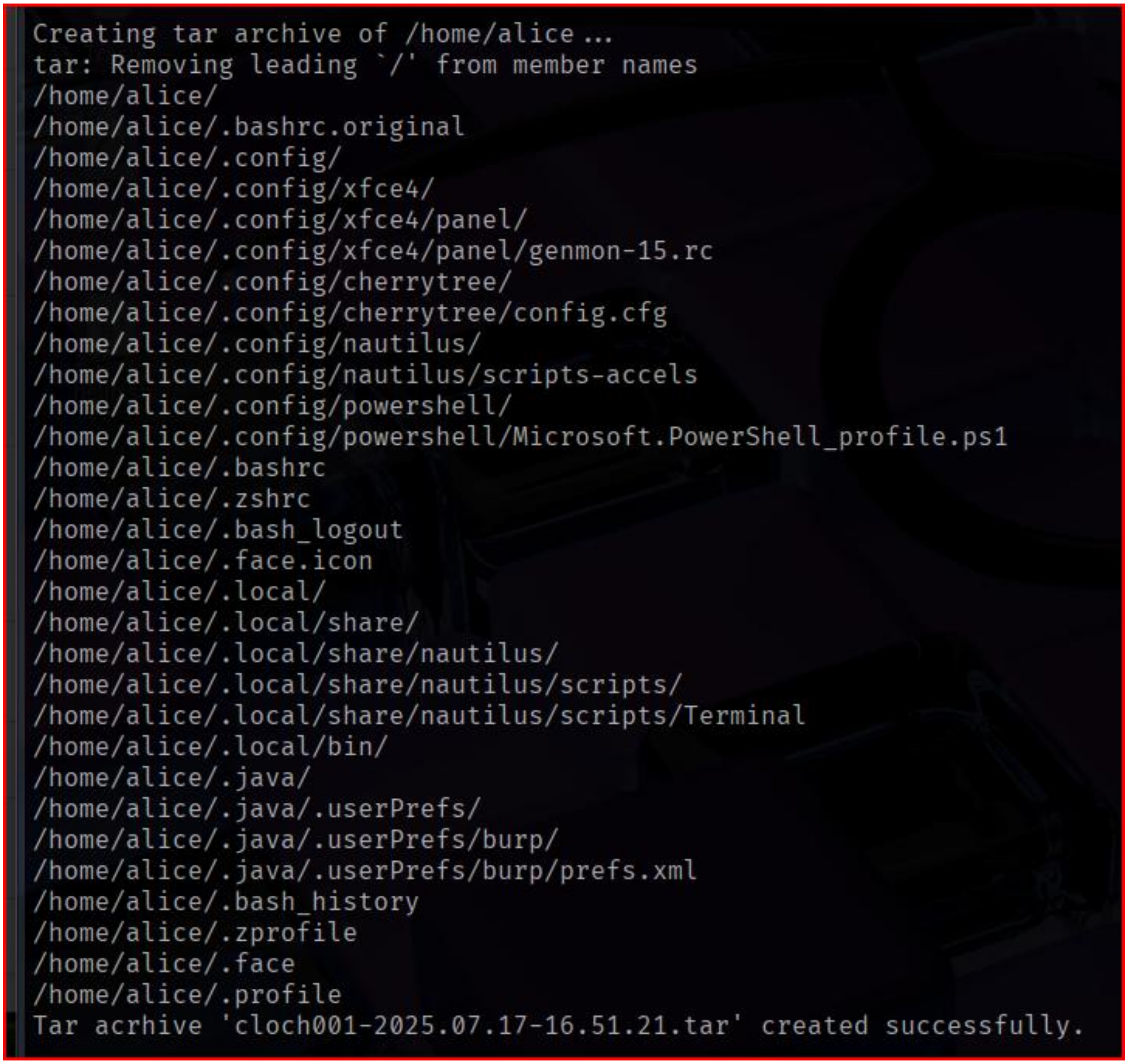


```
# —— Listing all files in /var/backups/ to confirm their existence ——
echo "Listing all files in ${BACKUP_DIR} for verification:"
ls -lh "${BACKUP_DIR}"
```

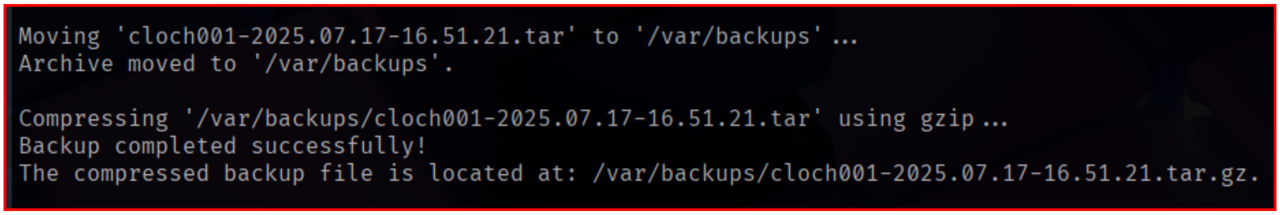
Manual Run of the Script to further verify everything was working correctly.



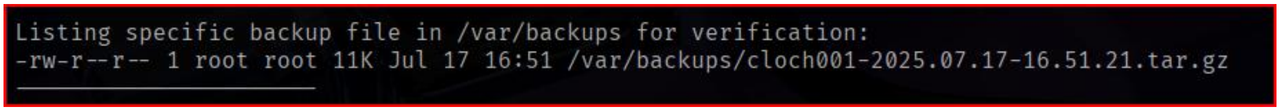
```
┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo ./alice_input.sh
Current midas name is cloch001.
Current date and time is 2025.07.17-16.51.21.

Preparing to backup /home/alice ...
Archive will be named: cloch001-2025.07.17-16.51.21.tar.
The compressed file version is cloch001-2025.07.17-16.51.21.tar.gz.
```



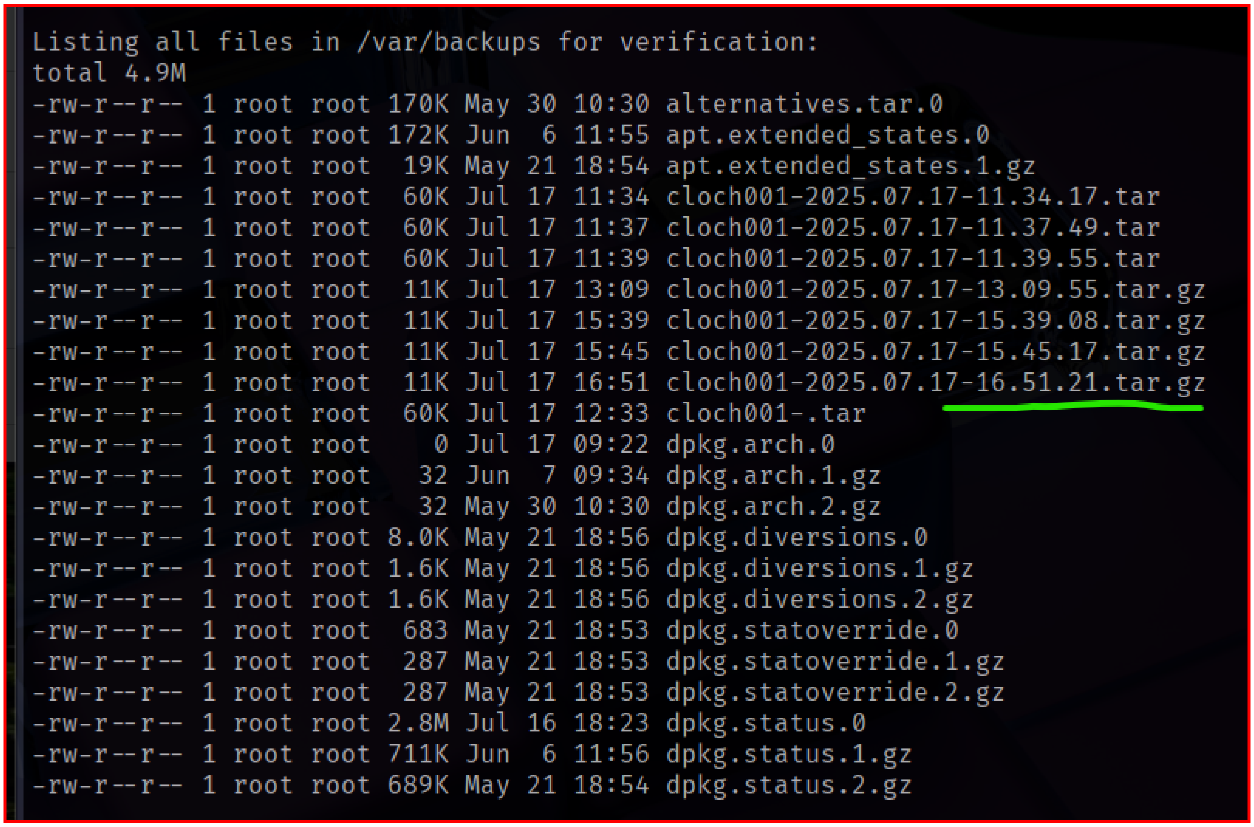
```
Creating tar archive of /home/alice ...
tar: Removing leading `/' from member names
/home/alice/
/home/alice/.bashrc.original
/home/alice/.config/
/home/alice/.config/xfce4/
/home/alice/.config/xfce4/panel/
/home/alice/.config/xfce4/panel/genmon-15.rc
/home/alice/.config/cherrytree/
/home/alice/.config/cherrytree/config.cfg
/home/alice/.config/nautilus/
/home/alice/.config/nautilus/scripts-accels
/home/alice/.config/powershell/
/home/alice/.config/powershell/Microsoft.PowerShell_profile.ps1
/home/alice/.bashrc
/home/alice/.zshrc
/home/alice/.bash_logout
/home/alice/.face.icon
/home/alice/.local/
/home/alice/.local/share/
/home/alice/.local/share/nautilus/
/home/alice/.local/share/nautilus/scripts/
/home/alice/.local/share/nautilus/scripts/Terminal
/home/alice/.local/bin/
/home/alice/.java/
/home/alice/.java/.userPrefs/
/home/alice/.java/.userPrefs/burp/
/home/alice/.java/.userPrefs/burp/prefs.xml
/home/alice/.bash_history
/home/alice/.zprofile
/home/alice/.face
/home/alice/.profile
Tar acrhive 'cloch001-2025.07.17-16.51.21.tar' created successfully.
```



```
Moving 'cloch001-2025.07.17-16.51.21.tar' to '/var/backups' ...
Archive moved to '/var/backups'.

Compressing '/var/backups/cloch001-2025.07.17-16.51.21.tar' using gzip ...
Backup completed successfully!
The compressed backup file is located at: /var/backups/cloch001-2025.07.17-16.51.21.tar.gz.
```

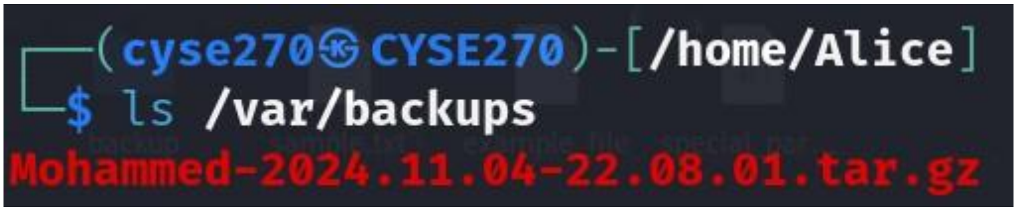


```
Listing specific backup file in /var/backups for verification:
-rw-r--r-- 1 root root 11K Jul 17 16:51 /var/backups/cloch001-2025.07.17-16.51.21.tar.gz
```

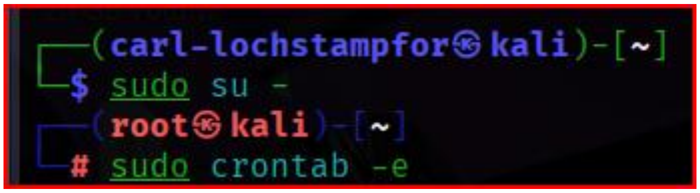


```
Listing all files in /var/backups for verification:
total 4.9M
-rw-r--r-- 1 root root 170K May 30 10:30 alternatives.tar.0
-rw-r--r-- 1 root root 172K Jun  6 11:55 apt.extended_states.0
-rw-r--r-- 1 root root  19K May 21 18:54 apt.extended_states.1.gz
-rw-r--r-- 1 root root  60K Jul 17 11:34 cloch001-2025.07.17-11.34.17.tar
-rw-r--r-- 1 root root  60K Jul 17 11:37 cloch001-2025.07.17-11.37.49.tar
-rw-r--r-- 1 root root  60K Jul 17 11:39 cloch001-2025.07.17-11.39.55.tar
-rw-r--r-- 1 root root  11K Jul 17 13:09 cloch001-2025.07.17-13.09.55.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 15:39 cloch001-2025.07.17-15.39.08.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 15:45 cloch001-2025.07.17-15.45.17.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 16:51 cloch001-2025.07.17-16.51.21.tar.gz
-rw-r--r-- 1 root root  60K Jul 17 12:33 cloch001-.tar
-rw-r--r-- 1 root root    0 Jul 17 09:22 dpkg.arch.0
-rw-r--r-- 1 root root   32 Jun  7 09:34 dpkg.arch.1.gz
-rw-r--r-- 1 root root   32 May 30 10:30 dpkg.arch.2.gz
-rw-r--r-- 1 root root 8.0K May 21 18:56 dpkg.diversions.0
-rw-r--r-- 1 root root 1.6K May 21 18:56 dpkg.diversions.1.gz
-rw-r--r-- 1 root root 1.6K May 21 18:56 dpkg.diversions.2.gz
-rw-r--r-- 1 root root  683 May 21 18:53 dpkg.statoverride.0
-rw-r--r-- 1 root root  287 May 21 18:53 dpkg.statoverride.1.gz
-rw-r--r-- 1 root root  287 May 21 18:53 dpkg.statoverride.2.gz
-rw-r--r-- 1 root root 2.8M Jul 16 18:23 dpkg.status.0
-rw-r--r-- 1 root root 711K Jun  6 11:56 dpkg.status.1.gz
-rw-r--r-- 1 root root 689K May 21 18:54 dpkg.status.2.gz
```

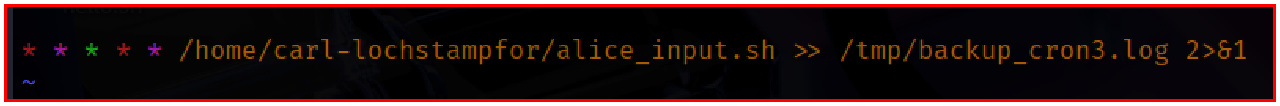
3. **(30 Points) Create a crontab file to keep the scheduled task running for 3 minutes**, then check the contents in the /var/backups directory. Your output should be look similar to the following:



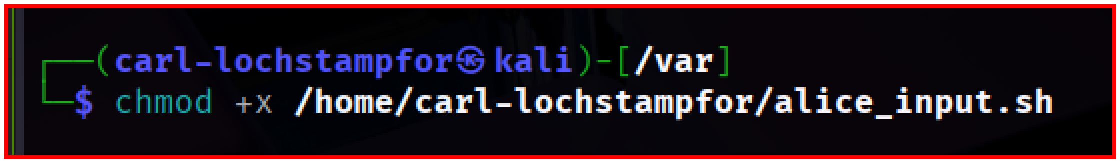
```
┌──(cyse270㉿CYSE270)-[/home/Alice]
└─$ ls /var/backups
Mohammed-2024.11.04-22.08.01.tar.gz
```



```
┌──(carl-lochstampfor㉿kali)-[~]
└─$ sudo su -
┌──(root㉿kali)-[~]
└─# sudo crontab -e
```



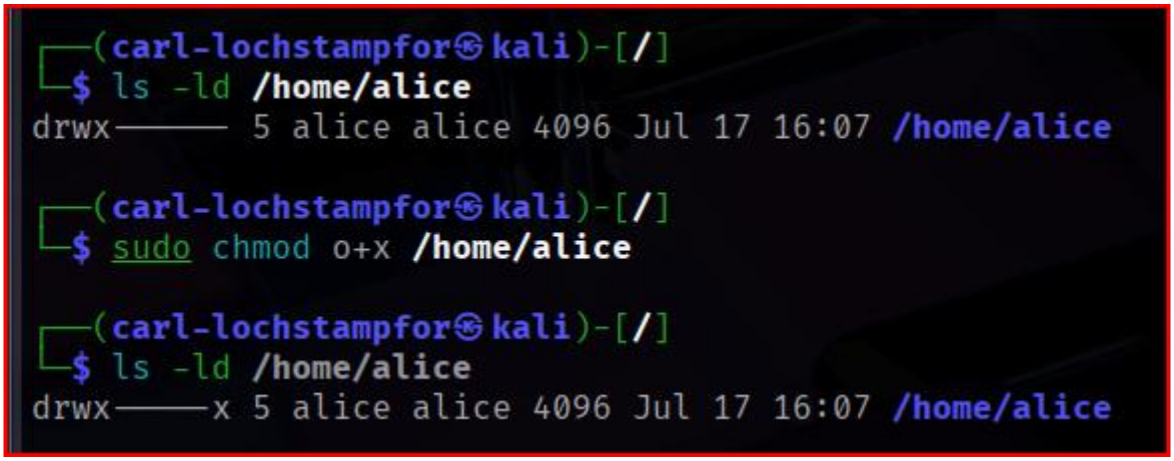
```
* * * * * /home/carl-lochstampfor/alice_input.sh >> /tmp/backup_cron3.log 2>&1
~
```



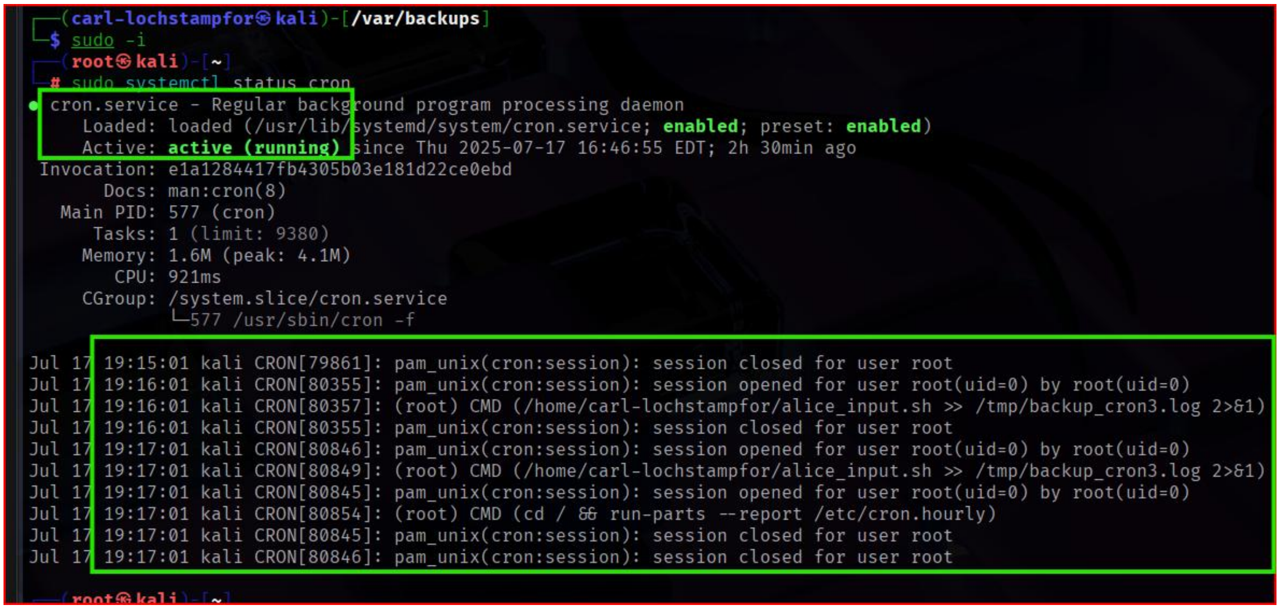
```
┌──(carl-lochstampfor㉿kali)-[/var]
└─$ chmod +x /home/carl-lochstampfor/alice_input.sh
```

NOTE:

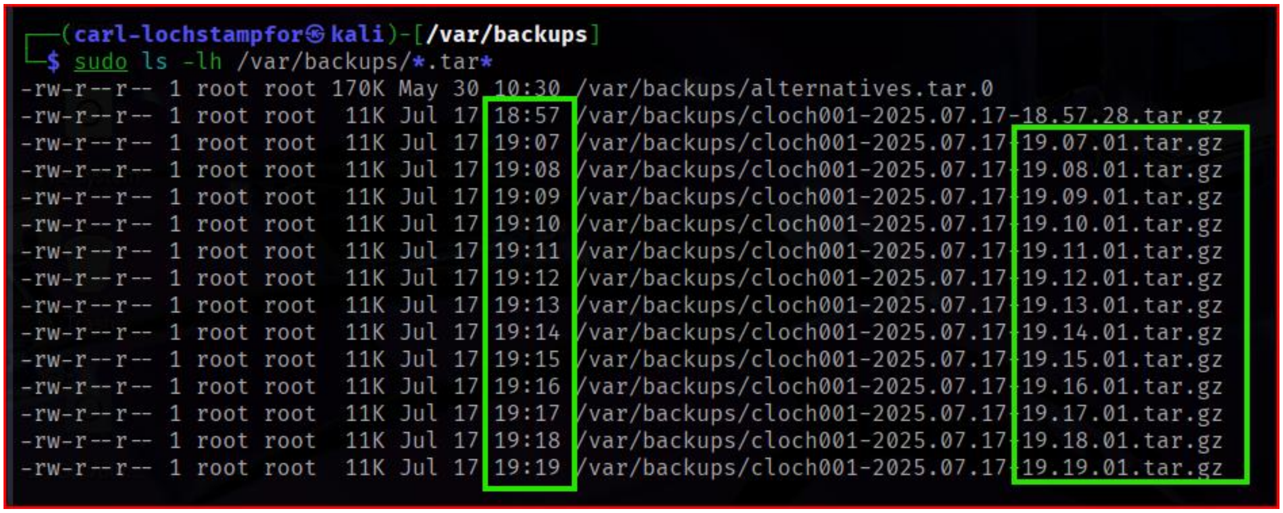
Had to change permissions (below) to run the backup script because root respected the sole permissions of /home/alice. An unknown bug was preventing me to run the crontab in the background, but I was still able to manually run the script via ./alice_input and run the tar file separately. Confirmed there were issues with the backup process beforehand through the root/sudo systemctl status cron. Once the bug was fixed, I returned to the root/sudo systemctl status cron to confirm everything was actively running and not abruptly ending.



```
┌──(carl-lochstampfor㉿kali)-[/]
└─$ ls -ld /home/alice
drwx------ 5 alice alice 4096 Jul 17 16:07 /home/alice

┌──(carl-lochstampfor㉿kali)-[/]
└─$ sudo chmod o+x /home/alice

┌──(carl-lochstampfor㉿kali)-[/]
└─$ ls -ld /home/alice
drwx-----x 5 alice alice 4096 Jul 17 16:07 /home/alice
```

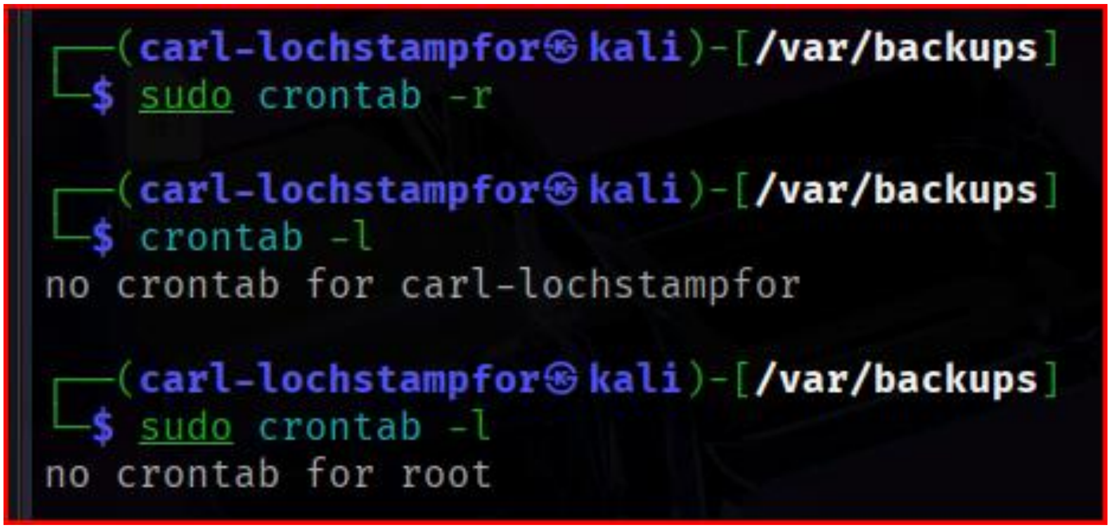


```
┌──(carl-lochstampfor㉿kali)-[/var/backups]
└─$ sudo -i
┌──(root㉿kali)-[~]
└─# sudo systemctl status cron
● cron.service - Regular background program processing daemon
     Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-07-17 16:46:55 EDT; 2h 30min ago
 Invocation: e1a1284417fb4305b03e181d22ce0ebd
       Docs: man:cron(8)
   Main PID: 577 (cron)
      Tasks: 1 (limit: 9380)
     Memory: 1.6M (peak: 4.1M)
        CPU: 921ms
     CGroup: /system.slice/cron.service
             └─577 /usr/sbin/cron -f

Jul 17 19:15:01 kali CRON[79861]: pam_unix(cron:session): session closed for user root
Jul 17 19:16:01 kali CRON[80355]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
Jul 17 19:16:01 kali CRON[80357]: (root) CMD (/home/carl-lochstampfor/alice_input.sh >> /tmp/backup_cron3.log 2>&1)
Jul 17 19:16:01 kali CRON[80355]: pam_unix(cron:session): session closed for user root
Jul 17 19:17:01 kali CRON[80846]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
Jul 17 19:17:01 kali CRON[80849]: (root) CMD (/home/carl-lochstampfor/alice_input.sh >> /tmp/backup_cron3.log 2>&1)
Jul 17 19:17:01 kali CRON[80845]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
Jul 17 19:17:01 kali CRON[80854]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Jul 17 19:17:01 kali CRON[80845]: pam_unix(cron:session): session closed for user root
Jul 17 19:17:01 kali CRON[80846]: pam_unix(cron:session): session closed for user root

┌──(root㉿kali)-[~]
```



```
┌──(carl-lochstampfor㉿kali)-[/var/backups]
└─$ sudo ls -lh /var/backups/*.tar*
-rw-r--r-- 1 root root 170K May 30 10:30 /var/backups/alternatives.tar.0
-rw-r--r-- 1 root root  11K Jul 17 18:57 /var/backups/cloch001-2025.07.17-18.57.28.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:07 /var/backups/cloch001-2025.07.17-19.07.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:08 /var/backups/cloch001-2025.07.17-19.08.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:09 /var/backups/cloch001-2025.07.17-19.09.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:10 /var/backups/cloch001-2025.07.17-19.10.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:11 /var/backups/cloch001-2025.07.17-19.11.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:12 /var/backups/cloch001-2025.07.17-19.12.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:13 /var/backups/cloch001-2025.07.17-19.13.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:14 /var/backups/cloch001-2025.07.17-19.14.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:15 /var/backups/cloch001-2025.07.17-19.15.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:16 /var/backups/cloch001-2025.07.17-19.16.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:17 /var/backups/cloch001-2025.07.17-19.17.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:18 /var/backups/cloch001-2025.07.17-19.18.01.tar.gz
-rw-r--r-- 1 root root  11K Jul 17 19:19 /var/backups/cloch001-2025.07.17-19.19.01.tar.gz
```

4. **(10 Points)** Cancel the crontab jobs.



```
┌──(carl-lochstampfor㉿kali)-[/var/backups]
└─$ sudo crontab -r

┌──(carl-lochstampfor㉿kali)-[/var/backups]
└─$ crontab -l
no crontab for carl-lochstampfor

┌──(carl-lochstampfor㉿kali)-[/var/backups]
└─$ sudo crontab -l
no crontab for root
```

TASK B: SYSTEM CLEANUP (EXTRA CREDIT) [20 Points]

Scenario: In the above scenario, your system disk will be filled up eventually without cleaning up the old backups. Therefore, in this optional task, create a script that checks the number of backups you created in Task A. If the number of the backup file is more than a pre-defined threshold, the script will delete the old archives to maintain the backups under a reasonable size.

This script should do the following:

1. Count the number of backups created in Task A and determine if this number is larger than 3.

2. Nothing should happen if the number of backups is less than the threshold, 3.

3. If more backup archives are detected, calculate the number of backups to delete. Then delete the old archives.

Note: As the script needs to write contents in the "/var/backups" folder, which is owned by root, you should consider the permission issue properly. (Using **sudo** to create crontab file)

Reference: How to Format Date for Display or Use In a Shell Script:
https://www.cyberciti.biz/faq/linux-unix-formatting-dates-for-display/

Reference: How to append date timestamp to filename:

https://crunchify.com/shell-script-append-timestamp-to-file-name/