Jasmyn Wilhelm 21 November 2024

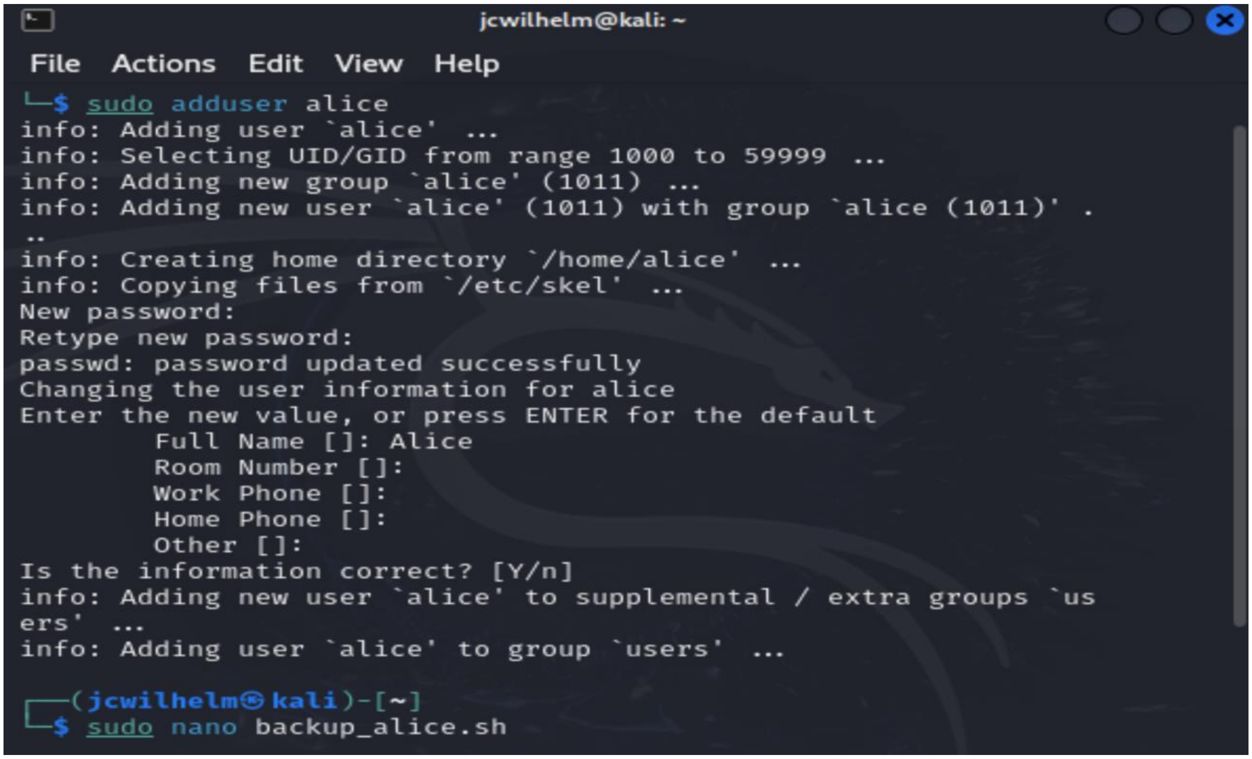
CYSE 270: Linux System for Cybersecurity

Assignment-9

Task A - Backup your system (Using crontab)

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

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

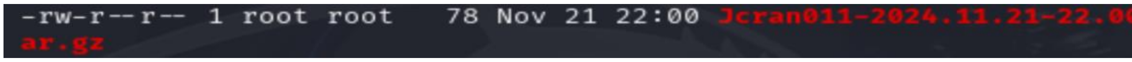


```
└─$ sudo adduser alice
info: Adding user `alice' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `alice' (1011) ...
info: Adding new user `alice' (1011) with group `alice (1011)' .
..
info: Creating home directory `/home/alice' ...
info: Copying files from `/etc/skel' ...
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]
info: Adding new user `alice' to supplemental / extra groups `us
ers' ...
info: Adding user `alice' to group `users' ...

┌──(jcwilhelm㉿kali)-[~]
└─$ sudo nano backup_alice.sh
```

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

   ❖ Do the following:

   - Take **2 inputs** with their values- your **MIDAS** name and **current date (for example, midas=Mohammed)**.
   - 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**).

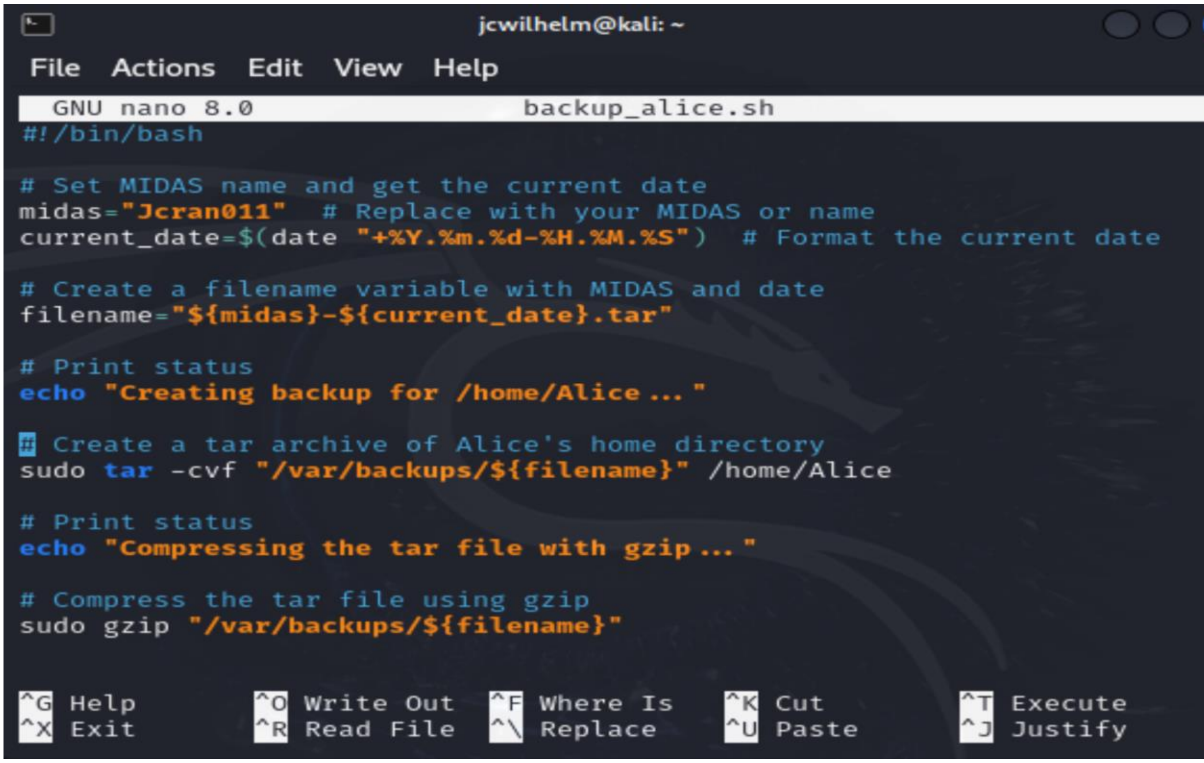


```
-rw-r--r-- 1 root root   78 Nov 21 22:00 Jcran011-2024.11.21-22.0
ar.gz
```

- 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)

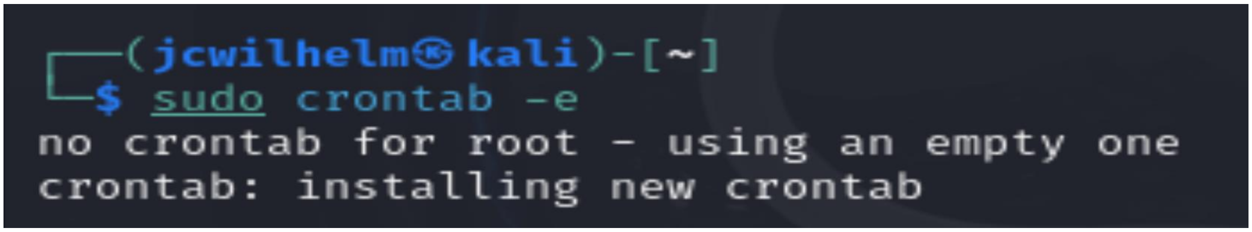
❖ Move the tape archive file/tar file (created in step 2-iii) to /var/backups/ directory using correct command in linux.

❖ 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.

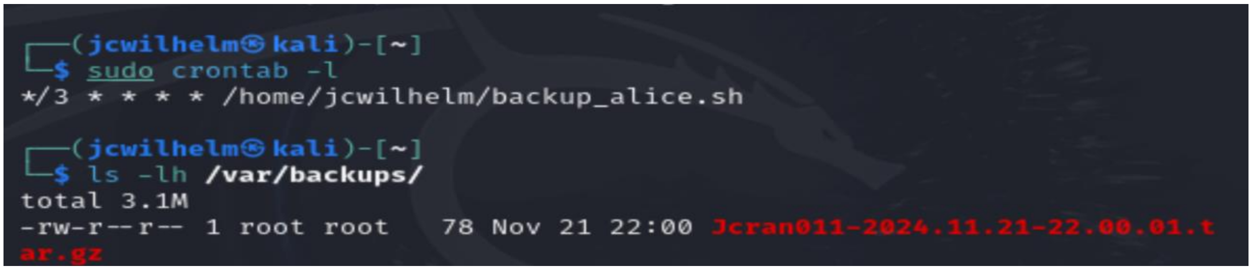


```
  GNU nano 8.0                  backup_alice.sh
#!/bin/bash

# Set MIDAS name and get the current date
midas="Jcran011"  # Replace with your MIDAS or name
current_date=$(date "+%Y.%m.%d-%H.%M.%S")  # Format the current date

# Create a filename variable with MIDAS and date
filename="${midas}-${current_date}.tar"

# Print status
echo "Creating backup for /home/Alice ..."

# Create a tar archive of Alice's home directory
sudo tar -cvf "/var/backups/${filename}" /home/Alice

# Print status
echo "Compressing the tar file with gzip ..."

# Compress the tar file using gzip
sudo gzip "/var/backups/${filename}"


^G Help      ^O Write Out  ^F Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify
```

3. **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:



```
┌──(jcwilhelm㉿kali)-[~]
└─$ sudo crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
```



```
┌──(jcwilhelm㉿kali)-[~]
└─$ sudo crontab -l
*/3 * * * * /home/jcwilhelm/backup_alice.sh

┌──(jcwilhelm㉿kali)-[~]
└─$ ls -lh /var/backups/
total 3.1M
-rw-r--r-- 1 root root    78 Nov 21 22:00 Jcran011-2024.11.21-22.00.01.t
ar.gz
```

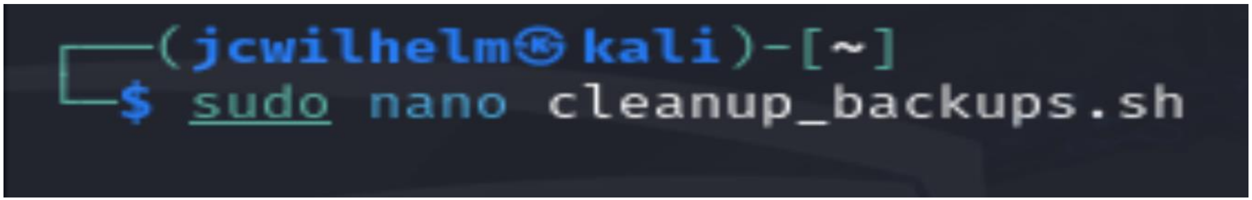
4. Cancel the crontab jobs.

- ❖ The command in the crontab file was deleted therefore canceling the job.

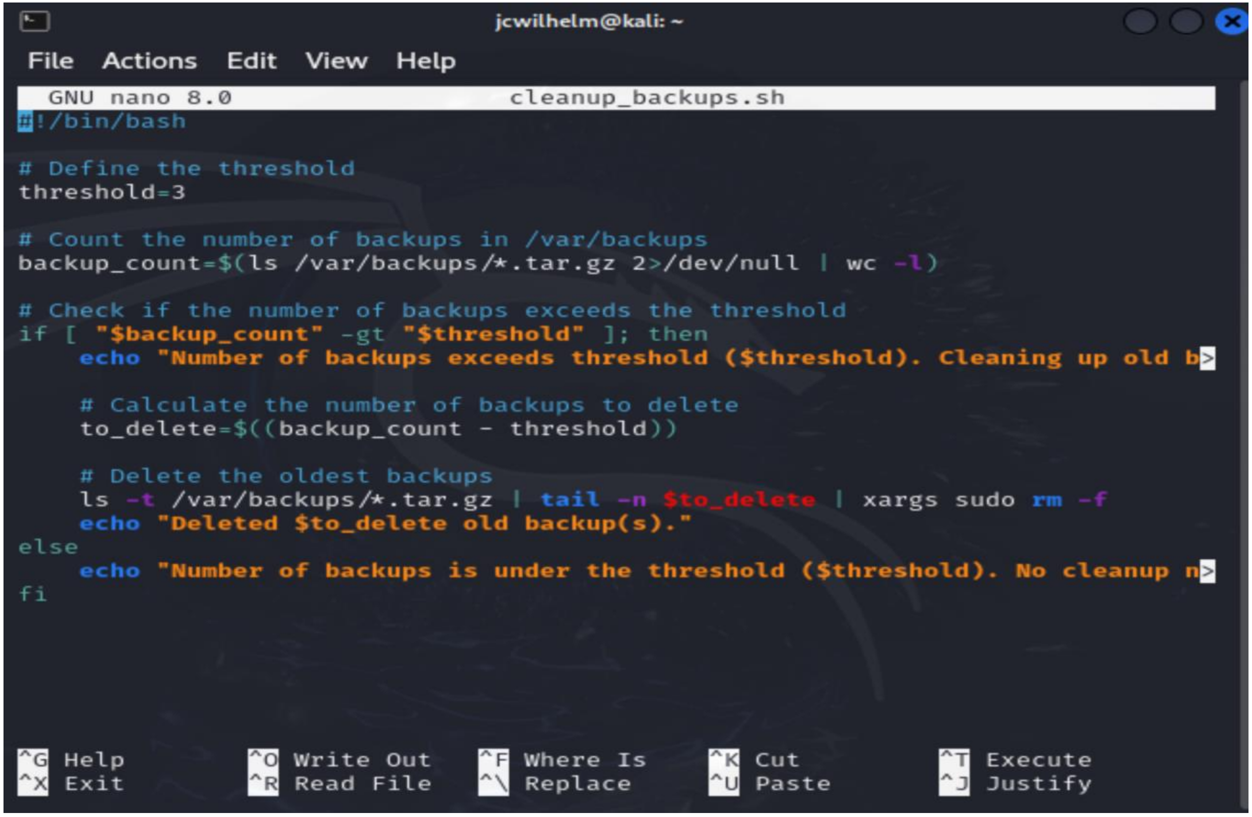
TASK B: SYSTEM CLEANUP (EXTRA CREDIT)

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:



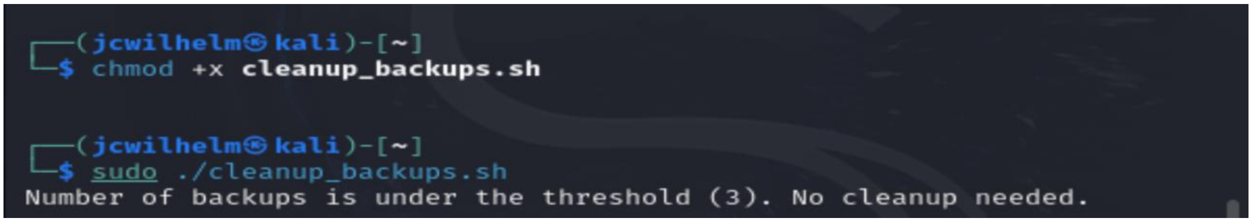
```
┌──(jcwilhelm㉿kali)-[~]
└─$ sudo nano cleanup_backups.sh
```



```
  GNU nano 8.0                    cleanup_backups.sh
#!/bin/bash

# Define the threshold
threshold=3

# Count the number of backups in /var/backups
backup_count=$(ls /var/backups/*.tar.gz 2>/dev/null | wc -l)

# Check if the number of backups exceeds the threshold
if [ "$backup_count" -gt "$threshold" ]; then
    echo "Number of backups exceeds threshold ($threshold). Cleaning up old b>

    # Calculate the number of backups to delete
    to_delete=$((backup_count - threshold))

    # Delete the oldest backups
    ls -t /var/backups/*.tar.gz | tail -n $to_delete | xargs sudo rm -f
    echo "Deleted $to_delete old backup(s)."
else
    echo "Number of backups is under the threshold ($threshold). No cleanup n>
fi




^G Help        ^O Write Out   ^F Where Is    ^K Cut         ^T Execute
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify
```

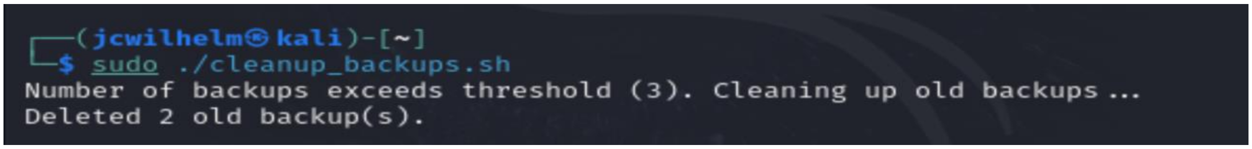
1. Count the number of backups created in Task A and determine if this number is larger than 3.

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



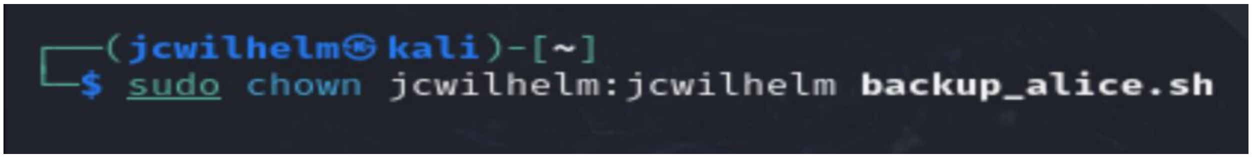
```
┌──(jcwilhelm㉿kali)-[~]
└─$ chmod +x cleanup_backups.sh

┌──(jcwilhelm㉿kali)-[~]
└─$ sudo ./cleanup_backups.sh
Number of backups is under the threshold (3). No cleanup needed.
```

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

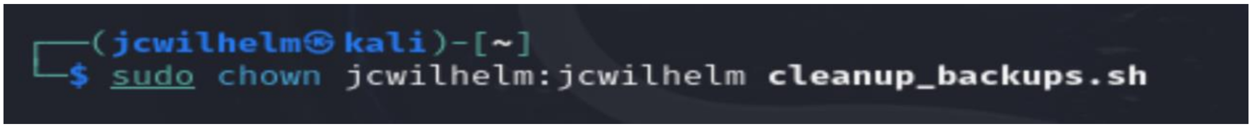


```
┌──(jcwilhelm㉿kali)-[~]
└─$ sudo ./cleanup_backups.sh
Number of backups exceeds threshold (3). Cleaning up old backups ...
Deleted 2 old backup(s).
```

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)



```
┌──(jcwilhelm㉿kali)-[~]
└─$ sudo chown jcwilhelm:jcwilhelm backup_alice.sh
```



```
┌──(jcwilhelm㉿kali)-[~]
└─$ sudo chown jcwilhelm:jcwilhelm cleanup_backups.sh
```

- ❖ The permissions stated that access was denied, but using this command allowed me to change the file's owner by repeating my username to allow access to execute the .sh files.