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).

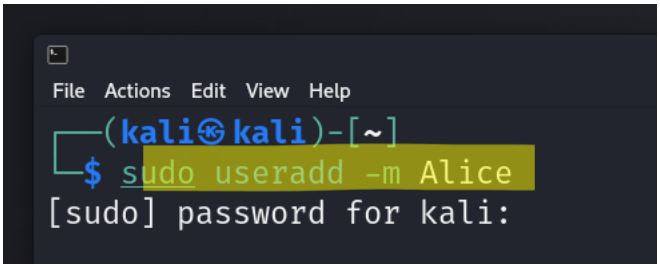
   Explanation: Added Alice with command **sudo useradd -m** Alice and set the password to kali. The -m option will create a home directory for the user.

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

   a. 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). Using tar command, create a tape archive for Alice's home directory (/home/Alice) and the filename created above (in step-2-ii).
   b. Move the tape archive file/tar file (created in step 2-iii) to /var/backups/ directory using correct command in linux.
   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.

   Explanation: Added Alice with command **sudo useradd -m** Alice and set the password to kali. The -m option will create a home directory for the user. Then I entered into nano from bash shell with command **nano** followed by my file name, which in this case is **backup.sh.** I code the requirements stated in *a. b. c.* and then exit nano **by pressing ctrl + x** and then saving the changes when prompted by pressing **y** and **enter** to confirm my current filename. Important commands to note is **tar -cvf** , **sudo gzip,** and the syntax involving the date and midas variables. Screenshots are continued on the next page.

   Screenshot:

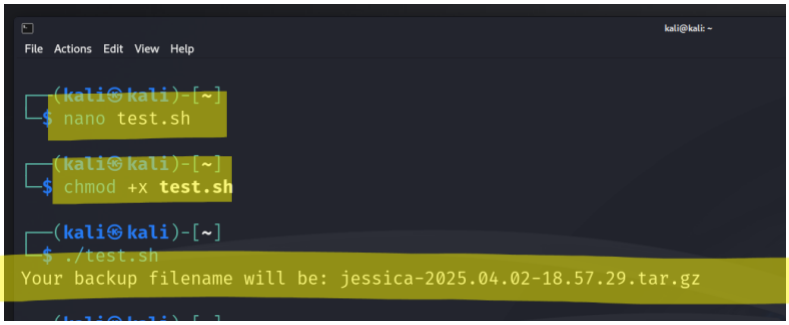


```
File  Actions  Edit  View  Help
  GNU nano 8.2
#!/bin/bash

# Step 1: Inputs
midas="jessica"
now=$(date +"%Y.%m.%d-%H.%M.%S")

# Step 2: Create filename
filename="${midas}-${now}.tar"

# Step 3: Archive Alice's home
tar -cvf "$filename" /home/Alice

# Step 4: Move archive to /var/backups
sudo mv "$filename" /var/backups/

# Step 5: Compress the archive
sudo gzip /var/backups/"$filename"
```

Explanation Continued: I then make my code executable eith command **chmod +x** followed by my file, **backup.sh** and run the script to check with command **sudo /backup.sh**. I then enter into nano again to code using **sudo nano test.sh**. I am checking the above code for output accuracy described in the lab with the new 'test' file. I make it executable with command **chmod +x test.sh** and then run it using command **./test.sh**. Screenshot can be found on the next page, I am specifically looking for my name, date, and .gz to indicate compression.

Screenshots Continued:

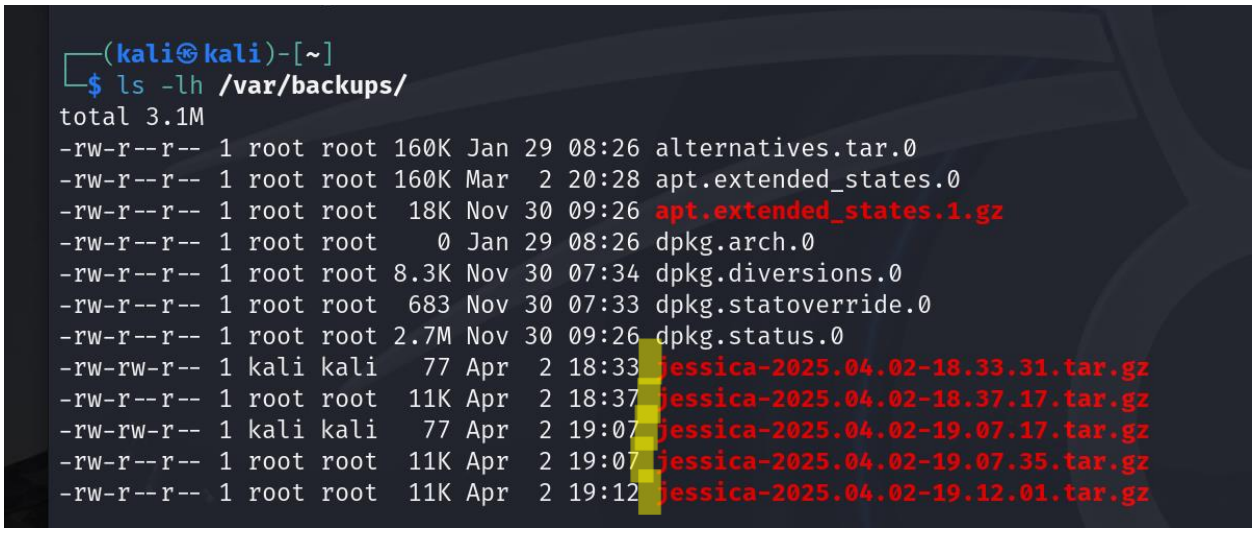
```
┌──(kali㉿kali)-[~]
└─$ sudo ./backup.sh
tar: Removing leading `/' from member names
/home/Alice/
/home/Alice/.zshrc
/home/Alice/.java/
/home/Alice/.java/.userPrefs/
/home/Alice/.java/.userPrefs/burp/
/home/Alice/.java/.userPrefs/burp/prefs.xml
/home/Alice/.zprofile
/home/Alice/.bashrc
/home/Alice/.bashrc.original
/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/.bash_logout
/home/Alice/.face.icon
/home/Alice/.config/
/home/Alice/.config/xfce4/
/home/Alice/.config/xfce4/panel/
/home/Alice/.config/xfce4/panel/genmon-15.rc
/home/Alice/.config/nautilus/
/home/Alice/.config/nautilus/scripts-accels
/home/Alice/.config/cherrytree/
/home/Alice/.config/cherrytree/config.cfg
/home/Alice/.config/powershell/
/home/Alice/.config/powershell/Microsoft.PowerShell_profile.ps1
/home/Alice/.profile
/home/Alice/.face
```



3. Create a crontab file to keep the scheduled task running for 3 minutes, then check the contents in the /var/backups directory.

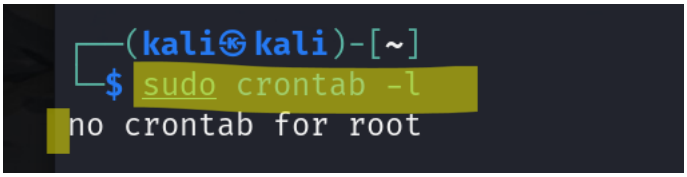
   Explanation: I enter into crontab with command **crontab -e** and I use **-e** to have full permissions. I pick option 1, which is nano. I code ***/3 * * * * /home/kali/backup.sh** to create a script that will run every 3 minutes. To exit nano I press **ctrl x** , **y** to confirm changes, and **enter** to confirm name. I need to check if If all my steps were successful so I use the command **ls -lh** followed by where I need to check, **/var/backups**. I check my output and I have successfully completed the objective.

   Screenshot:



4. Cancel the crontab jobs.
   Explanation: I use command **sudo crontab -r** to remove all root's scheduled cron jobs and check with command **sudo crontab -l** . I am successful.

   Screenshot:



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:
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)

Explanation: I enter into nano with command **nano** followed by the file name of the script **cleanup.sh** , I code the requirements for the extra credit outlined above, paying close attention to my if else statement and threshold. I also tailor output for user's because I want it to be easy to read. I keep it concise and simple. I make the command executable and run the script twice using commands **chmod +x** and **sudo ./** . I am looking for two different outputs, one where deletion happened and one where there were fewer than 3. Both outputs happened as expected according to my code.

Screenshots: