Erik Sorto

CYSE 270: Linux System for Cybersecurity

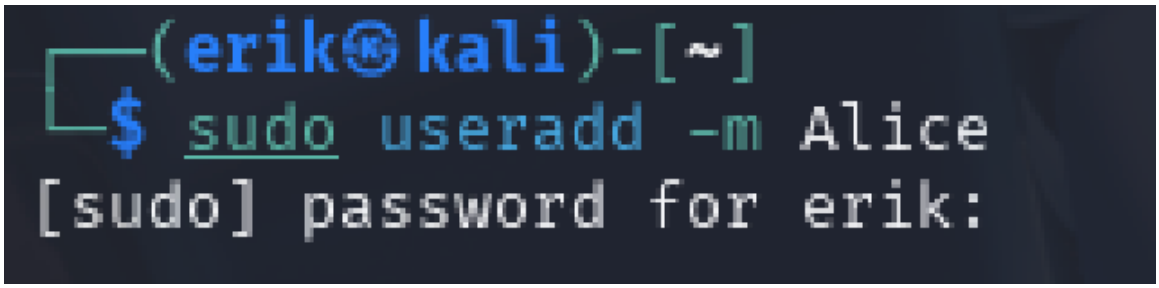
Lab 9: Task Automation

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**).



2. **(50 Points) 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**).

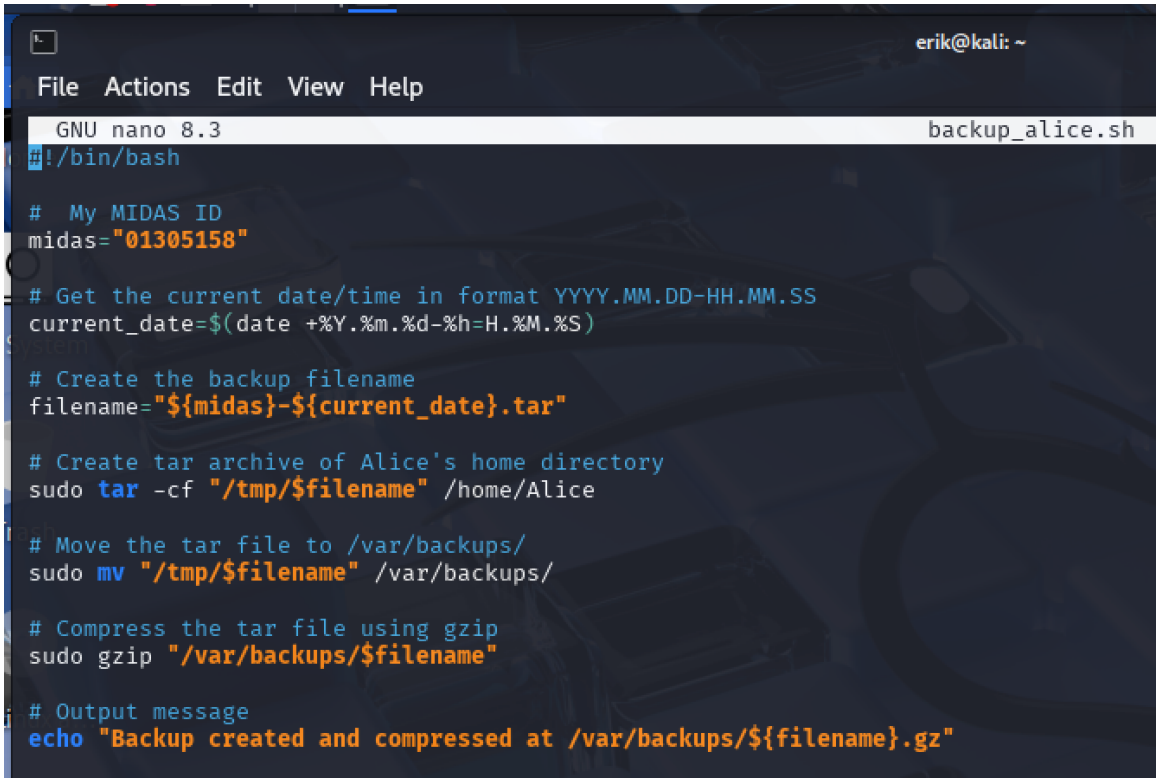
• Using **tar** command, create a tape archive for Alice's home directory

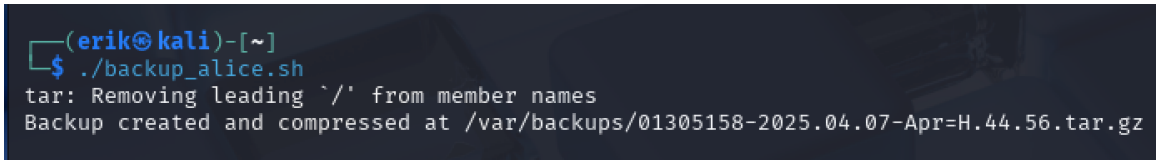
(/home/Alice) and the **filename** created above (in step-2-ii). (Please

learn about tar command in Linux for its usage)

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.





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
```

```
┌──(erik㉿kali)-[~]
└─$ sudo crontab -e
no crontab for root - using an empty one
Select an editor.  To change later, run select-editor again.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
crontab: installing new crontab
```

```
erik@kali: ~
File  Actions  Edit  View  Help
  GNU nano 8.3                                  /tmp/crontab.NjM
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/3 * * * * /home/erik/backup_alice.sh
```

```
┌──(erik㉿kali)-[~]
└─$ ls -l /var/backups
total 3056
-rw-rw-r-- 1 erik erik      81 Apr  7 21:36 '01305158-2025.04.07-Apr=H.36.26.tar.gz'
-rw-r--r-- 1 root root   10459 Apr  7 21:44 '01305158-2025.04.07-Apr=H.44.56.tar.gz'
-rw-r--r-- 1 root root   10459 Apr  7 21:57 '01305158-2025.04.07-Apr=H.57.01.tar.gz'
-rw-r--r-- 1 root root  174080 Apr  6 00:00  alternatives.tar.0
-rw-r--r-- 1 root root  161879 Apr  5 23:50  apt.extended_states.0
-rw-r--r-- 1 root root       0 Apr  6 00:00  dpkg.arch.0
-rw-r--r-- 1 root root    8218 Apr  5 23:51  dpkg.diversions.0
-rw-r--r-- 1 root root     683 Apr  5 23:49  dpkg.statoverride.0
-rw-r--r-- 1 root root 2740623 Apr  5 23:50  dpkg.status.0
```

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

```
erik@kali: ~
File  Actions  Edit  View  Help
  GNU nano 8.3                                  /tmp/crontab.60YOlV/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
```

just deleting the last line cancels the jobs.

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.

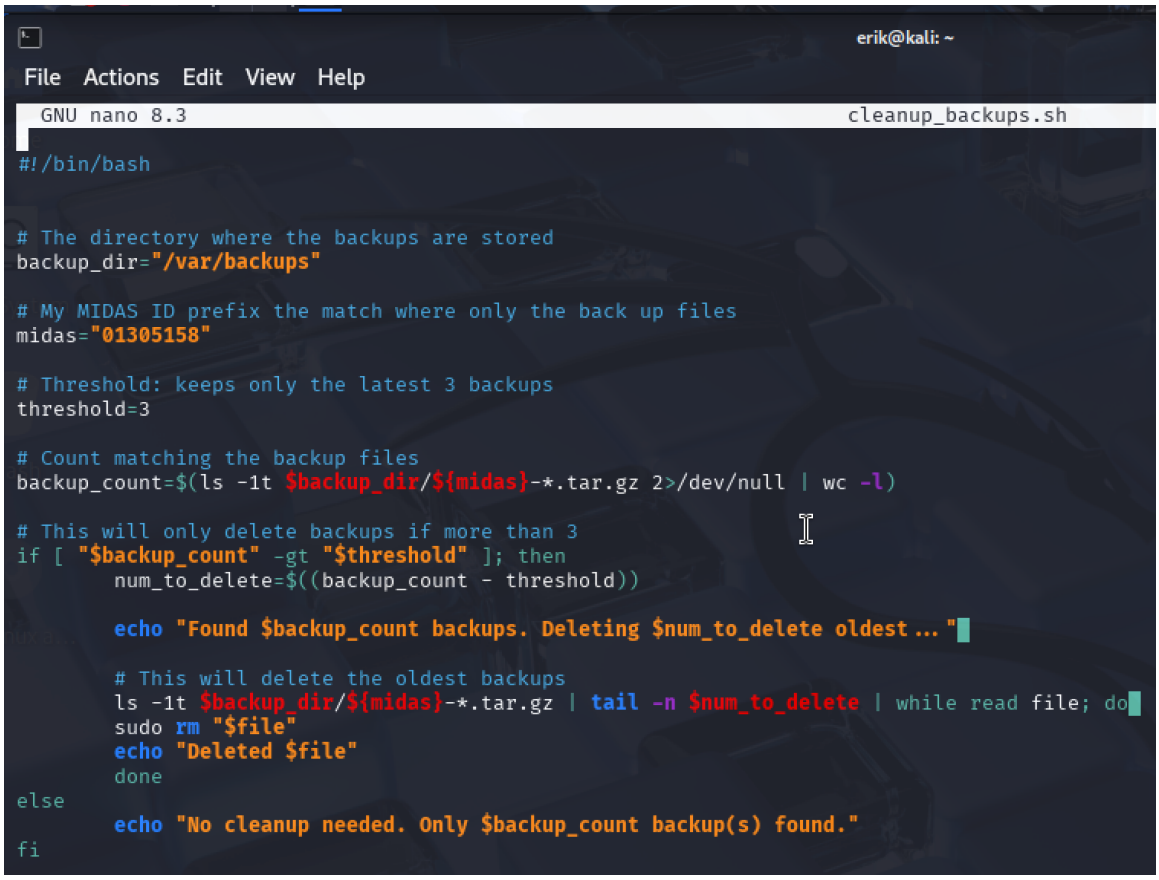
2. Nothing should happen if the number of backups is less than the threshold, 3.

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

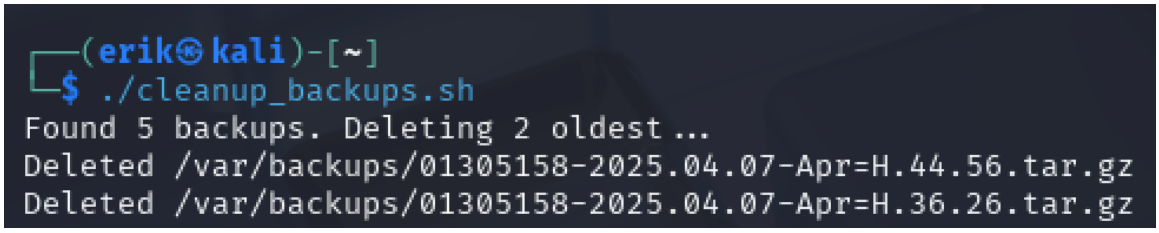
delete the old archives.

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/