CYSE 270 Lab Assignment 9

Derek D. Hillman
UIN:00861068

Task A

Step 1.) I created the user **Alice** with a home directory.

```
┌──(derek㉿kali)-[~]
└─$ sudo useradd -m -d /home/Alice -s /bin/bash Alice
sudo passwd Alice

New password:
Retype new password:
passwd: password updated successfully
```

Verified its privileges.

```
┌──(derek㉿kali)-[~]
└─$ id Alice
ls -ld /home/Alice

uid=1010(Alice) gid=1011(Alice) groups=1011(Alice)
drwx------ 5 Alice Alice 4096 Nov  9 07:57 /home/Alice
```

Step 2.) I created the file utilizing **vim.**

```
File  Actions  Edit  View  Help
┌──(derek㉿kali)-[~]
└─$ sudo vim /usr/local/bin/backup_alice.sh


┌──(derek㉿kali)-[~]
└─$ sudo chmod +x /usr/local/bin/backup_alice.sh
```

I gave the file executable privileges by using **chmod + x**

```
File  Actions  Edit  View  Help
┌──(derek㉿kali)-[~]
└─$ sudo vim /usr/local/bin/backup_alice.sh


┌──(derek㉿kali)-[~]
└─$ sudo chmod +x /usr/local/bin/backup_alice.sh
```

I executed the script to verify that it is working properly, and that the backup saved in the correct location

CYSE 270 Lab Assignment 9

Derek D. Hillman
UIN:00861068

Below are the contents of the script.

```
┌──(derek㉿kali)-[~]
└─$ cat /usr/local/bin/backup_alice.sh
#!/bin/bash
set -euo pipefail

# Require root so we can write to /var/backups without sudo inside the script
if [[ $EUID -ne 0 ]]; then
  echo "ERROR: Run this script with sudo or as root." >&2
  exit 1
fi

# ——— a) Two inputs: MIDAS and current date (you must enter both) ———
read -p "Enter your MIDAS (e.g., dhill036): " midas
read -p "Enter current date value (e.g., 2025.11.09-14.05.00): " datestr

if [[ -z "${midas}" || -z "${datestr}" ]]; then
  echo "ERROR: Both MIDAS and date are required." >&2
  exit 1
fi

# a) filename = MIDAS-date.tar  (we'll compress to .tar.gz in /var/backups)
filename="${midas}-${datestr}.tar"

src="/home/Alice"
dest="/var/backups"
tmp="/tmp/${filename}"

# sanity checks
if [[ ! -d "${src}" ]]; then
  echo "ERROR: ${src} does not exist. Create the user/home first." >&2
  exit 1
fi

mkdir -p "${dest}"
```

CYSE 270 Lab Assignment 9

Derek D. Hillman
UIN:00861068

```
mkdir -p "${dest}"

# a) Create tar of /home/Alice with the chosen filename
tar -cf "${tmp}" -C "/" "home/Alice"

# b) Move tar to /var/backups
mv "${tmp}" "${dest}/"

# c) Compress the tar in /var/backups (choose one algo – using gzip)
gzip -f "${dest}/${filename}"     # results in ${filename}.gz

echo "Created: ${dest}/${filename}.gz"
```

Step 3.) I created the cronjob to back up the system and verified that it worked.

```
#
# m h  dom mon dow   command
* * * * * echo -e "dhill036\n$(date +\%Y.\%m.\%d-\%H.\%M.\%S)" | /usr/local/bin/ba>
```

```
-rw-r--r-- 1 root root  11K Nov  9 08:30 '-e dhill036-2025.11.09-08.30.01.tar.gz'
-rw-r--r-- 1 root root  11K Nov  9 08:31 '-e dhill036-2025.11.09-08.31.01.tar.gz'
-rw-r--r-- 1 root root  11K Nov  9 08:32 '-e dhill036-2025.11.09-08.32.01.tar.gz'
```

Step 4.) I deleted the cron job and verified its deletion.

```
┌──(derek㉿kali)-[~]
└─$ sudo crontab -l

# 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


┌──(derek㉿kali)-[~]
```

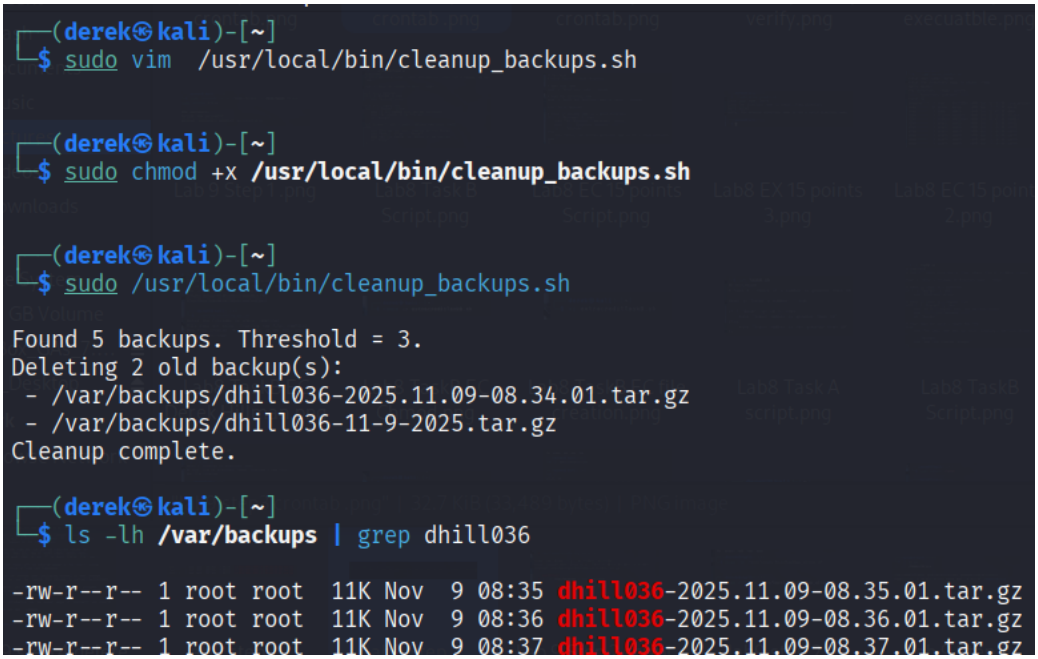
CYSE 270 Lab Assignment 9

Derek D. Hillman
UIN:00861068

Task B

Extra Credit.

I created the backup script utilizing **vim**, executed **chmod +x** to make the file executable, executed the script and verified the contents of the **/var/backups** folder.



Below are the contents of the clean-up script.