

**CYBR371 – Lab One**  
**Thomas Green – 300536064**

**Part 1 – Linux Account Management**

**Q1.1 What is the octal or string representation of the following permissions?**

- $\text{rwxrw-r-t} = 1765$
- $\text{r-S-wx-x} = 400\ 400\ 021\ 001 = 4431$
- $\text{rwxr-xr--} = 000\ 421\ 401\ 400 = 0754$
- $\text{r-Sr-sr-x} = 420\ 400\ 401\ 401 = 6455$
- $432 = 000\ 400\ 021\ 020 = \text{r---wx-w-}$
- $3532 = 021\ 401\ 021\ 020 = \text{r-x-ws-wT}$
- $6713 = 420\ 421\ 001\ 021 = \text{rws--s-wx}$
- $1530 = 001\ 401\ 021\ 000 = \text{r-x-wx-T}$

**Q1.2 If the unmask value for a user is 035, what are the default file and directory permissions set for the user? Write the permissions and how they were calculated.**

Unmask value subtracted from maximum permissions 777 for directories to determine default permissions. Mask is 035 so  $777-035=742$

- Directory = 777
- Mask = 035
- Equals = 742
- Converted =  $421\ 400\ 020 = \text{rwx r-- -w-}$

Unmask value subtracted from maximum permissions 666 for files to determine default permissions. Mask is 035 so  $666-035=631$

- File = 666
- Mask = 035
- Equals = 631
- Converted =  $420\ 021\ 001 = \text{rw- -wx -x}$

**Q1.3 If the default permissions give to directories that the user xyz creates are  $\text{rwxr-x--}$ , what are the default permissions set for the files created by the user? Write the permissions and how they were calculated.**

Default permission for directories is 777. Calculate unmask value from permissions.

Default permission: 777

$\text{rwxr-x---} = 421\ 401\ 000 = 750$

Unmask value =  $777-750= 27$

Default permission for directory is 666. Calculate directory permission

Default permission for directory:  $666-027 = 640 = \text{rw-r----x}$

**Q1.4 Find all the executables that have SUID set. Hint: you can use the find command with the right parameters**

**a. Provide the command that you used to find these programmes**

```
find / -type f -perm /u=s
```

**b. List all the programmes you found. For 3 of them, provide a brief description why it needs to be a SUID programme**

/usr/bin/sudo:

Elevated Privileges: 'sudo' allows a permitted user to execute a command as the superuser or another user, as specified by the security policy. Since many administrative tasks require elevated privileges, 'sudo' needs to execute with the permissions of the superuser (root) to perform these tasks/

/usr/bin/passwd:

Updating Password File: The 'passwd' command is used to change user passwords. It needs to update the system's password file which is typically owned by the superuser. The SUID bit allows non-privileged users to run 'passwd' and update their password entry in the password file without needing write permissions to the file.

/usr/bin/newgrp:

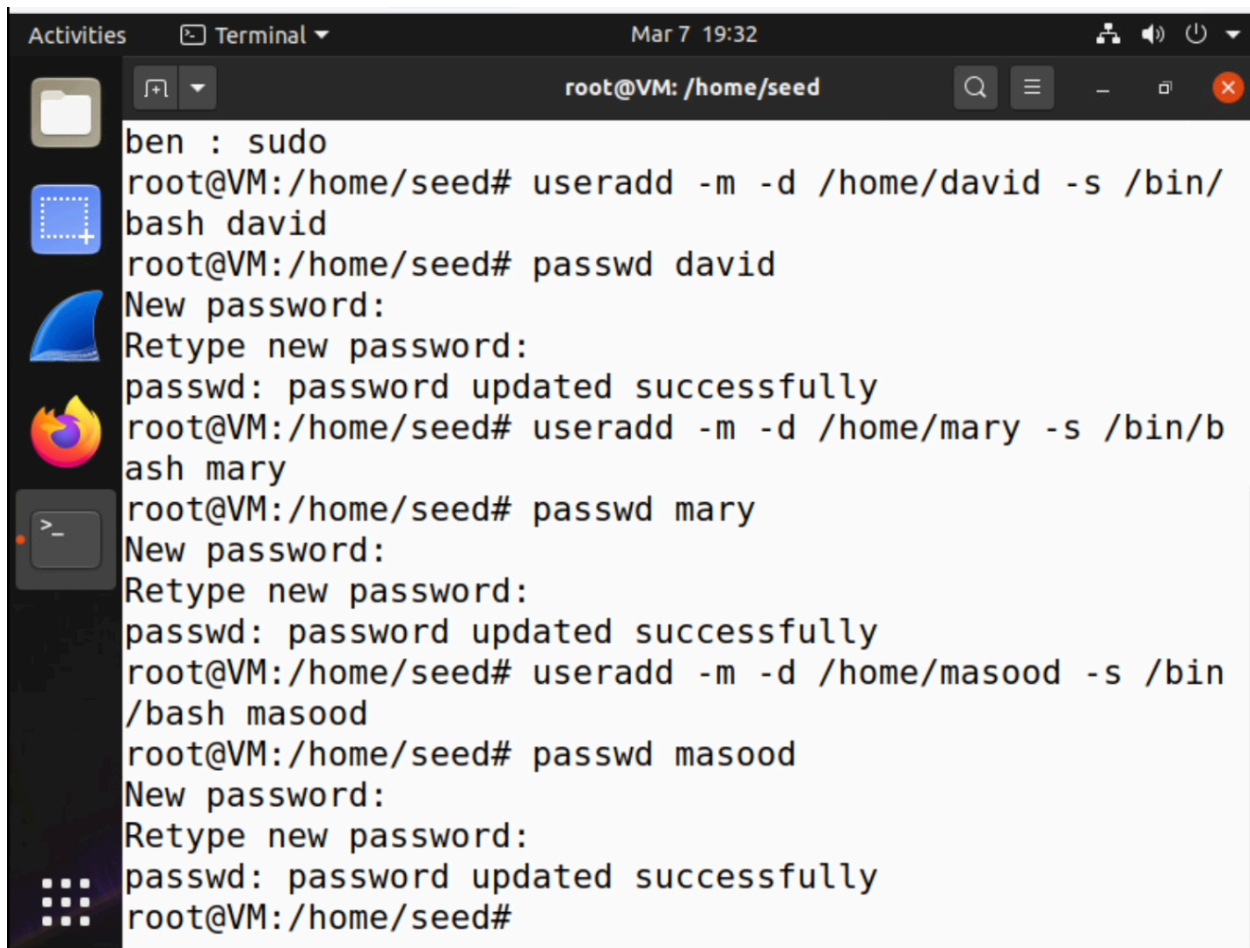
Changing Group Affiliation: The 'newgrp' command is used to change a user's effective group ID. Without the SUID bit, a user could not change their group affiliation to a group they are not a member of. The SUID bit allows the 'newgrp' command to change the group ID of the user, facilitating collaboration in shared directories where group membership is required.

## **2 Part 2 – Linux Access Control List (ACL)**

**Complete the lab "Linux Access Control List" and answer the questions highlighted in this document in order they appear in the lab document. Please note that the questions below are dependent on the sequence of the lab instructions and must be followed and answered step by step as they appear in the Linux Access Control List lab document.**

**Q2.1 Write the command(s) you used to add the users with their associated provided information**

```
root@VM:/home/seed# useradd -g sudo -d /home/cybr371 -m -s /bin/bash cybr371
root@VM:/home/seed# passwd cybr371
New password:
Retype new password:
passwd: password updated successfully
root@VM:/home/seed# useradd -g sudo -d /home/ben -m -s /bin/bash ben
root@VM:/home/seed# passwd ben
New password:
Retype new password:
passwd: password updated successfully
root@VM:/home/seed#
```



The screenshot shows a terminal window titled "Terminal" with the date and time "Mar 7 19:32". The terminal prompt is "root@VM: /home/seed". The terminal output shows the following commands and responses:

```
ben : sudo
root@VM:/home/seed# useradd -m -d /home/david -s /bin/bash david
root@VM:/home/seed# passwd david
New password:
Retype new password:
passwd: password updated successfully
root@VM:/home/seed# useradd -m -d /home/mary -s /bin/bash mary
root@VM:/home/seed# passwd mary
New password:
Retype new password:
passwd: password updated successfully
root@VM:/home/seed# useradd -m -d /home/masood -s /bin/bash masood
root@VM:/home/seed# passwd masood
New password:
Retype new password:
passwd: password updated successfully
root@VM:/home/seed#
```

**Q2.2** Login as user “ben” and write a command to append the line “This line is from ben” to myfile.txt file in the cybr371’s home directory (use absolute path). Write the command you used to append the line and explain the output (i.e. did you manage to append the line? Explain why the command was successful and/or why it failed)

```
ben@VM:~$ id
uid=1002(ben) gid=27(sudo) groups=27(sudo)
ben@VM:~$ su - cybr371
Password:
To run a command as administrator (user "root"), use "
sudo <command>".
See "man sudo_root" for details.

cybr371@VM:~$ id
uid=1001(cybr371) gid=27(sudo) groups=27(sudo)
cybr371@VM:~$ getfacl myfile.txt
# file: myfile.txt
# owner: cybr371
# group: sudo
user::rw-
group::rw-
other::r--
```

```
ben@VM:~$ echo "This line is from the user ben" >> /home/cybr371/myfile.txt
Terminal:~$ su - cybr371
Password:
To run a command as administrator (user "root"), use "
sudo <command>".
See "man sudo_root" for details.

cybr371@VM:~$ cat myfile.txt
This file was created by user cybr371
This line is from the user ben
cybr371@VM:~$ █
```

As we can see, Ben and Cybr371 share the same group (gid=27(sudo)) however ben does not own the same file myfile.txt. However we can see using the command getfacl myfile.txt

that myfile.txt has rw group permissions. So ben can both read and write to it as seen in the second image.

**Q2.3 Login as user “david” now and write a command to add a line “This is from david” to myfile.txt file in cybr371’s home directory. (Write the command and explain why the operation is either successful or not).**

```
ben@VM:~$ echo "This line is from the user ben" >> /home/cybr371/myfile.txt
ben@VM:~$ su - cybr371
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

cybr371@VM:~$ cat myfile.txt
This file was created by user cybr371
This line is from the user ben
cybr371@VM:~$ su - david
Password:
david@VM:~$ id
uid=1003(david) gid=1003(david) groups=1003(david)
david@VM:~$ echo "This line is from user david" >> /home/cybr371/myfile.txt
-bash: /home/cybr371/myfile.txt: Permission denied
david@VM:~$
```

As we can see David is not in the sudo group like cybr371 or David so he will have only read permissions for myfile.txt as it only has read permission for other users. This means that he will have been denied when writing to the file as seen in the image above.

**Q2.4 Login as the user masood and issue a command to read the content of the file myfile.txt in the cybr371's home directory. Can the user masood read the file? Write the commands and explain the output of the command**

```
masood@VM:~$ id
uid=1005(masood) gid=1005(masood) groups=1005(masood)
masood@VM:~$ id cybr371
uid=1001(cybr371) gid=27(sudo) groups=27(sudo)
masood@VM:~$ cat /home/cybr371/myfile.txt
This file was created by user cybr371
This line is from the user ben
This line is from user david
masood@VM:~$ getfacl /home/cybr371/myfile.txt
getfacl: Removing leading '/' from absolute path names
# file: home/cybr371/myfile.txt
# owner: cybr371
# group: sudo
user::rw-
user:david:rwx
group::rw-
mask::rwx
other::r--
```

As Masood is not in the group sudo like cybr371 or ben and is not the owner of myfile.txt he has the permissions as an other so he can view the file however cannot write to myfile.txt or execute.

Q2.5 after completion of step 12, Write a command to set an ACL to deny all access (read, write and execute) to myfile.txt for user David

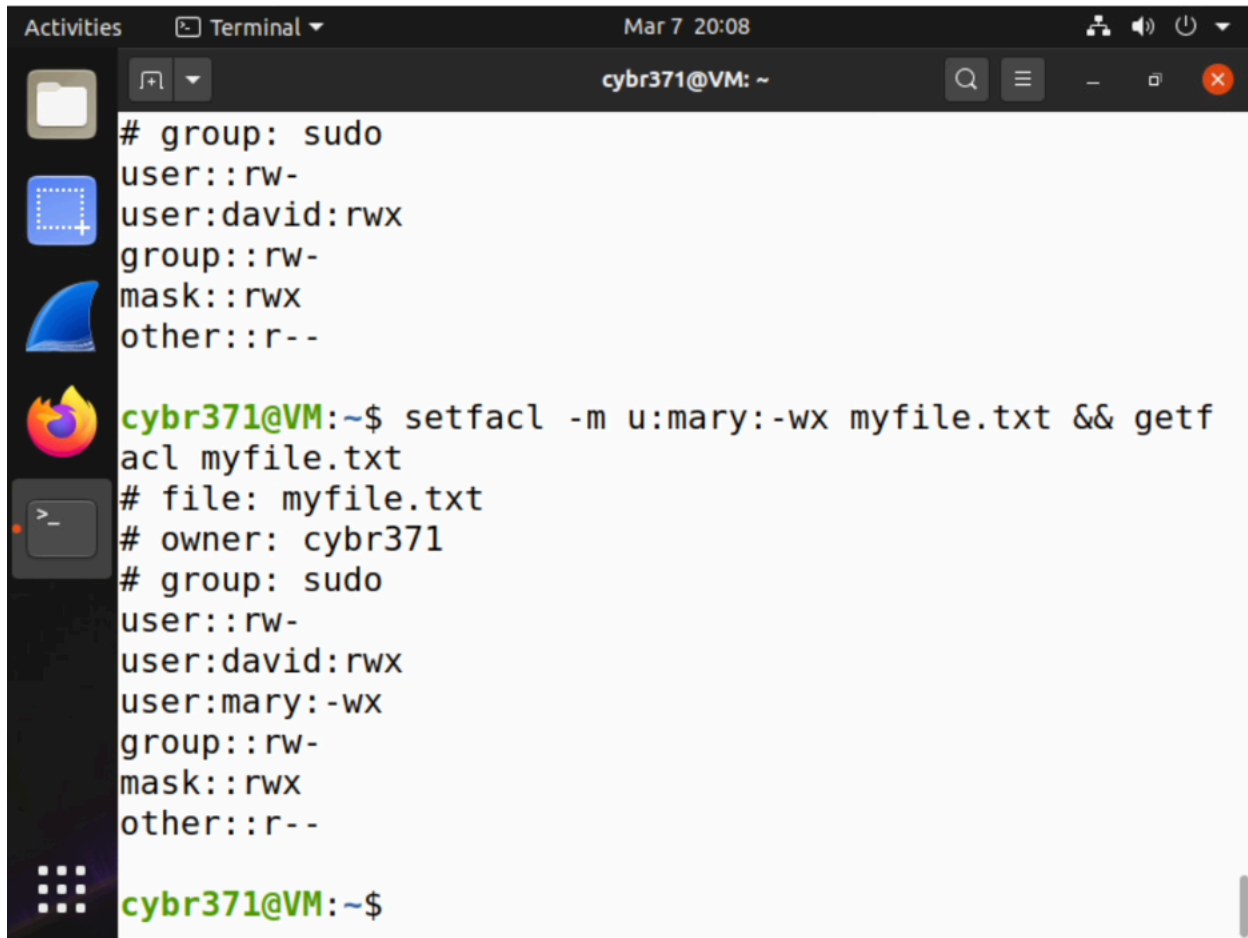
```
Terminal ▾ Mar 7 19:57
cybr371@VM: ~
# owner: cybr371
# group: sudo
user::rw-
user:david:rwx
group::rw-
mask::rwx
other::r--

cybr371@VM:~$ setfacl -m u:david:--- myfile.txt &&getf
acl myfile.txt
# file: myfile.txt
# owner: cybr371
# group: sudo
user::rw-
user:david:---
group::rw-
mask::rw-
other::r--

cybr371@VM:~$
```



**Q2.6 Write a command to create an ACL entry for user mary with write and execute permissions only on the file myfile.txt**

A terminal window titled 'cybr371@VM: ~' with a dark theme. The window shows the output of the 'ls -la' command, which lists permissions for 'myfile.txt'. The user then runs the command 'setfacl -m u:mary:-wx myfile.txt && getfacl myfile.txt'. The output shows the updated ACL for 'myfile.txt', including the new entry for user 'mary' with write and execute permissions. The terminal window has a sidebar with icons for Activities, Terminal, and a file manager. The top bar shows the date 'Mar 7 20:08' and system icons for network, volume, and power.

```
Activities Terminal Mar 7 20:08 cybr371@VM: ~  
# group: sudo  
user::rw-  
user:david:rwx  
group::rw-  
mask::rwx  
other::r--  
cybr371@VM:~$ setfacl -m u:mary:-wx myfile.txt && getfacl  
acl myfile.txt  
# file: myfile.txt  
# owner: cybr371  
# group: sudo  
user::rw-  
user:david:rwx  
user:mary:-wx  
group::rw-  
mask::rwx  
other::r--  
cybr371@VM:~$
```