**PROBLEM 2**

1. Copied the passwd file to desktop and then executed the passwd file as shown in the output screenshot.

Conclusion: unable to change the password using passwd. Hence we do not have the root privileges.

A screenshot of a computer

Description automatically generated with medium confidence

These commands need to be Set-UID programs because, we will need permission to change the password or even access a few files when necessary.

The user A has execute permissions on vi but user A doesn’t have permission to read the prog.txt file. And so when vi tries to read prog.txt, an error message is displayed. Now when the Set-UID bit is set on vi, the access to prog.txt is given. What happens within the UNIX system is that it thinks that the root user is accessing the prog.txt file and not user A, and hence access is granted.

In case the passwd command are not setUID programs, then a user (other than the root user) will not be able to change the password or change a user’s login shell attribute, etc.

B1) Login as root, copy /bin/zsh to /tmp and make it a set-root-uid program

And Login as user and execute the zsh.

Refer the image below.

Conclusion can be drawn that from the output below, we can conclude that root privilege is available.

The command whoami clearly states that the user has been changed to root.

A screenshot of a computer

Description automatically generated with medium confidence

B2)

A screenshot of a computer

Description automatically generated with medium confidence

Login as root, copy /bin/bash to /tmp and make it a set-root-uid program And Login as user and execute the bash. Refer the image above.

From the output above, it’s clear that root privilege is not available. The whoami command verifies this conclusion. This is because /bin/bash has certain built-in protection that prevents the abuse of the Set-UID mechanism.

C1) As bad\_ls runs with elevated root permissions, the script named ls will also run with elevated root permissions. So it is a bad idea for root to let normal users run bad\_ls as long as it has SUID privileges, because it would run any program named ls that comes first in the user's PATH.

OS Credential dumping : Adversaries may attempt to dump the contents of /etc/passwd and /etc/shadow to enable offline password cracking. Most modern Linux operating systems use a combination of /etc/passwd and /etc/shadow to store user account information including password hashes in /etc/shadow. By default, /etc/shadow is only readable by the root user.

The Linux utility, unshadow, can be used to combine the two files in a format suited for password cracking utilities such as John the Ripper:

# /usr/bin/unshadow /etc/passwd /etc/shadow > /tmp/crack.password.db

A screenshot of a computer

Description automatically generated with medium confidence

C2)

Graphical user interface, website

Description automatically generated

C3) Text

Description automatically generated