Homework 3

Due Date: By the end of Friday, 3/28/2025. Submit a PDF file on HuskyCT.
Total points: 100

In this homework assignment, we experiment with bash and Linux file access control on file systems, on the virtual machine (VM) for SEED labs. Create a directory, e.g., hw3, in an ext4 file system, for example, in the home directory of user 'seed'. Do NOT use the folder shared with the host OS. It is not mounted as an ext4 file system.

For each task, show and explain the commands you use and the results.

1. (50 points) We experiment with permissions on files in this exercise.
   a. Find out the permissions on /etc/passwd and explain the meaning of the permissions.
   b. Find out the permissions on command mount and explain the meaning of the permissions. You can find the path to the program with which command "which mount".
   c. What is the primary group of user seed? What are the supplementary groups?
   d. Create a new file `a`. Use a text editor or simply redirect output as "echo message > a". Find its permissions and associated user id and group id.
   e. The command chmod can change permissions on files and directories. Use the command to change the permissions of file `a` so that group and other users do not any permissions.
   f. The command chown can change the owner and group associated with a file. Change the group associated with file `a` to `sudo`.
   g. Change file `a`'s group to `ssh`. If you do not run the command with `sudo`, you will see an error message. Why?
   h. Now change file `a`'s group back to `seed`. Do you have to run the command with `sudo`? Why?
   i. The default permissions on files can be controlled by umask in bash. Find out the umask value in your current bash session and explain its meaning.
   j. How would you set the umask so the group and other users do not have any permissions on new files as in d)? Demonstrate that your method works.

Continued on next page.

2. (50 points) In this exercise, we experiment with scripts, more specifically, bash scripts. First, create two files `myls` and `ls` in the directory, with your favorite text editor.

   `myls` has the following three lines. The first line that starts with a hash tag is called shebang, which specifies the program that will interpret the script. The second line shows the username who runs the script and the third line invokes ls command.

   ```
   #!/bin/bash
   whoami
   ls $*
   ```

   `ls` has the following lines.

   ```
   #!/bin/bash
   echo "This is not the real ls."
   ```

   a. Change the permissions of the two files so that everyone can run the scripts and only the owner and the group can write to the files. For example, seed should be able to run `myls` by typing `./myls`. Note that one needs to have the read permission to run a script.

   b. `bash` searches executables in the directories listed in environment variable PATH. The directories are separated by colons. Find out the list of directories in PATH in your bash session.

   c. `bash` allows an environment variable set for specific commands. What happens if your run the following command? Explain the result.

      ```
      PATH=. myls
      ```

   d. How would you change `myls` so that the correct commands are executed even if someone sets PATH as in c)?

   e. Change the owner of `myls` to root and set the SUID bit on `myls`. Confirm the SUID bit is set correctly. Run the script as user seed. What's the output of the script? Do you see anything unexpected? Try to list the files under directory "/root" and compare results from running the script directly and running it with sudo.

Links on security issues on scripts:

http://www.faqs.org/faqs/unix-faq/faq/part4/section-7.html

https://www.drdobbs.com/dangers-of-suid-shell-scripts/199101190