CIS 418/518 – Secure Software Engineering

Set-UID Privileged Programs (Chapter 2)

1. What is a Set-UID program?
   In Linux, a Set-UID program is a privileged program that has its Set-UID bit on.

2. Which ID of a process is used to determine its access privileges?

   Real user ID (`uid`) **Effective user ID (`euid`)**

3. When a Set-UID program is run, what determines the privileges granted to it?

   a) Privileges of the user running the program
   b) Privileges of the owner of the program

4. When a Set-UID program owned by `root` user is executed by user `bob` (user ID 1000), what is the effective user ID of the corresponding process? **`euid` will be 0**

5. When a non-Set-UID program owned by `root` is executed by user `bob` (user ID 1000), what is the effective user ID of the corresponding process? **`euid` will be 1000**

6. Is `/bin/passwd` a Set-UID program? What command can help you answer this question?
   Yes; `$ ls -l /bin/passwd`

7. Is `/bin/sudo` a Set-UID program? What command can help you answer this question?
   Yes; `$ ls -l /bin/sudo`

8. Is `/bin/su` a Set-UID program? What command can help you answer this question?
   Yes; `$ ls -l /bin/su`

9. Is `/bin/cat` a Set-UID program? What command can help you answer this question?
   No; `$ ls -l /bin/cat`

10. Alice runs a Set-UID program that is owned by Bob. The program tries to read from `/tmp/x`, which is readable to Alice, but not to anybody else. Can this program successfully read from the file? **NO**

11. A process tries to read a file for read. The process's effective user ID is 1000 and real user ID is 2000. The file is readable to user ID 2000, but not to user ID 1000. Can this process successfully open the file? **NO**

12. We are trying to turn a program `prog` owned by the `seed` user into a Set-UID program that is owned by `root`. Can running the following commands achieve the goal?

```
$ sudo chmod 4755 prog
```

```
$ sudo chown root prog
```

No; you have to flip the order of these command to achieve the goal.

13. Both `system()` and `execve()` can be used to execute external programs. Why is `system()` unsafe while `execve()` is safe?

    `system()` launches a shell program and the shell runs the command. Shell parses the command string and executes the command(s) it finds in the string. The command string does not isolate/separate the command from its arguments/data. It is easy to inject additional (malicious) commands into the string.

    `execve()` function takes three arguments: command, an array of arguments to the command (including the command), and an array of environment variables to pass to the process that runs the command.

    `execve()` clearly separates command from its arguments/data. It enforces the *principle of isolation* (code and the data that the code operates on is separated), thus making it a safe way to run external programs from inside a program (set-uid or not).

    Read sections 2.5.1, 2.5.2, and 2.5.4 in the textbook.