HW 10

Joshua Thompson

April 8, 2018

Questions

1. (5 points) Why must there be a de-escelation of privilege when the login program executes the shell for an authenticated user?

   So that the authenticated users doesn't get root priveleges

2. (10 points) Consider the following program with the following permission strings below, if you (as your username) were to run these, programs, hat capabilites (group and user permissions) would the executing program have?

   (a) `-rwxr-x--x 1 aviv scs 8622 Mar 30 10:40 a.out`

   The user running the program only gets the permission they have

   (b) `-rwsr-x--x 1 aviv scs 8622 Mar 30 10:40 a.out`

   The user running the program gets the setbit for user execution but retains the real id for group execution

   (c) `-rwxr-s--x 1 aviv scs 8622 Mar 30 10:40 a.out`

   The user running the program has the real id for user execution but has set bit for group execution

   (d) `-rwsr-s--x 1 aviv scs 8622 Mar 30 10:40 a.out`

   The user running the program has setbit for both user and group execution

3. (5 points) What is the difference between the real and effective user and group id of a running process?

   A user's real id is unique and has their own permissions but when running a program, the effective user id is who ever's id it was that made the progam. When one runs the program they take the id of the owner and assume their permissions.

4. (15 points) Provide a short, Plain english description of each of the system calls below.

   (a) `getuid()`

Returns the real user id of the calling process

(b) `getgid()`

Returns the real group id of the calling process

(c) `geteuid()`

return the effective user id for the calling process

(d) `getegid()`

return the effective group id for the calling process

(e) `setuid(uid)`

change the effictive user id of a process to uid given in the parameter

(f) `setgid(uid)`

change the effective group id of a process to gid given in the parameter

(g) `setreuid(uid, euid)`

Sets the real user id and the effective user id

(h) `setregid(gid, egid)`

Sets the real group id and the effective group id

5. (10 points) Consider the following chmod statements, provide the permission string, that is the permission string rwxrwxrwx represents 777. Be careful about setbits

(a) `chmod 6750 a.out`

rwsr-s—

(b) `(chmod 4750 a.out`

rwsr-x—

(c) `(chmod 2750 a.out`

rwxr-s—

6. (5 points) Supposed you are writing a setuid program, and you want downgrade the effective permission of the program back to the user who is execuing the program. Provide on line of c that can do that.

setuid(uid);

7. (5 points) What does the library call system() do?

The system call executes a command as if it were written on the command line using the effective user id.

8. (5 points) Explain how the environment variable PATH is used to select which program to execute when using execvp() or system() or in a shell?

The system calls check the current directory to see if there is a command that matches the one that the user wants to call. If there is not a program that matches the call, then check the parent directory and keep doing that until in the home directory. It then defaults to the /bin directory of calls to call the program that the user wants.

9. (10 points) The following program has multiple security flaws. Describe how to exploit and provide at least one way to change the program to protect it from attack.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{system("cat smaple.db | cut -d ',' -f 3 | sort | uniq")}
```

We can exploit this program by writing our own "cat" or "cut" or "sort" or "uniq" programs to execute whatever we want in the system. We can protect from this attact by specifying that we want the commands to be called from the /bin directory

10. (10 points) The following program has two security flaws, describe how to exploit them.

```
int main()
{
char cmd[1024];
char fname[40];
printf("Enter file name:\n");
scanf(%s, fname);
snprintf(cmd, 1024, /bin/cat %s", fname);
system(cmd);
}
```

The first security flaw is buffer overflow for the command. The second is to close off the command using the ";cmd" technique.

11. (10 points) Describe a solution to each of the security flaws you identified in the previous question

We can bound check the command string to make sure that it is withing the right size.

12. Consider yourself as a software developer designing a tool for your organization that takes advantage of different UNIX system tools. As such you wish to make use of the system() and popen() calls to inter operate with your tool and teh standard UNIX tools. Whie the tool needs to have high privelege leves, individual users may ned varying lesser levels, but not necessarily equal across users.

    (a) (5 points) Describe three potential ethical and legal impacts on your organization (with respect to actions attackers could take) if your software was designed insecurely.

Some potential impacts could be that an unauthorized user could elevate their privileges and do things that they shouldn't be doing.

(b) (5 points) Describe three coding practices you can employ to reduce the ethical and legal impacts of insecurity in your software.