EE487: Applications of Cyber Engineering Name(s):
SX-6: Note Taking Program and POSIX File Permissions

Discussion: This SX is intended to familiarize students with permissions and access control. While SX-5 taught the basics of authentication, this SX will focus on the next steps – **authorization and access**. As with authentication, mechanisms for authorization and access started out in very simple form but have now evolved into very complicated algorithms. If you're curious, look up "Kerberos" on the Internet and ponder why we need such complicated systems. In this lab, however, we're going to introduce "POSIX" permissions.

Materials: The example source code from the Erickson text, *notetaker.c* and *notesearch.c*, will be used for this lab, along with the Live CD VM.

Reading assignment: Erickson, section 0x282 and 0x283, pages 87-96.

1. Let us first revisit those ten fields we deferred from the last SX. Type this command and write down the output.

   ```
   ls –lha /etc | grep passwd
   ```

   Focus on the ten characters on the far left. They are the "POSIX" permissions. POSIX is an IEEE standard for operating systems and covers many features, but in this case it allows many different operating systems to have compatible file permissions. As a result, this method is by far the most widely used (and crucially important).

2. Look at the file permissions (the sequence of up to ten letters/dashes at the far left of the output). The very first dash indicates that these are files (and not directories or links). The remaining nine positions indicate the file permissions for the "user," "group," and "other." Do a "**`man chmod`**". Write down the file permission differences between the passwd file and its backup version (with the '-'). Why might this be the case?

3. Type:

```
ls –lha /usr/sbin | grep john
```

The top line is the program file "John the Ripper" that you ran in SX-5. Why are there x's in the permissions?

Now type:

```
sudo chmod o-x /usr/sbin/john
```

Then perform the detailed listing command from above again (**Hint**: try pressing the up arrow twice and see if this might save you some typing). What changed?

Now try to run john again like you did in SX-5. What happened and why?

Note: It is somewhat advanced to use `chmod` with octal permissions as in the book. This method specifies absolute permissions, while using the +/- method is a bit safer because it only changes one feature at a time. For a true hacker, however, the octal method can be very useful.

4. What's the difference between a text file and a program? Think in terms of the file contents <u>AND</u> the file properties and permissions.

5. If not already done, compile the *simplenote.c* program and create the note "I hack for charity". Then from section 0x283 on page 88, follow with the author and carefully execute **ALL** of the commands just as he does up through page 96. <u>This section is crucial</u>! Answer the questions below as you progress.

6. Type:

   `cat /etc/passwd`

   Examine the UID's and GID's in that file. What are yours? What about root?

   Then type:

   `cat /etc/group`

   Can a user be in several groups? Is one of them primary?

7. **SETUID** (Important!)
   a. What does the "**`setuid`**" permission allow? Why might both a programmer (white hat) and a malicious (black-hat) hacker find this feature useful?

   b. When a program runs with "**`setuid`**" set to UID 0, which user does the system believe is running the program?

   c. In this case, it is possible to tell who the <u>actual</u> user is that initiated the program?

8. Why is it that when you execute the *notesearch* program as 'jose' you do not see the notes written by 'reader' despite the fact that notes for both users exist in the file?

9. If you execute *notesearch* as 'root' would you expect to see the notes of all users? Try it out.

EE487: Applications of Cyber Engineering Name(s):
SX-6: Note Taking Program and POSIX File Permissions

Conclusion and Results:

Your typed lab report will consist of two paragraphs, in the first paragraph:
• Briefly describe what you did in the lab in your own words.
• Discuss something new that you learned.

In the second paragraph, answer the questions:
• How could an adversary use this knowledge or these tools for malicious purposes?
• How could you use your new understanding to protect your systems and personnel from attack?

Staple the completed report to the back of your original lab and turn it in to your instructor at the beginning of the next class.