Homework 2

Due: April 26, 2013 at 11:55pm **Points:** 100

Questions

1. (*20 points*) Classify each of the following vulnerabilities using the PA model. Assume that the classification is for the implementation level. Remember to justify your answers.
   (a) The presence of the "wiz" command in the *sendmail* program (see Section 23.2.).
   (b) The failure to handle the **IFS** shell variable by *loadmodule* (see Section 23.2.8).
   (c) The failure to select an *Administrator* password that was difficult to guess (see Section 23.2.9).
   (d) The failure of the Burroughs system to detect offline changes to files (see Section 23.2.6).
   (*text* §23.9, exercise 2).

2. (*20 points*) Suppose Alice has *r* and *w* rights over the file *book*. Alice wants to copy *r* rights to *book* to Bob.
   (a) Assuming there is a copy right *c*, write a command to do this.
   (b) Now assume the system supports a copy flag; for example, the right *r* with the copy flag would be written as *rc*. In this case, write a command to do the copy.
   (c) In the previous part, what happens if the copy flag is *not* copied?

3. (*20 points*) Classify each of the following as an example of a mandatory, discretionary, or originator controlled policy, or a combination thereof. Justify your answers.
   (a) The file access control mechanisms of the UNIX operating system
   (b) A system in which no memorandum can be distributed without the author's consent
   (c) A military facility in which only generals can enter a particular room
   (d) A university registrar's office, in which a faculty member can see the grades of a particular student provided that the student has given written permission for the faculty member to see them.
   (*text* §4.11, exercise 5)

4. (*40 points*) This problem asks you to implement a buffer overflow attack on a program. In the Resources area of SmartSite (or the Homework area of the nob.cs.ucdavis.edu class web site) is a program *bad.c*. This program contains a buffer overflow vulnerability; see the call to *gets*(3) at line 13. Your job is to exploit the overflow by providing input to the running process that will cause the program to invoke the function `trap` (which, you may notice, is not called anywhere else). You will know you've succeeded when you run the program, give it your input, and it prints "`Gotcha!`"

   The following questions will help guide you. Please turn in your answers to them, a hex dump of the input you use to call `trap`, and a typescript or screen shot of you running the program *bad*, giving it your input, and showing its output.
   (a) What is the address of the function `trap`()? How did you determine this?
   (b) What is the address on the stack that your input must overwrite (please give both the address of the memory location(s), and their contents)? How did you locate this address?
   (c) What is the address of `buf`?
   (d) The *sled* is the input you give to alter the return address stored on the stack. What is the minimum length your sled must be?

Extra Credit

1. (*14 points*) One expert noted that the PA model and the RISOS model are isomorphic. Show that the PA vulnerability classifications correspond to the RISOS vulnerability classifications and *vice versa*.
   (*text* §23.9, exercise 10).