Homework 2

Due Date: April 27, 2011 **Points:** 100

Corrections

- In Problem 1, the set of rights should be $\{r, w, x, a, l, m, o\}$, where m is the *modify* right. Also, o is the *own* right.
- In the Extra Credit problem, "problem 5" should be "problem 4".

These are corrected below.

Questions

1. (*25 points*) Consider the set of rights $\{r, w, x, a, l, m, o\}$.

   (a) Using the syntax in Section 2.3, write a command *delete_all_rights*(p, q, o). This command causes p to delete all rights the subject q has over an object o.

   (b) Modify your command so that the deletion can occur only if p has *modify* (m) rights over o.

   (c) Modify your command so that the deletion can occur only if p has *modify* (m) rights over o and q does not have *own* (o) rights over o.

   (*text*, §2.8, exercise 4).

2. (*15 points*) A common error on UNIX systems occurs during the configuration of *bind*, a directory name server. The time-to-expire field is set at 0.5 because the administrator believes that this field's unit is minutes (and wishes to set the time to 30 seconds). However, *bind* expects the field to be in seconds and reads the value as 0—meaning that no data is ever expired.

   (a) Classify this vulnerability using the RISOS model, and justify your answer.

   (b) Classify this vulnerability using the PA model, and justify your answer.

   (c) Classify this vulnerability using Aslam's model, and justify your answer.

   (*text*, §23.9, exercise 4).

3. (*15 points*) Given the security levels TOP SECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED (ordered from highest to lowest), and the categories A, B, and C, specify what type of access (read, write, or both) is allowed in each of the following situations. Assume that discretionary access controls allow anyone access unless otherwise specified.

   (a) Paul, cleared for (TOP SECRET, { A, C }), wants to access a document classified (SECRET, { B, C }).

   (b) Anna, cleared for (CONFIDENTIAL, { C }), wants to access a document classified (CONFIDENTIAL, { B }).

   (c) Jesse, cleared for (SECRET, { C }), wants to access a document classified (CONFIDENTIAL, { C }).

   (d) Sammi, cleared for (TOP SECRET, { A, C }), wants to access a document classified (CONFIDENTIAL, { A }).

   (e) Robin, who has no clearances (and so works at the UNCLASSIFIED level), wants to access a document classified (CONFIDENTIAL, { B }).

   (*text*, §5.8, exercise 2).

4. (*45 points*) This problem has you implement a buffer overflow attack on a program. In the "Program" folder of the "Resources" area of SmartSite (or `http://nob.cs.ucdavis.edu/classes/ecs153-2011-02/programs`) is a program called *bad.c*. This program contains a buffer overflow vulnerability; see the call to *gets*() 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 have 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 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) What is the *minimum* length your sled must be? Remember, the sled is the input you give to alter the return address stored on the stack.

Extra Credit

1. (*20 points*) Augment your solution to problem 4 to execute code you place on the stack. Have the code do something interesting, like create a shell. You will need to check that the loader will allow code on the stack to be executed (the linker/loader switch—for the Fedora Core systems in the CSIF, the option `-Wl,-z,execstack` to *gcc* will do this.