Homework 4

Due Date: May 29, 2001
Points: 120

1. (*20 points*; text, exercise 12.9) The designers of the UNIX password algorithm used a 12-bit salt to perturb the first and third sets of 12 entries in the E-table of the UNIX hashing function (the DES). Which would most greatly increase the expected time to guess a password chosen at random: adding 8 more characters to the password, or 12 more bits to the salt? Please justify your answer.

2. (*20 points*; text, exercise 13.3) In their wonderful book *Software Tools*, Kernighan and Plauger argue a minimalist philosophy of tool building. Their thesis is that each program should perform exactly one task, and more complex programs should be formed by combining simpler programs. Please discuss how this philosophy fits in with the principle of economy of mechanism. In particular, how does the advantage of the simplicity of each component of a software system offset the disadvantage of a multiplicity of interfaces among the various components?.

3. (*20 points*; text, exercise 14.1) The web site *www.widget.com* requires users to supply a username and a password. This information is encoded into a cookie and sent back to the browser. Whenever the user connects to the web server, the cookie is sent. This means the user need only supply a password once. The name of the cookie is "identif".
   a. Assume the password is kept in the clear in the cookie. What should the settings of the secure and expires fields be, and why?
   b. Assume the name and password are hashed and the hash stored in the cookie. What information must the server store to determine the user name associated with the cookie?
   c. Is the cookie storing state or acting as an authentication token or both? Please justify your answer.

4. (*20 points*; text, exercise 15.6) It is said that UNIX uses access control lists. Does the UNIX model include capabilities as well as access control lists? (*Hint*: consider file descriptors. If a file is opened, and its protection mode changed to exclude access by the opener, can the process still access the file using the file descriptor?)

5. (*20 points*; text, exercise 15.7) Suppose a user wishes to edit the file *xyzzy* in a capability-based system. How can he be sure that the editor cannot access any other file? Could this be done in an ACL-based system? How, or why not?

6. (*20 points*; text, exercise 16.2) Let $L = (S_L, \leq_L)$ be a lattice. Prove that the structure $IL = (S_{IL}, \leq_{IL})$, where:
   a. $S_{IL} = \{ [a, b] \mid a, b \in S_L \wedge a \leq_L b \}$;
   b. $\leq_{IL} = \{ ([a_1, b_1], [a_2, b_2]) \mid a_1 \leq_L a_2 \wedge b_1 \leq_L b_2 \}$;
   c. $lub_{IL}([a_1, b_1], [a_2, b_2]) = (lub_L(a_1, a_2), lub_L(b_1, b_2))$; and
   d. $glb_{IL}([a_1, b_1], [a_2, b_2]) = (glb_L(a_1, a_2), glb_L(b_1, b_2))$,
   is a lattice.