Homework 4

Due Date: March 10, 2006   Points: 100

1. (*16 points*) Text, problem 5.2.
2. (*14 points*) Text, problem 6.10.
3. (*5 points*) Text, problem 9.6.
4. (*20 points*) Text, problem 9.18.
5. (*45 points*) A computer science student is helping out on a project that involves monitoring a network. The student's job is to write a program that will pluck the URLs for all HTTP traffic from the network. This requires *root* privileges, but for policy reasons the professor who is running the project cannot give the student those privileges. His solution is to create a small program, called *mksuid*, that will make a small program called *sniff* setuid to *root*.

   The program *runpriv* works as follows:

   1. Check that the student is running the program by comparing the real UID of the process with that of the student. (Assume you are the student for this testing.) If the test fails, print an error message and exit.
   2. Prompt the user for his or her password, and validate it against the one stored in the password file. If the password entered is incorrect, print an error message and exit.
   3. If the current working directory does not contain a file called *sniff*, print an error message and exit.
   4. If the *sniff* is not owned by the student, or is not executable by the owner of the file, or can be read, written, or executed by anyone else (except, of course, *root*), print an error message and exit.
   5. If *sniff* was created or modified over 1 minute ago, print an error message and exit.
   6. Change the ownership of *sniff* to *root* (UID 0), its group to *proj* (GID 95), and its protection mode to 04550 (meaning setuid to owner, and **only** readable and executable by the owner and group members).

   Please write this program.

   One goal of this program is to teach you how to program with security in mind. To that end, we are experimenting with a "secure programming clinic". The purpose of this clinic is to review programs **before** you turn them in, with an eye to helping you understand potential security problems and working around them. So, once you have your first **working** version of this program, please send a note to progreview@cs.ucdavis.edu. Ask to meet with the clinic (currently it's a graduate student). Please attach your program. The graduate student will respond and arrange a meeting time, and go over your program with you. ***You must do this before you turn in your program, or your program will not be graded and you will lost 45 points!*** Then, please revise your program as appropriate, before you submit it.

   Please submit your program as a "tar" file. Don't forget to include a *Makefile* that will automatically compile your program!

Extra Credit

1. (*20 points*) Text, problem 9.13.
2. (*25 points*) In the programming problem above, explain the reason for each of the first five steps in the way the program works. What assumption(s) overall is the professor making?