Homework #1

Due: April 7, 2021 **Points:** 100

Questions

1. (*16 points*) The aphorism "security through obscurity" suggests that hiding information provides some level of security. Give an example of a situation in which hiding information does not add appreciably to the security of a system. Then give an example of a situation in which it does.

2. (*24 points*) The program *su* enables a UNIX user to access another user's account. Unless the first user is the superuser, *su* requires that the password of the second user be given. A (possibly apocryphal) version of *su* would ask for the user's password and, if it could not determine if the password was correct because the password file could not be opened, *immediately* grant superuser access so that the user could fix the problem. Discuss which of the design principles this approach meets, and which ones it violates.

3. (*30 points*) The *Robust Programming* handout points out that multiplication can cause overflows. The obvious way to test for overflow in ab is to multiply the absolute value of a and b and see if the result is smaller than the absolute value of either a or b (because if $|ab| < |a|$ when $|a| > 1$ and $|b| > 1$, then overflow has occurred). Does this always work—if so, say why, and if not, give a counterexample? Assuming it works, what problems would it introduce? (Hint: think about architectures allowing arithmetic overflow to cause a trap.) Suggest an alternate method without these problems.

4. (*10 points*) On a Linux or UNIX-like system, how does *ftell*(3) use *errno* to distinguish failure from success?

5. (*20 points*) The story "Diabologic" by Eric Frank Russell presents an explorer making first contact with an alien race. That race thinks very logically. He proceeds to confound them.
   (a) Why can the newly-contacted race not cope with the explorer's tactics?
   (b) What key theme of this story relates to attacking or defending a computer system, and how does it do so?