Sample Final Exam

1. The following routine reads a file name from the standard input and returns its protection mode. It treats the argument as a file name, and returns the protection mode of the file as a short integer. Identify three non-robust features of this routine, and state how to fix them.

```
/* return protection mode of the named file */
short int protmode(void)
{
      struct stat stbuf;
      char inbuf[100];

      gets(&inbuf);
      stat(inbuf, &stbuf);
      return(stbuf.st_mode&0777);
}
```

2. Define each of the following terms in one short sentence:
   a. public key cryptosystem
   b. challenge-response
   c. ciphertext
   d. end-to-end encryption
   e. principle of fail-safe defaults
3. Show how ACLs and C-Lists are derived from an access control matrix.
4. Discuss the revocation problem with respect to access control lists and capabilities. How might one efficiently implement a command to revoke access to an object by one particular user?
5. Consider the Bell-LaPadula multilevel security model. If a subject with security label (*L*, *C*) can read an object with security label (*L'*, *C'*), then (*L*, *C*) is said to *dominate* (*L'*, *C'*). Prove that this *dominates* relation is reflexive, antisymmetric, and transitive.
6. Consider the problem of managing certificates. One expert said that a hierarchical scheme, such as that employed by PEM, is more likely to be used for business than the Web of Trust employed by PGP. What specific features of the hierarchical system as implemented for PEM (and for other Internet applications) led him to make this assertion? Why might these features lead him to make this statement?
7. Why is a precise statement of security requirements critical to the determination of whether a given system is secure?
8. System vendors often add security features to strengthen the security of their systems. These additions are not designed into the system, but rather are added after the system has been shipped. Discuss whether adding security features to a large, complex operating system not designed with security in mind (such as the UNIX operating system or Windows 95) violates any of Saltzer's and Schroeder's design principles. (Go through all 8 design principles.)