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. Show how ACLs and C-Lists are derived from an access control matrix.
3. 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?
4. Consider a system that used the Bell-LaPadula model to enforce confidentiality and the Biba model to enforce integrity.
   a. If the security classes were the same as integrity classes, what objects could a given process (with some security class that also served as its integrity class) access?
   b. Why is this scheme not used in practice?
5. 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?
6. Please describe how the vulnerabilities models are used during the Flaw Hypothesis Methodology. Be explicit: which phase of the methodology uses them, and how?
7. Into which category or categories of the Program Analysis classification do the following fall? Please justify your answer.
   a. Buffer overflow causing a return into the stack?
   b. Allowing an ordinary user to alter the password file?
   c. Simultaneous writes to a shared database?
   d. Reading a UNIX file by directly accessing the raw device and reading first the superblock, then the file's inode, and finally the file's data blocks?