Answers to Sample Final

1. Represent a security compartment label using the notation

   <*security level*; *set of categories*>.

   Can a user cleared for secret; { dog, cat, pig } have read or write access (or both) to documents classified in each of the following ways under the military security model?

   a. <top secret; { dog }>

   b. <secret; { dog }>

   c. <secret; { dog, cow }>

   d. <secret; { moose }>

   e. <confidential; { dog, pig, cat }>

   Answer:

   a. No read (as top secret > secret); no write (as { dog, cat, pig } $\not\subseteq$ { dog })

   b. Read (as secret = secret and {dog} $\subseteq$ { dog, cat, pig }); no write (as { dog, cat, pig } $\not\subseteq$ { dog })

   c. No read (as { dog, cow } $\not\subseteq$ { dog, cat, pig }); no write (as { dog, cat, pig } $\not\subseteq$ { dog, cow })

   d. No read (as { moose } $\not\subseteq$ { dog, cat, pig }); no write{ dog, cat, pig } $\not\subseteq$ { dog, moose })

   e. Read (as confidential < secret and { dog, pig, cat } $\subseteq$ { dog, cat, pig }); no write (as confidential < secret)

2. 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?

   Answer:

   a. Assume the security classes were the same as the integrity classes. Let A and B be the labels of security compartments, where A *dom* B. Then under the Bell-LaPadula model, a subject with label B cannot read an entity with label A. Under Biba's model, a subject with label A cannot read an entity with label B. A similar set of conditions holds for writing. However, if $A = B$, then both models allow reads and writes. And, of course, if there is no dominance relation between any two labels, entities with those labels can neither read nor write one another. Thus, if the security classes are the same as the integrity class, a given process can only access objects in its own compartment (class).

   b. This scheme is far too restrictive to be used in practice. The processes are completely confined to their compartments, and often processes need to be able to read data in compartments that their compartment dominates. This is not possible in this scheme.

3. Define each of the following terms in one short sentence:

   a. public key cryptosystem

   b. challenge-response

   c. computer worm

   d. end-to-end encryption

   e. web cookie

   Answer:

   a. A public key cryptosystem is a cryptosystem with two keys, one of which is known to everyone (the public key) and the other of which is known only to one person (the private key).

   b. A challenge-response refers to a mechanism in which the client is challenged to demonstrate knowledge of a secret to a server. The client's response must demonstrate to the server that the client knows the secret.

c. A computer worm is a program that replicates and copies itself from system to system.

d. End-to-end encryption occurs when a message is enciphered, and sent to its destination, and then deciphered at its destination. It is enciphered as it transits the network, leading to the term "end-to-end encryption."

e. A web cookie is a construct that encapsulates the state of a session between a browser client and a web server.

4. Show how ACLs and C-Lists are derived from an access control matrix.

   Answer: ACLs correspond to the columns of an access control matrix, and C-Lists correspond to the rows of an access control matrix.

5. 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?

   Answer: Suppose we wanted to revoke subject *s*'s access rights *r* to a file *f*. If the system used access control lists, one would revoke the access by going to *f*'s ACL and deleting *s*'s rights *r*. If the system used capability lists, one would revoke the access by going to *s*'s capability list and remove the capability that gives *s* the *r* rights over *f*.

   With ACLs, it is trivial to remove all rights to a given object from all subjects. With C-lists it is much more difficult. For example, suppose we want to remove all users' rights to read a file*f*. We need traverse only one ACL, but will need to traverse every process' C-List to see if that process has read rights over *f*. Conversely, with C-Lists, it is easy to remove all rights to all objects from a given subject. With ACLs, it is much harder. For example, we want to remove a subject *s*'s rights to all objects. We need traverse only one C-List, but would need to traverse every file's ACL.

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?

   Answer: The primary consideration was the standard for signing certificates. In the PEM hierarchy, each Certification Authority (the entity that issues certificates to individuals) is legally bound to comply with the policies for issuing certificates that its Policy Certification Authority states. Therefore, when one gets a certificate, one can determine the methods used to authenticate the subject (the entity to whom the certificate was issued) by examining the document describing the standards that the PCA requires the CA to use. The Web of Trust, on the other hand, leaves the level of trust that a certificate signer (the PGP equivalent of a certificate issuer) has in the subject to the discretion of the issuer, and the standards that the signer uses need not be written down, or even be consistent among certificates signed by that certificate signer. Hence one cannot determine from the certificate how the signer validated the subject. This makes trust in identity more problematic than for certificates issued under the PEM hierarchical model.

7. Into which category or categories of the Protection 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?

   Answer:

   a. Buffer overflow causing a return into the stack is an example of improper validation. It arises from not checking the length of the string being loaded into the buffer. One could also argue that it is an example of improper choice of operation or operand, because the operand (the string) is meant to be data, but is in reality instructions.

   b. Allowing an ordinary user to alter the password file is an example of improperly setting the initial protection domain, because the user should not be able to alter the password file.

c. Two processes updating a database simultaneously can cause inconsistencies. As they must synchronize themselves, failing to do so is improper synchronization.

d. Accessing a file by reading the raw disk bypasses the abstraction of "file" in the UNIX operating system. Hence this is an improper isolation of implementation detail.