Name: _____ UNI: _____

COMS W4187: Security Architecture and Engineering
October 2007

Rules

- Remember to write your name and UNI on the blue exam book.
- **Important: also write your name on this paper**
- You must turn in *both* the exam sheet and the blue book
- Books and notes are allowed during this examination; computers are not permitted.
- This is a time-limited test. All papers must be turned in 75 minutes after the beginning of the test.
- The total points add up to 90 .
- Good luck, and may the Force be with you.

1		20
2		10
3		15
4		10
5		15
6		10
7		10
Total		90

Questions

1. (20 points) All parts of this question concerns the class's homework submission script, which is not setUID. The question does *not* assume that you have read it.

a. (5 points) The directories to which assignments are written have permission `rwx,-wx,-wx`. Why are the directories world-writable?

The script is running with the students' permissions. Thus, the directory has to be writable by the students. I could have relied on group-write permission if I were certain that all students were in the proper group.

b. (5 points) The file names used for assignments contain many random characters. What is the purpose of these characters?

It is necessary to prevent people from deleting other people's assignments. Since the parent directory is world-writable, anyone who knows the name of a file can delete it. The random characters prevent guesses, and the lack of read permission on the directories prevents them from being listed.

c. (10 points) The script does not add entries to a logfile for submitted assignments, even though such a log would be desirable. I omitted that feature because I couldn't do it securely. What is the problem?

As in part (a), writing to the log file would have to be done with the students' permissions. This means that any student would have the ability to overwrite the entire log

file, or to create bogus entries. If there were some form of append-only permission, the situation would be different, but there isn't.

2. (10 points) In a discussion of random number generation (Lecture 12, from October 11), I wrote

- An application can keep a file with a few hundred bytes of random numbers
- Generate some true-random bytes, mix with the file, and extract what you need
- Write the file back to disk — read-protected, of course — for next time

Describe an attack that could happen if this file were world-*writable*? (Assume that the application needs to be secure.)

Most of the randomness in the next number to be generated comes from this file. If someone overwrites it, they know most of those bits, and only have to guess at the (presumably small number) of true-random bytes mixed in.

3. (15 points) I found a web page with some declassified NSA documents (`http://cryptome.org/nsa-nse/nsa-nse-01.htm`, if you want to look later). The classifications of a subset of them are as follows:

Document	*Level*	*Compartment*
60	Top Secret	Umbra, Noforn, X1
62	Secret	
65	Confidential	
66	Confidential	MR
63	Top Secret	Umbra, Laconic, Nocon

The hierarchy of classifications, from least to most sensitive, is Confidential, Secret, Top Secret.

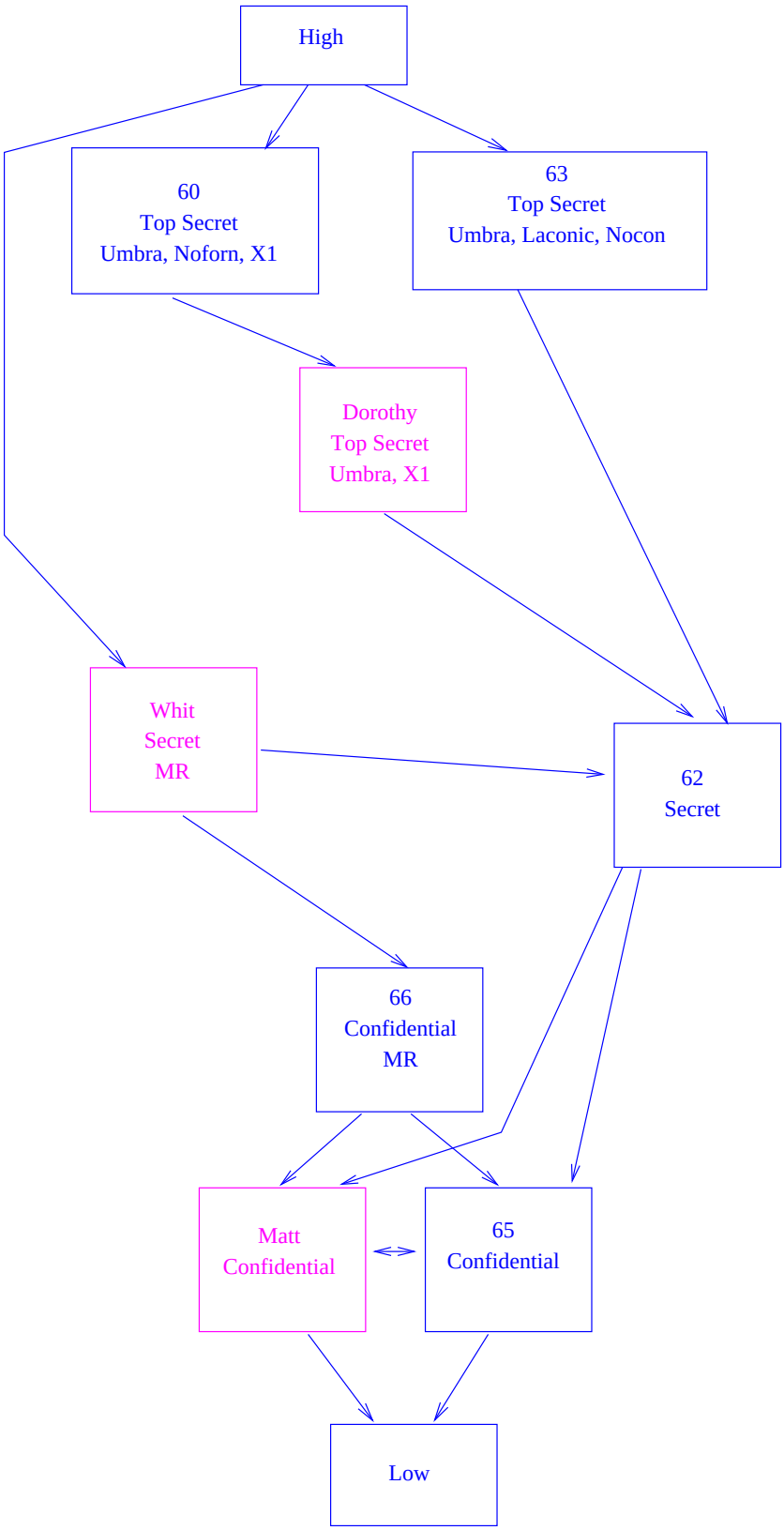
Assume that the system has the following users:

User	*Level*	*Compartment*
Whit	Secret	MR
Dorothy	Top Secret	Umbra, X1
Matt	Confidential	

a. (10 points) Draw the lattice showing the relationships among these documents and individuals.
See the following page

b. (5 points) For each individual, list which documents they can read
Whit: 62, 65, 66 (wrong level for 60, 63)
Dorothy: 62, 65 (wrong compartments for 60, 63, 66)
Matt: 65 (wrong level and compartments for the others)

As a footnote: after I made up the exam, I learned that X1 and MR are not compartments. X codes refer to material exempt from automatic declassification; MR means "manual review". And the three names were selected in honor of prominent security and cryptography researchers.


High
60
Top Secret
Umbra, Noforn, X1
63
Top Secret
Umbra, Laconic, Nocon
Dorothy
Top Secret
Umbra, X1
Whit
Secret
MR
62
Secret
66
Confidential
MR
Matt
Confidential
65
Confidential
Low

4. (10 points) Ross, a wealthy businessman, has some very, very sensitive data that he accesses once per year. To protect it, he encrypts it with multiple keys, so that you need all of the keys to decrypt the file:

$$\{\{\{\{\{\text{file}\}_{K_5}\}_{K_4}\}_{K_3}\}_{K_2}\}_{K_1}$$

K_1 is stored on a flash disk he wears on a string around his neck. K_2 is stored in a bank vault. K_3 is split into two pieces; one is held by his lawyer and one by his accountant. K_4 is a passphrase memorized by his grandfather. K_5 is a passphrase Ross has memorized.

Is this a good way to protect the file? Why or why not?

It's a very bad way to do things, because there's a tremendous risk of loss of one of the keys. If he loses the flash disk or forgets his key, or if he switches lawyers or accountants, or if his grandfather dies or forgets the key, there's no way to decrypt the file.

5. (15 points) A university proposes replacing its ID card with a series of fingerprint readers. These would be used to verify the students' identity for access to the lounge, signing on to administrative computer systems, and buying computers at the student store. For *each of these three uses*, is this a good idea? Explain why or why not *for each one*.

Access to the lounge is a low-value resource. A fingerprint reader is probably acceptable.

Administrative computer systems (i.e., SSOL) are high-value targets, and they're remote from the users. Remote biometrics without observation don't work very well; all the far side sees is a set of bits.

If you're buying a computer in person, you're under observation by the sales clerk, so fingerprints can work well. However, the student should be asked for his/her name or ID number, since all biometrics work better on a match/no match basis than on picking out one person from a large database.

6. (10 points) Reviewing the test logs for a new product, you note that during testing the program dumped core 10% of the time when passed invalid input. What are the security implications of that observation?

A core dump like that generally indicates an unexpected, unhandled error condition. Since this happens in response to invalid input, it suggests that the program is not prepared to handle invalid input properly. It would be prudent to assume that some bad guy could trick the system into doing something, perhaps by a buffer overflow.

7. (10 points) On Linux, for every process n there is a file `/proc/`n`/mem`, which corresponds to the memory of process n. What permissions should this file have?

If the file is writable by anyone else, they can change the data in a program being run by someone else. This is bad... If others can read that process' memory, they can spy on confidential data, including passwords and cryptographic keys. Accordingly, it should not be readable or writable by any others.

Making the file writable by owner — the user running the program — is more complex. Linux does, in fact permit this, for use by debuggers, but it's far from obvious that this is the right decision. For this exam, either answer is acceptable on this point.