Name: _____ UNI: _____

COMS W4187: Security Architecture and Engineering
October 2010

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.
- Most questions can be answered in just a paragraph or two; if you think you need to write several pages, you're writing too much and may be on the wrong track entirely.
- The total points add up to 75.
- Good luck, and may the Force be with you.

Question	Points	Score
1	20	
2	15	
3	15	
4	10	
5	15	
Total:	75	

1. (20 points) Suppose I were concerned that substitutes were taking this exam, instead of the actual enrolled students. What form of authentication could I use? Note: the total enrollment in the class is 40. I know of no secret university resources I could use, such as fingerprint databases; anything you specify has to be contained to this class.

   Answer:

   The best solution is probably looking at the picture on the CUID, and perhaps comparing a signature with that on some other piece of identification such as a driver's license or passport.

   It is worth noting that for exams with more serious security threats — e.g., the LSAT (Law School Admission Test), there are stringent authentication measures, up to and including fingerprinting.

2. You're working in software development environment for a medium-sized company. There is a development copy of the code; there is also a separate copy for each release in the field, plus a patched version of each release that incorporates bug fixes.

Developers can make changes to the development copy. The manager of the development group is the only person allowed to create new releases and at some point copy the development version to a new release directory. Testers look only at release directory trees. Maintenance programmers develop patches for each release.

(a) (5 points) What form of access control mechanisms would you use for this setup? (Don't bother talking about version control systems; that's not what any part of this question is about.)

Answer:
Access control lists, with groups used for each role. Simple user/group/other doesn't work because some files need different group permissios for different roles.

(b) (10 points) I've identified four roles: developers, manager, testers, and maintainers. I've also identified three sets of files: development versions, release versions, and patched versions. Draw the access control matrix and show who has which permissions.

Answer:

	devel	*release*	*patch*
Developers	rw	-	-
Manager	r	rw	-
Testers	-	r	-
Maintainers	-	r	rw

Other permissions, such as 'x', aren't really relevant here.

You may make reasonable assumptions that don't contradict what I've said if you document them.

3. Suppose (in an environment with no network email) I want to be able to encrypt email sent to me, but (for some reason) without using public key cryptography. Instead, I write a program that processes all mail sent to me.

(a) (5 points) What are the risks I run?

Answer:
The program has to run as me while accepting input from others; this is difficult. The key file must be online; if the program is compromised, the key to all encrypted email will be revealed.

(b) (5 points) This program needs access to an encryption key. How should this key be protected?

Answer:
Make it readable only by "owner". It has to be online; it can't itself be encrypted.

(c) (5 points) Suppose, instead, that I write a setuid program that everyone will run to send me mail. What are the additional risks of this alternative? (Assume that everyone else will cooperate and actually run it to send me mail.)

Answer:
It will be directly invoked by possible attackers, which means that they can control the environment and not just the inputs.

4. There's a very small village on the border to another country. Rather than staff an immigration booth, two different biometric schemes are proposed to control a turnstile. Will the schemes work? Explain.

(Ignore issues of whether or not someone could just climb over or go around the fence. I'm also not interested in things like customs; the question is about the security of the schemes I describe.)

(a) (5 points) People walk in to the turnstile; a camera looks at their faces and decides whether or not to admit them.

Answer:
Biometrics do not do well at finding matches against large sets of data, so it will not work well. Facial recognition is not very accurate.

(b) (5 points) People insert use a mag stripe card with their name and digitized fingerprints; if their fingerprints match what's on the card, the name field is used to decide if they're allowed to enter.

Answer:
If the digitized fingerprints are digitally signed, this is a pretty good scheme; if they're not, it's easy to replace the contents of the mag stripe.

If you need to make any assumptions, state them explicitly.

5. Random numbers are very important in cryptography. Reportedly, the National Security Agency likes to use cryptographic algorithms, rather than strange hardware, to generate these.

(a) (5 points) What is the advantage of this policy?

Answer:
It's high assurance — algorithms don't suffer from the hard-to-trace failure modes of random number hardware.

(b) (10 points) What else do they need to do to avoid serious risks?

Answer:
They need to pick a true-random seed, and they need to make sure that no insiders and steal this seed.