Name: ______________________________ UNI: _____________

COMS W4187: Security Architecture and Engineering
October 2013

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	20	
3	15	
4	20	
Total:	75	

OKAY, CLASS, I'VE TURNED IN YOUR EXAMS FOR GRADING. NOW-
MISS LENHART?
I USED A #3 PENCIL INSTEAD OF A #2. WILL THAT MESS ANYTHING UP?
YOU WHAT?
AIEEE
BLAM
OH GOD!
OH GOD!
I'VE NEVER SEEN SO MUCH BLOOD!

1. Because of the hassles of installing cryptographic software that works everywhere, some secure email web sites (e.g., hushmail.com) offer "downloadable crypto": a web page with Java or Javascript that implements the necessary functions to send and receive secure email. There's a serious risk, though: you don't know what you're executing. Is it the real crypto code, or does it have a back door installed by the provider because the government ordered them to?

   (a) (10 points) To avoid this problem, a new site decides to offer digitally-signed crypto. That is, the code they send your browser will be digitally signed and will be accompanied by a certificate. This does not solve the problem. Explain why it fails. (Assume that the browser is in fact capable of verifying such signatures. This question is about security architecture, not implementation limitations.)

   Answer:
   A digital signature of the code only proves the code's authenticity, not that the code is benign. If the government ordered the provider to insert a back door, it would presumably also order the provider to digitally sign the new one.

   (b) (10 points) Design a better scheme that in principle can work. (Again, ignore the question of what browsers can or cannot do.)

   Answer:
   There are several possible answers here. Among them:

   - Allow for signatures by trusted endorsers, e.g., civil liberties groups. If they audit the code and find it benign, they can sign it, too.
   - Check that the code hasn't changed. This doesn't help if the back door has been there for a long time; it does let you know if there's been a change since you started using it.

2. (20 points) The MTA wants to replace the Metrocard used to pay subway and bus fares in New York. Suppose that someone suggested using iris scans: when you board a bus or enter a turnstile, something scans your eye and deducts the appropriate fare from your account. Is this a good architecture or not? Explain.

   Answer:
   No, it isn't.

   - It would probably be too slow
   - It would be privacy-invasive
   - It can only operate online
   - It requires someone to set up an account at a location with an iris scanner; this is inconvenient for casual riders, such as tourists
   - It makes it hard for one person to have multiple accounts, e.g., one for personal travel, one for work, etc.

   You would certainly need a system to preload user accounts with money, but you need that with more or less anything else other than accepting cash or credit cards.

3. Suppose I want to build an online exam server: students take the exam via their own web browsers on their own computers. Also assume that for some reason I'm not worried about students collaborating via their computers; consider only the questions I'm explicitly asking.

   (a) (10 points) What sort of authentication should be used? Justify your choice.

   Answer:

   Any solution has to rely on existing hardware. It's not possible to use biometrics or smart card readers, since (a) most students' computers don't have such things, and (b) remote, unobserved biometrics aren't a good idea. This means we're back to passwords, client side certificates, or some form of two-factor authentication. This could be done with client-side software or with a hardware token (If you specify this latter, you need to state this explicity.)

   Note that this isn't the weak link: the risk is someone else taking the exam with the cooperation of the student who should take it; such a student could share any necessary authentication details.

   (b) (5 points) Discuss other sorts of programming precautions you should take in building the server.

   Answer:

   Apart from the usual, the big risk here is someone seeing another student's answers. The best solution is to encrypt all answers in the instructor's public key.

   The usual attacks, such as bad input values and buffer overflows, must be considered.

4. (20 points) There's a new multiplayer game out that runs on distributed systems that share a file system, e.g., something like the CLIC Lab. There several different roles here:

   - The game company, which is constantly adding new features to the already-installed game
   - The local game administrator
   - The game itself, with secret files, score files, etc.
   - Many individual players; new ones can join at any time

   plus of course the site's usual system administrators. Describe the access control and permission architecture you would use.

   Answer:

   If you use message-passing, the game should run as its own service with its own userid, to prevent hacks from affecting anyone else. If you use setuid, it should have its userid and groupid; per below, the game is probably best run setgid, not setuid.

   All score files should be writable only by this group. Depending on the game needs, score files may be restricted to being readable only by that group.

   The game administrator needs to have the ability to overwrite the game program, and probably to modify the score files and and other administrative files. This suggests that the administrator should be in the game's group and in the group that owns the programs; those in turn should be group-writable.

   If there are some resources accessible only by players, either all players can be in a some group, or there should be some game program—setgid to the game's group—that reads those resources.

   There are three ways to handle updates to the game. First, the game can auto-install those. Second, the game administrator can download them and manually install them. Third, the game company can do it directly. In the first two cases, the updates should be digitally signed. In the third situation, the game company should be in the appropriate group to overwrite those files.