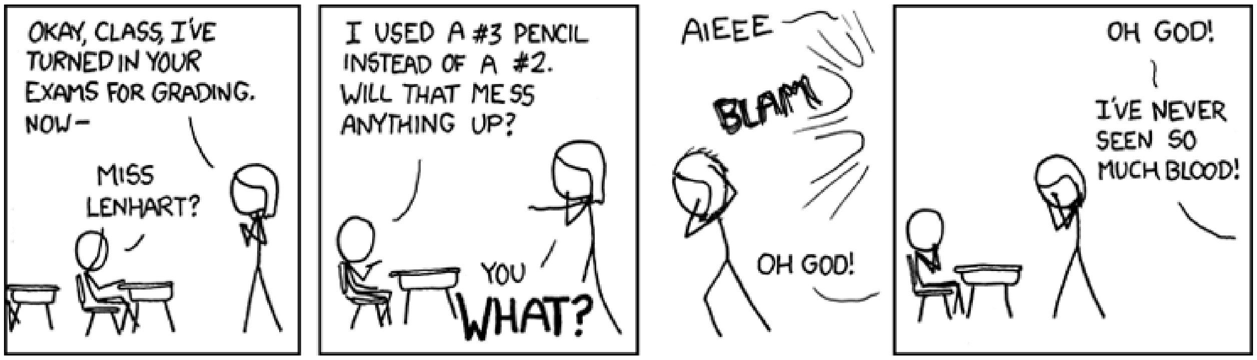
Name: ______________________________ UNI: _____________

COMS W4187: Security Architecture and Engineering
October 2015

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 65.
- Good luck, and may the Force be with you.

Question	Points	Score
1	15	
2	20	
3	15	
4	15	
Total:	65	


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. (15 points) As discussed in class, storage of private keys is a difficult challenge. Someone suggests using a database. Specifically, all user keys are stored in a database; the database is on a separate machine with no user accounts. To retrieve a key, users log into the database via an encrypted connection. Database access control lists restrict users to retrieving their own keys.

   Is this a good scheme or a bad one? Explain.

   Answer:

   It's both good and bad.

   Bad: the keys are stored unencrypted. If someone hacks the machine, everyone's keys are exposed. Alternately, there may be errors in the ACLs.

   Good: this is really just about what a key distribution center is, and with no user accounts the machine is more secure than an ordinary system.

   Not mentioned in class or the question: the machine needs to be "hardened": all unnecessary services turned off.

   There are availability issues—and installing a backup database requires care.

2. There is a start-up that thinks there's money to be made from people who have lost the keys to their house. Briefly, you upload a picture of your house key to their server; when you need a replacement, you go to one of their kiosks and it will produce one for you. (This much is true, I might add.)

   (a) (5 points) What are the security risks that this service must defend against?

   Answer:

   The biggest problem is authenticating the customers who want keys retrieved. Passwords? They're too weak. Use a credit card as a token? A stolen wallet or purse will have the credit card, driver's license, and the owner's address—and that's a weakness.

   How do you know the uploader is the owner of the house?

   Someone can eavesdrop on the link.

   In addition: what if the central site is hacked?

   Lesser risk: hack the kiosk.

   (b) (15 points) Design an authentication system for the service. Justify the design, given the threats you've just pointed out.

   Answer:

   Two-factor authentication is clearly needed. A credit card is a good choice for one factor, though don't forget the common loss scenario; so is a text message to a phone. A password is not a bad idea for the second factor.

   Credit cards prevent guessing attacks against passwords. A phone protects against a hacked kiosk; the challenge will be different each time.

   Use an encrypted connection, of course.

   This is a good spot for encrypting the file with the image—and if someone forgets their password, they get to call a locksmith.

   Don't store addresses—why make it easy for someone who hacks the database?

Do take a picture of someone who wants a key printed; the police might be interested if there's a burglary...

An answer no one suggested: use an app to upload the picture of the key; have the app upload location information as well. Then insist on a driver's license or some such when retrieving a key.

Putting humans into the loop is far too expensive—someone could just call a locksmith.

All you can do about the central site is strongly defend it, but the details are more for 4180 than for this class.

See `http://www.wired.com/2014/07/keyme-let-me-break-in/` and `http://dl.acm.org/citation.cfm?id=1455830`.

3. (15 points) In class, we discussed the difficulties and dangers of `setuid` programs. To deal with them, I propose a new privilege: `setuid` execution. Users have to have that privilege to execute a `setuid` program. Is this a useful defense? Explain.

Answer:

This doesn't really do very much. The risk is to the owner of the setuid program (or rather, the resources accessible to the owner); if the program is poorly written, those can be accessed improperly. Besides, many people need to use *some* `setuid` program; this scheme would let them use all of them.

4. (15 points) Suppose, in an effort to prevent cheating, a university decides to use mandatory access control. That is, modifies the OS of every machine on campus to have faculty-set access control lists on student files; these ACLs prevent students from sharing their files with any other students.

   Clearly, this won't work. Explain some of the problems with the scheme.

Answer:

There are so many other ways to cheat. Most simply, students could use their own laptops. They could also email files around, upload them to web sites, copy them to flash drives, develop them in subdirectories of `/tmp`, etc. *Maybe* some of these schemes could be blocked by tight ACLs, though I doubt it, but there's no way to stop people from using their own computers.