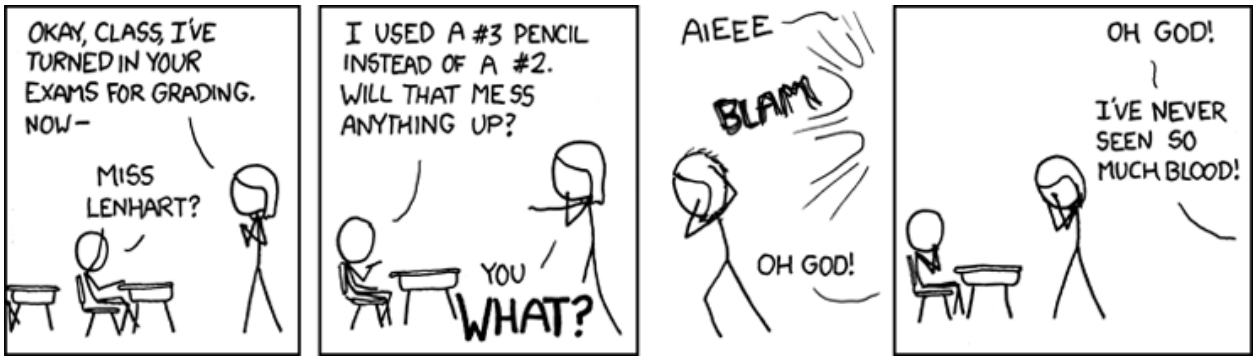
Name: _____ UNI: _____

COMS W4187: Security Architecture and Engineering
October 2016

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	



1. (15 points) In class (October 3), we discussed HSMs—Hardware Security Modules—as a way to securely store cryptographic keys. Could we produce a secure system by putting the kernel in an HSM? Explain.

Answer:

In a word, no.

HSMs are supposed to provide both physical and software security. However, the attacks we see against our systems are based on software, not physical attacks. If we don't do anything to change the interface to the kernel—which today is via system calls—we'll see the same sorts of problems.

2. (20 points) New York's Citi Bike bike-sharing system uses a token system for identification and authentication of annual members: you insert a "key" into a slot next to the bike you want. Explain the advantages and disadvantages of using passwords instead. (Note: I'm talking *only* about annual members, not the pay-as-you-go option.)

Answer:

Note that to get full credit, you needed some part of the answer that was specific to this scenario. That is, a generic answer about passwords versus tokens is not adequate.

The big advantage of passwords is that you don't have to have a physical object with you when you want to ride. This is important, because one use case is spur-of-the-moment decisions to ride: you've walked farther than you'd planned, or the subway is experiencing delays, etc. The second big advantage is that they don't have to cope with replacing lost physical items. Finally, the scheme as described is single-factor authentication: if a key is lost or stolen, whoever finds it can take a bike.

On the other hand, passwords present serious security and usability problems. You have to have user names, too; otherwise, you force each person to have a unique password. That latter is *guaranteed* to result in lots of forgotten passwords, with all of the expense of recovery; there's also the risk of spoofed recovery questions, and the bikes are expensive. Beyond that, people would have to type long strings into a kiosk to get a bike. It's a lot slower than simply inserting a key. Given that you're entering this password in a public place, there's also the risk of "shoulder-surfing".

(For details on how the keys work, see `http://tomorrow-lab.com/lab25`.)

3. (15 points) Someone claims that the best way to make a file unreadable by anyone else is to use a pair of privileged programs to store and retrieve the files. That is, to protect a file you pass it to, say, `secure_store`; to recover it, you invoke `secure_retrieve`. The `secure_retrieve` program will only let the original user (as determined by UID) recover a stored file. Is this scheme better or worse than just using the OS permission mechanisms? Explain.

Answer:

This is much worse.

If you rely on file system permissions, the attacker either have to obtain full system privileges (e.g., `root`) or crack a particular user's account. In the latter case, only that user's files are at risk.

By contrast, with `secure_store` and `secure_retrieve`, there are the same risks from cracking root or cracking a given user's files. In addition, the attacker can go after whatever UID those programs run as—remember, they're privileged. If that account is cracked, all files are exposed.

It is also possible that these two programs are less secure than the kernel.

4. (15 points) Because of the dangers posed by certain characters if they get anywhere near a shell, a certain web site decides to restrict the characters in passwords to only letters and digits. Are they right or wrong? Explain.

Answer:

This is wrong.

As explained in class, the proper way to store passwords is to salt and hash them. There's no reason to invoke a shell to do that. Furthermore, restricting the character set that way makes life easier for an attacker: it drastically reduces the search space.