



Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2011

Quiz II

You have 80 minutes to answer the questions in this quiz. In order to receive credit you must answer the question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name on this cover sheet.

THIS IS AN OPEN BOOK, OPEN NOTES EXAM.

Please do not write in the boxes below.

| I (xx/16) | II (xx/12) | III (xx/8) | IV (xx/8) | V (xx/8) |
|-------------------|--------------------|---------------------|------------------|-----------------------|
| | | | | |
| VI (xx/22) | VII (xx/10) | VIII (xx/10) | IX (xx/6) | Total (xx/100) |
| | | | | |

Name:

Username from handin site:

I Android

To help people keep track of todo items on their Android phone, Ben Bitdiddle writes *Ben's Todo Manager*. Ben wants to allow other applications to access todo items stored by his todo manager, so he implements a public *content provider* component that stores all of the todo items. To protect the todo items, Ben's application defines two new permissions, `com.bitdiddle.todo.read` and `com.bitdiddle.todo.write`, which are meant to allow other applications to read and modify todo items, respectively. Ben also sets the read and write labels of his todo content provider to these two permissions, respectively.

1. [4 points]: What permission type (“normal”, “dangerous”, or “signature”) should Ben choose for the two new permissions declared by his application? Explain why.

2. [4 points]: Ben decides to implement a notification feature for todo items: a day before a todo item is due, Ben's application sends a broadcast intent containing the todo item, notifying other applications so that they may tell Ben to start working on that todo item. Ben sends the broadcast intent by using the `sendBroadcast(intent)` function. Explain what security problem Ben may have created by doing so, and specifically how he should fix it.

3. [8 points]: Ben discovers that Android does not control which application declares which permission name. In particular, this means that another malicious application, installed before Ben's Todo Manager, could have already declared a permission by the name of `com.bitdiddle.todo.read`, and this will not prevent Ben's Todo Manager application from being installed. Explain the specific steps an adversary could take to attack Ben's application given this weakness.

II BitLocker

Recall that a trusted platform module (TPM) contains several platform configuration registers (PCRs). The `extend(n, v)` operation updates the value of PCR register n by concatenating the old value of that PCR register (call it x) with provided data, v , and hashing the result, i.e.,

$$x' = H(x||v)$$

where H is SHA-1, $||$ is the concatenation operator, and x' is the new value of PCR register n .

Suppose that H were instead an insecure hash function that admitted a *preimage attack*, that is, given some value a it is easy to find another value $b \neq a$ for which $H(a) = H(b)$, and with high probability, it's easy to find such a b that starts with a specific prefix.

4. [6 points]: Could an attacker who has stolen a computer defeat BitLocker protection on its hard drive, with high probability? Explain how, or argue why not.

5. [6 points]: Could an attacker who has stolen the hard drive, but not the computer, defeat BitLocker protection on that drive, with high probability? Explain how, or argue why not.

III Side channel attacks

Ben Bitdiddle wants to secure his SSL server against RSA timing attacks, but does not want to use RSA blinding because of its overhead. Instead, Ben considers the following two schemes. For each of the schemes, determine whether the scheme protects Ben's server against timing attacks, and explain your reasoning.

6. [4 points]: Ben proposes to batch multiple RSA decryptions, from different connections, and have his server respond only after all the decryptions are done.

7. [4 points]: Ben proposes to have the server thread sleep for a (bounded) random amount of time after a decryption, before sending the response. Other server threads could perform computation while this thread is asleep.

IV Tor and Privacy

8. [4 points]: An “Occupy Northbridge” protestor has set up a Twitter account to broadcast messages under an assumed name. In order to remain anonymous, he decides to use Tor to log into the account. He installs Tor on his computer (from a trusted source) and enables it, launches Firefox, types in `www.twitter.com` into his browser, and proceeds to log in.

What adversaries may be able to now compromise the protestor in some way as a result of him using Tor? Ignore security bugs in the Tor client itself.

9. [4 points]: The protestor now uses the same Firefox browser to connect to another web site that hosts a discussion forum, also via Tor (but only after building a fresh Tor circuit). His goal is to ensure that Twitter and the forum cannot collude to determine that the same person accessed Twitter and the forum. To avoid third-party tracking, he deletes all cookies, HTML5 client-side storage, history, etc. from his browser between visits to different sites. How could an adversary correlate his original visit to Twitter and his visit to the forum, assuming no software bugs, and a large volume of other traffic to both sites?

V Security economics

10. [8 points]: Which of the following are true?

- A. **True / False** To understand how spammers charge customers' credit cards, the authors of the paper we read in lecture (Levchenko et al) had to collaborate with one of the credit card association networks (e.g., Visa and MasterCard).
- B. **True / False** The authors of the paper (Levchenko et al) expect it would be costly for a spammer to switch acquiring banks (for credit card processing), if the spammer's current bank was convinced to stop doing business with the spammer.
- C. **True / False** The authors of the paper (Levchenko et al) expect it would be costly for a spammer to switch registrars (for registering domains for click support), if the spammer's current registrar was convinced to stop doing business with the spammer.
- D. **True / False** If mail servers required the sending machine to solve a CAPTCHA for each email message sent, spammers would find it prohibitively expensive to advertise their products via email.

VI Trusted hardware

11. [8 points]: Ben Bitdiddle decides to manufacture his own TrInc trinkets. Each one of Ben's trinkets is a small computer in itself, consisting of a processor, DRAM memory, a TPM chip, a hard drive, and a USB port for connecting to the user's machine.

To make his trinket tamper-proof, Ben relies on the TPM chip. Ben's trinket uses the TPM to seal (i.e., encrypt) the entire trinket state (shown in Figure 1 in the TrInc paper) under the PCR value corresponding to Ben's trinket software. The TPM will only unseal (i.e., decrypt) this state (including counter values and K_{priv}) if the processor was initially loaded with Ben's software. When the trinket is powered off, the sealed state is stored on the trinket's hard drive.

Ben's simplified trinket does not implement the symmetric key optimization from TrInc, and does not implement the crash-recovery FIFO Q .

Assume Ben's software perfectly implements the above design (i.e., no bugs such as memory errors), and that the TPM, processor, and DRAM memory are tamper-proof.

How can an adversary break Ben's trinket in a way that violates the security guarantees that a trinket is supposed to provide?

Alice works for a bank that wants to implement an electronic currency system. The goal of the electronic currency system is to allow users to exchange *coins*. There is exactly one type of coin, worth one unit of currency. Alice's bank maintains one server that initially hands out coins. The system should allow user *A* to give user *B* a coin even if the two users are disconnected from the rest of the world (i.e., cannot talk to the bank or to any previous holders of that coin). Furthermore, it should be possible for user *B* to now give a coin to user *C* without having to contact anyone else. It should be impossible for user *A* to "double-spend", that is, to give the same coin to two different users.

12. [14 points]: Design an electronic currency system assuming each user has a TrInc trinket. Assume each trinket's public key is signed by the bank, that everyone knows the bank's public key, and that all trinkets are tamper-proof and trustworthy.

Explain three aspects of your design:

- What is the representation of a coin that a user has to store?
(It's OK if this representation is not constant size.)
- How does user *A* send a coin to user *B*?
- What should user *B* do to verify that it has received a legitimate coin?

VII Usability

13. [10 points]: Alice's bank gives up on the trinket idea as being too costly. Instead, Alice is now designing a banking application for Android. She is worried that users of her banking application may be tricked into entering their bank account information into another look-alike application, because there's no reliable way for a user to tell what application he or she may be interacting with.

For example, there's no way for a user to look at the screen and tell what application is currently running. Even if a user initially runs on a legitimate banking application, a malicious application can start an activity right after that, and display an identical screen to the user. Finally, applications can use full-screen mode to completely replace the entire Android UI.

Propose a design in which users can safely enter their credentials into a banking application. Your proposed design can involve changes to the Android system itself. Unmodified existing Android applications *must* continue to work in your new design (though if you change the UI as part of your design, it's OK if the applications look slightly different as a result). It's fine to require sensitive applications (e.g., Alice's new banking application) to do things differently in your design.

VIII Zoobar

14. [4 points]: After turning in lab 5, Ben Bitdiddle remarks that it is strange that the browser allowed him to call the lab e-mail script in an `` tag, and suggests that browsers should protect against this attack by refusing to load images from URLs that contain query strings. If Ben's proposal is implemented, can an adversary still use `` tags to email user cookies after exploiting a cross-site scripting bug?

15. [6 points]: Ben is working on a Javascript sandboxing system similar to FBJS and lab 6, and he is worried that bugs in the Javascript parsing library that is used to rewrite code (e.g., Slimit) can make his sandbox insecure. Suppose that Ben's Javascript parsing library has some bug in the code that *parses* Javascript code into an AST. Can an adversary exploit such a bug to subvert the sandbox? Give a sketch of how, or explain why not.

IX 6.858

We'd like to hear your opinions about 6.858 to help us improve the class for future years. Please answer the following questions. (Any answer, except no answer, will receive full credit.)

16. [2 points]: What other topics would you have wanted to learn about, either in lectures or in labs?

17. [2 points]: What is your favorite paper from 6.858, which we should keep in future years?

18. [2 points]: What is your least favorite paper, which we should get rid of in the future?

End of Quiz