Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Spring 2022

Quiz I

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 and submission website email address on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

This quiz is printed double-sided.

Please do not write in the boxes below.

I (xx/8)	II (xx/9)	III (xx/9)	IV (xx/8)	V (xx/15)	VI (xx/10)	VII (xx/9)	VIII (xx/10)	IX (xx/2)	Total (xx/80)

Name:

Submission website email address:

You can answer the feedback questions on the back of the quiz before the official start time.

This page intentionally left blank.

I Google security architecture

1. [8 points]: Suppose an adversary compromises the storage service responsible for storing the Gmail data for some Google user (call them X), meaning that the adversary's code is already running on that storage service. In which of the following scenarios would the adversary be able to obtain X's Gmail messages?

Circle True or False for each choice. We will give 2 points for a correct answer, 1 point for no answer, and 0 points for the wrong answer.

A. True / False If the inter-service communication configuration allows the adversary to issue RPCs to the compromised storage service.

B. True / False If user X logs into their Google account and accesses another service (such as Google Calendar).

C. True / False If user X logs into their Google account and accesses their Gmail service.

D. True / False If the adversary compromises the Gmail service in addition to the storage service (but user X never logs in).

This page intentionally left blank.

II U2F

Consider the U2F protocol as described in "Security Keys: Practical Cryptographic Second Factors for the Modern Web." Ben Bitdiddle wants to optimize U2F by not requiring the server to send a randomly-generated challenge during the authentication phase. Instead, in Ben's modified U2F protocol, the client generates its own challenge, uses it in the same way as regular U2F, and sends it to the server along with the signature. The server then uses the challenge sent by the client for verification. Assume that Ben makes no other changes to the server except for using the client-supplied challenge.

2. [9 points]: Describe an attack that an adversary could perform, within the U2F threat model, against Ben's modified U2F which is prevented in the original U2F design. Be specific: describe what the attacker does, what steps they take, and what they achieve.

This page intentionally left blank.

III Baggy bounds

Consider the 32-bit version of Baggy Bounds as described in the paper "Baggy Bounds Checking: An Efficient and Backwards-Compatible Defense against Out-of-Bounds Errors" by Akritidis et al (plus the errata). Assume a slot size of *256* bytes (instead of 16 as in the paper).

3. [9 points]: An application runs the following code:

```
1 char *q = malloc(130);
2 char *r = q + 300;
3 char *s = r - 100;
4 char a = *s;
5 *r = a;
6 char *n = NULL;
7 *n = a;
```

Assume that the call to `malloc` succeeded. What line will the program crash on, and why?

This page intentionally left blank.

IV OKWS

Ben Bitdiddle is using OKWS to build his web application, but he is writing all of his services using Rust, a memory-safe programming language that does not allow programmers to make mistakes that lead to buffer overflows. (He avoids any "unsafe" code in his Rust applications, which otherwise could have allowed programmer mistakes to lead to memory corruption.) As a result, he decides he doesn't need to run the services separate from one another: he runs all services in one process, with one connection to the database.

4. [8 points]: Does Ben's choice of running all services together in one process reduce security, given that all of his service code is written in Rust? Explain why the split-service design is no longer needed, or explain what attacks a split-service design might still prevent. Be specific.

This page intentionally left blank.

V Firecracker

Consider the paper "Firecracker: Lightweight Virtualization for Serverless Applications." For the following scenarios, describe what security goals of the AWS Lambda service running Firecracker an adversary might be able to violate (and how they might be able to do that), or explain why the scenario does not give the adversary the ability to do any additional damage. For each question, assume that this is the only exploitable bug that the adversary knows about.

Circle at most one of "Can violate" or "Cannot violate", and explain.

5. [5 points]: The adversary finds an exploitable buffer overflow in the Linux kernel implementation of the `read()` system call.

Can violate / Cannot violate

6. [5 points]: The adversary finds an exploitable buffer overflow in how the virtio network device implementation in the Firecracker process (say, in the unsafe portions of the Rust code) handles the guest-VM-accessible virtual device memory.

Can violate / Cannot violate

7. [5 points]: The adversary finds an exploitable buffer overflow in the Linux KVM handler for VM traps.

Can violate / Cannot violate

This page intentionally left blank.

VI WebAssembly

Consider the paper "Bringing the Web up to Speed with WebAssembly."

8. [10 points]: Alyssa P. Hacker extends WebAssembly to support de-allocating memory, by adding a `shrink_memory` *n* instruction that reduces the allocated memory size by *n* bytes, the opposite of `grow_memory` *n*. How can this change lead to a security problem in a JIT-based WebAssembly runtime that was secure before this change? Assume the runtime executes just one WebAssembly program, once.

This page intentionally left blank.

VII iOS security

9. [9 points]: Suppose that Apple suddenly discovers that many iPhones manufactured since 2020 were assigned the same ECID. An adversary physically steals someone's iPhone, with the same ECID as the attacker's phone, and wants to break into it by installing an older version of iOS, from 2015, which has an exploitable vulnerability. Explain how the adversary can perform this attack, or explain why it's not possible.

This page intentionally left blank.

VIII EXE

Consider the following code fragment in C in which `stepN` is a symbolic variable, running in the EXE system as described in the paper "EXE: Automatically generating inputs of death".

```
1  char buf[10];
2
3  int f() {
4    int stepN;
5    int sum = 0;
6    int i;
7
8    markSymbolic(&stepN);
9
10   for (i = 0; i < 10/stepN; i++) {
11     sum += buf[i*stepN];
12   }
13   return sum;
14 }
```

10. [10 points]: How many execution paths does EXE explore when executing `f()`?

IX 6.858

We'd like to hear your opinions about 6.858. Any answer, except no answer, will receive full credit.

11. [2 points]: Is there one paper out of the ones we have covered so far in 6.858 that you think we should definitely remove next year? If not, feel free to say that.

End of Quiz