Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Spring 2022

Quiz II

You have 120 minutes to answer the questions in this quiz. In order to receive credit you must answer each 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 COMMUNICATION OR COLLABORATION DURING THE QUIZ.

This quiz is printed double-sided.

Please do not write in the boxes below.

I (xx/16)	II (xx/12)	III (xx/24)	IV (xx/12)	V (xx/32)	VI (xx/10)	VII (xx/10)	VIII (xx/4)	Total (xx/120)

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 Web security

Ben Bitdiddle has an account at BensBank, `https://bensbank.com/`. Like most web sites, BensBank uses cookies to track the sessions of logged-in users. However, BensBank developers forgot to set the `Secure` attribute on the session cookie for `bensbank.com`.

1. [8 points]: How can a network adversary gain access to Ben's account? Describe what a network adversary would need to do, and what they would need to trick Ben into doing. Assume that Ben will not type his password into any site other than `https://bensbank.com/`, and will not type any Javascript code into his browser; assume there are no bugs in the BensBank web application or in the web browser.

This page intentionally left blank.

Alyssa P. Hacker is developing a web application hosted at `https://alyssa.com/`. She uses a Javascript spell-checking library from her friend Carl, by including the following HTML tag in the `https://alyssa.com/` web page:

```
<SCRIPT SRC="https://carls-code.com/spell.js"></SCRIPT>
```

Unfortunately, an adversary was able to corrupt the `spell.js` file hosted at `carls-code.com`.

Now, a victim user visits `https://alyssa.com/`.

2. [8 points]: Can the adversary described above obtain this victim's cookies for `alyssa.com`? Explain how or why not.

This page intentionally left blank.

II SSL and TLS

3. [12 points]: Eric uses HTTPS (HTTP over TLS) to download a file with contents F from server S, whose name and public key are well known to everyone. Can Eric provide cryptographic evidence to convince his friend Fred that the server really sent Eric the bytes for F, without having Fred download the file from the server himself? Explain how or why not.

This page intentionally left blank.

III Messaging security

Ben Bitdiddle develops a messaging protocol as follows (assume A is the sender of a message and B is the recipient). B generates an ephemeral public/private key pair (PK$_e$, SK$_e$) and sends $\langle$PK$_e$, Sign(SK$_B$, H(PK$_e$))$\rangle$ to A, where SK$_B$ is B's signing key, and H is a secure hash function (say, SHA-256). When A receives B's message, she checks the signature using Verify(PK$_B$, ...); assume that A knows the correct PK$_B$ corresponding to B's SK$_B$. A then sends B $\langle$Enc(PK$_e$, m), Sign(SK$_A$, H(m))$\rangle$, where *m* is the message she wants to send and SK$_A$ is A's signing key. B then decrypts the message and checks the signature on the hash of *m* (assume that B knows PK$_A$). In short, the protocol is:

- A $\leftarrow$ B: PK$_e$, Sign(SK$_B$, H(PK$_e$))
- A $\rightarrow$ B: Enc(PK$_e$, m), Sign(SK$_A$, H(m))

Ben gets many users to install and use his messaging application. In the following questions, assume an adversary who controls the network, and controls the computer of one of the friends of A and B.

4. [12 points]: Describe how an adversary can violate confidentiality – that is, how can an adversary get the contents of a message sent by A, even if the adversary is not B?

5. [12 points]: Describe how an adversary can violate authenticity – that is, how can an adversary send a message to B as if the message came from A, even if A did not send that message to B?

This page intentionally left blank.

IV Spectre

Ben Bitdiddle is developing a new CPU and is worried about Spectre attacks. He decides to implement the following defense strategy. The CPU tracks all of the memory addresses accessed during speculative instructions (regardless of whether they hit in the cache or not), and if the instruction ends up being aborted (due to mis-speculation), the CPU will evict all of the cache lines for addresses accessed by that instruction from the CPU caches.

6. [12 points]: Describe how an adversary can still mount a Spectre-like attack against a CPU with Ben's defense to learn arbitrary memory contents. Be specific.

This page intentionally left blank.

V Guest lectures

7. [8 points]: According to Max Burkhardt's guest lecture, what is the main attack vector that Figma's security engineers worry about?

(Circle the best choice; we subtract points for incorrect answers.)

A. Exploiting buffer overflow vulnerabilities in Figma's C++ server software.

B. Guessing the password of one of Figma's security engineers.

C. Phishing attacks against Figma's employees.

D. Fraudulent TLS certificates for Figma's domain.

8. [8 points]: Based on Galen Hunt's guest lecture about Microsoft Azure Sphere, which of the following are security benefits for an IoT device manufacturer from using Azure Sphere?

(Circle True or False for each choice; we subtract points for incorrect answers.)

A. True / False Azure Sphere ensures the Linux OS is updated to fix newly discovered security vulnerabilities.

B. True / False Azure Sphere ensures that the device-specific application code does not have buffer overflow vulnerabilities.

C. True / False Azure Sphere manages the private keys that identify the IoT device.

D. True / False Azure Sphere prevents compromised device-specific application code from gaining control of the entire IoT device.

NOTE: There are more questions on the back side of this page.

9. [8 points]: Based on bunnie's guest lecture about hardware security, which of the following are correct statement about the precursor device?

(Circle True or False for each choice; we subtract points for incorrect answers.)

A. True / False Using a physical keyboard avoids the need to trust the firmware of a touch-screen controller.

B. True / False Disassembling the device allows the user to detect supply-chain attacks where an adversary adds a new chip to the motherboard.

C. True / False Using an FPGA makes it impossible for an adversary to interpose on the system's I/O using a thru-silicon via (TSV) physical implant.

D. True / False Using an FPGA guarantees that the RISC-V CPU has not been modified to add any backdoors.

10. [8 points]: Based on Max Krohn's guest lecture about Zoom security, what is the most significant reason for why Zoom does not support end-to-end encryption in Zoom's web-based client?

(Circle the best choice; we subtract points for incorrect answers.)

A. It is difficult to implement cryptographic operations in a web browser using Javascript.

B. It is difficult to prevent a web server from sending different Javascript code to different users.

C. It is difficult to reliably encrypt audio and video in a web browser.

D. It is difficult to avoid cross-site scripting vulnerabilities.

VI Lab 3

11. [10 points]: Consider the following code snippet, similar to `check-symex-int.py` from lab 3.

```
def f(x):
    if x > 500:
        return 33
    if x // 7 == 7:
        return 1234
    if x*2 == x+1:
        return 100
    return 40

def test_f():
    i = fuzzy.mk_int('i', 0)
    v = f(i)
    return v

f_results = fuzzy.concolic_execs(test_f, verbose=1)
```

Daniel is working on the lab and has **correctly** implemented everything. The call to `concolic_execs` will loop through the branches and make calls to `concolic_find_input` on several constraints. Circle all such possible constraints, then explain why the uncircled constraints are not explored. (Pay careful attention to the `Not`'s)

A. `And(True, Not((i > 500) == False))`

B. `And(And(True, (i > 500) == False), Not((i/7 == 7) == False))`

C.
```
And(And(And(True, (i > 500) == False), (i/7 == 7) == False),
    Not((i*2 == i + 1) == False))
```

D.
```
And(And(And(True, (i > 500) == False), Not((i/7 == 7) == False)),
    Not((i*2 == i + 1) == False))
```

E.
```
And(And(And(True, Not((i > 500) == False)), Not((i/7 == 7) == False)),
    Not((i*2 == i + 1) == False))
```

This page intentionally left blank.

VII Lab 4

Ben Bitdiddle implements an additional layer of security to protect against the cookie-stealing attacks you implemented in lab 4. Ben uses the `SameSite=strict` attribute of the cookie to prevent the browser from sending this cookie along with cross-site requests. Ben also sets the `path` attribute to the zoobar login page so that the cookie can only be accessed from the login page.

12. [10 points]: Can an attacker still obtain the victim's cookie with these new protections, using techniques from lab 4? Either explain why this is not possible, or describe a specific attack (with HTML and Javacsript) snippets that the attacker could implement.

NOTE: The feedback question is on the back side of this page.

VIII 6.858

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

13. [2 points]: Are there any papers or guest lectures in the second part of the semester that you think we should definitely remove next year? If not, feel free to say that.

14. [2 points]: Are there topics that we didn’t cover this semester that you think 6.858 should cover in future years?

End of Quiz