6.858 Quiz 2

Huy C. Dai

TOTAL POINTS

106 / 120

QUESTION 1

1 8 / 8

✓ **+ 8 pts Correct**

+ 6 pts Tricking Ben to visit `http://bensbank.com/` is missing

+ 6 pts Malicious site with request in form/image

+ 4 pts Knows Secure not being set enables cookies to be sent over `HTTP`.

+ 0 pts Empty / Wrong

QUESTION 2

2 8 / 8

✓ **+ 8 pts Correct**

+ 0 pts Wrong / Empty

+ 4 pts Yes, but no mention of SOP exemption / insufficient explanation

QUESTION 3

3 6 / 12

- 0 pts Correct answer: No, TLS provides deniability.

✓ **- 6 pts Incorrect/incomplete answer, but showed correct understanding of TLS.**

- 12 pts No answer or didn't show understanding of TLS.

QUESTION 4

4 6 / 12

- 0 pts Correct

✓ **- 6 pts Incorrect/incomplete/used different threat model, but showed some reasoning.**

- 12 pts No answer or no reasoning.

QUESTION 5

5 12 / 12

✓ **- 0 pts Correct**

- 6 pts Incorrect/incomplete/used different threat model, but showed some reasoning.

- 12 pts No answer or no reasoning.

QUESTION 6

6 12 / 12

✓ **+ 12 pts Correct**

+ 4 pts It is unclear how the attacker probes the cache before the instruction is aborted (cache line is evicted). To get full points, more details are needed.

+ 0 pts Incorrect

+ 12 pts Correct

+ 2 pts The answer needs to be more specific.

+ 12 pts Correct - the attack solution is possible.

+ 2 pts It is unclear how the attacker can infer the secret in Ben's design.

+ 0 pts No answer or no solution was provided.

+ 8 pts The attacker can initialize the cache state to make the measurement accurate.

+ 8 pts Need some details, e.g., what specific steps the attacker needs to take.

QUESTION 7

7 8 / 8

✓ **+ 8 pts Correct**

+ 2 pts Blank answer

+ 0 pts Wrong answer

QUESTION 8

8 pts

8.1 8A **2 / 2**

✓ **+ 2 pts Correct**

+ 0 pts Incorrect

+ 1 pts Blank answer

8.2 8B **2 / 2**

✓ **+ 2 pts Correct**

+ **0 pts** Incorrect

+ **1 pts** Blank answer

+ **2 pts** OK because of assumption about software updates

8.3 8C 2 / 2

✓ **+ 2 pts Correct**

+ **0 pts** Incorrect

+ **1 pts** Blank answer

8.4 8D 2 / 2

✓ **+ 2 pts Correct**

+ **0 pts** Incorrect

+ **1 pts** Blank answer

QUESTION 9

8 pts

9.1 9A 0 / 2

+ **2 pts** Correct

✓ **+ 0 pts Incorrect**

+ **1 pts** Blank answer

9.2 9B 2 / 2

✓ **+ 2 pts Correct**

+ **0 pts** Incorrect

+ **2 pts** OK

+ **1 pts** Blank answer

9.3 9C 2 / 2

✓ **+ 2 pts Correct**

+ **0 pts** Incorrect

+ **1 pts** Blank answer

9.4 9D 2 / 2

✓ **+ 2 pts Full credit for any answer**

QUESTION 10

10 8 / 8

+ **0 pts** Wrong answer

✓ **+ 8 pts Correct**

+ **2 pts** Blank answer

QUESTION 11

11 10 / 10

✓ **+ 10 pts Correct with reasoning e.g. "returns early", "short circuit", "over-constrained"**

+ **6 pts** Correct answer (A, B, C) but wrong reasoning

+ **3 pts** Some reasoning, but incorrect/incomplete

+ **0 pts** No answer or wrong answer w/ no explanation or wrong explanation

QUESTION 12

12 10 / 10

✓ **+ 10 pts Correct**

+ **7 pts** Mentions attack possible (insufficient explanation)

+ **5 pts** Mentions attack not possible because of same-site constraint

+ **5 pts** Mentions same-site attack (e.g., clickjacking) is possible

+ **0 pts** Incorrect

QUESTION 13

13 2 / 2

✓ **+ 2 pts Correct**

+ **0 pts** No answer.

QUESTION 14

14 2 / 2

✓ **+ 2 pts Correct**

+ **0 pts** No answer

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: Huy Dai

Submission website email address: huydai@mit.edu

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.

An adversary would need to trick Ben into first logging into the website normally (via HTTPS connection) and then have him browse to the HTTP version of the page. Since "Secure" attribute is not set on the login cookie, the cookie identifying Ben's session would be transmitted over HTTP, which the adversary can grab and log in as Ben.

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.

Yes. The adversary can change the spell.js to ~~perform a CSRF~~ retrieve the user's cookie, which it can do b/c scripts are an exception to SOP and thus any 3rd-party code will run as if it is from that ~~domain~~ origin. It can then inject an image to the page with an href link that logs the victim's cookie, upon loading, which we did in lab 4, and sends it back to the adversary.

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.

Yes. Eric can present contents F to Eric along with the MAC(s) that the server S provided in the packet(s) for verifying the integrity of the file contents. ~~Fred can initiate this conn. use the known S's public key to~~ Eric can then present the MAC key used during the connection along w/ the chain of trust showing how that key was transmitted from S to Eric (e.g. over established & secure TLS channel using S's known public key and transmitting the MAC key over that channel). Thus, with just knowing the S's public key, Fred can be sure the rest of the communication, and subsequently file contents F, came from S. by recalculating the MAC using the now-trusted MAC key.

This page intentionally left blank.

III Messaging security

$A \longrightarrow B$ (PK_B, SK_B) (PK_e, SK_e)

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 $\langle PK_e, \text{Sign}(SK_B, H(PK_e))\rangle$ 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 $\langle \text{Enc}(PK_e, m), \text{Sign}(SK_A, H(m))\rangle$, 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 ← B: PK_e, Sign(SK_B, H(PK_e))
- A → 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.

say C

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?

The adversary, call him C, ~~intercepts a conn~~ starts a conversation w/ B. Instead of sending a normal msg, C sends B its secret emphemeral key SK_{CE}, of which B will reply in the response as $\text{Sign}(SK_B, H(SK_{CE}))$. C then talks to A as B, sending: PK_{CE}, $\text{Sign}(SK_B, H(SK_{CE}))$. Thus C can then decrypt any message ~~se~~ A sends to B b/c it knows the secret key SK_{CE}.

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?

We sniff A's network traffic to see when it connects to B and read the ~~publicly known~~ cleartext PK_e. We then initiate our own connection w/ A and send the same PK_e as the emphemeral key to our conversation. Thus when A sends us a message m', it will have the form $\text{Enc}(PK_e, m')$, $\text{Sign}(SK_a, H(m'))$ which we can then forward it on to B, spoofing as A, to get B to think A sent it message m'. Note we still are ~~can only send~~ limited in terms of messages of only what A says to us.

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.

An adversary can still execute a Spectre-like attack by first training a branch predictor to ~~take~~ speculate into take a wrong branch, and instead of causing a load into the CPU cache, ~~a data~~ w/in the wrong branch we can cause a side-effects in other side-channels, specifically the Translation lookaside Buffer (TLB). Thus, even when the CPU evict the cache lines we can still make measurements in the TLB to figure out what ~~byte~~ value was accessed/loaded into memory from the speculative execution, using methods like TLBleed for the TLB.

Final answer

Alternatively, we can at start fill the cache with our own data, train the branch predicate to speculate into taking a wrong branch, ~~wait for the CPU to evict wrongly~~ within the branch tell CPU to load some kernel data into a cache line (where the cache line # corresponds to value of the data), wait for CPU to later evict it, and then iterate over the cache to figure out what line # was evicted to figure out the loaded value.

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 precusor 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. Screens can be scraped.

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. → it's his biggest fear.

D. True / (False) Using an FPGA guarantees that the RISC-V CPU has not been modified to add any backdoors. → Need to compile CPU again to check, ~~Also~~ user

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.

Compromised web server may send JS w/ backdoor code to users that hijack authentication flow.

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))
```

D and E are not explored b/c anytime concolic_force_branch is called to test the other side of a given branch, it will make a new constraint consisting of conditions (0 thru b-1) and !b. So it will never negate (NOT()) on more than one branch condition from the original code, since each of the if statements are set to return a value within their if condition, & so the concolic execution won't encounter a new branch after going into an if statement.

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.

Description of attack:

~~From the login,~~ The attacker can inject code into their profile page via an XSS attack, which opens an iframe to the login page and then injects Javascript code into that page (iframe) to peryorm a CSRF attack where it embeds an image that sends back the user's cookie ~~to the attacker~~.

Code

```
<iframe id="1" src="/zoobar/index.cgi/login"></iframe>
<script>
document.getElementById("1").appendChild("<script>
    [... code to embed "hidden" img w/ href of link
     that contains cookie & send it to attacker....]
    document.getElementById("hidden_img").click();
<script>");
```

Since profile and login page share domain, SOP is bypassed in code injection to iframe. SameSite only prevents CSRF from other domains, and path attribute is bypassed b/c we load login page as an iframe and runs exploit from it.

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.

Max Burkhardt's guest lecture

Bunnie's guest lecture

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

Kerberos!,

SQL injection,

SUNDR

End of Quiz