Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.566 Spring 2026

Quiz II Solutions

Mean TBD Standard deviation TBD

Double-sided quiz: check the back of this page!

I Guest lectures [16 points]

1. [8 points]: Based on the lecture by Russ Cox, which of the following are examples of open-source supply chain attacks?

(Circle True or False for each choice.)

A. True / False An adversary makes a change to a popular open-source web server library, adding a special URL that returns a dump of the server's memory, and that change is included in the next release of that library. **Answer:** True.

B. True / False An adversary takes advantage of a bug in the automated release pipeline of a popular open-source image-processing library to inject code into the released library that sends a copy of the process's memory to the adversary's server over the network. **Answer:** True.

C. True / False An adversary takes advantage of a bug in a common logging library used by many network servers, causing those servers to run shell commands supplied by the adversary. **Answer:** False, the adversary did not alter the supply chain.

D. True / False An adversary takes advantage of a vulnerability in a web browser running on the computer of one of the OpenSSH developers, and uses that to insert a backdoor into the OpenSSH source code. **Answer:** True.

2. [8 points]: Based on the lecture by Colby Morgan, what makes it easier for an adversary to gain access to a typical company through a phishing attack on developers, as opposed to other employees? **(Circle the one best choice.)**

A. Developers are used to sending and receiving email messages from external addresses.

B. Developer workstations make it easier for adversaries to run attack code.

C. Developers are increasingly busy and will miss the warning signs of a phishing attack.

D. Developers use cron jobs that allow an adversary to make their attack persistent.

Answer: B: Developer workstations allow compiling code and running interpreters, bypassing malware defenses that would otherwise help protect computers of other employees.

II Network security [8 points]

Ben Bitdiddle decides to change the TCP stack on his network server: when a SYN packet comes in, the server will choose SN_S as $SN_S = H(secret||SN_C) \bmod 2^{32}$, where *secret* is some random secret chosen at boot time, and $H(\cdot)$ is the SHA-512 cryptographic hash function.

3. [8 points]: Describe how an adversary can spoof a connection (i.e., establish a TCP connection and send data over that connection) to Ben's network server from any IP address. Assume that the adversary cannot monitor any network traffic (other than packets sent to the adversary's own address), and that the adversary's ISP does not implement reverse-path filtering.

Answer: Initiate a connection from the adversary's own IP address with some choice of SN_C, observe the chosen SN_S, now spoof a SYN packet from an arbitrary IP address with the same SN_C, and spoof subsequent ACK and DATA packets with SN_S starting from the same SN_S value as observed in the first connection from the adversary's IP address. The server will choose SN_S the same way for both the first and second connections.

III TLS [8 points]

Ben Bitdiddle is building a chess application, with a single central server that interacts with clients running on each user's computer.

A client sends each chess move from the user to the server as a POST request, where the body of the POST contains the game ID (a globally unique value chosen at random when the game is first created), the player position (representing either the white or black chess pieces), the move number (starting from 0), and the actual move itself (the "from" and "to" locations on the chess board). The server authenticates the client's request, checks that the client is allowed to participate in the specified game as the requested player position, checks that the move number is the next one expected from that player position, and checks that the move itself is valid.

4. [8 points]: Ben proposes that, for performance, he will use TLS 1.3's 0-RTT session resumption to send each move request to the server. Explain what specific security problem would arise for Ben's application, or explain why this is safe.

Answer: TLS 1.3 0-RTT resumption allows an adversary to potentially replay requests, but in this case, replaying moves doesn't cause a problem, because moves are sequentially numbered and the server will reject duplicate move requests (with the same move number).

IV Certificates [10 points]

Let's Encrypt now implements multi-perspective validation: when a user asks for a certificate, mulitple Let's Encrypt validation servers will request the challenge from the user's server, from different vantage points on the Internet.

5. [10 points]: Suppose an adversary, Alyssa P. Hacker, wants to get a TLS certificate from Let's Encrypt for a victim's domain, `bitdiddle.com`. For each of the following scenarios, will Alyssa be able to get a certificate from Let's Encrypt? Assume Let's Encrypt uses multi-perspective validation.

(Circle True or False for each choice.)

A. True / False Alyssa gains access to the authoritative DNS server for `bitdiddle.com`. **Answer:** True, can redirect DNS record for `bitdiddle.com` to Alyssa's own server.

B. True / False Alyssa can see a log of all HTTP requests received by the web server running at `bitdiddle.com`. **Answer:** False, cannot use this log to successfully pass challenge.

C. True / False Alyssa gains access to the network link of one of the Let's Encrypt validation servers (i.e., she can monitor, modify, and inject packets going over that network link). **Answer:** False, cannot use this to successfully pass challenge from other validators.

D. True / False Alyssa finds a buffer-overflow vulnerability in the web server running at `bitdiddle.com` that can be exploited to run arbitrary code. **Answer:** True, can break into the server and use it to successfully respond to ACME challenges.

E. True / False Alyssa discovers that Let's Encrypt generates challenges using a sequential counter: each certificate request gets the next integer value as its challenge. **Answer:** False, cannot get `bitdiddle.com` to respond to challenge request, even if Alyssa can predict what the challenge will be.

V User authentication [8 points]

6. [8 points]: Suppose that Ben Bitdiddle opens the web browser on his computer, goes to his bank web site at `https://bank.com`, and logs into his account, which uses U2F. In which of the following scenarios will the adversary be able to gain access to Ben's account on `https://bank.com`?

(Circle True or False for each choice.)

A. True / False Adversary is able to hijack the DNS queries from Ben's computer, causing Ben's computer to resolve `bank.com` to the adversary's IP address. **Answer:** False: the adversary would not have a TLS certificate for `bank.com`.

B. True / False Adversary tricks Ben into visiting `http://bank.com` instead of `https://bank.com`. **Answer:** False: the browser would not generate a U2F response from the security key for a non-HTTPS origin.

C. True / False Adversary tricks Ben into registering for an account at `https://baank.com`, which is controlled by the adversary, including registering Ben's U2F security key. **Answer:** False: any signed messages while intereacting with `baank.com` would have that origin rather than `bank.com`, and moreover, the security key generates different public keys at each registration attempt.

D. True / False Adversary tricks Ben into installing a modified version of the web browser on his computer, supplied by the adversary. **Answer:** True: once Ben logs into his bank account, the modified web browser can forward the resulting session credentials to the adversary.

VI Decentralized key management [8 points]

Answer the following question based on the lecture and paper by Max Krohn on FOKS.

Suppose we modify the FOKS client so that, when a user deletes a device, the client posts a new entry to the user's sigchain, containing the deletion operation, but keeps the same PUK.

7. [8 points]: Describe a scenario in which this change leads to a security problem. Be specific about the steps that must take place for your attack to succeed, and what the adversary is able to achieve.

Answer: The victim user loses their phone, and revokes that device in FOKS. A new document is then shared with the victim user. An adversary later gains access to the victim's lost phone. With the modified FOKS design, the adversary can get the PUK from the lost phone, which is still the user's current PUK, and use that to decrypt the new document. With the original FOKS design, the new document would be accessible only to the new PUK, which would not be accessible to the lost phone.

VII AI agent security [8 points]

Answer the following questions based on the lecture by Anish Athalye on CaMeL.

Recall the dual LLM example from lecture, where the user issues the following prompt to their agent:

> *How many papers has Anish written? You can find the list here: https://x.anish.io/publications*

and the agent uses the dual LLM pattern to generate code such as:

```
contents = fetch("https://x.anish.io/publications")
count = query_quarantined_llm("How many papers are included here: " + contents)
return count
```

8. [8 points]: What ensures that *prompt injection attacks* against the quarantined LLM cannot cause damage?

(Circle the one best choice.)

A. `query_quarantined_llm` must be executed in a sandbox, such as a container.

B. `query_quarantined_llm` must use a model that does not support injection of commands.

C. `query_quarantined_llm` must not implement an agent loop.

D. `query_quarantined_llm` must inform the model that it is quarantined.

Answer: C.

VIII Messaging security [8 points]

Answer the following question based on the lecture and paper by Cohn-Gordon et al on the analysis of the Signal messaging protocol. The names of keys in the question below are based on the paper's notation.

9. [8 points]: Alice plans to start a conversation with Bob via Signal, and send him a secret message. Alice and Bob's Signal clients know each other's long-term public keys (ipk^A and ipk^B).

Suppose the adversary discovers that Bob's Signal client writes $prek^B$ (the medium-term private key) to a log file that's stored privately on Bob's device (but no other private keys are logged). Suppose the adversary also controls the Signal server itself. The adversary expects that they will be able to compromise Bob's device sometime in the next year. What should the adversary do so that, when they compromise Bob's device, they will be able to learn Alice's secret message?

Assume there are no other software bugs. The adversary cannot modify the Signal client software running on Alice and Bob's devices, or trick Alice or Bob into performing any actions on their devices.

Answer: The server should avoid sending $eprek^B$ to Alice's device when Alice asks for Bob's prekey bundle, so that the root key can be derived just from knowing $prek^B$ and ik^B, which the adversary will learn upon compromising Bob's device.

IX Anonymous communication [8 points]

Ben Bitdiddle is disappointed that Tor hidden services require going through 6 onion relay hops (3 chosen by the client and 3 chosen by the server). Ben's idea to fix this is to have the client prove to the server that it picked its 3 relays at random. The way this works is that the client picks a random integer r, and chooses the first relay in the circuit as number $H(r) \bmod N$ from the directory's list of N total relays, second relay as $H(r+1) \bmod N$, and the third relay as $H(r+2) \bmod N$. The client then sends its rendezvous point address, its 3 chosen relays, and the random value r, to the introduction point of the hidden service. The server can now check that the chosen relays are indeed chosen based on the random value r, and connect to the client's circuit at $H(r+2) \bmod N$ without having to build its own circuit of 3 relays.

10. [8 points]: How does Ben's proposed fix allow an adversary obtain the IP address of the computer running a hidden service, with the adversary sending just one request to that hidden service? Assume the adversary's capabilities are the same as what the Tor paper targets.

Answer: Set up an OR controlled by the adversary, iterate over random values r until finding one where $H(r+2) \bmod N$ matches the adversary's OR, set up an RP on that OR and send a request to the introduction point of the hidden service. The computer running the hidden service will connect to the adversary's OR asking to use the RP address.

X 6.566 [8 points]

We'd like to hear your opinions about 6.566. Any answer, except no answer, will receive full credit.

11. [4 points]: Out of the papers and guest lectures that we have covered in the second half of the semester, listed below, mark the one that you think we should remove next year (or mark none if you think all papers and guest lectures should stay).

- ☐ Supply chain security: guest lecture by Russ Cox.
- ☐ TCP/IP network security.
- ☐ Secure channels: TLS.
- ☐ Certificates: Let's Encrypt.
- ☐ User authentication: U2F.
- ☐ Decentralized key management: FOKS, guest lecture by Max Krohn.
- ☐ AI agent security: CaMeL, guest lecture by Anish Athalye.
- ☐ Messaging security: Signal.
- ☐ Information security in real life: guest lecture by Colby Morgan.
- ☐ Anonymous communication: Tor.

Answer: TBD.

12. [4 points]: What new lab assignment should we add next year?

Answer: TBD.

End of Quiz