CS155: Computer Security **Spring 2024**

Final Exam

Instructions:

- **Please answer all six questions. You have three hours.**
- You may take the exam at any time during the exam window. You have three hours from the moment you begin until the moment you submit your answers on Gradescope.
- The exam is open book, open notes, and open laptops. However, you are expected to do the exam on your own. You may not interact, collaborate, or discuss the exam with another person during the exam window. **You may not use an AI tool or a search engine.**
- To submit your answers please either (i) use the provided LaTeX template, or (ii) print out the exam and write your answers in the provided spaces, or (iii) write your answers on blank sheets of paper, but please **make sure to start each question on a new page.** When done, please upload your solutions to Gradescope (course code `5JE2KR`).
- The LaTeX template for the final is available here. Please do not share the link with others.
- If you have questions, please post them privately on Ed and we will answer them as quickly as we can.
- Students are bound by the Stanford honor code. In particular, you are expected to do the exam on your own and without use of an AI tool or a search engine.

1. (*10 points*) .... True or False

For each question, please write `T` or `F` in the space provided. No explanation needed.

______ (a) Control hijacking attacks are possible only when there is an overflow of a buffer located on the stack.

______ (b) It is fundamentally harder to create a program analysis tool with a low rate of false positives than a low rate of false negatives.

______ (c) Fuzzing is a bug-finding technique that finds all security vulnerabilities in the system tested, without requiring access to source code.

______ (d) Web servers can instruct browsers to store a given cookie for many years.

______ (e) Client-side validation of form data is as secure as server-validation; the only difference is that it happens on the client's machine.

______ (f) Implementing your web application using prepared statements is effective against SQL injection attacks.

______ (g) Bot nets can be used for denial of service attacks.

______ (h) The boot ROM on an iPhone will only boot an OS boot loader that is signed by Apple.

______ (i) TCP-based protocols are easier to abuse to mount reflection-based denial of service attacks than UDP-based protocols.

______ (j) Deploying SMS-based two factor authentication (2FA) is a strong protection against phishing attacks on users.

2. (*20 points*) .... Questions from all over with a short answer

(a) (*4 points*) You are setting up a home computer for a friend and you want to provide some firewall protection. One option is a small hardware device that operates as a stand-alone firewall. Another is firewall software that runs on your friend's computer. What are the relative advantages of each?

Your answer:

(b) (*4 points*) In the mobile platform security lecture we discussed how AirTags periodically transmit a unique identifier that is received and forwarded to Apple servers by Apple devices near the AirTag. Explain why every AirTag needs to periodically choose a new random identifier. What would go wrong if an AirTag used the same identifier forever?

Your answer:

(c) (*4 points*) Do the IP and TCP checksums provide protection against packet tampering by an active network attacker? Justify your answer.

Your answer:

(d) (*4 points*) Which of the following technologies can help defend against an attack that uses a buffer overflow in the heap: stack canaries, ASLR, the NX bit, prepared SQL? Briefly explain how the technologies you chose help.

Your answer:

(e) (*4 points*) How do Linux capabilities improve system security?

Your answer:

3. (*15 points*) .... Stubborn vulnerabilities

MITRE recently published a list of vulnerabilities that appear over and over every year. Below we list the top five types. For each type, briefly explain (a) what is the vulnerability and (b) how it can be exploited by an attacker.

(a) (*3 points*) Out-of-bounds Write

Your answer:

(b) (*3 points*) Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Your answer:

(c) (*3 points*) Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Your answer:

(d) *(3 points)* Use After Free

Your answer:

(e) *(3 points)* Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

Your answer:

4. (*24 points*) .... Web Security

(a) (*4 points*) How does explicitly setting the `Domain` attribute on a cookie change its scope compared to an implicitly inferred domain?

Your answer:

(b) (*4 points*) How can the website `https://cs.stanford.edu/dabo` steal a cookie set by the website `https://cs.stanford.edu/zakir`? What should you do to prevent this?

Your answer:

(c) (*4 points*) If Gmail set its authentication cookie that keeps you logged in to have the following option `SameSite=Strict`, what experience would break for normal users?

Your answer:

(d) (*4 points*) What makes physical security keys like Yubikeys that implement the U2F protocol secure against phishing attacks?

Your answer:

(e) (*4 points*) How does the Content Security Policy `default-src 'self';` protect against XSS attacks despite allowing Javascript to be loaded from the same domain?

Your answer:

(f) (*4 points*) Does the following Javascript code trigger a pre-flight CORS request? Why or why not?

```
const http = new XMLHttpRequest()
http.open(‘POST’, ‘/login’)
http.setRequestHeader(‘Content-type’, ‘application/json’)
http.send(JSON.stringify({‘login’:‘john’,‘password’:‘123’}))
```

Your answer:

5. (*24 points*) .... Network Security

(a) (*4 points*) What practically prevents (unecrypted) HTTP responses from being spoofed by an off-path attacker who only knows client's and server's IP addresses?

Your answer:

(b) What are two reasons that you should use HTTPS even if the website you're building is static and does not transfer any sensitive data (e.g., no login form)?

Your answer:

(c) *(4 points)* Suppose that a certificate authority is hacked and issues a rogue certificate for `gmail.com`. Explain why this is a problem, and how it is mitigated.

Your answer:

(d) *(4 points)* What information was leaked to a passive network observer during the TLS handshake prior to TLS 1.3? What privacy concern does this pose?

Your answer:

(e) (*4 points*) What attack can be mounted on a local network to intercept and eavesdrop on network traffic that cannot be mounted over the Internet? What can you do if you are building an Internet service to protect against this potential eavesdropping?

Your answer:

(f) (*4 points*) Why is it easy to recover from a BGP hijacking of `124.48.72.0/16` but difficult to recover from a BGP hijacking of `13.23.89.0/24`?

Your answer:

6. (*10 points*) .... Certificates

In class we explained that a certificate is issued by a CA and is used to bind a public key to an entity such as a website. In practice, a CA provides the website with a certificate chain, say $A \to B \to C$, which means that a trusted CA called A issued a certificate to another trusted CA called B; later B issued a certificate to website C. The website presents its certificate chain $(A \to B \to C)$ to the browser. The browser checks that the $A \to B$ certificate and the $B \to C$ certificate are both valid. Then, if it trusts the CA called A, then it also trusts the CA called B, and therefore can trust the public key of C. Here A is called the top-level CA and B is called an intermediate CA.

Let's see a potential problem with this mechanism. John Smith generates a public/private key pair $(\mathrm{sk}, \mathrm{pk})$, and obtains a certificate from a trusted CA binding his public key pk to the domain `johnsmith.com` which he owns. He then generates a second public/private key pair $(\mathrm{sk}', \mathrm{pk}')$, and uses his first private key sk to sign a certificate that binds pk' to the domain `www.amazon.com`. In effect John is acting as a CA.

Now John has a certificate chain for `www.amazon.com`, where `johnsmith.com` plays the role of the intermediate CA. Browsers might incorrectly accepts this fake Amazon certificate because there is a valid certification path to a trusted top-level CA.

(a) (*4 points*) Why does this scenario present an attack? How can it be exploited?

Your answer:

(b) (*6 points*) How would you fix this problem? That is, how can we support certificate chains without enabling the attack from part (a)?

Your answer: