VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

EXAMINATIONS – 2024

TRIMESTER 2

FRONT PAGE

CYBR 271

SECURE CODING
07/11/2024

Time Allowed: TWO HOURS (120 minutes)

Instructions:

- Attempt **ALL** questions in this booklet.
- Only silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.
- Printed foreign to English language dictionaries are permitted.
- Write answers in the spaces provided **in the examination booklet**.
- **Hand in the examination booklet.**

Questions	**Marks**
1. Security Principles	[5]
2. STRIDE Threat Modeling	[10]
3. Risk Assessment and Attack Trees	[10]
4. Security Reviews, Testing, and Supply Chain Attacks	[10]
5. Privilege Escalation	[10]
6. Buffer Overflow and Return-to-libc	[20]
7. Format String	[12]
8. Cross-site Scripting (XSS)	[10]
9. Cross-site Request Forgery (CSRF)	[8]
10. SQL-Injection	[5]

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

1. **Security Principles** **(5 marks)**

Consider a smart home system that allows users to remotely control their home's lighting, temperature, door locks, and security cameras via a mobile app.

(a) **(2 marks)** **Define** the security principle of "least privilege" and give an example of how this principle can be applied in the context of the smart home system.

(b) **(2 marks)** **Define** the security principle of "fail securely" and give an example of how this principle can be applied in the context of the smart home system.

(c) **(1 mark)** **Briefly** explain why applying these security principles is important in the context of smart home systems.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

2. **STRIDE Threat Modeling** **(10 marks)**

Consider an online voting system for a university student election. The system allows students to log in, view candidate information, and cast their votes securely.

(a) **(5 marks)** **Draw** a simple DFD representing the system. It must include at least one of each type of element e.g., data flow, data store, process, interactors and trust boundary.

(b) **(5 marks)** **Apply** the STRIDE model to the online voting system scenario. **Identify** five potential threats, each from a different STRIDE category, and **briefly** explain how they could impact the system.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

3. **Risk Assessment and Attack Trees** **(10 marks)**

Consider a cloud-based file storage service that allows users to upload, store, and share files.

(a) **(5 marks)** **Create** a simple attack tree for an attacker trying to gain unauthorized access to a user's files in the cloud storage service. Include at least **two levels** in your tree and **one** AND node.

(b) **(5 marks)** Pick a threat from your tree and apply **DREAD** to come up with an overall risk score (0-50). Make sure that you explain your reasoning and assumptions.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

4. **Security Reviews, Testing, and Supply Chain Attacks** **(10 marks)**

Consider a software company developing a new e-commerce platform integrating various third-party components for payment processing, inventory management, and customer analytics.

(a) **(5 marks)** **Briefly** explain why manual testing might be more effective than automatic testing for identifying vulnerabilities in the payment processing integration.

(b) **(3 marks)** **Briefly** explain what a supply chain attack is and why it's a significant concern for this e-commerce platform. Provide an example related to one of the platform's components.

(c) **(2 marks)** **Briefly** describe a measure the e-commerce platform developers could implement to mitigate the risks of supply chain attacks.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

5. **Privilege Escalation** **(10 marks)**

Consider a Unix-based system where multiple users have access to various applications and resources.

(a) **(2 marks)** **Briefly** explain what privilege escalation is and why it's a significant security concern.

(b) **(3 marks)** **Briefly** describe the concept of Set-UID programs in Unix systems. What is their purpose, and how do they work?

Consider the following C code snippet for a Set-UID program:

```c
#include <stdlib.h>
#include <stdio.h>
int main(int argc, char *argv[]) {
    char command[256];
    sprintf(command, sizeof(command), "/bin/echo %s", argv[1]);
    system(command);
    return 0;
}
```

(c) **(5 marks)** **Briefly** explain why the Set-UID program shown is vulnerable to privilege escalation, how an attacker might exploit it and how the risk can be mitigated.

SPARE PAGE FOR EXTRA ANSWERS

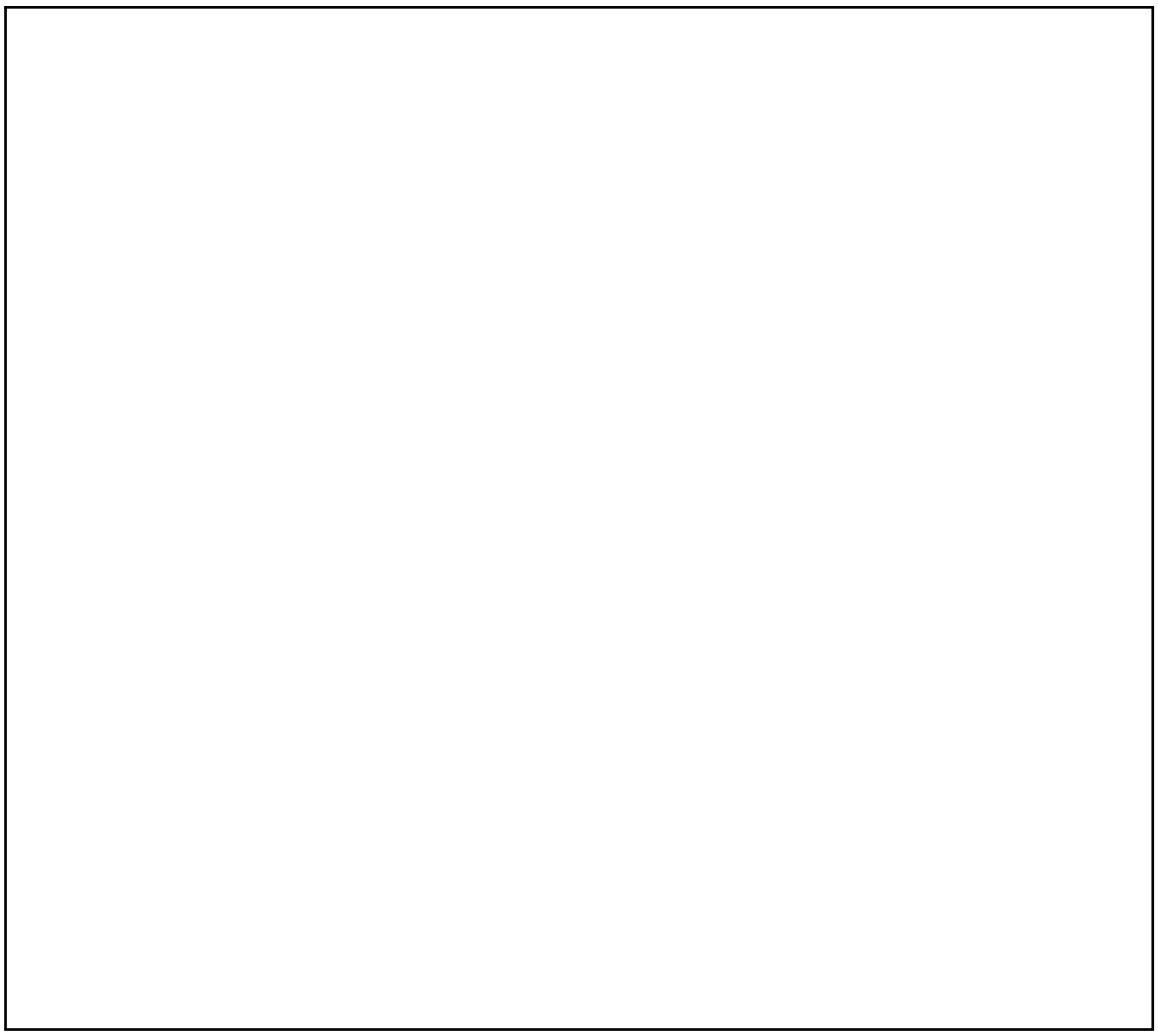
Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

6. **Buffer Overflow and Return-to-libc** **(20 marks)**

***Note**: Unless stated otherwise, assume an Intel x86 32-bit computer system is being used.*

(a) **(6 marks)** Draw the **stack frame** of the `foo()` function assuming line number 5 is being executed and **cdecl** calling convention is used.

```
1  int foo(int a, int b, int c, int d)
2  {
3      static char buffer;
4      int i, j;
5      i = a - d;
6      j = b + i;
7      return j;
8  }
9
10 int bar(char arg)
11 {
12     int result;
13     int str = 5;
14     static int k=5;
15     result = foo(arg, 3, 2, str);
16     return result+k;
17 }
```



(b) **(4 marks)** How many bytes is the stack frame of the function `bar()`? **Briefly** justify your answer by providing a breakdown of the stack frame contents and their respective sizes. You can ignore any alignment requirements in your answer.

(c) **(2 marks)** Calculate the memory address of the "Return Address" field for the `qux()` function given below. Assume the offset between the start of the `buffer` variable and `ebp` is 64 bytes, and `ebp` is pointing to the address `0xbffea190`. Justify your answer by providing a working of the solution.

```
1 int buf(char *str)
2 {
3     int x;
4     char buffer[8];
5     strcpy(buffer, str);
6
7     return 1;
8 }
```

(d) **(2 marks)** **Briefly** describe the role of **canary** as a buffer overflow countermeasure.

(e) **(2 marks)** Which countermeasure does return-to-libc aim to defeat? **Briefly** explain how return-to-libc is able to overcome the said countermeasure.

(f) **(3 marks)** **Briefly** explain the three main tasks in a return-to-libc attack.

(g) **(1 mark)** **Briefly** explain how an attacker can pass a custom string to a program which can then be used in the return-to-libc attack.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

7. **Format String** **(12 marks)**

***Note**: Unless stated otherwise, assume an Intel x86 32-bit computer system is being used.*

(a) **(8 marks)** Consider the following program:

```
#include <stdio.h>

void fmtstr()
{
    char input[100];
    int var = 0xbeefbeef;
    int secret = 0x11223344;
    int i = 0;

    printf("Please enter a string: ");
    fgets(input, sizeof(input) , stdin);
    printf(input);
}

void main ()
{
    fmtstr ();
}
```

Assume that the distance between the variable `i` and `va_list` is `20` bytes.

i. **(2 marks)** **Construct** an input that is guaranteed to crash the program. **Justify** your answer.

ii. **(3 marks)** **Construct** an input that will print out the variable `secret`. **Justify** your answer.

iii. **(3 marks)** **Construct** an input that will change the value of the variable `var` to any value. Assume that `var` is at address `0x41424344`. **Justify** your answer.

Hint: You may use the fact that `0x41` is the ASCII character `'A'`.

(b) **(4 marks)** **Briefly** describe **two** countermeasures that can prevent format string attacks.

8. **Cross-site Scripting (XSS)** **(10 marks)**

(a) **(2 marks)** **Briefly** explain **one** particular *website behavior* that can make it vulnerable to non-persistent XSS attack.

(b) **(2 marks)** **Briefly** explain the **two** approaches that a Javascript code can use to self-propagate.

(c) **(4 marks)** A website (`cybr271.org`) utilises the Content Security Policy (CSP) mechanism to prevent XSS attacks and allows script to be executed based on the following CSP.

Explain the meaning of this CSP.

```
Content-Security-Policy: default-src 'self';
   script-src 'self' 'nonce-3efsdfsdff' victoria.ac.nz';
```

(d) **(2 marks)** The *nonce* and *hashing* mechanisms can be used to allow inline script code to run on a page. **Briefly** discuss **one** advantage of nonce over hashing.

9. **Cross-site Request Forgery (CSRF)** **(8 marks)**

(a) **(2 marks)** A CSRF attack involves three parties: a *victim user*, a *targeted website*, and a *malicious website* that is controlled by an attacker. What important requirement regarding the user session with respect to the targeted website should be satisfied for the attack to be successful? **Briefly** explain what will happen if this condition is not satisfied.

(b) **(3 marks)** **Briefly explain** what the `referer` header is, **how** it can prevent a CSRF attack, and **one** of its drawbacks.

(c) **(3 marks)** **Briefly** explain how same-site cookies can enhance CSRF protection. **Briefly** discuss **one** disadvantage of `SameSite=Strict` attribute value.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

10. **SQL-Injection** **(5 marks)**

(a) **(2 marks)** **Briefly** explain what "prepared statements" are, and the underlying principle that it uses to mitigate SQL-injection attacks.

(b) **(3 marks)** The following SQL statement is sent to the database to add a new user to the database, where the content of the `$name` and `$passwd` variables are provided by the user, but the `EID` and `Balance` field are set by the system.

Assume the malicious user has the following credentials:

- username is "`malice`"
- password is "`abc123`"

The SQL query in the web page is:

```
1  $sql = "INSERT INTO account (Name, EID, Password, Balance)
2          VALUES ('$name', 'EID6000', '$passwd', 0)";
```

Suppose in the new user creation page, it asks for the username and password. What values should a malicious user use to set his/her balance to a particular value, say 1000000?

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

* * * * * * * * * * * * * * *