VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

EXAMINATIONS – 2024

TRIMESTER 2

FRONT PAGE

CYBR 271

SECURE CODING
07/11/2024

***** WITH SOLUTIONS *****

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.

> Provide least privilege definition in the context of the smart home, for example only required access to some systems for specific reasons.

(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.

> Provide fail securely definition in the context of the smart home, for example, in event of a power failure the doors default to locked to prevent attackers accessing the house.

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

> We should recognize that failures and security breaches are inevitable and that these principles will limit the security impact when they occur.

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.

Must use notation from the lectures or label clearly to distinguish each element. Must label all elements including 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.

Each of the STRIDE categories must apply to element found within the diagram above and be applicable to the class for element.

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.

```
The tree must have a clear hierarchy, the attack
paths have to make sense and relate to security
threats relevant to cloud storage and must include
at least one case of two threats must be present to
allow a successful attack.
```

(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.

```
Must relate to a threat found in the tree, must pick
a value for each DREAD component, must explain why
chosen the value, must come up with an overall score
and an overall risk rating.  An excellent answer will
consider an attack path through the tree rather than
a single threat.
```

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.

Must discuss context sensitivity of manual versus automatic testing, the difficulty of testing authorization and authentication, make a case that payment processing will rely upon these two elements because it involves managing money, the difficulty of business logic testing using automatic testing, and the difficulty of testing complex workflows with multiple components using automatic testing.

(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.

Define the concept of a supply chain attack with an emphasis on the fact that it is not only software but services or processes that might be compromised. Any example related to a component.

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

Technical control such as SBOM described and how it could be applied to mitigate the risk of supply chain attacks. Minimal or part marks for answers that are generic risk management approaches than the approaches discussed in class for mitigating supply 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.

```
Define in terms of gaining higher privileges than
should have through exploitation of a vulnerability
in a system.
```

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

```
Describe the mechanism (setuid bit, access control
settings and setting the owner to the ID associated
with the elevated privileges).  Also explain that
the operating system will set a setuid program's
effective user ID to the owner's user ID while the
program is run and restore it to the user's original
ID after the program has completed running.
```

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.

```
Key issue is that the command is injected into the
string "/bin/echo" without any input sanitization.
This means that an arbitrary command can be passed as
the input as long as it is preceded by ";" character
that can be used to put multiple commands on a single
line in UNIX. The SetUID program will execute the
injected command at the privilege level of what ever
user is associated with the running program, for
example in some cases it might be a server program
running as root user thereby providing free access to
the entire 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.

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 }
```

The stack frame drawing should contain the following:
variable d
variable c
variable b
variable a
return address
previous EBP
variables i and j (any order)

(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.

The stack frame of bar() includes: arg (1 byte), return address (4 bytes), previous EBP (4 bytes), result (4 bytes), str (4 bytes). Hence the answer is 1 + 4*4 = 17 bytes.

(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.

```c
int buf(char *str)
{
    int x;
    char buffer[8];
    strcpy(buffer, str);

    return 1;
}
```

0xbffea190 + 4 = 0xbffea194

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

A canary is a small value placed between buffer and control data (e.g. return address). If canary is overwritten, then buffer overflow is detected.

(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.

Return-to-libc aims to defeat the non-executable stack countermeasure. It does so by overwriting the return address with the address of a libc function such as system().

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

The 3 main tasks are:
1) Find the address of the system() function.
2) Find the address of the "/bin/sh" string.
3) Construct arguments for the system() function.

(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.

The string can be passed through an environment variable.

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.

6 %s will guarantee that the program will crash. This is because the sixth %s will use the value of i as address which will surely cause segmentation fault.

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

Need 6 %x to move the pointer and one more %x to printout the value. 6 %x are needed since the distance is 20 and we also need to traverse i before reaching secret, i.e., 20/4 + 1 = 5 + 1 = 6.

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'`.

The input is: DCBA followed by 7 %x and then %n. Need 7 %x to move the pointer and %n to overwrite the value at var.

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

The two approaches are:
1) Developer approach: Avoid using untrusted user inputs for format strings in functions like printf(), sprintf(), fprintf(), etc.
2) Compiler approach: Use compilers that can detect potential format string vulnerabilities (e.g. gcc and clang).
Address randomization can also help by making the attack more difficult.

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.

The website has a *reflective behavior*: it takes an input from a user, conduct some activities, and then sends a response to the user with the original user input included in the response.

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

The two approaches are:
1) DOM approach: JavaScript code can get a copy of itself directly from DOM via DOM APIs
2) Link approach: JavaScript code can be included in a web page via a link using the src attribute of the script tag.

(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’;
```

This CSP permits scripts to be executed if it is from the same domain, i.e., cybr271.org, from victoria.ac.nz, has the nonce “3efsdfsdff”. All other scripts are blocked.

(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.

Although using hashes is a good approach, if anything inside the script tag is changed due to formatting or adding spaces, the hash will be different and the script will not execute.

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.

The user must be logged in. If the user is not logged in, they will be re-directed to another page (login page) which can alert the user.

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

This an HTTP header field identifying the address of the web page from where the request is generated. Using this, a server can check whether a request originated from its own pages or not. Some browsers (or their extensions) and proxies remove this field to protect users' privacy.

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

Same-site cookies help protect against CSRF by ensuring that cookies are only sent in same-origin requests, thus preventing them from being sent in cross-site requests (such as those initiated by an attacker).

If a cookie is set with the SameSite=Strict attribute, browsers will not send it in any cross-site requests. Although this is the most secure option, it can negatively impact the user experience in cases where cross-site functionality is desirable.

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.

Prepared statements are a database query method that pre-compiles SQL queries with placeholders for input values, ensuring that user data is treated as data rather than executable code. They help mitigate SQL injection attacks by ensuring that user-supplied data is treated strictly as data and not executable code.

(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:

```
 $sql = "INSERT INTO account (Name, EID, Password, Balance)
          VALUES ('$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?

The user should provide the correct username and password then add the value for the salary, complete the statement, and comment out the remaining part, that is:
username: malice
password: abc123’, 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.

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