



VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

EXAMINATIONS – 2022(draft)

TRIMESTER 2

CYBR 271

SECURE PROGRAMMING

Time Allowed: TWO HOURS

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

CLOSED BOOK

Permitted materials: 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.

No other material is permitted.

Instructions: Attempt ALL questions.

There are 100 marks total.

Write answers in the spaces provided in the examination booklet.

Hand in the examination booklet.

Questions	Marks
1. Security Principles	[10]
2. Security Threat Modelling	[20]
3. Security Testing	[20]
4. Privilege escalation	[10]
5. Buffer Overflow	[10]
6. Format String	[10]
7. Web security	[20]

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.

SECTION A Set-UID**[20 marks]**

1. Unix uses the “9-bit permission mechanism” and groups users into different groups where access permission can be managed per group. Using the `ls -l` command, you have got the following output. Identify the different groups and their corresponding permissions regarding this file. **(3 marks)**

```
-r-xrwxrwx- 1 john  staff  290 12 Oct 15:03 script
```

- **Owner:** read only.
- **Group:** read, write and execute.
- **Others:** read and write.

2. What values should be used with the `chmod` command to grant the following permissions to the script file? **(3 marks)**

- **Others:** Cannot read, write and execute.
- **Owner:** Can read and execute but cannot write.
- **Group:** Only allowed to write

```
$chmod __ __ __ script
$chmod 527 script
```

3. Converting a program to be a Set-UID program requires setting the Set-UID bit and changing the ownership of the program. After executing the following two commands on the `stack` program, the system is not showing it is a privileged program. Explain what could be the reason. **(5 marks)**

```
seed@VM$ sudo chmod 4755 stack
seed@VM$ sudo chown root stack
```

As a countermeasure, `chown` will reset the Set-UID bit before changing the ownership of a program. Hence, `chmod` must be executed after `chown`.

4. A developer created the following privileged program to display the calendar by invoking the `cal` program using the `system()` function. Briefly explain how Environment Variables can be used to execute a `cal` program developed by the attacker instead of the one provided by the system. **(5 marks)**

```
1  #include <stdlib.h>
2  int main()
3  {
4      system("cal");
5      return 0;
6  }
```

The parent process will pass the environment variables, including the newly created and modified ones by the user, to the child process. The user can instruct the process to find the cal program in the current folder instead of the one on the system by adding "." to the beginning of the PATH environment variable.

5. Briefly explain the Principle of Isolation

(5 marks)

The data and command should never be mixed. Hence, using multiple channels one for data and one for command is important.

This page intentionally left blank.

SECTION B Buffer Overflow**[20 marks]**

6. Draw the stack frame for the following function.

(5 marks)

```

1  int func(int g, int x, int f)
2  {
3      int d, b, a;
4      static str;
5      static y=5;
6      a = g - x;
7      d = a + x;
8      return d;
9  }

```

f, x, g, return address, old frame, d, b, a

7. For each of the following variables, specify on which part of the memory it will be located. **(5 marks)**

1. Initialised local variable
2. static ver_foo = 7
3. Uninitialised global variable
4. int *ptr = (int *) malloc(5*sizeof(int)); //(*ptr)
5. static hello;

- 1.
- 2.
- 3.
- 4.
- 5.

1. Stack
2. Data Segment
3. BB Segment
4. Stack
5. BSS Segment

8. Calculate the memory address of the “Return Address” field if the offset between the start of the buffer and `ebp` is 73 bytes, and `ebp` is pointing to the address `0xbffea130`. (2 marks)

```
0xbffea130 + 0x49 = 0xbffea179
```

9. An attacker has identified a vulnerable program that reads from a file and has created `abadfile` with the aim of getting a shell via a buffer overflow attack. The attack surface is the `strcpy()` function that copies the user input into a buffer.

After checking the memory layout for the compiled code using the debugging mode (`gdb`), she prepared a `badfile` and injected the new return address based on `ebp+75` (hex), where `ebp` is `0xbfffc8b`. She has turned off address randomisation, made the stack executable and disabled all stack countermeasures. Her attack should be successful; however, a `Segmentation fault` error message has been received instead of a shell when the vulnerable program executed on the input. (10 marks)

- (a) Explain the role of the NOP (`0x90`) in the attack and how it increases the chance of a successful attack. (3 marks)

```
The NOP slide provides a larger attack surface where the attacker only needs to jump to one of these addresses instead the exact address of the shellcode to have a successful attack.
```

- (b) Explain what is the main cause of the attack failing. (7 marks)

```
The injected value into the Return Address field is 0xbfffe00 (=0xbfffc8b + 75). As the last byte of the address is 00, strcpy() will terminate the copying of the input at that point. Hence, the shellcode will not be copied to the stack.
```

10. Consider buffer-overflow countermeasures. Answer the following based upon what you observe in the following figure. (4 marks)

```
[09/18/22] seed@VM$ ./stack
Segmentation fault (core dumped)
[09/18/22] seed@VM$ █
```

- (a) State the name of the countermeasure used above.

(1 mark)

Non-executable Stack

- (b) Discuss how the countermeasure mechanism works.

(3 marks)

The system will not allow any command on the stack to be executed, where all the values will be treated as data only.

This page intentionally left blank.

SECTION C Format String**[20 marks]**

11. Construct a string that will printout a secret value that is stored at a specific location on the stack.

Assume that a value is stored 48 bytes above the `va_list` pointer on a 32-bit machine(each memory block is 4 bytes). **(3 marks)**

Need 12 %x to move the pointer and one more %x to printout the value

12. Construct a string that would print out secret value that has its *address* stored on the stack.

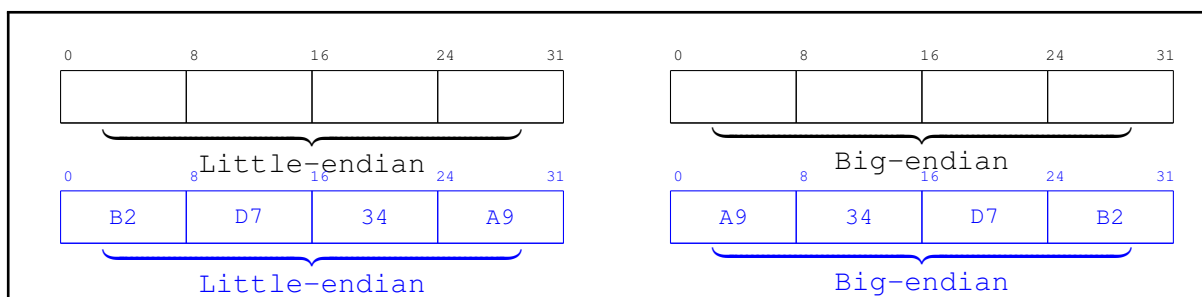
Assume that a secret value's *address* is 8 bytes above the `va_list` pointer on a 32-bit machine (each memory block is 4 bytes). **(3 marks)**

%x%x%s

13. Briefly explain why StackGuard cannot prevent format string attacks. **(5 marks)**

Unlike buffer overflow, format string does not require to overwrite the area before the Return Address field. Hence, the guard/secret will not be overwritten by any value, which represents the main factor to detect buffer overflow attacks.

14. What is the “Little-endian” and “Big-endian” representations of the 32-bit integer 0xA934D7B2? **(4 marks)**



15. The “width” modifier allows you to control the number of characters to be printed out. Using the %x and %n specifiers along with the width modifier, provide a format string to write 400 (decimal) into a variable on the stack if you know its address is 16 bytes above the va_list pointer. **(5 marks)**

```
%.8x%.8x%.8x%.Dx%n where D=400-(3*8)
```

16. An attacker targeted a format string vulnerability on system, where the aim was to print out a secret value stored on the system. Based on some investigations, the attacker was able to find the distance between the address of the secret value and the va_list pointer. Accordingly, she provided 20 %s specifiers that is enough to advance the pointer to the right position and print out the value. **(3 marks)**

- (a) Explain why her attack was not successful. **(7 marks)**

```
The system will treat each %s as an address to a value. If any of the address contains a value, e.g., 0x00, that is in a restricted region (kernel) or not mapped to a physical address, the program will crash.
```

- (b) How can you fix the problem? **(3 marks)**

```
Apart from the last %s, replace all the specifiers with %x.
```

This page intentionally left blank.

SECTION D XSS and CSRF**[20 marks]**

17. An attacker has decided to inject a script code into the brief description of his profile that will change the brief description of every user visits the attackers profile by replacing the content with “CYBR271 IS MY FAVOURITE COURSE”. Explain why the attacker needs to check that the current visitor of the profile is not he himself before carrying out the changes. **(3 marks)**

ANSWER

18. 3. A website (cybr271.com) utilises the ‘Content Security Policy’ (CSP) mechanism to prevent XSS attacks and allows script code to be executed based on the following CSP role Which of the following java script codes will, or will not, be allowed to run? **(3 marks)**

Content-Security-Policy: default-src 'self;' script-src 'self'
 'nonce-3fX254s' google.com

1. <script>... JavaScript code ...</script>
2. <script src="stdudents.js">... JavaScript code ...</script>
3. <script src="http://abc.com/sss.js">... JavaScript code ...</script>
4. <script nonce="3fX25DD">... JavaScript code ...</script>
5. <script nonce="3fX254s">... JavaScript code ...</script>

- 1.
- 2.
- 3.
- 4.
- 5.

ANSWER

19. To defeat XSS attacks, a developer decides to implement filtering on the browser side. Basically, the developer plans to add JavaScript code on each page, so before data are sent to the server, it filters out any JavaScript code contained inside the data. Let’s assume that the filtering logic can be made perfect. Can this approach prevent XSS attacks? **(3 marks)**

ANSWER

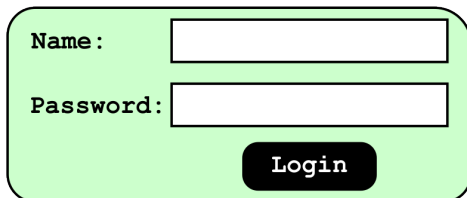
This page intentionally left blank.

SECTION E SQL-Injection**[20 marks]**

20. Data sanitisation (including both filtering and encoding approaches) can mitigate SQL-injection attacks, but does not address the fundamental cause of the problem. Discuss why “Prepared Statements”, on the other hand, is a perfect solution for this type of attacks. **(3 marks)**

ANSWER

21. A web application uses a the login form presented in the following figure to allow access only for those who have a record in the application’s database. The system does not sanitize the input before forming and submitting the query to the database. **(3 marks)**



```
<?php
    $sql = "SELECT *
            FROM users
            WHERE uid='$input_name' AND password='$input_password'";
    $result = $conn->query($sql);
?>
```

ANSWER

* * * * *