

EXAMINATIONS – 2019

TRIMESTER 2

CYBR 271

SECURE PROGRAMMING

Time Allowed: TWO HOURS

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 60 marks total.

Write answers in the spaces provided in the examination booklet.

Hand in the examination booklet.

Questions	Marks
1. Threat Modelling	[8]
2. Security Principles	[8]
3. Buffer Overflow	[11]
4. Format String Vulnerability	[5]
5. SQL Injection	[4]
6. Cross-Site Scripting (XSS)	[6]
7. Cross-Site Request Forgery	[8]
8. Input Validation	[6]
9. Security Testing	[4]



SPARE PAGE FOR EXTRA ANSWERS

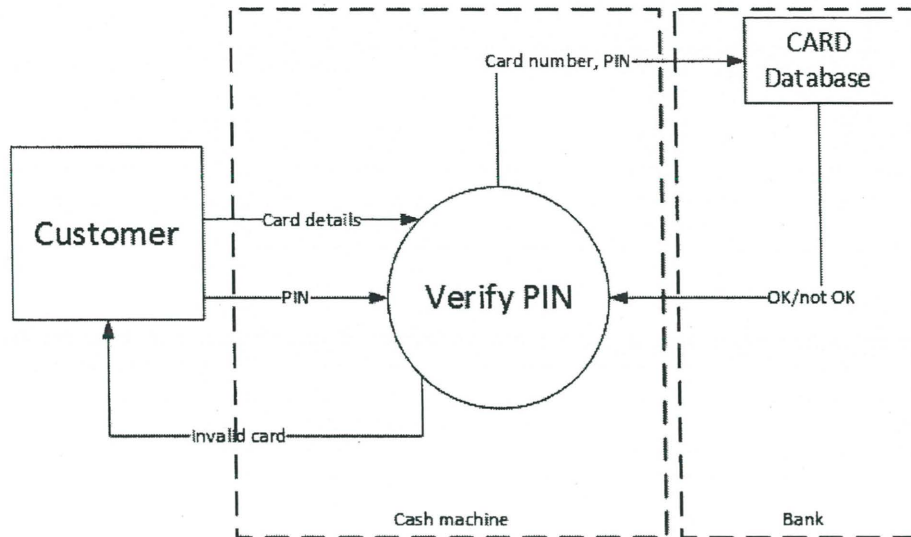
Cross out rough working that you do not want marked.

Specify the question number for work that you do want marked.

1. Threat Modelling.

(8 marks)

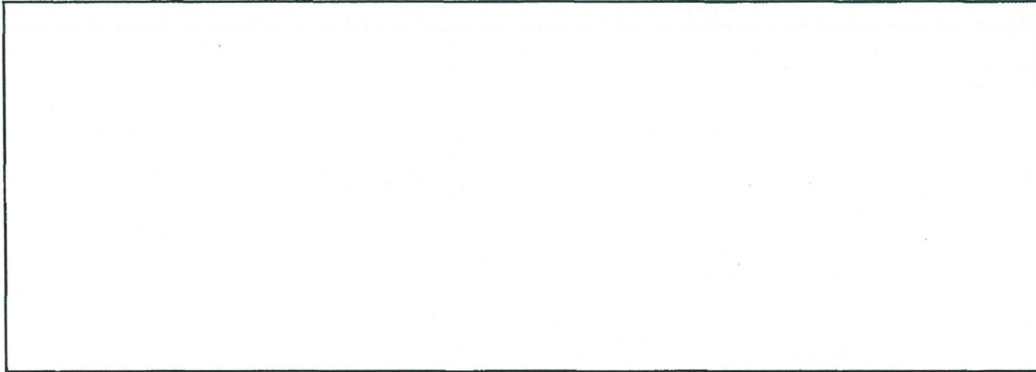
Consider the following data flow diagram for a cash machine showing what happens when a customer presents their card to the machine before being able to withdraw cash. Apply the STRIDE methodology to develop a bottom-up threat model for the cash machine.



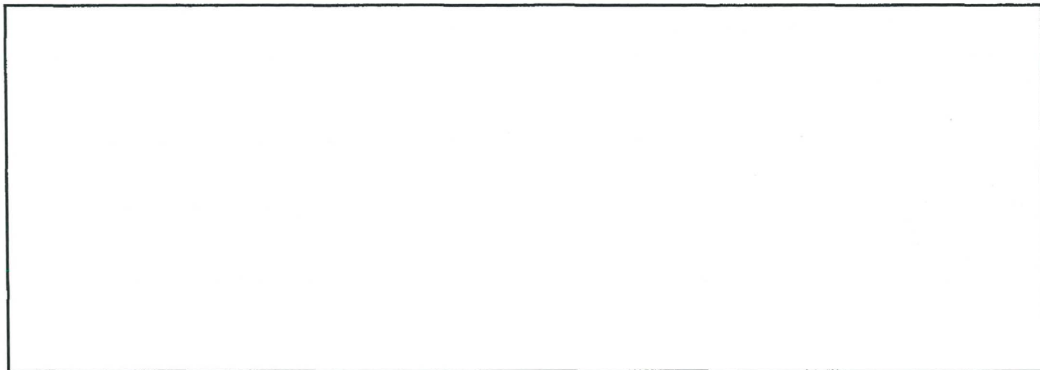
- i. (2 marks) Name an external entity, process, data store and dataflow shown in the diagram.

- ii. (2 marks) Consider STRIDE. Name two threats that are applicable to a data store.

- iii. **(2 marks)** Give an example of a threat applicable to an external entity in the diagram. Make sure that you make clear the type of threat as well as how it might be carried out.



- iv. **(2 marks)** Outline how you could *transfer* a risk associated with an information disclosure threat where an attacker observes the customer entering their PIN number?



2. Security Principles.

(8 marks)

- (a) (3 marks) Briefly outline the main difference between a *fail-secure* and *fail-safe* computer-controlled door lock system for a car.

- (b) (2 marks) Imagine that you have to fix a security bug for a server that you created several years ago. Briefly outline the advantages and disadvantages of maintaining *backward compatibility* with older client software.

- (c) (3 marks) What security principle helps protect against losses due to users failing to “do the right thing”? What is a potential problem with applying it in all cases?



SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.

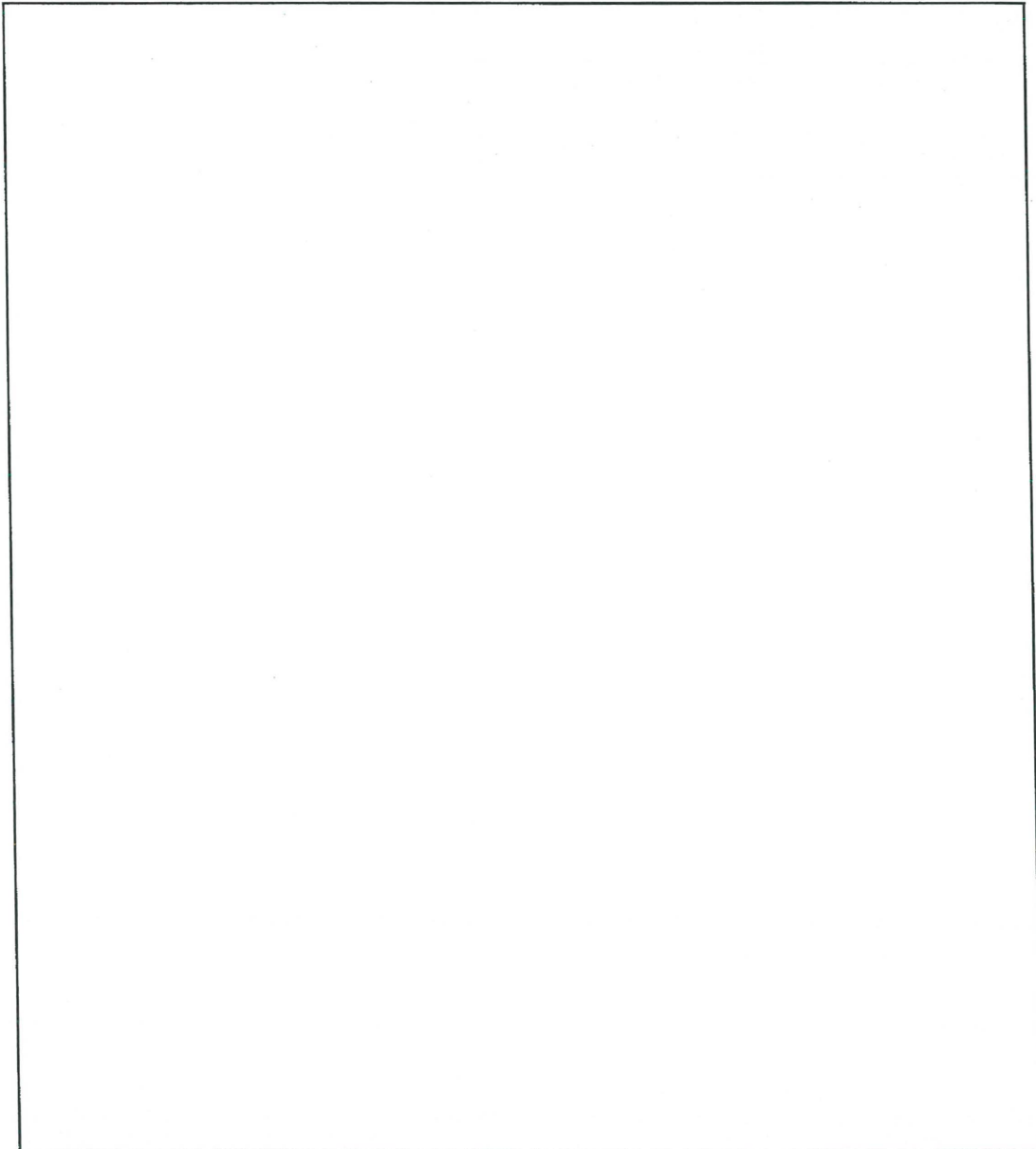
Specify the question number for work that you do want marked.

3. Buffer Overflow.

(11 marks)

(a) (6 marks) Draw the stack frame for the following function.

```
int func(int a, int b, int c)
{
    int m, n, k;
    m = a - c;
    k = m + b;
    return k;
}
```



- (b) (5 marks) Explain what is the MAIN cause of the following buffer attack failing.

The victim program reads from a file and has created a `badfile` with the aim of getting a shell via a buffer overflow attack. The target of the attack 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 `0xbffff200` as the new return address (the highlighted 4 bytes in the snippet below).

Note that address randomisation has been turned off, the victim program is compiled with executable stack protection turned off as well as made the stack executable and disabled all stack countermeasures.

```

1 00000000 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 |.....|
2 *
3 00000020 90 90 90 90 00 f2 ff bf 90 90 90 90 90 90 90 90 |.....|
4 00000030 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 |.....|
5 *
6 000001e0 90 90 90 90 90 90 90 90 90 90 90 90 31 c0 50 68 |.....1.Ph|
7 000001f0 2f 2f 73 68 68 2f 62 69 6e 89 e3 50 53 89 e1 99 |//shh/bin..PS...|
8 00000200 b0 0b cd 80 00 |.....|

```


4. Format String Vulnerability.

(5 marks)

- (a) (2 marks) Construct a string that will print out a secret value that is stored at a specific location on the stack. Assume that a value is stored 28 bytes above the `va_list` pointer on a 32-bit machine (each memory block is 4 bytes).

- (b) (3 marks) Explain why an attacker implementing a code injection attack constructs the attack in terms of two separate operations. First overwriting the return address and secondly writing the malicious code to memory.



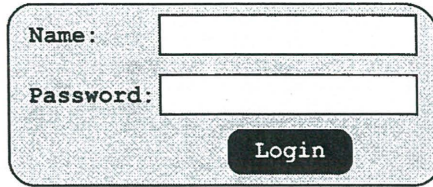
SPARE PAGE FOR EXTRA ANSWERS

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

5. SQL Injection.

(4 marks)

- (a) (2 marks) A web application uses 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.



The figure shows a login form with a light blue background and rounded corners. It contains two input fields: the first is labeled 'Name:' and the second is labeled 'Password:'. Below these fields is a dark blue button with the word 'Login' in white text.

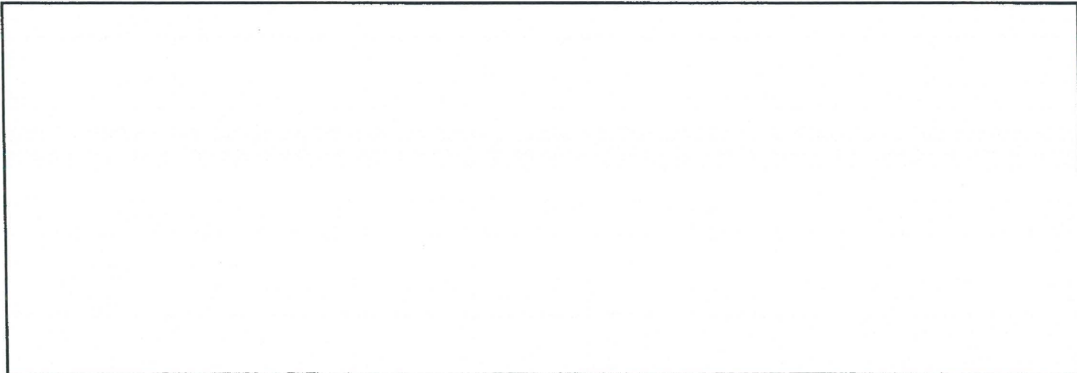
```
<?php
    $sql = "SELECT *
            FROM users
            WHERE uid='$input_name' AND password='$input_password'";
    $result = $conn->query($sql);
?>
```

Construct an input that will let you to access the system even if you are not registered.

Name:

Password:

- (b) (2 marks) Briefly explain the fundamental cause of SQL-injection attacks, and list three approaches to mitigate/prevent this type of attack.



A large empty rectangular box with a black border, intended for the student to write their answer to part (b).



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. Cross-Site Scripting (XSS).

(6 marks)

- (a) **(4 marks)** Briefly explain the main difference between *reflected* and *stored* XSS attacks and comment upon which can have a wider impact.

- (b) **(2 marks)** Explain why a XSS defence based upon filtering out `script` tags from user-provided data is easily evaded and give an example of how to do this.

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. Cross-Site Request Forgery (CSRF).

(8 marks)

- (a) (6 marks) List THREE main differences between CSRF and XSS attacks.

- (b) (2 marks) Explain why the use of HTTPS does not protect against CSRF attacks?

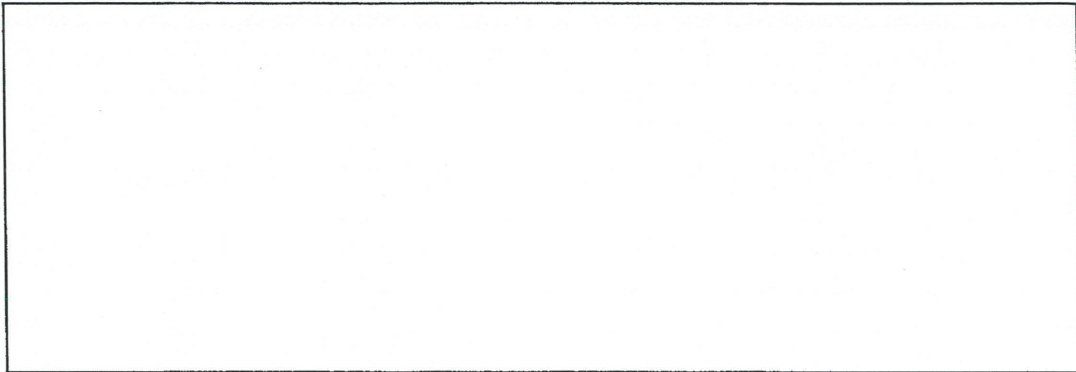
SPARE PAGE FOR EXTRA ANSWERS

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

8. Input validation.

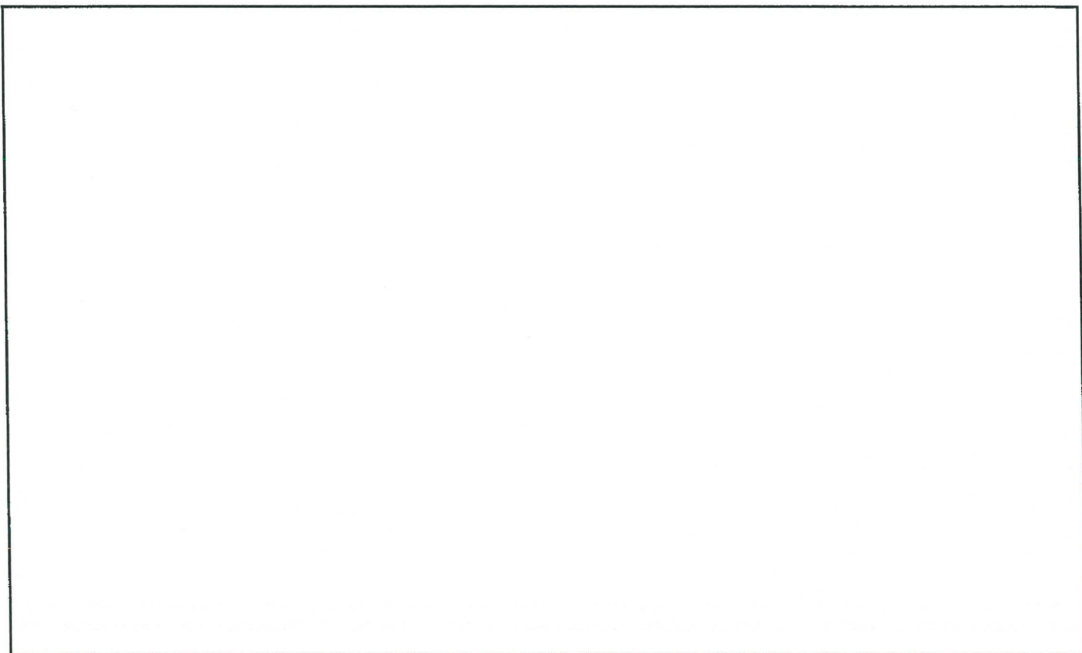
(6 marks)

- (a) **(2 marks)** Explain why *whitelisting* is usually preferred to *blacklisting*?



- (b) **(4 marks)** Consider a web application that allows the uploading of images. The program performs input validation on the file name to check that it has an image file extension such as GIF, PNG or JPG.

Briefly outline TWO vulnerabilities related to canonical file name issues that might apply in this case.





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.

9. Security testing.

(4 marks)

Briefly discuss ONE advantage and ONE disadvantage that *input fuzzing* has over manual data mutation.

