22

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **WINTER SEMESTER, 2022-2023**

DURATION: 1 HOUR 30 MINUTES **FULL MARKS: 75**

SWE 4503: Software Security

Programmable calculators are not allowed. Do not write anything on the question paper.
Answer **all 3 (three)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

[For all the questions, assume 32-bit system unless otherwise mentioned.]

1. a) Identify and explain the type of violated security properties (CIA) for each of the incidents below. 3×3 (CO2) (PO2)
   i. Attackers stole sensitive files from a third-party server belonging to National Student Clearinghouse of UK in September 2023.
   ii. Systems under City of Dallas (Texas) suffered a ransomware attack where attackers ex-filtrated 1.17 TB of data and encrypted sensitive files.
   iii. Due to a recent cyberattack on Auckland Transport of New Zealand, users can not recharge their transportation cards.

   b) What does the graph in Figure 1 represent? Explain the insights that you get from each segment of the graph in Figure 1. 8 (CO2) (PO2)

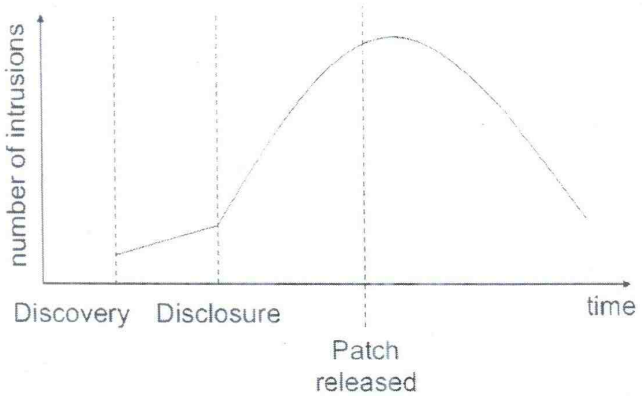

Figure 1: "number of intrusions vs time" graph for Question 1.b)

   c) "Attacks happen more during a recession and difficult economic times". Justify this statement with proper reasoning and examples. 8 (CO1) (PO1)

2. a) The Skipjack encryption algorithm designed by NSA was initially kept secret for sole use of federal agencies. Despite the secrecy, independent researchers were able to figure out the algorithm and found possible attacks against it. $5 + 4$ (CO2) (PO2)
   i. Considering the case of Skipjack algorithm, explain why keeping a system secret doesn't increase its security in light of Shannon's Maxim.
   ii. Provide one example with proper reasoning where secrecy might be a good choice for security.

b) Two C functions and their corresponding x86 assembly are given in Code Snippet 1 and Code Snippet 2.

```
1 int callee(int a, int b){
2     int c = 69;
3     return (a + b) - c;
4 }
5 int main(){
6         int a = 1569;
7         int b = 48;
8         callee(a, b);
9 }
```

Code Snippet 1: C code for Question 2.b)

```
1  <main>:
2  .....
3  804919a: movl   $1569,-0x4(%ebp)
4  80491a1: movl   $48,-0x8(%ebp)
5  80491a8: push   -0x8(%ebp)
6  80491ab: push   -0x4(%ebp)
7  80491ae: call   8049166 <callee>
8  80491b3: add    $0x8,%esp
9  .....
10
11 <callee>:
12 8049166: push   %ebp
13 8049167: mov    %esp,%ebp
14 8049169: sub    $0x10,%esp
15 .....
16 8049176: movl   $69,-0x4(%ebp)
17 804917d: mov    0x8(%ebp),%edx
18 8049180: mov    0xc(%ebp),%eax
19 8049183: add    %edx,%eax
20 8049185: sub    -0x4(%ebp),%eax
21 8049188: leave
22 8049189: ret
```

Code Snippet 2: x86 assembly for Question 2.b)

i. Show the exact locations of three local variables a, b & c in their respective stack frames. The locations must be shown with respect to each stack frame's Saved Frame Pointer. 5 (CO1) (PO1)

ii. Based on Code Snippet 2, illustrate with diagrams how the function prologue and epilogue work together to set up and restore the stack frame for the callee and caller respectively. Show the changes in `%esp` and `%ebp`. 12 (CO1) (PO1)

3. a) "In 2016 Bangladesh Bank Cyber Heist attackers used a series of malware like DRIDEX, NestEgg, MackTruck and SierraCharlie to enter into Bank's internal network and compromised the machines. Attackers also used malicious ZIP file sent as email to infect employee machines and hacked the printer to hide their activity."
From the above scenario, identify the attack vectors used by attackers and the assets which should have been protected. 5 (CO2) (PO2)

b) What is the major source of memory unsafety problems? Mention three mitigations that Google is trying to increase memory safety in Android codebase. 5 (CO1) (PO1)

c) Consider the following vulnerable C code given in Code Snippet 3. Assume, EBP holds `0xbfffcd54` and ESP holds `0xbfffcd48` marking the top and bottom of stack frame of function *vuln*() and sizeof shellcode is 32.

```
1 void vuln(){
2     char buf[12];
3     gets(buf);
4 }
```

Code Snippet 3: Vulnerable C code for Question 3.c)

i. Why the code in Code Snippet 3 is vulnerable? What could be done to remove this vulnerability? 4 (CO3) (PO2)

ii. Draw the stack diagram with appropriate memory addresses showing the location of shellcode, `buf` variable and return address. 6 (CO3) (PO2)

iii. Write the payload in Python which would be used as input to the vulnerable code in Code Snippet 3. 4 (CO3) (PO2)