

Roll Number:

Thapar Institute of Engineering and Technology, Patiala
Department of Computer Science & Engineering

BE- CoE, CSE (VI Semester) MST
8 April 2022

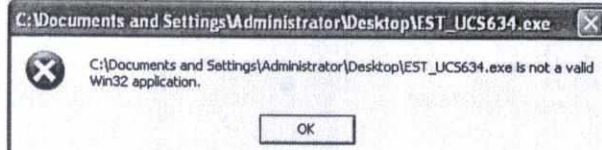
Time: 02 Hours; MM: 35

Note: Attempt any five questions.

UCS638: Secure Coding

Name of Faculty: Dr. Maninder Singh

Q1. ✓ Dr. Singh edited EST_UCS634.exe file with LordPE and got this error message, what could have been the reason? Help him to fix this error.



b) ✓ Comment about this memory map capture of EST_UCS634 file.

004550F0	6A 60	PUSH 60	00400000	00001000	EST_UCS6	PE header
004550F2	68 08814700	PUSH EST_UCS6.00478108	00401000	0005C000	EST_UCS6	code
004550F7	E8 08210000	CALL EST_UCS6.00457204	0045D000	0001E000	EST_UCS6	imports
004550FC	BF 94000000	MOV EDI, 94	0047B000	00006000	EST_UCS6	data
00455101	8BC7	MOV EAX, EDI	00481000	00003000	EST_UCS6	data
			00484000	00002000	EST_UCS6	.rsrc
						resources
						NewSec

(2, 5)

Q2. Refer to the memory map given above:

a) ✓ What changes are to be made in order to jump to the "Code Cave address"?

b) ✓ What should be the startup instructions at 00484000 memory address?

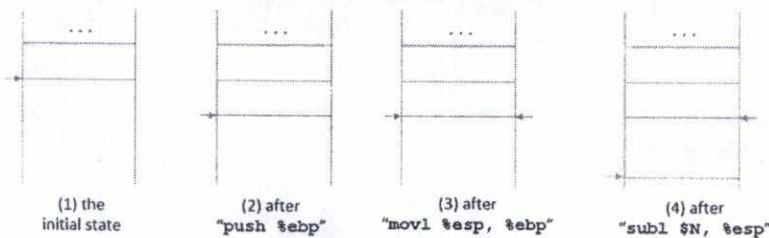
c) ✓ Dr. Singh started a putty session on port 22 to 192.168.240.137 but connection failed, give appropriate reason for the same and offer solution to overcome this issue. Dr. Singh wrote one liner to create shell with RAW payload for reverse TCP connection to port 443. Produce same one liner code.

d) ✓ Give instructions with proper addresses to proceed with normal flow of execution after injection & realignment.

(1, 2, 2, 2)

Q3.

a) ✓ Fill the blank spaces and arrow positions for the function prologue execution:



b) ✓ In the function epilogue, the previous frame pointer, which is stored in the area below the return address, will be retrieved and assigned to the *ebp* register. However, when we overflow the return address, the previous frame pointer

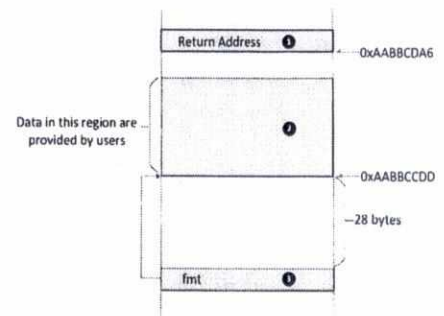
c) ✓ region is already modified, so after the function epilogue, *ebp* contains some arbitrary value. Does this matter? As we know, the *system()* function calls */bin/sh*, which is a symbolic link to */bin/bash*. Recent versions of *bash* will drop the privilege if it detects that the effective user ID and the real user ID are different. Assume that we still want to use *system()* in our Return-to-libc attack, please describe how you can overcome this challenge. You are allowed to have zeros in your input (assume that *memcpy()* is used for memory copy, instead of *strcpy()*)

(2, 2, 3)

Q4. a) ✓ When *printf(fmt)* is executed, the stack (from low address to high address) contains the following values (4 bytes each), where the first number is the content of the variable *fmt*, which is a pointer pointing to a format string. If you can decide the content of the format string, what is the smallest number of format specifiers that you can use to crash the program with a 100 percent probability?

0xAABBCDD, 0xAABDDFF, 0x22334455, 0x00000000, 0x99663322

b) ✓ A server program takes an input from a remote user, saves the input in a buffer allocated on the stack. The address of this buffer is then stored in the local variable *fmt*, which is used in the following statement in the server program: *printf(fmt)*; When the above statement is executed, the current stack layout is depicted in Figure. If you are a malicious attacker, can you construct



the input, so when the input is fed into the server program, you can get the server program to execute your code? Please write down the actual content of the input (you do not need to provide the exact content of the code; just put "malicious code" in your answer, but you need to put it in the correct location).

(2, 5)

Q5. a) There are three kinds of Honeypots used in security implementation: elaborate each of one of these with appropriate use case.

b) Rootkits use various methods to hide malware from anti-malware software, what are these methods, explain in detail working of rootkits with appropriate example.

(3, 4)

Q6. It was a morning ritual. Mrs. Singh sipped her coffee as she quickly went through the email that arrived during the night. One of the messages caught her eye, because it was clearly spam that somehow got past the email filter. The message extolled the virtues of being slim and contained a link to the on-line slim-tea offerings web site. "Do people really fall for this stuff?" Mrs. Singh thought. She was curious to know how the website would convince its visitors to make the purchase, so she clicked on the link. The website was slow to load and seemed to be broken. There was no major content on the page, it only said "Animated Cursor Demo". Disappointed, Mrs. Singh closed the browser's window and continued with her day. "She didn't realize that her Windows XP computer just got infected."

You are a Secure Coding Professional. You possess the following capture.

```
00 00 00 00 52 49 46 46 EB 77 00 00 41 43 4F 4E 61 6E 69 68
00 00 00 10 24 00 00 00 24 00 00 00 02 00 00 00 EB 16 00 00
00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 30 00 00 00 00 01 00 00 00 61 6E 69 68 58 00 00 00
00 00 00 40 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00 00 00 50 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00 00 00 60 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
00 00 00 70 41 41 41 41 41 41 41 41 41 41 41 41 00 00 00 00
00 00 00 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 90 90 90 90 90 CC CC CC CC
```

a) Help Mrs. Singh to craft a stream of hexadecimal numbers, which will be used to create ANI file (give one line command with proper use of piping).

b) Also, your mission is to understand what probably would have happened to Mrs. Singh's system after she clicked the link. Your analysis should reveal intent. Give technical commentary step by step and conclusion, emphasizing each of the markings on the figures: like EIP: 77D822C0.

```
27 20.400149 192.168.72.130 192.168.72.128 HTTP GET /exp.html HTTP/1.1
30 20.414699 192.168.72.130 192.168.72.128 HTTP GET /demo.ani HTTP/1.1
31 20.415158 192.168.72.130 192.168.72.128 HTTP HTTP/1.1 200 OK (text/plain)
```

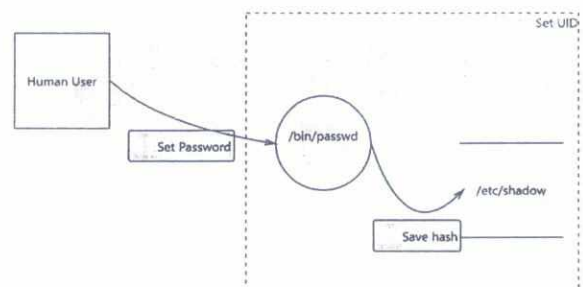
```
01850095 90 NOP
01850096 90 NOP
01850097 90 NOP
01850098 FC CLD
01850099 E8 89000000 CALL 01850127
0185009A 60 PUSHAD
0185009B 89E5 MOV EBP,ESP
0185009C 31D2 XOR EDX,EDX
0185009D 64:8B52 30 MOV EDX,DWORD PTR FS:[EDX+30]
0185009E 8B52 0C MOV EDX,DWORD PTR DS:[EDX+C]
0185009F 8B52 14 MOV EDX,DWORD PTR DS:[EDX+14]
018500A0 8B72 28 MOV ESI,DWORD PTR DS:[EDX+28]
018500A1 0FB74A 26 MOVZX ECX,WORD PTR DS:[EDX+26]
018500A2 31FF XOR EDI,EDI
018500A3 31C0 XOR EAX,EAX
018500A4 AC LODS BYTE PTR DS:[ESI]
018500A5 3C 61 CMP AL,61
018500A6 7C 02 JL SHORT 018500BF
```

```
01850000 52 PUSH EDX
01850001 49 DEC ECX
01850002 46 INC ESI
01850003 4C INC ESI
01850004 EB 16 JMP SHORT 0185001C
01850005 0000 ADD BYTE PTR DS:[EAX],AL
01850006 41 INC ECX
01850007 43 INC EBX
01850008 4F DEC EDI
01850009 4E DEC ESI
0185000A 61 POPAD
0185000B 6E OUTS DX, BYTE PTR ES:[EDI]
0185000C 6968 24 0000002 IMUL EBP,DWORD PTR DS:[EAX+24],24
0185000D 0000 ADD BYTE PTR DS:[EAX],AL
0185000E 0002 ADD BYTE PTR DS:[EDX],AL
0185000F 0005 ADD BYTE PTR DS:[EAX],AL
01850010 00EB ADD BL,CH
01850011 77 00 JA SHORT 0185001F
```

```
77D822C0 -FF25 JMP DWORD PTR DS:[ESI]
77D822C1 8B45 E9 MOV BYTE PTR DS:[EDI+17],AL
77D822C2 5E DEC ECX
77D822C3 5E DEC ECX
77D822C4 74 0F JZ 77D822CF
77D822C5 FFBE MOV EBX,EBP
77D822C6 9E SRAH
77D822C7 0000 ADD BYTE PTR DS:[EAX],AL
77D822C8 00E9 ADD CL,CH
77D822C9 49 DEC ECX
77D822CA 5E POP ESI
77D822CB FE DEC EDI
77D822CC FF8B C1E97E5E DEC DWORD PTR DS:[EBX+5E7E5C]
77D822CD FE DEC EDI
77D822CE FF66 81 JMP DWORD PTR DS:[ESI-7F]
77D822CF 7D 10 JBE SHORT USER32.77D822EE
77D822D0 00040F ADD BYTE PTR DS:[EDI+ECX],AL
77D822D1 82E2 68 AND DL,68
```

c) Study the given design view of the system being developed, apply your know-how of STRIDE at each actor and boundary place and produce a) threat list and b) threat properties.

(2, 5)



(3, 4)