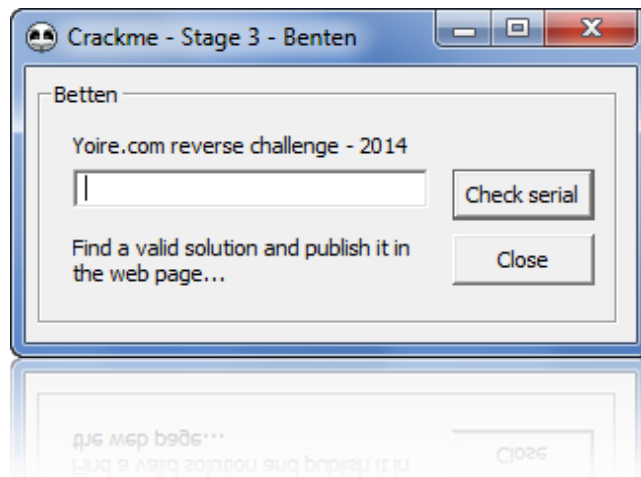


Solución para el CrackMe Benten de Yoire.com

Fuerza Bruta



By deurus
28/08/2014

ÍNDICE

1. Analizando.....	2
2. Aplicando fuerza bruta	3
3. Enlaces	4

Equipo utilizado:

S.O: Windows 7 x32 /Windows 8 x64

Depurador: Ollydbg 1.10 (32bits) con plugins

Analizador: PEiD 0.95

1. Analizando

Abrimos el crackme con **Olllydbg** y vamos a las **referenced strings**.

Text string
ASCII "aClass"
(Initial CPU selection)
ASCII "GREAT!!! Your serial is valid :)"
ASCII "Response"
ASCII "Sorry, the serial is invalid :("

Pinchamos sobre cualquiera.

00401183	50	LEA EAX,LOCAL:2503	Arg1 = 7580EE0A
00401184	E8 07FFFFFF	CALL 01_crack.00401090	01_crack.00401090
00401189	3D 3E769EB7	CMP EAX,B79E763E	
0040118E	75 16	JNZ SHORT 01_crack.004011A6	
00401190	6A 40	PUSH 40	Style = MB_OK!MB_ICONASTERISK!MB_APPLMODAL
00401192	68 20304000	PUSH 01_crack.00403020	Title = "Response"
00401197	68 29304000	PUSH 01_crack.00403029	Text = "GREAT!!! Your serial is valid :)"
0040119C	6A 00	PUSH 0	hOwner = NULL
0040119E	FF15 70414000	CALL DWORD PTR DS:[<&USER32.MessageBoxA>]	MessageBoxA
004011A4	EB 14	JMP SHORT 01_crack.004011BA	
004011A6	6A 30	PUSH 30	Style = MB_OK!MB_ICONEXCLAMATION!MB_APPLMODAL
004011A8	68 20304000	PUSH 01_crack.00403020	Title = "Response"
004011AD	68 00304000	PUSH 01_crack.00403000	Text = "Sorry, the serial is invalid :("
004011B2	6A 00	PUSH 0	hOwner = NULL
004011B4	FF15 70414000	CALL DWORD PTR DS:[<&USER32.MessageBoxA>]	MessageBoxA
004011B8	50	POP EAX	

Vemos un "Call" donde seguramente se generará un **SUM** en función del serial metido ya que después del Call vemos una comprobación contra "**B79E763E**" lo que nos da una pista de que vamos a tener que utilizar **fuerza bruta** para llegar a ese valor. Vamos a explorar el Call.

004010A0	BF ED5E0000	MOV EDI,5EED	
004010B2	EB 23	JMP SHORT 01_crack.004010D7	
004010B4	89F8	MOV EAX,EDI	
004010B6	C1E0 05	SHL EAX,5	
004010B9	0FB613	MOVZX EDX,BYTE PTR DS:[EBX]	
004010BC	31D0	XOR EAX,EDX	
004010BE	89C7	MOV EDI,EAX	
004010C0	81F7 ED1E0B1D	XOR EDI,1D0B1EED	
004010C6	43	INC EBX	
004010C7	6A 64	PUSH 64	Timeout = 100. ms
004010C9	FF15 EC404000	CALL DWORD PTR DS:[<&KERNEL32.Sleep>]	Sleep
004010CF	FF15 E8404000	CALL DWORD PTR DS:[<&KERNEL32.GetTickCount>]	GetTickCount
004010D5	89C6	MOV ESI,EAX	
004010D7	803B 00	CMP BYTE PTR DS:[EBX],0	
004010DA	75 D8	JNZ SHORT 01_crack.004010B4	
004010DC	2B75 FC	SUB ESI,LOCAL:1	
004010DF	81FE EE020000	CMP ESI,2EE	
004010E5	77 08	JA SHORT 01_crack.004010EF	
004010E7	81FE 5E010000	CMP ESI,15E	
004010ED	73 04	JNB SHORT 01_crack.004010F3	
004010EF	31C0	XOR EAX,EAX	
004010F1	EB 02	JMP SHORT 01_crack.004010F5	
004010F3	89F8	MOV EAX,EDI	
004010F5	5F	POP EDI	01_crack.00401189
004010F6	5E	POP ESI	01_crack.00401189
004010F7	5B	POP EBX	01_crack.00401189
004010F8	89EC	MOV ESP,EBP	
004010FA	5D	POP EBP	
004010FB	C2 0400	RETN 4	01_crack.00401189

Lo que resalto con la flecha son una par de **Calls** que podemos **NO**pear ya que lo único que hacen es **ralentizar** la generación del SUM. A continuación vamos a analizar el algoritmo de generación del SUM.

1	MOV EDI,SEED	- EDI = SEED
2	JMP SHORT 01_crack.004010D7	
3	/MOV EAX,EDI	<----Bucle
4	ISHL EAX,5	- SEED * 32 = BDDA0
5	IMOVZX EDX,BYTE PTR DS:[EBX]	- Coge el dígito
6	IXOR EAX,EDX	- BDDA0 XOR dígito
7	IMOV EDI,EAX	
8	IXOR EDI,1D0B1EED	- XOR 1D0B1EED
9	INC EBX	
10	...	
11	IMOV ESI,EAX	
12	CMP BYTE PTR DS:[EBX],0	
13	JNZ SHORT 01_crack.004010B4	- Bucle ---->

Para un serial de tres dígitos la secuencia sería ésta (valores en hexadecimal):

1ºDigit—> BDDA0 XOR 1D0B1EED XOR 1ºDigit XOR 1D0B1EED = Temp
 2ºDigit—> Temp = Temp * 20 Xor 487268077 XOR 2ºDigit
 3ºDigit—> Temp = Temp * 20 Xor 487268077 XOR 3ºDigit
 ...
 CMP Temp, B79E763E

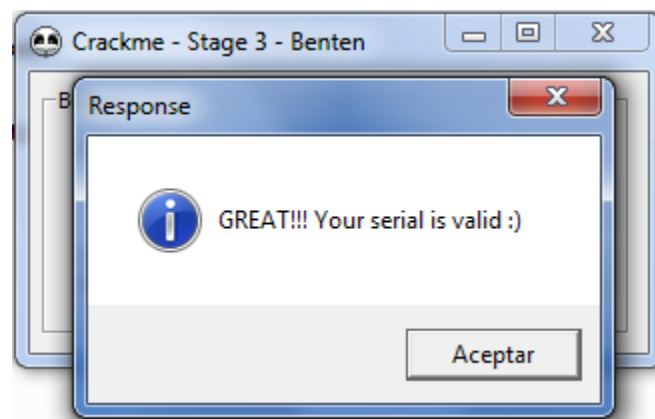
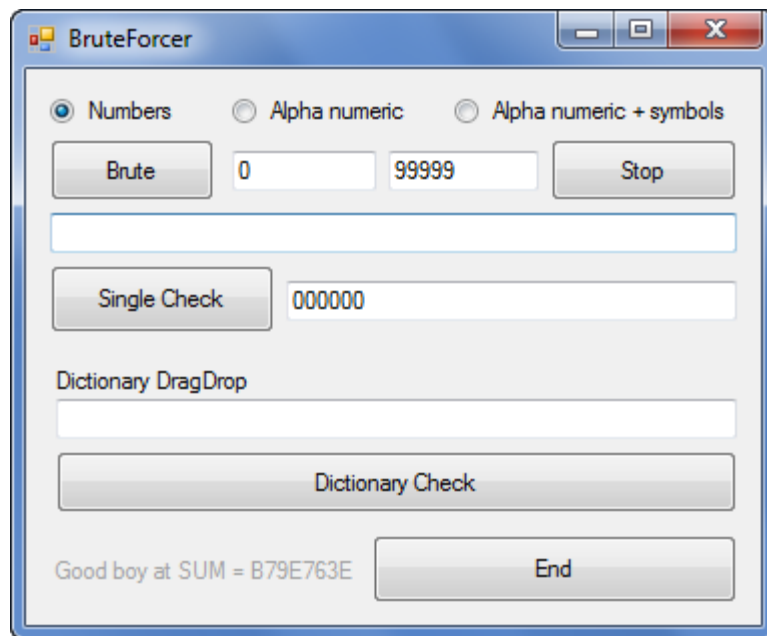
2. Aplicando fuerza bruta

La creación del “**BruteForcer**” os la dejo a vosotros. Aquí tenéis un fragmento hecho en VB.Net.

```

1 Dim temp As Long
2 Dim temp2 As String
3 Dim letter As Integer
4 Dim brute As String
5 brute = TextBox4.Text
6 temp = 0
7 temp = Asc(Mid(brute, 1, 1)) Xor 487268077 Xor 777632
8 temp2 = Hex(temp)
9 temp2 = Microsoft.VisualBasic.Right(temp2, 8)
10 temp = Convert.ToUInt64(temp2, 16)
11 For i = 2 To Len(brute)
12 letter = Asc(Mid(brute, i, 1))
13 temp = temp * 32
14 temp2 = Hex(temp)
15 temp2 = Microsoft.VisualBasic.Right(temp2, 8)
16 temp = Convert.ToUInt64(temp2, 16)
17 temp = temp Xor 487268077
18 temp2 = Hex(temp)
19 temp2 = Microsoft.VisualBasic.Right(temp2, 8)
20 temp = Convert.ToUInt64(temp2, 16)
21 temp = temp Xor letter
22 '
23 temp2 = Hex(temp)
24 Next

```



3. Enlaces

<https://deurus.info/2014/08/yoire-pe-stage-3-reversing-challenge-benten-fuerza-bruta/>