

Keygen para el CrackmeMe#02 de LaFarge

KeyGen por injerto

By deurus
14/09/2014

ÍNDICE

1. Introducción	2
2. Injerto Light	2
3. KeyGen a partir de la víctima	3
4. Notas finales.....	5
5. Enlaces	5
6. Crackeando Crackmes by deurus.....	5

Equipo utilizado:

S.O: Windows 7 x32 /Windows 8 x64

Depurador: Ollydbg 1.10 (32bits) con plugins

Analizador: PEiD 0.95

1. Introducción

Hoy vamos a hacer algo diferente, vamos a hacer un **keygen con la propia víctima**. El término anglosajón para esto es “**selfkeygening**” y no es que esté muy bien visto por los reversers pero a veces nos puede sacar de apuros.

La víctima elegida es el **Crackme 2 de LaFarge**. Está hecho en ensamblador.

2. Injerto Light

Primeramente vamos a realizar un injerto light, con esto quiero decir que vamos a **mostrar el serial bueno** en la **MessageBox de error**.

Abrimos **Ollly** y localizamos el código de comprobación del serial, tenemos suerte ya que el serial se muestra completamente y no se comprueba byte a byte ni cosas raras. En la imagen inferior os muestro el serial bueno para el nombre deurus y el mensaje de error. Como podeis observar el **serial bueno** se saca de memoria con la instrucción **PUSH 406749** y el mensaje de error con **PUSH 406306**.

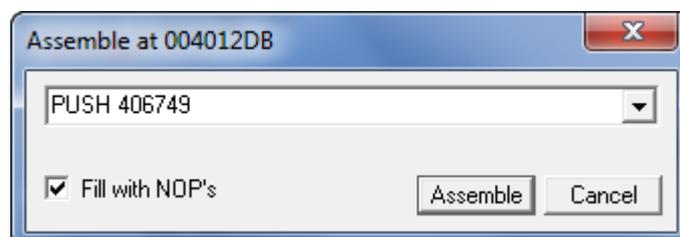
```
00401286 .< 75 F8 JNZ SHORT CrackME.00401279
00401288 . 68 49674000 PUSH CrackME.00406749
0040128D . 68 49654000 PUSH CrackME.00406549
00401292 . E8 5F010000 CALL <JMP.&kernel32.lstrcpyA>
00401297 . 68 00020000 PUSH 200
0040129C . 68 49694000 PUSH CrackME.00406949
004012A1 . 6A 64 PUSH 64
004012A3 . FF75 08 PUSH DWORD PTR SS:[EBP+8]
004012A6 . E8 E5000000 CALL <JMP.&user32.GetDlgItemTextA>
004012AB . 68 49654000 PUSH CrackME.00406549
004012B0 . 68 49694000 PUSH CrackME.00406949
004012B5 . E8 36010000 CALL <JMP.&kernel32.lstrcpyA>
004012BA . 0BC8 OR EBX, EBX
004012BE .> 75 16 JNZ SHORT CrackME.004012D4
004012C0 . 6A 40 PUSH 40
004012C8 . 68 DB624000 PUSH CrackME.004062DB
004012CD . 68 AC624000 PUSH CrackME.004062AC
004012D2 . FF75 08 PUSH DWORD PTR SS:[EBP+8]
004012D4 . E8 CA000000 CALL <JMP.&user32.MessageBoxA>
004012D8 . JMP SHORT CrackME.004012E8
004012D9 .> 6A 10 PUSH 10
004012DB . 68 06634000 PUSH CrackME.00406306
004012DE . 68 E7624000 PUSH CrackME.004062E7
004012E0 . FF75 08 PUSH DWORD PTR SS:[EBP+8]
004012E3 . E8 B4000000 CALL <JMP.&user32.MessageBoxA>
004012E8 .> 68 00020000 PUSH 200
004012ED . 68 49654000 PUSH CrackME.00406549

String2 = "1964087121"
String1 = CrackME.00406549
lstrcpyA
Count = 200 (512.)
Buffer = CrackME.00406949
ControlID = 64 (100.)
hWnd = 002F077A ('LaFarge's crackme #2',class='#32770')
GetDlgItemTextA
String2 = "1964087121"
String1 = "12345"
lstrcpyA

Style = MB_OK|MB_ICONASTERISK|MB_APPLMODAL
Title = "Good boy..."
Text = "Yep, thats the right code!\n\rGo write a keygen!"
hOwner = 002F077A ('LaFarge's crackme #2',class='#32770')
MessageBoxA

Style = MB_OK|MB_ICONHAND|MB_APPLMODAL
Title = "bad boy..."
ASCII = "Nope, thats not it!\n\rTry again"
hOwner = 002F077A ('LaFarge's crackme #2',class='#32770')
MessageBoxA
Length = 200 (512.)
Destination = CrackME.00406549
```

Si cambiamos el PUSH del serial por el de el mensaje de error ya lo tendríamos. Nos situamos encima del **PUSH 406306** y pulsamos **espacio**, nos saldrá un diálogo con el push, lo modificamos y le damos a **Assemble**.



Ahora el crackme cada vez que le demos a **Check it!** nos mostrará:



3. KeyGen a partir de la víctima

Pero no nos vamos a quedar ahí. Lo interesante sería que el serial bueno lo mostrara en la caja de texto del serial. Esto lo vamos a hacer con la función **user32.SetDlgItemTextA**.

The **SetDlgItemText** function sets the title or text of a control in a dialog box.

```
BOOL SetDlgItemText(
  HWND hDlg,           // handle of dialog box
  int nIDDlgItem,      // identifier of control
  LPCTSTR lpString     // text to set
);
```

Parameters

hDlg
Identifies the dialog box that contains the control.

nIDDlgItem
Identifies the control with a title or text that is to be set.

lpString
Points to the null-terminated string that contains the text to be copied to the control.

Según dice la función **necesitamos el handle de la ventana, el ID de la caja de texto y el string a mostrar**. La primera y segunda la obtenemos fijándonos en la función **GetDlgItemTextA** que recoge el serial introducido por nosotros. La string es el **PUSH 406749**.

```
00401296 | . E8 01010000 CALL 0040129E ;CALL USER32.GetUser32Copy
00401297 | . 68 00020000 PUSH 200
0040129C | . 68 49694000 PUSH 00406949
004012A1 | . 6A 64 PUSH 64
004012A3 | . FF75 08 PUSH DWORD PTR SS:[EBP+8]
004012A6 | . E8 E5000000 CALL USER32.GetUser32Copy
004012A7 | . 68 40674900 PUSH 406749
```

CALL USER32.GetUser32Copy
MaxCount = 512.
String = "12345"
ItemID = 100.
hDialog

Con esto ya tenemos todo lo que necesitamos excepto el espacio dentro del código, en este caso lo lógico es parchear las MessageBox de error y acierto. Las seleccionamos, click derecho y **Edit > Fill with NOPs**.

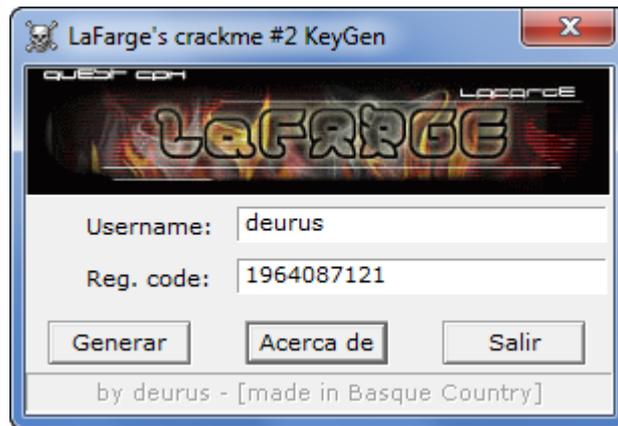
004012B5	E8 42074000	PUSH CrackME.00406749	MessageBoxA
004012B6	E8 36010000	CALL <JMP.&kernel32.lstrcpA>	lstrcpA
004012B7	00C0	OR EAX,EAX	kernel32.BaseThreadInitThunk
004012B8	75 16	JNZ SHORT CrackME.004012D4	Style = MB_OK MB_ICONASTERISK MB_APPLMODAL
004012B9	6A 40	PUSH 40	Title = "Good boy..."
004012BA	68 0B624000	PUSH CrackME.004062DB	Text = "Yep, thats the right code!\n\nGo write a keygen!"
004012BB	68 AC624000	PUSH CrackME.004062AC	hOwner = 7FFD9000
004012BC	FF75 08	PUSH DWORD PTR SS:[EBP+8]	MessageBoxA
004012BD	E8 CA000000	CALL <JMP.&user32.MessageBoxA>	Style = MB_OK MB_ICONHAND MB_APPLMODAL
004012BE	EB 14	JMP SHORT CrackME.004012E8	Title = "Bad boy..."
004012BF	6A 10	PUSH 10	ASCII "Hope, thats not it!\n\nTry again"
004012C0	68 06634000	PUSH CrackME.00406306	hOwner = 7FFD9000
004012C1	68 E7624000	PUSH CrackME.004062E7	MessageBoxA
004012C2	FF75 08	PUSH DWORD PTR SS:[EBP+8]	Length = 200 (512.)
004012C3	E8 B4000000	CALL <JMP.&user32.MessageBoxA>	MessageBoxA
004012C4	E8 00020000	PUSH 200	hOwner = 7FFD9000
004012C5	00C0	OR EAX,EAX	Length = 200 (512.)

004012B5	E8 36010000	CALL <JMP.&kernel32.lstrcpA>	lstrcpA
004012B6	90	NOP	
004012B7	90	NOP	
004012B8	90	NOP	
004012B9	90	NOP	
004012BA	90	NOP	Style
004012BB	90	NOP	Title
004012BC	90	NOP	
004012BD	90	NOP	Text
004012BE	90	NOP	
004012BF	90	NOP	hOwner
004012C0	90	NOP	
004012C1	90	NOP	MessageBoxA
004012C2	90	NOP	
004012C3	90	NOP	
004012C4	90	NOP	Style
004012C5	90	NOP	Title
004012C6	90	NOP	
004012C7	90	NOP	Text
004012C8	90	NOP	
004012C9	90	NOP	hOwner
004012CA	90	NOP	
004012CB	90	NOP	MessageBoxA
004012CC	90	NOP	
004012CD	90	NOP	
004012CE	90	NOP	
004012CF	90	NOP	
004012D0	90	NOP	
004012D1	90	NOP	
004012D2	90	NOP	
004012D3	90	NOP	
004012D4	90	NOP	Style
004012D5	90	NOP	Title
004012D6	90	NOP	
004012D7	90	NOP	
004012D8	90	NOP	
004012D9	90	NOP	
004012DA	90	NOP	
004012DB	90	NOP	Text
004012DC	90	NOP	
004012DD	90	NOP	
004012DE	90	NOP	
004012DF	90	NOP	
004012E0	90	NOP	
004012E1	90	NOP	
004012E2	90	NOP	
004012E3	90	NOP	MessageBoxA
004012E4	90	NOP	
004012E5	90	NOP	
004012E6	90	NOP	
004012E7	90	NOP	
004012E8	68 00020000	PUSH 200	Length = 200 (512.)
004012E9	E8 42074000	CALL <JMP.&user32.MessageBoxA>	MessageBoxA

Ahora escribimos el injerto.

004012B8	90	NOP	
004012B9	90	NOP	
004012BA	68 49674000	PUSH CrackME_.00406749	Text = ""
004012BB	6A 64	PUSH 64	ControlID = 64 (100.)
004012BC	FF75 08	PUSH DWORD PTR SS:[EBP+8]	hWnd = 7FFD0000
004012BD	E8 AE5D6D75	CALL user32.SetDlgItemTextA	SetDlgItemTextA
004012BE	90	NOP	
004012BF	90	NOP	

Finalmente con **Resource Hack** cambiamos el aspecto del programa para que quede más profesional y listo. Tenemos pendiente hacer el keygen puro y duro, venga agur.



4. Notas finales

He probado el keygen en varios sistemas y **solo funciona en versiones de 32 bits**, en las de 64 bits da error y se cierra.

5. Enlaces

- Crackme
- Keygen + Injertos
- Entrada en el blog

6. Crackeando Crackmes by deurus

- https://deurus.info/archivos/crackmes/CrackME2_LaFarge.rar
- https://deurus.info/archivos/keygens/Lafarge_crackme2_injerto_keygen_deurus.rar