

Comment fisher CMSoldier de Craft.

Par Sp0ke

Outils :

- [CMSoldier](#)
- [Ollydbg 1.10](#)



- Un peu de cervelle

Bonjour à toutes et à tous dans ce tutorial nous allons tenter de comprendre le crackme CMSoldier de Craft.

Après avoir ouvert le crackme dans Ollydbg on l'exécute (F9) et on entre cinq caractères « 12345 » on ne peut pas en mettre plus et de ce fait on voit au bas du crackme le message ASCII « !! » »Access » »Denied » »!! ».

On regarde dans les SDR si on ne voit pas ce message, et il n'y est pas enfin du moins pas en clair lol .Donc on sort des SDR et on va faire une recherche dans la mémoire.

Pour cela on clique sur le bouton « M » en haut dans la fenêtre d'Ollydbg ou <ALT+M>., une fois dedans on fait clic droit et « Search » ou <Ctrl+B > et on entre notre message « Access Denied ».

On voit un premier message qui n'est pas le bon, un second qui lui aussi n'est pas bon et le troisième finalement qui est le bon c'est-à-dire avec la même syntaxe majuscule et les signes « !! ».

La même apparence en fait !

On pose donc un <break point on memory access> sur le premier caractère « ! » et dans le champs d'enregistrement on retire un caractère et on le remet pour qu'Ollydbg break et on fait « F9 ».

Et Bamm !! Ollydbg break à nouveau et on regarde dans la fenêtre des STACK dans Ollydbg et on voit le caractère « ! » et dessous on voit une chaîne de caractères ASCII « ## » »Caagqq » »Fglkgf » »## » qui est évidemment le message crypté de ASCII « !! » »Access » »Denied » »!! » .

Bon maintenant que l'on sait à quoi correspond cette chaîne cryptée. On retourne dans les SDR et on fait une recherche sur celle-ci en recherchant juste « Caagqq ».

Et on tombe sur ceci:

```
Text strings referenced in CMSoldie:CODE, item 2196 Address=004528D8 Disassembly=MOV EAX,CMSoldie.00452A50 Text string=ASCII "##"Caagqq"Tcnkfcvg"##"
```

Visiblement on s'aperçoit très vite que la terminaison de cette chaîne n'est pas celle recherchée alors que juste dessous celle-ci se trouve celle que l'on recherchait et en 5 occurrences de surcroît.

Comme toutes ces adresses sont rapprochées on double-clique sur la première Text string=ASCII « ## » »Caagqq » »Tcnkfcvg » »## » qui est sûrement le message « GoodBoy » lol.

```
String cryptée ==>ASCII « ## » »Caagqq » »Fglkgf » »## »
String non cryptée ==>ASCII « !! » »Access » »Denied » »!! »

String cryptée ==>ASCII « ## » »Caagqq » »Tcnkfcvg » »## »
String non cryptée ==>ASCII « !! » »Access » »Validate » »!! »
```

On arrive donc dans cette partie du code :

```
004528D8 |. B8 502A4500 MOV EAX,CMSoldie.00452A50 ; ASCII « ## » »Caagq » »Tcnkfcvg » »## »
```

Juste au dessus de l'adresse **004528D8** on peut déjà apercevoir 5 sauts conditionnels que l'on peut appeler aussi branchements conditionnels qui sont ici très significatifs aux adresses **0045284B**, **00452867**, **00452883**, **0045289F**, **004528BB** encadrées en **vert**, tous sautent vers le message «!! » »Access » »Denied » »!! » réparti sur 5 adresses différentes (**5 occurrences** dont je parlais juste au dessus) :ici encadrées en **rouge** surlignées en **jaune** image ci-dessous.

branchements conditionnels:	
instruction	état du flag
JNC	C=0
JC	C=1
JNO	O=0
JO	O=1
JNS	S=0
JS	S=1
JNZ	Z=0
JZ	Z=1

0045284B	3D FF000000	CMP EAX,0FF
0045284E	0F85 7E010000	JNZ CMSoldie.004529CF
00452851	8B86 3C030000	MOV EHX,DWORD PTR DS:[ESI+33C]
00452857	8B80 6C010000	MOV EAX,DWORD PTR DS:[EAX+16C]
0045285D	E8 EA9EFCFF	CALL CMSoldie.0041C74C
00452862	3D FF000000	CMP EAX,0FF
00452867	0F85 2D010000	JNZ CMSoldie.0045299A
0045286D	8B86 04030000	MOV EHX,DWORD PTR DS:[ESI+304]
00452873	8B80 6C010000	MOV EAX,DWORD PTR DS:[EAX+16C]
00452879	E8 CE9EFCFF	CALL CMSoldie.0041C74C
0045287F	3D FF000000	CMP EAX,0FF
00452883	0F85 DC000000	JNZ CMSoldie.00452965
00452889	8B86 30030000	MOV EHX,DWORD PTR DS:[ESI+330]
0045288F	8B80 6C010000	MOV EAX,DWORD PTR DS:[EAX+16C]
00452895	E8 B29EFCFF	CALL CMSoldie.0041C74C
0045289D	3D FF000000	CMP EAX,0FF
0045289F	0F85 88000000	JNZ CMSoldie.0045292D
004528A5	8B86 40030000	MOV EHX,DWORD PTR DS:[ESI+340]
004528AB	8B80 6C010000	MOV EAX,DWORD PTR DS:[EAX+16C]
004528B1	E8 969EFCFF	CALL CMSoldie.0041C74C
004528B6	3D FF000000	CMP EAX,0FF
004528BE	75 38	JNZ SHORT CMSoldie.004528F5

Nous allons donc poser un Break point sur chaque adresse de ces cinq sauts conditionnels en double-cliquant dessus ou en faisant « F2 ».

On fait <Ctrl+F2> pour recharger le crackme dans Ollydbg et <F9> pour l'exécuter à nouveau.

On entre ensuite notre serial bidon dans l'editbox pour moi « 12345 » et à la frappe du cinquième chiffre Ollydbg Break sur le premier saut en **0045284B**.

Donc c'est bon signe on est dans la partie cruciale du code pour obtenir un bon serial.

0045284E	0F85 7E01000	JNZ CMSoldie.004529CF	
00452851	8B86 3C03000	MOV EAX,DWORD PTR DS:[ESI+33C]	
00452857	8B80 6C01000	MOV EAX,DWORD PTR DS:[EAX+16C]	
0045285D	E8 EA9EFCFF	CALL CMSoldie.0041C74C	
00452862	3D FF000000	CMP EAX,0FF	
00452867	0F85 2D01000	JNZ CMSoldie.0045299A	
0045286D	8B86 0403000	MOV EAX,DWORD PTR DS:[ESI+304]	
00452873	8B80 6C01000	MOV EAX,DWORD PTR DS:[EAX+16C]	
00452879	E8 CE9EFCFF	CALL CMSoldie.0041C74C	
0045287E	3D FF000000	CMP EAX,0FF	
00452883	0F85 DC00000	JNZ CMSoldie.00452965	
00452889	8B86 3003000	MOV EAX,DWORD PTR DS:[ESI+330]	
0045288F	8B80 6C01000	MOV EAX,DWORD PTR DS:[EAX+16C]	
00452895	E8 829EFCFF	CALL CMSoldie.0041C74C	
0045289A	3D FF000000	CMP EAX,0FF	
0045289F	0F85 8800000	JNZ CMSoldie.0045292D	
004528A5	8B86 4003000	MOV EAX,DWORD PTR DS:[ESI+340]	
004528AB	8B80 6C01000	MOV EAX,DWORD PTR DS:[EAX+16C]	
004528B1	E8 969EFCFF	CALL CMSoldie.0041C74C	
004528B6	3D FF000000	CMP EAX,0FF	
004528BB	75 38	JNZ SHORT CMSoldie.004528F5	
004528BD	8B86 5403000	MOV EAX,DWORD PTR DS:[ESI+354]	
004528C3	8B40 68	MOV EAX,DWORD PTR DS:[EAX+68]	
004528C6	BA 00FF0000	MOV EDI,0FF00	
004528CB	E8 B99EFCFF	CALL CMSoldie.0041BF88	
004528D0	8D4D B8	LEA ECX,[LOCAL.20]	
004528D3	BA 9A020000	MOV EDI,29A	
004528D8	B8 502A4500	MOV EAX,CMSoldie.00452A50	
004528DD	E8 BAF3FFFF	CALL CMSoldie.00451C9C	
004528E2	8B55 B0	MOV EDI,[LOCAL.20]	
004528E5	8B86 5403000	MOV EAX,DWORD PTR DS:[ESI+354]	

ASCII "###""Caagqq""Tenkfvg""###"

La ligne rouge nous indique en fait que le premier caractère du serial entré précédemment n'est pas bon.

Donc on fait <F9> jusqu'à obtenir le message « !! » »Access » »Denied » »!! », et on modifie notre serial entré « 12345 » par « 22345 ».

Et là il break à nouveau mais cette fois-ci avec la flèche grise ce qui est bien pour nous car cela veut dire que notre premier caractère « 2 » est bon.

On fait <F9> on arrive sur le deuxième saut conditionnel en 00452867 qui lui par contre nous indique par la présence de la flèche rouge,

Que le deuxième caractère du serial entré n'est pas bon. Même manip que tout à l'heure <F9> jusqu'à obtenir le message « !! » »Access » »Denied » »!! »,

On retire le premier break point en 0045284B et on modifie notre serial entré « 22345 » par « 24444 ».

Là il break à l'adresse 00452867 avec cette fois-ci la flèche grise donc le début de notre bon serial est « 24 ».

Et on refait encore cette manip pour le troisième, quatrième, et cinquième caractère de notre serial jusqu'à ce que le serial soit valide.

Donc après quelques essais au clavier relativement rapides j'arrive au résultat suivant avec un serial valide comme « 24Y14 » que je m'empresse de tester hors Ollydbg.

Et voilà ce que m'affiche le crackme : « !! » »Access » »Validate » »!! » on a trouvé cool.

Et on remarque aussi que les petits carrés rouge répartis dans cinq colonnes sont disposés d'une certaine façon ce qui n'est pas un hasard !

Au contraire ce petit détail va nous aider à trouver très facilement et manuellement tous les autres bons caractères pour former des serials valides.

En fait pour le premier caractère on va simplement tester tous les caractères du clavier un à un (Chiffres, Lettres majuscule et minuscule),

Et à chaque fois que la première colonne ressemble à ceci :



Cela veut dire que les caractères sont bons et qu'il faut les noter soit sur papier ou alors dans Bloc Note de Windows ce qui formera une chaîne de caractères.

Et ainsi de suite pour les autres colonnes. Autrement dit chaque colonne possède sa propre table de chaîne de caractères,

que j'ai détaillé ci-dessous intitulé **Tableau chaînes de caractères**.==> "Quelle originalité !!! Ha!Ha!Ha!" 😊

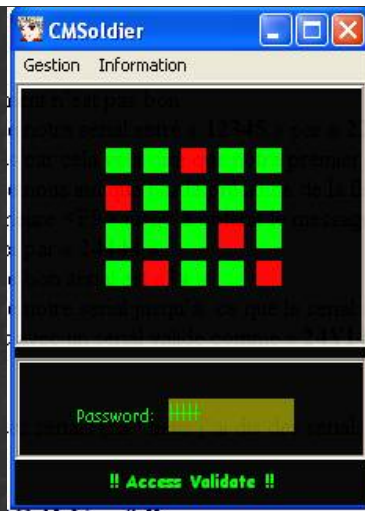


Tableau chaînes de caractères

Tableau des chaînes de caractères correspondant aux caractères N°1,N°2,N°3,N°4,N°5 des serials (ha ! tiens j'ai dis des serials !) avec les repères graphiques pour les retrouver facilement manuellement sans Ollydbg 5 minutes montre en main :

Repères graphiques du carré rouge correspondant à la chaîne			Chaînes	caractères du serial	5 CHAINES DE CARACTERES																											
		Chaîne 1	1er caractère du serial	2	9	Z	P	Q	4	L	V	p	v	&	"	ç																
		Chaîne 2	2e caractère du serial	4	5	7	0	R	O	H	X	B	è	à)	\$	*	!	;	,	a	z	e	r	t	y	g	j	k	l	c	n
		Chaîne 3	3e caractère du serial	Y	U	I	D	K	W	(=	m	:	x																		
		Chaîne 4	4e caractère du serial	1	3	6	8	A	E	T	S	F	G	J	M	C	N	é	'	-	_	u	i	o	h	f	d	s	q	w	b	
		Chaîne 5	5e caractère du serial	4	5	7	0	R	O	H	X	B	è	à)	\$	*	!	;	,	a	z	e	r	t	y	g	j	k	l	c	n

Dans ce tableau on peut constater que la Chaîne 2 et la Chaîne 5 sont en fait exactement les mêmes.

Si on fait un teste en essayant un caractère de chaque chaîne au hasard exemple « **QOWMX** » cela affiche le message de réussite.

Maintenant qu'on a tous les éléments requis il ne reste plus qu'à coder un petit « keygen » en utilisant la fonction Randomize (Aléatoire) :

Randomize ==> Rand sur chaque chaîne de caractères.

Ici il est codé en VB.NET 2008:

Voici le [KeygenCMSoldier](#) avec son source==> exemple ci-dessous:

Code source VB.NET 2008 :

```
Public Class Form1
```



```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

CMSoldier()

TextBox1.Text = serial

End Sub

#Region "Déclarations, fonction Rand() "

Dim serial As String

Public Function Rand(ByVal Low As CharacterRange, ByVal High As CharacterRange) As Long ' Cette fonction est utilisée pour créer un random integers(Nombres Aléatoires).

End Function

#End Region

#Region "Génération sérials [CMSoldier()]"

Private Function CMSoldier()

Dim String1 As String = "29ZPQ4LVpv&ç"
Dim String2 As String = "4570ROHXBèà$*!.,azertygjkln"
Dim String3 As String = "YUIDKW(=m:x"
Dim String4 As String = "1368AETSFGJMCNé'-_uiohfdsqwb"
Dim String5 As String = "4570ROHXBèà$*!.,azertygjkln"
Dim Char1 As Char
Dim Char2 As Char
Dim Char3 As Char
Dim Char4 As Char
Dim Char5 As Char

Char1 = String1.Chars(Rnd(Rnd() * 11) * 11)
Char2 = String2.Chars(Rnd(Rnd() * 28) * 28)
Char3 = String3.Chars(Rnd(Rnd() * 10) * 10)
Char4 = String4.Chars(Rnd(Rnd() * 27) * 27)
Char5 = String5.Chars(Rnd(Rnd() * 11) * 11)

serial = Char1 & Char2 & Char3 & Char4 & Char5

End Function

#End Region

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
If TextBox1.Text = "" Then ' Rien à copier fait planter le keygen, alors création d'une messageBoxe sous forme de SplashScreen
Dim ab As New SplashScreen3()
ab.ShowDialog(Me)
Else ' Copie le serial dans le Press-Papier
My.Computer.Clipboard.SetText(TextBox1.Text)
End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
Close()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
Dim ab As New SplashScreen1()
ab.ShowDialog(Me)
End Sub

```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
End Sub

Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
End Sub

Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label2.Click
End Sub

End Class
```

Merci à [Craft](#) pour ce crackme super sympa et très original, j'ai bien apprécié. 🤖

Remerciements à :

[Coolmen](#), [Kirjo](#), [ZIPPer](#), [mars](#), [taloche](#), [Hibou28](#), [Burner](#), [Néo](#) ainsi que tous les membres d' [xTx](#),
[uLysse_31/FFF](#) , [Dynasty](#), [donald](#), [baboon](#) , [Ezéqui3l](#) et tous les membres de [DeezDynasty](#)

[Sp0ke](#) [xTx TEAM](#) Janvier 2009.

Bonne et heureuse Année 2009 à tous.

