

Tutoriel du serialfishing et Selfkeygening de VERBE 5.6

Niveau moyen

Le soft est fourni avec ce tutoriel :

Outils requis :

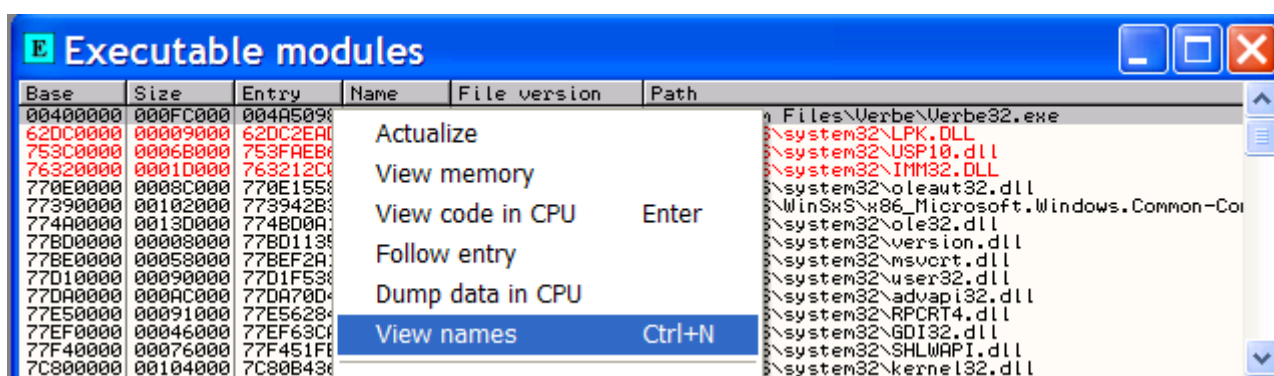
-Ollydbg

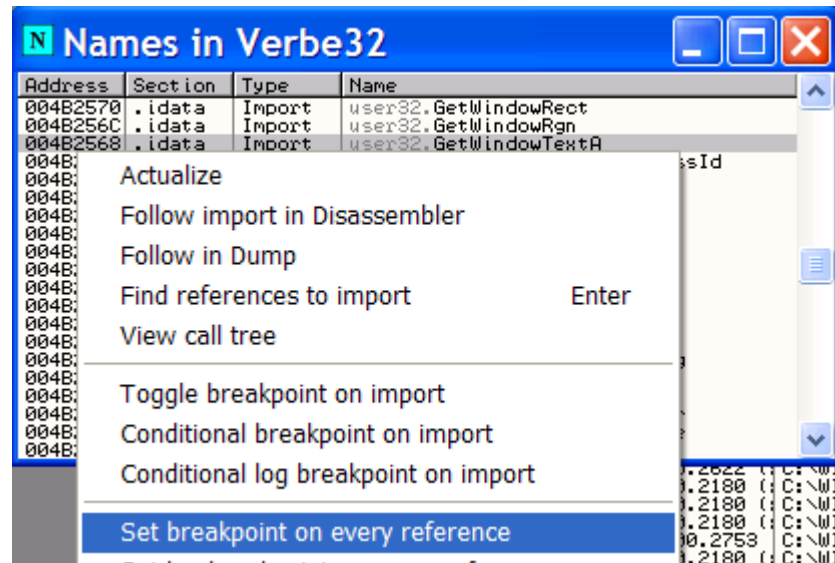
Théorie

Nous allons partir à la pêche au sérial, nous allons identifier la routine de vérification du sérial. On va utiliser les fonctions appelées « API » de Windows. Quand un programme veut afficher une boîte de dialogue ou un message, il utilise par exemple la fonction MessageBoxA. Pour récupérer les données entrées au clavier par l'utilisateur et pour connaître le nb de caractère entrées, le programme utilise la fonction GetDlgItemTextA ou précisément dans cet exemple GetWindowTextA. Cette fonction demande des paramètres pour savoir dans l'ordre, quel est le nb maximum de caractères, ou mettre le texte récupéré, l'identifiant de contrôle, le handle de la boîte de dialogue. Le programme va transférer ces paramètres à Windows par la pile. A la fin, la fonction retourne dans le registre eax le nb de caractères saisis le bon sérial.

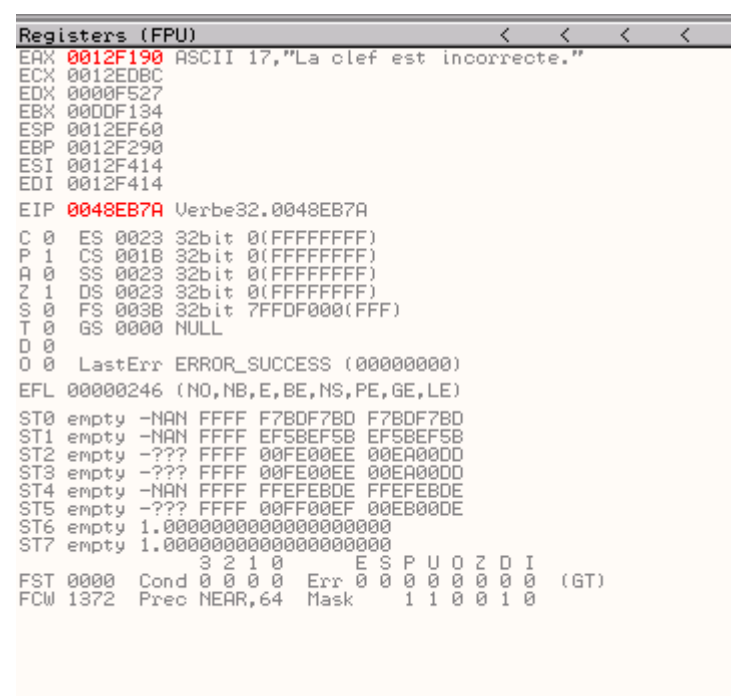
Pratique

Pour mettre un breakpoint sur une fonction, cliquez sur le menu « **View names** » puis sur **Exécutable modules**. Sélectionnez le programme VERBE32.EXE, clic droit et sélectionnez view names. Vous obtenez une liste des fonctions (si les fonctions ne sont pas triées par ordre alphabétique, clic droit et sort by names). Faites un clic droit sur la fonction getwindowtexta et sélectionnez « **set breakpoint on every références** ».





J'entre dans la boîte d'enregistrement pour mettre mon Nom et Serial bidon
Et je valide et Ollydbg break dessus sans problème.



Donc Ollydbg Break sur cette Api je trace avec F8 pour voir si notre serial bidon apparaît et notre bon serial aussi, donc rien n'y fait je tombe directement sur le message « La clef est incorrecte » image au-dessus à la fin du traçage en 0048EB74.

0048EB39	> 8080 00FFFFFF	LEA ECX,DWORD PTR SS:[EBP-100]	
0048EB3F	. BA 7CEC4800	MOV EDX,Verbe32.0048EC7C	
0048EB44	. B8 20EC4800	MOV EAX,Verbe32.0048EC20	
0048EB49	. E8 1E93FCFF	CALL Verbe32.00457E6C	ASCII 0B,"78547526984"
0048EB4E	. A1 10C84A00	MOV EAX,DWORD PTR DS:[4AC810]	
0048EB53	. 8B00	MOV EAX,DWORD PTR DS:[EAX]	
0048EB55	. FE80 10030000	INC BYTE PTR DS:[EAX+310]	
0048EB5B	> 33D2	XOR EDX,EDX	
0048EB5D	. 8B83 D8020000	MOV EAX,DWORD PTR DS:[EBX+2D8]	
0048EB63	. 8B08	MOV ECX,DWORD PTR DS:[EAX]	
0048EB65	. FF51 60	CALL DWORD PTR DS:[ECX+60]	
0048EB68	. B8 D0070000	MOV EAX,7D0	
0048EB6D	. E8 96FCFFFF	CALL Verbe32.0048E808	
0048EB72	. 6A 00	PUSH 0	
0048EB74	. 8085 00FFFFFF	LEA EAX,DWORD PTR SS:[EBP-100]	La clef est incorrecte.
0048EB7A	. 66:8B0D 24EC4	MOV CX,WORD PTR DS:[48EC94]	
0048EB81	. B2 02	MOV DL,2	
0048EB83	. E8 1495FCFF	CALL Verbe32.0045809C	
0048EB88	. A1 10C84A00	MOV EAX,DWORD PTR DS:[4AC810]	
0048EB8D	. 8B00	MOV EAX,DWORD PTR DS:[EAX]	
0048EB8F	. 80B8 10030000	CMF BYTE PTR DS:[EAX+310],0	
0048EB96	. 0F97C0	SETA AL	
0048EB99	. 8B15 10C84A00	MOV EDX,DWORD PTR DS:[4AC810]	Verbe32.004B1390
0048EB9F	. 8B12	MOV EDX,DWORD PTR DS:[EDX]	
0048EBA1	. 8B92 14030000	MOV EDX,DWORD PTR DS:[EDX+314]	
0048EBA7	. 0A82 80030000	OR AL,BYTE PTR DS:[EDX+380]	
0048EBAD	< 74 11	JE SHORT Verbe32.0048EBC0	
0048EBAF	. 8BC3	MOV EAX,EBX	
0048EBB1	. E8 E6AFFBFF	CALL Verbe32.00449B9C	
0048EBB6	. B8 D0070000	MOV EAX,7D0	
0048EBBB	. E8 48FCFFFF	CALL Verbe32.0048E808	
0048EBC0	> 33C0	XOR EAX,EAX	
0048EBC2	. 5A	POP EDX	
0048EBC3	. 59	POP ECX	
0048EBC4	. 59	POP ECX	
0048EBC5	. 64:8910	MOV DWORD PTR FS:[EAX],EDX	
0048EBC8	. 68 FBEB4800	PUSH Verbe32.0048EBFB	
0048EBCD	> 8085 F0FCFFFF	LEA EAX,DWORD PTR SS:[EBP-310]	
0048EBD3	. BA 02000000	MOV EDI,2	
0048EBD8	. E8 9B52F7FF	CALL Verbe32.00403E78	
0048EBDD	. 8085 F8FCFFFF	LEA EAX,DWORD PTR SS:[EBP-308]	
0048EBE3	. E8 6C52F7FF	CALL Verbe32.00403E54	
0048EBE8	. 8085 FCDFFFFF	LEA EAX,DWORD PTR SS:[EBP-204]	
0048EBF5	. 6A 00	PUSH 0	

Address	ASCII dump
0012F190	La clef est incorrecte. L:*.EFD.
0012F1B0	3c../.0*+.fr.†g±.4g0w+*.3c..
0012F1D0	/..0*+.!*i.=% ... (=*.!*i.T=.
0012F1F0	-i0w.-²ΔT=*.Zë0w¶=*.#ë0w3c..¿3l.
0012F210	"3l.¶...0.....>...../..0*
0012F230	0.....fr.†g±=*.â=*.¶g.ëw0ë0w
0012F250	*ë0w@0w.....Ui:wâ=*.0ë0w
0012F270	â5l.....rë0w....Ui:wâë0w!=*.....

Donc j’applique la méthode du deadlisting c’est bien aussi parfois je remonte donc un peu plus haut pour voir et je tombe ici sur TEST AL, AL heumm !! Intéressant un test suivi d’un saut et au dessus un CALL je pose un Breackpoint sur ce CALL et je recharge VERBE.EXE et j’entre un nom bidon BPX_27 et un serial tout aussi bidon 123456789 BINGO !!!!! Ollydbg break et m’affiche mon Non et mon serial bidon deuxième image ci-dessous : Ce CALL est sûrement celui de la vérification et génération du bon serial.

0048E918	. E8 8277FFFF	CALL Verbe32.0048E900	
0048E91A	. 8D95 FCFCFFFF	LEA EDX,DWORD PTR SS:[EBP-304]	
0048E920	. A1 10C84A00	MOV EAX,DWORD PTR DS:[4AC810]	
0048E925	. 8B00	MOV EAX,DWORD PTR DS:[EAX]	
0048E927	. 8B80 14030000	MOV EAX,DWORD PTR DS:[EAX+314]	
0048E92D	. 59	POP ECX	
0048E92E	. E8 C13CF0FF	CALL Verbe32.004525F4	
0048E933	. 84C0	TEST AL,AL	
0048E935	. 0F84 FE010000	JE Verbe32.0048EB39	
0048E938	. 8D95 FCFCFFFF	LEA EDX,DWORD PTR SS:[EBP-204]	
0048E941	. 8B83 00020000	MOV EAX,DWORD PTR DS:[EBX+200]	
0048E947	. E8 931CF0FF	CALL Verbe32.004305E4	
004525F4=Verbe32.004525F4			

Registers (FPU)	
EAX	00086734
ECX	0012F090 ASCII 09,"123456789"
EDX	0012EF8C ASCII 06,"BPX_27"
EBX	000E0070
ESP	0012EF64
EBP	0012F290
ESI	0012F414
EDI	0012F414
EIP	0048E92E Verbe32.0048E92E

J'entre dans le CALL avec F7 et je trace, j'arrive sur un autre CALL ! je pose un BP dessus également, j'entre dans ce CALL avec F7 et trace comme tout à l'heure, et là je vois que mon Nom BPX_27 est traité pour créer mon vrai serial, je tombe sur un 1er sérial \$2282E46E que je m'empresse de tester. Mais hélas il ne fonctionne pas donc je refais la même manipulation. Je vois apparaître un autre sérial : \$E9F2DDA4, je le test aussi, mais il n'est pas bon non plus, décidément ! Je commence à penser que le bon sérial ne va pas être si facile que ça à trouver. Enfin bref, je ne me décourage pas, je continue donc et là que vois-je apparaître ? un autre serial \$71A1FCFA, je teste à nouveau et là BINGO !!! Le bon sérial donc je prends des notes comme d'habitude :

00452520	. E8 6DFFFFFF	CALL Verbe32.0045239C	
0045252F	. 8D9424 840000	LEA EDX,DWORD PTR SS:[ESP+84]	
00452536	. 8BC6	MOV EAX,ESI	
00452538	. B1 09	MOV CL,9	
0045253A	. E8 E507FBFF	CALL Verbe32.00402D24	
0045253F	. 81C4 90000000	ADD ESP,90	
00452545	. 5F	POP EDI	
00452546	. 5E	POP ESI	
00452547	. 5B	POP EBX	
Stack address=0012EF2C, (ASCII 09,"\$71A1FCFA")			
EDX=0012EE94, (ASCII 09,"\$71A1FCFA")			

Bon d'accord, j'ai mon serial ok ! Mais ma satisfaction n'est que de courte durée, car avoir trouvé mon sérial c'est cool, mais je n'arrive pas à m'en contenter. Alors je décide de faire autre chose de plus drôle.

Comme je ne suis pas un expert en langage ASM pour coder un keygen, il me vient une idée, si j'essayais de modifier le code du programme de telle manière qu'en entrant notre Nom et serial bidon il m'affiche le bon serial correspondant à mon Nom, au lieu du message « La clef est incorrecte » un selfKegen en fait, ce serait cool non ?

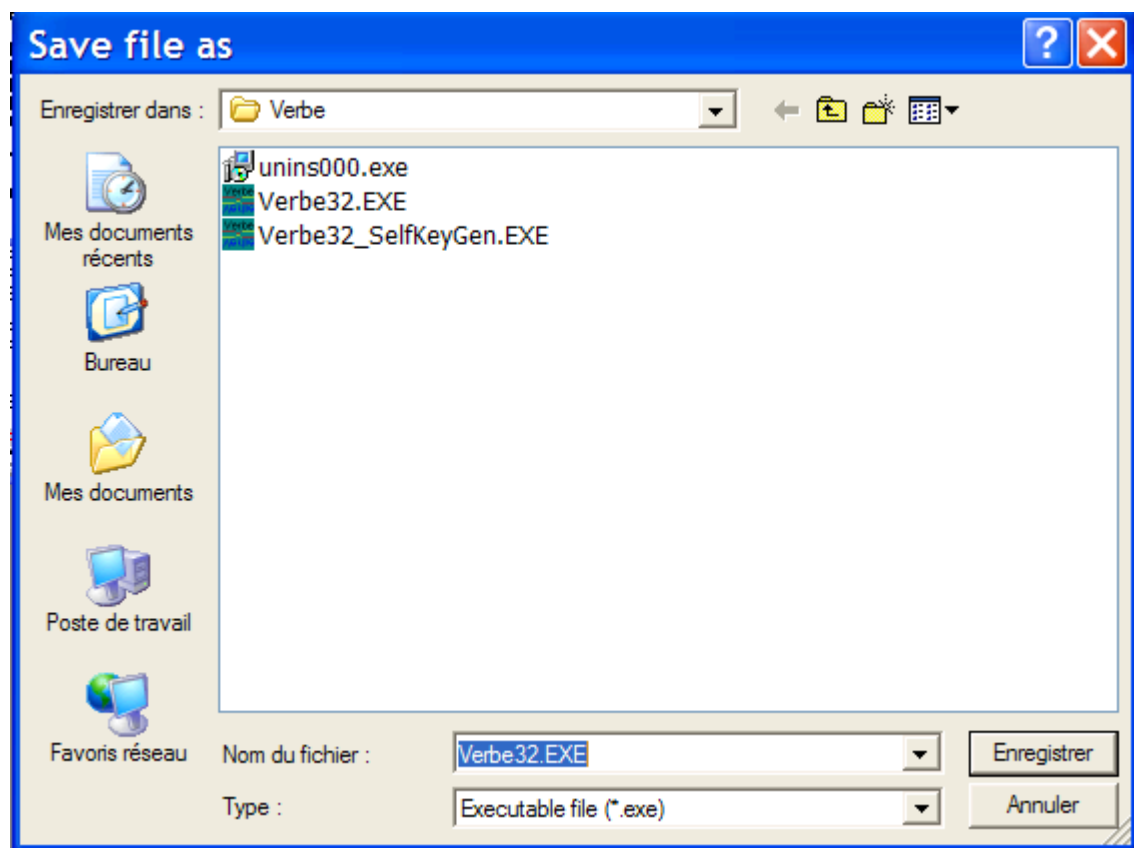
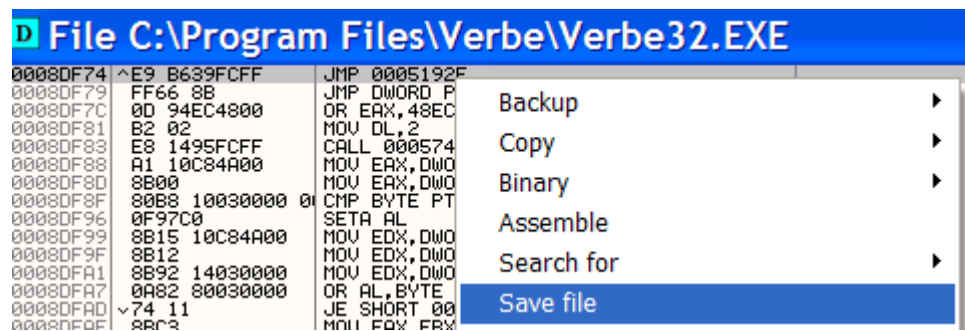
Donc sachant que l'adresse 0048EB74 LEA EAX, DWORD PTR SS : [EBP-100] nous affiche la boîte de dialogue « La clef est incorrecte ». Et que l'adresse 0045252F LEA EDX, DWORD PTR SS : [ESP+84] affiche le bon sérial, il faut alors procéder à quelques modifications du code :

La ligne de code à cette adresse devient alors : 0045252F **JUMP 0048EB74**

Et il faut modifier aussi l'adresse : 0048EB74 LEA EAX, DWORD PTR SS : [EBP-100] par 0048EB74 **LEA EAX, DWORD PTR SS : [ESP+84]**.

Cela à pour but donc d'effectuer un saut direct sur le message de la boite de dialogue, j'enregistre la modification comme cela :

0040EB72	6A 00	PUSH 0	
0040EB74	8D 84 24 84 00 00	LEA EAX,DWORD PTR SS:[ESP+84]	message "La clef est
0040EB7B	90	NOP	
0040EB7C	0D 94 EC 48 00	OR EAX,48EC94	
0040EB81	B2 02	MOV DL,2	
0040EB83	E8 14 95 FC FF	CALL Verbe32.0045809C	
0040EB88	A1 10 C8 4A 00	MOV EAX,DWORD PTR DS:[4AC810]	
0040EB8D	8B 00	MOV EBX,DWORD PTR DS:[EAX]	
Jump from 0045252F			



Je sauvegarde ma modification en donnant un nouveau nom au Verbe32.EXE en Verbe32_SelfKeyGen.EXE je ferme Ollydbg et essayes mon nouvel EXE



Voilà j'ai bien mon serial qui s'affiche dans cette boite dialogue et en plus sans plantage de l'EXE.

Génial non ?

Copyright © 2006

Sp0ke.

@++