'FC search' Reversme - Revme #1(Neitsa)

solution proposée par BeatriX

Le projet ultra secret 'FC search' est un moteur surpuissant de recherche dans la base du forum de FC. Malheureusement on a engager des programmeurs au rabais, et ceux ci n'ont pas été capable de terminer notre programme convenablement.Par dessus le marché, un virus très virulent s'est introduit dans notre réseau et a effacé nos code sources et modifié quelque peu le programme.

Nous avons besoin d'un Reverser qui puisse finir le boulot, après différent contact dans les milieux underground, nous pensons que vous êtes à même d'accomplir cette tâche.

Héhé ! Voilà donc une mission que nous relevons avec courage. Je vous propose donc d'étudier une façon de résoudre ce reverseme qui nous amènera à effectuer les tâches suivantes :

1) Détourner le programme de son exécution «normal» et écrire du code ASM pour parvenir à nos fins.
2) Modifier manuellement les ressources.
3) Ajouter une API en modifiant l' Image Import Descriptor et la table des jumps.

1 . PREPARATION

Je tiens à préciser que pour lire ce tutorial, il est préférable d'avoir pris un peu connaissance du fonctionnement de l'exe et éventuellement d'avoir le programme sous les yeux (papier ou debugger).

Nous allons tout d'abord jeter un oeil à la structure du fichier. Avec LordPE, nous pouvons voir les sections suivantes :

.code	RVA = 1000
.rdata	RVA = 2000
.data	RVA = 3000
.rscr	RVA = 4000

Nous pouvons aussi regarder les ressources et nous obtenons 5 ressources :

```
1 ) ICON
2 ) MENU
3 ) DIALOG
4 ) ICON GROUP
5 ) VERSION INFO
```

Regardons de plus près la ressource MENU (avec ResHack) :

```
10000 MENUEX
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
{
POPUP "&File", 10001, MFT_STRING, MFS_ENABLED, 0
{
      MENUITEM "&Search", 10006, MFT_STRING, MFS_ENABLED
      MENUITEM "", 0, MFT_SEPARATOR, MFS_ENABLED
      MENUITEM "&Exit", 10002, MFT_STRING, MFS_ENABLED
}
POPUP "Browser", 10003, MFT_STRING, MFS_ENABLED, 0
{
```

```
        MENUITEM "&FireFox", 10004, MFT_STRING, MFS_ENABLED
        MENUITEM "&IE", 10005, MFT_STRING, MFS_ENABLED
}
POPUP "&About", 10007, MFT_STRING, MFS_DISABLED, 0
{
        MENUITEM "&Greetz", 10008, MFT_STRING, MFS_ENABLED
        MENUITEM "&About", 10009, MFT_STRING, MFS_GRAYED
}
}
```

Tiens donc, le menu About est désactivé et le sous menu About est grisé ! Avant d'aller plus loin dans l'investigation, occupons nous de ces deux options !

2) ACTIVATION DE TOUS LES MENUS

Allez, on va la faire à la loyale ! Avec l'éditeur héxadécimal de OllyDbg, on obtient ceci pour le début de la section .rscr :

```
00404000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 00  ..............#.
00404010  03 00 00 00 38 00 00 80 04 00 00 00 50 00 00 80  #...8..€#...P..€
00404020  05 00 00 00 68 00 00 80 0E 00 00 00 80 00 00 80  #...h..€#...€..€
00404030  10 00 00 00 98 00 00 80 00 00 00 00 00 00 00 00  #...˜..€........
00404040  00 00 00 00 00 00 01 00 01 00 00 00 B0 00 00 80  ......#.#...°..€
00404050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
00404060  10 27 00 00 C8 00 00 80 00 00 00 00 00 00 00 00  #'..È..€........
00404070  00 00 00 00 00 00 01 00 65 00 00 00 E0 00 00 80  ......#.e...à..€
00404080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
00404090  C8 00 00 00 F8 00 00 80 00 00 00 00 00 00 00 00  È...ø..€........
004040A0  00 00 00 00 00 00 01 00 01 00 00 00 10 01 00 80  ......#.#...##.€
004040B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
004040C0  09 04 00 00 28 01 00 00 00 00 00 00 00 00 00 00  .#..(#..........
004040D0  00 00 00 00 00 00 01 00 09 04 00 00 38 01 00 00  ......#..#..8#..
004040E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
004040F0  09 04 00 00 48 01 00 00 00 00 00 00 00 00 00 00  .#..H#..........
00404100  00 00 00 00 00 00 01 00 09 04 00 00 58 01 00 00  ......#..#..X#..
00404110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
00404120  09 04 00 00 68 01 00 00 D0 45 00 00 A8 0C 00 00  .#..h#..ÐE..¨...
00404130  00 00 00 00 00 00 00 00 A0 44 00 00 2C 01 00 00  ........ D..,#..
00404140  00 00 00 00 00 00 00 00 80 41 00 00 DC 01 00 00  ........€A..Ü#..
00404150  00 00 00 00 00 00 00 00 78 52 00 00 14 00 00 00  ........xR..#...
00404160  00 00 00 00 00 00 00 00 60 43 00 00 40 01 00 00  ........`C..@#..
00404170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00404180  01 00 FF FF 00 00 00 00 00 00 00 00 40 08 CA 10  #.ÿÿ........@#Ê#
```

Que de zéros !!

Analysons cette partie de la section pour retrouver les informations concernant le menu. Ça ne sert pas vraiment à grand chose mais je crois, sans vouloir trop m'avancer, que peu de monde sait vraiment comment fonctionne clairement cette partie du format PE.

Voilà donc un premier tableau composé de 6 parties : les characteristics, le TimeDateStamp, la MajorVersion, la MinorVersion, le NumberOfNameEntries et le NumberOfIdEntries qui vaut 5. Ceci signifie que nous avons 5 ID dans nos ressources.

```
00404000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 00  ..............#.
```

Maintenant, nous avons une ***première structure*** qui regroupe les 5 ID. Ici, pour chaque ID, nous disposons de son TYPE. Je rappelle pour mémoire les différents TYPES :

```
0x0001 = Cursor
0x0002 = Bitmap
0x0003 = Icon
0x0004 = Menu
0x0005 = Dialog
0x0006 = String Table
0x0007 = Font Directory
0x0008 = Font
0x0009 = Accelerators Table
0x000A = RC Data (custom binary data)
0x000B = Message table
0x000C = Group Cursor
0x000E = Group Icon
0x0010 = Version Information
0x0011 = Dialog Include
0x0013 = Plug'n'Play
0x0014 = VXD
0x0015 = Animated Cursor
0x2002 = Bitmap (new version)
0x2004 = Menu (new version)
0x2005 = Dialog (new version)
```

Par exemple, nous voyons donc en 404010 le nombre 03 qui indique ICON. Cherchons le menu. Son ID = 04h et se trouve en 404018. Le DWORD suivant contient la valeur 80000050. Ne retenons que les bits de poids faible : 50h qui indique l'offset du prochain tableau concernant les menus.

```
00404010  03 00 00 00 38 00 00 80 04 00 00 00 50 00 00 80  #...8..€#...P..€
00404020  05 00 00 00 68 00 00 80 0E 00 00 00 80 00 00 80  #...h..€#...€..€
00404030  10 00 00 00 98 00 00 80 00 00 00 00 00 00 00 00  #...˜..€........
00404040  00 00 00 00 00 00 01 00 01 00 00 00 B0 00 00 80  ......#.#...°..€
```

Voici le ***deuxième tableau*** qui indique l'ID du menus. En 404060, on peut voir l'ID = 2710h du menu. En 404064, le C8h indique l'offset du prochain tableau traitant des menus.

```
00404050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
00404060  10 27 00 00 C8 00 00 80 00 00 00 00 00 00 00 00  #'..È..€........
00404070  00 00 00 00 00 00 01 00 65 00 00 00 E0 00 00 80  ......#.e...à..€
00404080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
00404090  C8 00 00 00 F8 00 00 80 00 00 00 00 00 00 00 00  È...ø..€........
004040A0  00 00 00 00 00 00 01 00 01 00 00 00 10 01 00 80  ......#.#...##.€
004040B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
```

Voici le ***troisième tableau*** qui indique la langue. En 4040D8, la langue = 409h, soit 1033 en décimal. En 4040DC, le 138h est l'offset du prochain sous-tableau traitant des menus.

```
004040C0  09 04 00 00 28 01 00 00 00 00 00 00 00 00 00 00  .#..(#..........
004040D0  00 00 00 00 00 00 01 00 09 04 00 00 38 01 00 00  ......#..#..8#..
004040E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
```

```
004040F0  09 04 00 00 48 01 00 00 00 00 00 00 00 00 00 00  .#..H#..........
00404100  00 00 00 00 00 00 01 00 09 04 00 00 58 01 00 00  ......#..#..X#..
00404110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ..............#.
00404120  09 04 00 00 68 01 00 00 D0 45 00 00 A8 0C 00 00  .#..h#..ÐE..¨...
```

Voici le ***dernier tableau*** qui indique l'offset des datas des ressources ainsi que leur taille. L'offset = 44A0h et la taille = 12Ch.

```
00404130  00 00 00 00 00 00 00 00 A0 44 00 00 2C 01 00 00  ........ D..,#..
00404140  00 00 00 00 00 00 00 00 80 41 00 00 DC 01 00 00  ........€A..Ü#..
00404150  00 00 00 00 00 00 00 00 78 52 00 00 14 00 00 00  ........xR..#...
00404160  00 00 00 00 00 00 00 00 60 43 00 00 40 01 00 00  ........`C..@#..
00404170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00404180  01 00 FF FF 00 00 00 00 00 00 00 00 40 08 CA 10  #.ÿÿ........@#Ê#
```

Donc les datas du menu se trouvent en 4044A0h. Allons voir ce qu'il s'y passe :

```
004044A0  01 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00  #.#.............
004044B0  11 27 00 00 01 00 26 00 46 00 69 00 6C 00 65 00  #'..#.&.F.i.l.e.
004044C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004044D0  16 27 00 00 00 00 26 00 53 00 65 00 61 00 72 00  #'....&.S.e.a.r.
004044E0  63 00 68 00 00 00 00 00 00 08 00 00 00 00 00 00  c.h.....#.......
004044F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00404500  12 27 00 00 80 00 26 00 45 00 78 00 69 00 74 00  #'..€.&.E.x.i.t.
00404510  00 00 00 00 00 00 00 00 00 00 00 00 13 27 00 00  ............#'..
00404520  01 00 42 00 72 00 6F 00 77 00 73 00 65 00 72 00  #.B.r.o.w.s.e.r.
00404530  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00404540  14 27 00 00 00 00 26 00 46 00 69 00 72 00 65 00  #'....&.F.i.r.e.
00404550  46 00 6F 00 78 00 00 00 00 00 00 00 00 00 00 00  F.o.x...........
00404560  15 27 00 00 80 00 26 00 49 00 45 00 00 00 00 00  #'..€.&.I.E.....
00404570  00 00 00 00 02 00 00 00 17 27 00 00 81 00 26 00  ....#...#'.. .&.
00404580  41 00 62 00 6F 00 75 00 74 00 00 00 00 00 00 00  A.b.o.u.t.......
00404590  00 00 00 00 00 00 00 00 18 27 00 00 00 00 26 00  ........#'....&.
004045A0  47 00 72 00 65 00 65 00 74 00 7A 00 00 00 00 00  G.r.e.e.t.z.....
004045B0  00 00 00 00 01 00 00 00 19 27 00 00 80 00 26 00  ....#...#'..€.&.
004045C0  41 00 62 00 6F 00 75 00 74 00 00 00 00 00 00 00  A.b.o.u.t.......
```

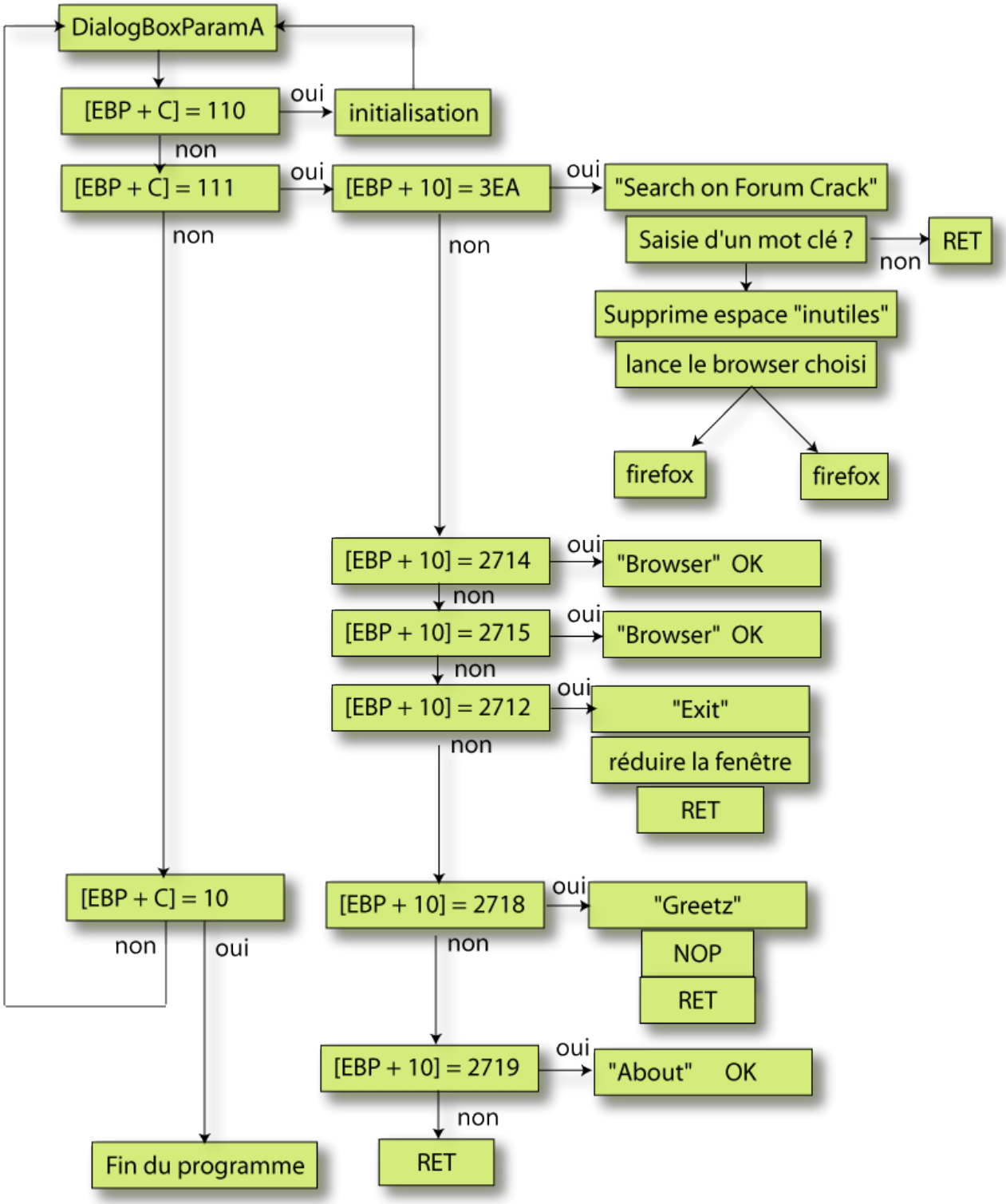
Voici donc chaque menu et chaque sous-menu. Pour chaque menu, on dispose de trois informations :

1er DWORD = état du menu. (Enabled = 0 ; disabled = 2 ; grayed = 1) (orange foncé)
2me DWORD = ID du menu.(orange clair)
3me : Chaîne au format Unicode qui représente le texte. (jaune)

On passe donc tous les états à 00 et nos deux menus deviennent donc accessibles.

3) ANALYSE DU CODE

Pour simplifier les explications, j'ai réalisé un petit schéma qui illustre le comportement du programme. Il s'agit donc d'une simple DialogBox. Vous pouvez repérer sa procédure grâce à la fonction DialogBoxParamA. Tout commence en 401119.



Nous avons donc la traditionnel série de tests pour savoir si un contrôle a été activé et le cas échéant, lequel. On peut dors et déjà détecter les «grosses» erreurs.

Bouton «Search On Forum Crack» : Nous sommes censé lancer le browser choisi or, dans la configuration actuelle, on ne peut lancer que firefox.

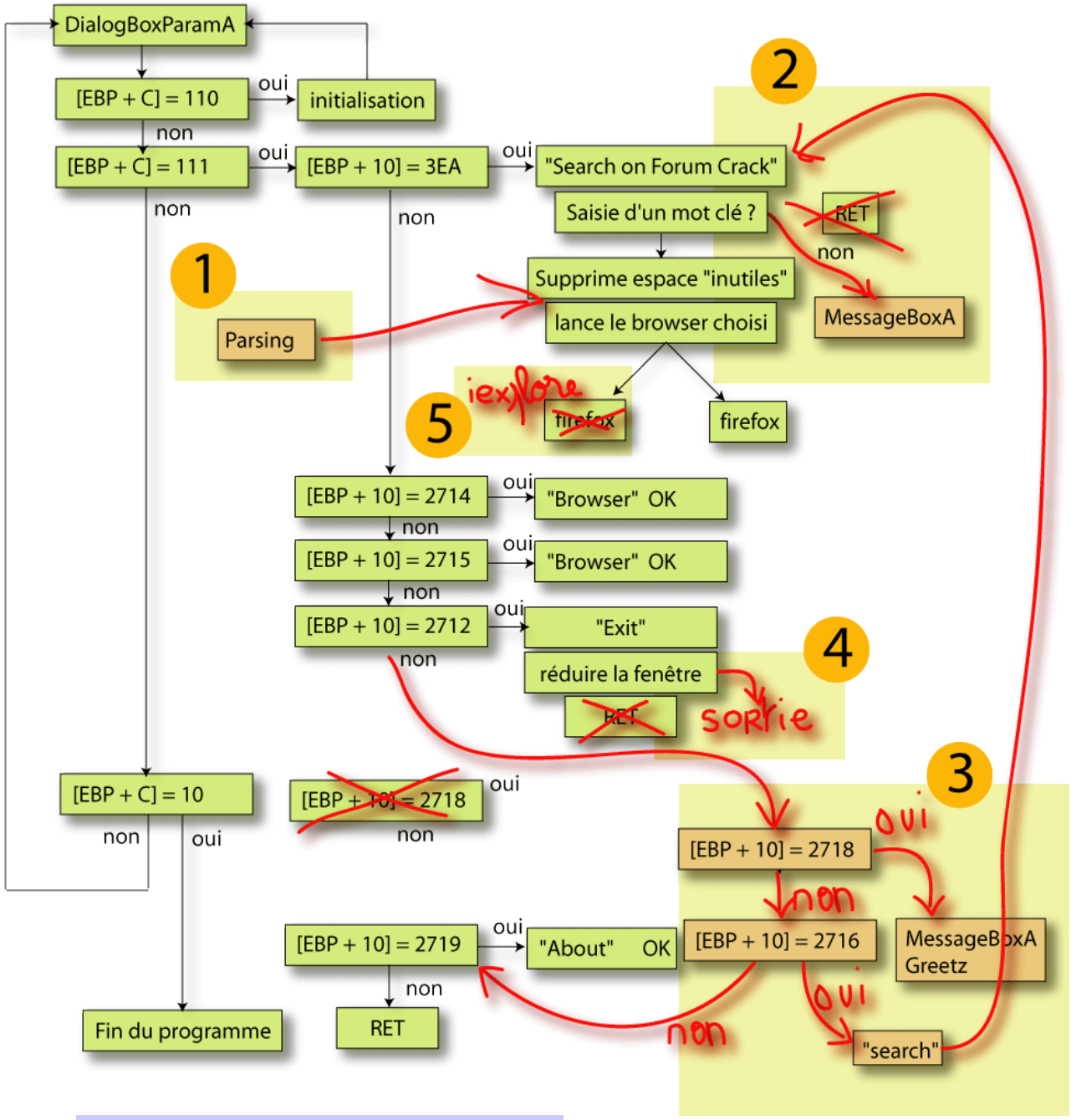
Menu «Exit» : Il ne fait que réduire la fenêtre !

Menu «Greetz» : Il ne fait rien ! (NOP)

Menu «Search» : Son ID = 2716 n'apparaît pas dans la série de tests !

4) REPARATIONS DU PROGRAMME.

Voilà donc les différentes modifications que j'ai apporté au programme :



Remarquez l'aspect artisanal du travail !

4 - 1) LE PARSING

Il s'agit de transformer une saisie de mot clé du style «IAT Process» en «IAT+Process». Pour ça, nous allons détourner le programme juste avant qu'il lance le browser, scanner la chaîne à la recherche d'espace et le remplacer par + . Le lancement du browser commence en 401088.

```
00401088  PUSH 3EB                                  ; /ButtonID = 3EB (1003.)
0040108D  PUSH DWORD PTR SS:[EBP+8]                 ; |hWnd
00401090  CALL <JMP.&user32.IsDlgButtonChecked>     ; \IsDlgButtonChecked
00401095  CMP EAX,1
00401098  JNZ SHORT FCSearch.004010BA
0040109A  PUSH 3                                    ; /IsShown = 3
0040109C  PUSH 0                                    ; |DefDir = NULL
0040109E  PUSH DWORD PTR DS:[403354]                ; |Parameters = NULL
004010A4  PUSH FCSearch.00403132                    ; |FileName = "firefox"
004010A9  PUSH FCSearch.0040312D                    ; |Operation = "open"
004010AE  PUSH DWORD PTR SS:[EBP+8]                 ; |hWnd
004010B1  CALL <JMP.&shell32.ShellExecuteA>         ; \ShellExecuteA
004010B6  LEAVE
004010B7  RETN 4
004010BA  PUSH 3EC                                  ; /ButtonID = 3EC (1004.)
004010BF  PUSH DWORD PTR SS:[EBP+8]                 ; |hWnd
004010C2  CALL <JMP.&user32.IsDlgButtonChecked>     ; \IsDlgButtonChecked
004010C7  CMP EAX,1
004010CA  JNZ SHORT FCSearch.004010E7
004010CC  PUSH 3                                    ; /IsShown = 3
004010CE  PUSH 0                                    ; |DefDir = NULL
004010D0  PUSH FCSearch.004030EE                    ; |Parameters =
"http://www.hackatak.org/forumcrack/search.php?search_keywords="
004010D5  PUSH FCSearch.00403132                    ; |FileName = "firefox"
004010DA  PUSH FCSearch.0040312D                    ; |Operation = "open"
004010DF  PUSH DWORD PTR SS:[EBP+8]                 ; |hWnd
004010E2  CALL <JMP.&shell32.ShellExecuteA>         ; \ShellExecuteA
004010E7  LEAVE
004010E8  RETN 4
```

Détournons le programme vers 4012AA (la fameuse «cave»):

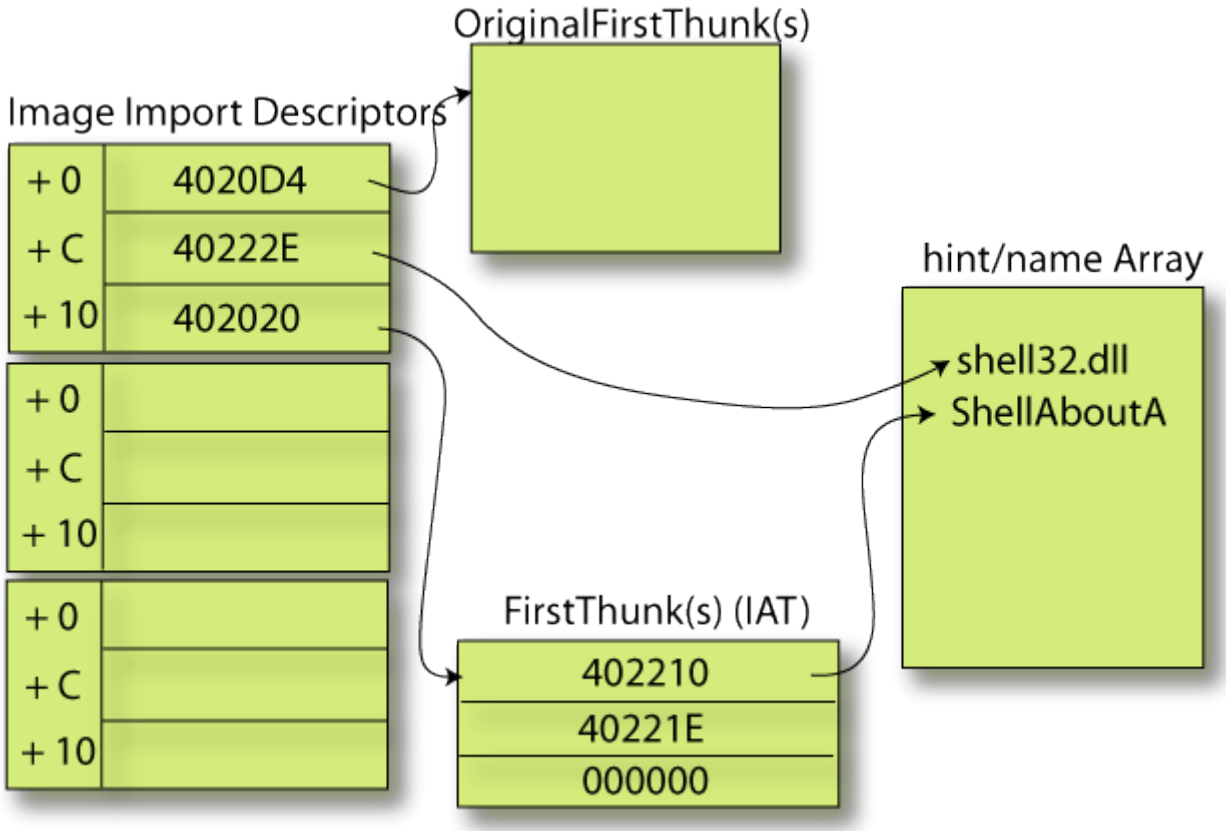
```
00401088  JMP FCSearch.004012AA
```

En 4012AA, nous allons donc remplacer les 20h par des 2Bh :

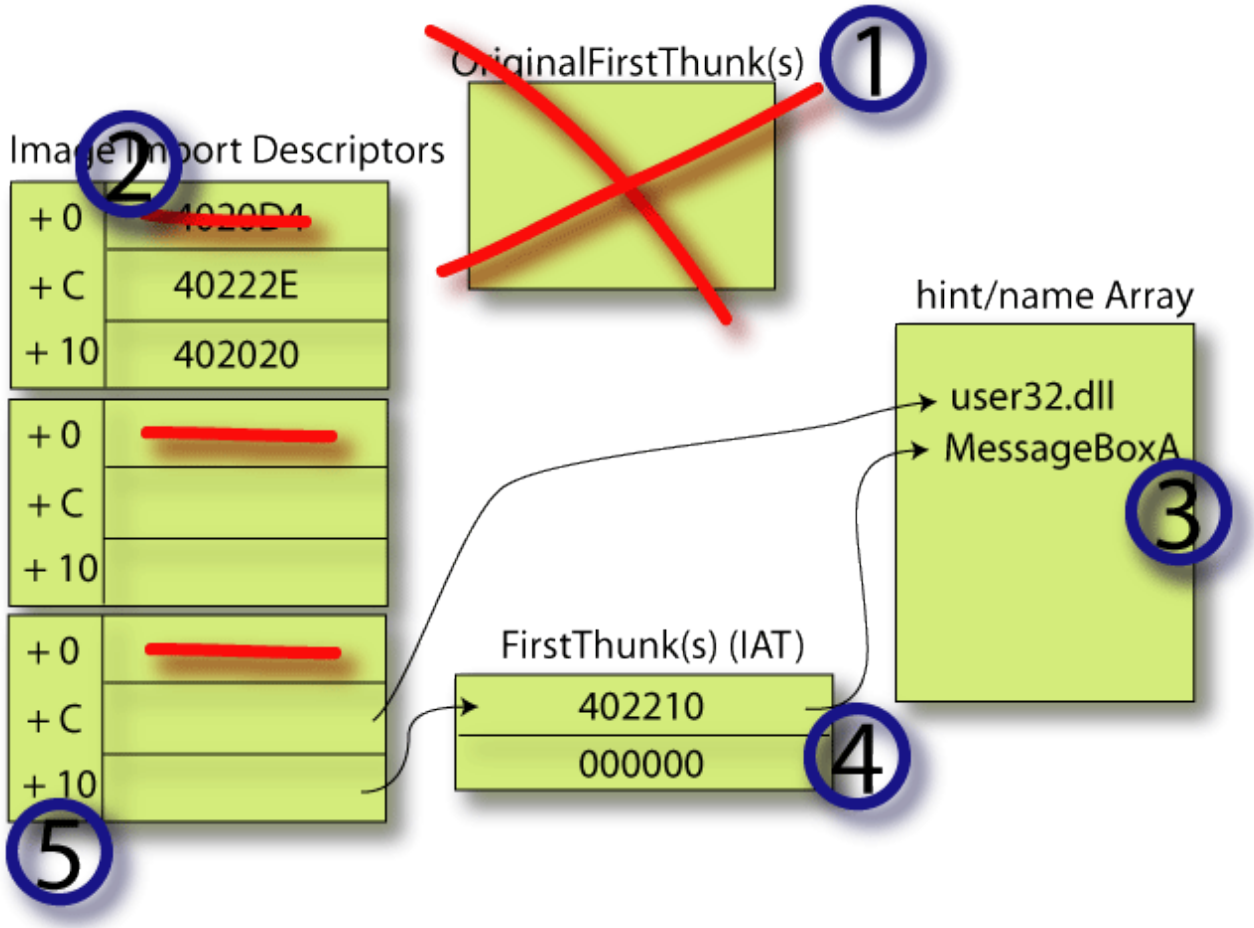
```
004012AA  MOV EDX,DWORD PTR DS:[403354]         ; récupère l'offset de la chaîne
004012B0  CMP BYTE PTR DS:[EDX],20              ; cherche un espace
004012B3  JNZ SHORT FCSearch.004012B8
004012B5  MOV BYTE PTR DS:[EDX],2B              ; remplace par un +
004012B8  INC EDX
004012B9  CMP BYTE PTR DS:[EDX],0
004012BC  JNZ SHORT FCSearch.004012AB
004012BE  PUSH 3EB                              ; partie écrasée en 401088
004012C3  JMP FCSearch.0040108D
```

4 - 2) Mot clé non saisi !

Lorsque l'on clique sur le bouton «Search on FC», il faut avoir saisi auparavant un mot clé. Dans le cas contraire, il faut avertir l'utilisateur de l'oubli. Pour cela, je vous propose d'afficher une simple MessageBox. Le problème serait simple s'il ne s'agissait que de cela...mais, le gros problème est que la fonction MessageBoxA ne fait pas partie des imports ! On va devoir la rajouter à la main. En fait, voilà un petit schéma qui illustre le fonctionnement de la section imports (.rdata) :



Je n'illustre que le cas de la DLL SHELL32. Comme le sous-entend ce schéma, l'OriginalFirstThunk n'est pas nécessaire. Nous n'avons besoin que de l'IID, du l'IAT et du hint/name Array. Voici donc les modifications à apporter :

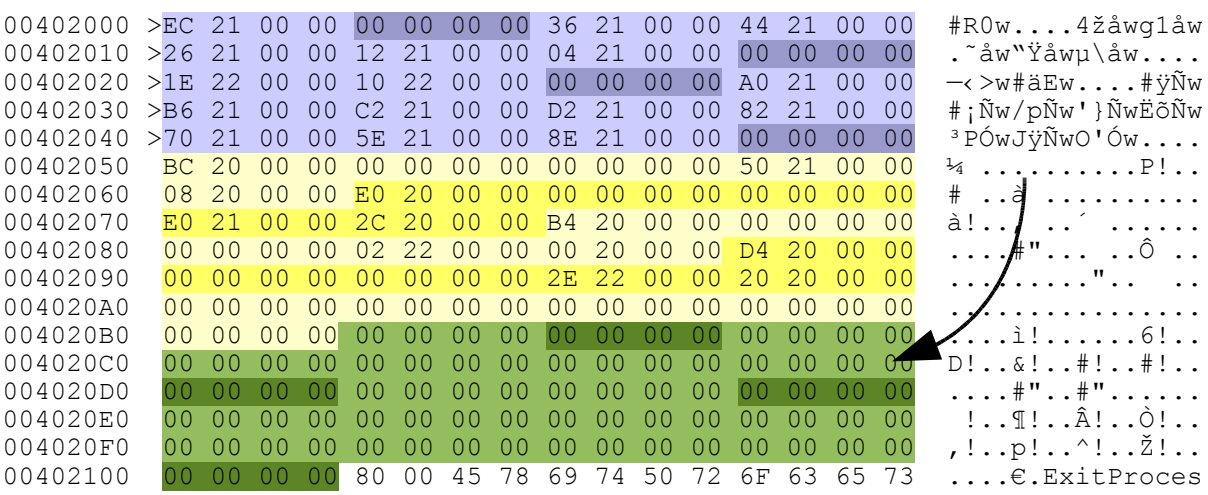


> 1 .Suppression de l' OriginalFirstThunk

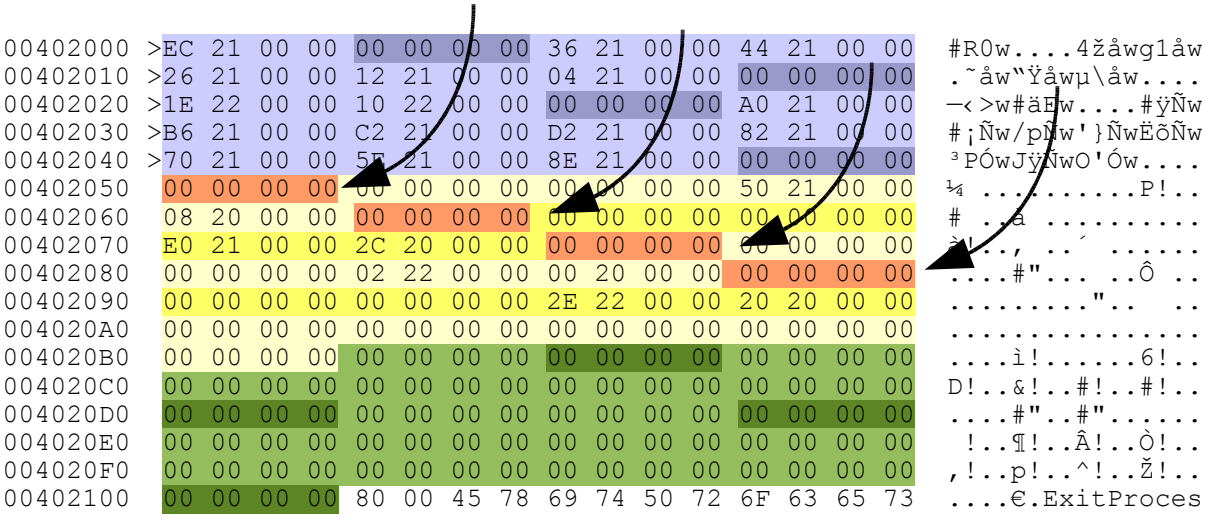
Vous pouvez voir les différentes parties ci-dessous : IAT, IID, OriginalFirstThunk.
Le hint/name Array se voit clairement juste après.

```
00402000 >EC 21 00 00 00 00 00 00 36 21 00 00 44 21 00 00  #R0w....4žåwg1åw
00402010 >26 21 00 00 12 21 00 00 04 21 00 00 00 00 00 00  .˜åw"Ÿåwµ\åw....
00402020 >1E 22 00 00 10 22 00 00 00 00 00 00 A0 21 00 00  —‹>w#äEw....#ÿÑw
00402030 >B6 21 00 00 C2 21 00 00 D2 21 00 00 82 21 00 00  #¡Ñw/pÑw'}ÑwËõÑw
00402040 >70 21 00 00 5E 21 00 00 8E 21 00 00 00 00 00 00  ³PÓwJÿÑwO'Ów....
00402050  BC 20 00 00 00 00 00 00 00 00 00 00 50 21 00 00  ¼ ..........P!..
00402060  08 20 00 00 E0 20 00 00 00 00 00 00 00 00 00 00  # ..à ..........
00402070  E0 21 00 00 2C 20 00 00 B4 20 00 00 00 00 00 00  à!..,  ..´ ......
00402080  00 00 00 00 02 22 00 00 00 20 00 00 D4 20 00 00  ....#"... ..Ô ..
00402090  00 00 00 00 00 00 00 00 2E 22 00 00 20 20 00 00  .........".."  ..
004020A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004020B0  00 00 00 00 EC 21 00 00 00 00 00 00 36 21 00 00  ....ì!......6!..
004020C0  44 21 00 00 26 21 00 00 12 21 00 00 04 21 00 00  D!..&!..#!..#!..
004020D0  00 00 00 00 1E 22 00 00 10 22 00 00 00 00 00 00  ....#"..#"......
004020E0  A0 21 00 00 B6 21 00 00 C2 21 00 00 D2 21 00 00   !..¶!..Â!..Ò!..
004020F0  82 21 00 00 70 21 00 00 5E 21 00 00 8E 21 00 00  ,!..p!..^!..Ž!..
00402100  00 00 00 00 80 00 45 78 69 74 50 72 6F 63 65 73  ....€.ExitProces
00402110  73 00 09 01 47 65 74 4D 6F 64 75 6C 65 48 61 6E  s..#GetModuleHan
00402120  64 6C 65 41 00 00 81 02 56 69 72 74 75 61 6C 41  dleA..#VirtualA
00402130  6C 6C 6F 63 00 00 83 02 56 69 72 74 75 61 6C 46  lloc..ƒ#VirtualF
00402140  72 65 65 00 BB 02 6C 73 74 72 63 70 79 41 00 00  ree.»#lstrcpyA..
00402150  6B 65 72 6E 65 6C 33 32 2E 64 6C 6C 00 00 30 00  kernel32.dll..0.
00402160  43 68 65 63 6B 44 6C 67 42 75 74 74 6F 6E 00 00  CheckDlgButton..
00402170  8A 00 44 69 61 6C 6F 67 42 6F 78 50 61 72 61 6D  Š.DialogBoxParam
00402180  41 00 AD 00 45 6E 64 44 69 61 6C 6F 67 00 F4 00  A..EndDialog.ô.
00402190  47 65 74 44 6C 67 49 74 65 6D 54 65 78 74 41 00  GetDlgItemTextA.
004021A0  72 01 49 73 44 6C 67 42 75 74 74 6F 6E 43 68 65  r#IsDlgButtonChe
004021B0  63 6B 65 64 00 00 84 01 4C 6F 61 64 49 63 6F 6E  cked..„#LoadIcon
004021C0  41 00 E2 01 53 65 6E 64 4D 65 73 73 61 67 65 41  A.â#SendMessageA
004021D0  00 00 2D 02 53 68 6F 77 57 69 6E 64 6F 77 00 00  ..-#ShowWindow..
004021E0  75 73 65 72 33 32 2E 64 6C 6C 00 00 3E 00 49 6E  user32.dll..>.In
004021F0  69 74 43 6F 6D 6D 6F 6E 43 6F 6E 74 72 6F 6C 73  itCommonControls
00402200  00 00 63 6F 6D 63 74 6C 33 32 2E 64 6C 6C 00 00  ..comctl32.dll..
00402210  65 00 53 68 65 6C 6C 41 62 6F 75 74 41 00 67 00  e.ShellAboutA.g.
00402220  53 68 65 6C 6C 45 78 65 63 75 74 65 41 00 73 68  ShellExecuteA.sh
00402230  65 6C 6C 33 32 2E 64 6C 6C 00 00 00 00 00 00 00  ell32.dll.......
00402240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00402250  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00402260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

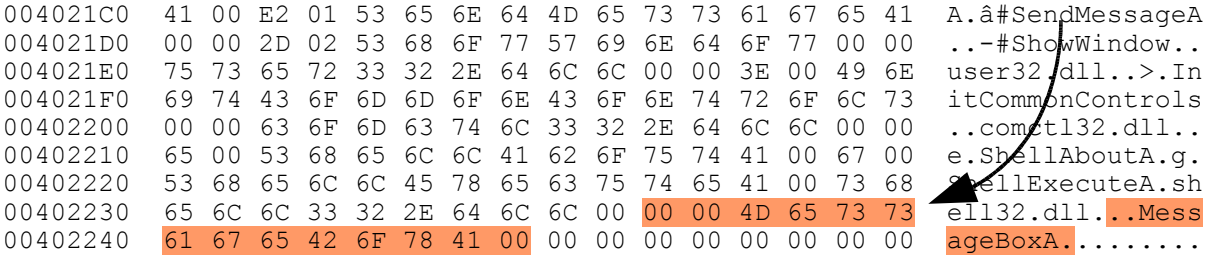
Nous supprimons donc l'OriginalFirstThunk : (remplissage par des zéros)



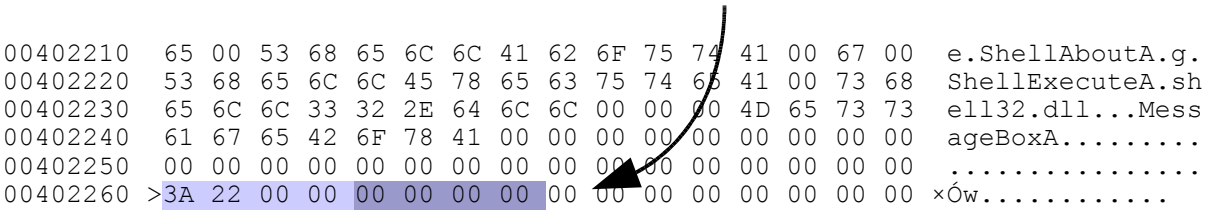
> 2 . Suppression des Champs OriginalFirstThunk dans les IIDs



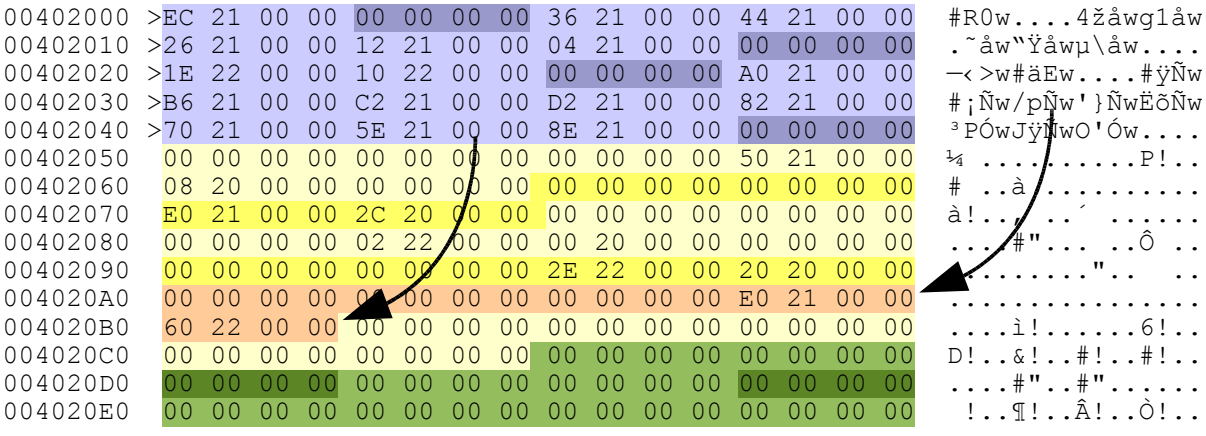
> 3 . Ajout du hint/name «MessageBoxA» en 40223A.



> 4 . Ajout de la FirstThunk en 402260.



> 5. Ajout d'une IID et remplissage de ses champs.



> 6. Ajout d'une entrée dans la table des jumps (non nécessaire).

```
00401480  JMP DWORD PTR DS:[<&kernel32.ExitProcess>;  kernel32.ExitProcess
00401486  JMP DWORD PTR DS:[<&kernel32.GetModuleHa>;  kernel32.GetModuleHandleA
0040148C  JMP DWORD PTR DS:[<&kernel32.VirtualAllo>;  kernel32.VirtualAlloc
00401492  JMP DWORD PTR DS:[<&kernel32.VirtualFree>;  kernel32.VirtualFree
00401498  JMP DWORD PTR DS:[<&kernel32.lstrcpyA>]  ;  kernel32.lstrcpyA
0040149E  JMP DWORD PTR DS:[<&user32.CheckDlgButto>;  user32.CheckDlgButton
004014A4  JMP DWORD PTR DS:[<&user32.DialogBoxPara>;  user32.DialogBoxParamA
004014AA  JMP DWORD PTR DS:[<&user32.EndDialog>]   ;  user32.EndDialog
004014B0  JMP DWORD PTR DS:[<&user32.GetDlgItemTex>;  user32.GetDlgItemTextA
004014B6  JMP DWORD PTR DS:[<&user32.IsDlgButtonCh>;  user32.IsDlgButtonChecked
004014BC  JMP DWORD PTR DS:[<&user32.LoadIconA>]   ;  user32.LoadIconA
004014C2  JMP DWORD PTR DS:[<&user32.SendMessageA>>;  user32.SendMessageA
004014C8  JMP DWORD PTR DS:[<&user32.ShowWindow>]  ;  user32.ShowWindow
004014CE  JMP DWORD PTR DS:[<&comctl32.InitCommonC>;
comctl32.InitCommonControls
004014D4  JMP DWORD PTR DS:[<&shell32.ShellAboutA>>;  shell32.ShellAboutA
004014DA  JMP DWORD PTR DS:[<&shell32.ShellExecute>;  shell32.ShellExecuteA
004014E0  ASCII "Kikoo Reverser :"
004014F0  ASCII "-)",0
004014F3  JMP DWORD PTR DS:[<&user32.MessageBoxA>] ;  user32.MessageBoxA
```

On ajoute maintenant notre MessageBox :

```
00401017  OR EAX,EAX
00401019  JNZ SHORT FCSearch.00401020
```

devient :

```
00401017  JMP FCSearch.00401300
```

Et on ajoute le code suivant :

```
00401300  OR EAX,EAX
00401302  JNZ FCSearch.00401020
00401308  PUSH 0                          ; /Style = MB_OK|MB_APPLMODAL
0040130A  PUSH FCSearch.004030D2          ; |Title = "FC Search -Error-"
0040130F  PUSH FCSearch.004030BA          ; |Text = "Entrez un mot clé SVP !"
00401314  PUSH 0                          ; |hOwner = NULL
00401316  CALL <JMP.&user32.MessageBoxA> ; \MessageBoxA
0040131B  JMP FCSearch.0040101C
```

4 - 3) GREETZ et SEARCH

Le travail est nettement plus simple cette fois : (on ajout ce code)

```
004012CE  CMP EAX,2718
004012D3  JNZ SHORT FCSearch.004012F0
004012D5  PUSH 0                           ; /Style = MB_OK|MB_APPLMODAL
004012D7  PUSH FCSearch.004030E4           ; |Title = "FC Search"
004012DC  PUSH FCSearch.00403000           ; |Text = "Greetz - Merci à tout les
habitants de FC ! Vous êtes trop nombreux
004012E1  PUSH 0                           ; |hOwner = NULL
004012E3  CALL <JMP.&user32.MessageBoxA>   ; \MessageBoxA
004012E8  JMP SHORT FCSearch.00401273
004012EA  DB 00
004012EB  DB 00
004012EC  DB 00                           ; ces zéros, c'est moche !
004012ED  DB 00                           ; j'ai codé à l'arrache sans trop
004012EE  DB 00                           ; regarder où je plaçais le code
004012EF  DB 00
```

```
004012F0  CMP EAX,2716
004012F5  JNZ FCSearch.00401235
004012FB  JMP FCSearch.00401191
```

4 - 4) EXIT

Une petite modification s'impose. On se jette sur le EndDialog qui se situe en 40125B et ça donne ceci :

```
0040121F  PUSH 6                                  ; /ShowState = SW_MINIMIZE
00401221  PUSH DWORD PTR SS:[EBP+8]               ; |hWnd
00401224  CALL <JMP.&user32.ShowWindow>           ; \ShowWindow
00401229  JMP SHORT FCSearch.0040125B              ; jump modifié

0040125B  PUSH 0                                  ; /Result = 0
0040125D  PUSH DWORD PTR SS:[EBP+8]               ; |hWnd
00401260  CALL <JMP.&user32.EndDialog>            ; \EndDialog
00401265  JMP SHORT FCSearch.00401273
```

4 - 5) IEXPLORER

Bon, le problème est assez clair ici :

```
00401088  PUSH 3EB                                ; /ButtonID = 3EB (1003.)
0040108D  PUSH DWORD PTR SS:[EBP+8]               ; |hWnd
00401090  CALL <JMP.&user32.IsDlgButtonChecked>   ; \IsDlgButtonChecked
00401095  CMP EAX,1
00401098  JNZ SHORT FCSearch.004010BA
0040109A  PUSH 3                                  ; /IsShown = 3
0040109C  PUSH 0                                  ; |DefDir = NULL
0040109E  PUSH DWORD PTR DS:[403354]              ; |Parameters = NULL
004010A4  PUSH FCSearch.00403132                  ; |FileName = "firefox"
004010A9  PUSH FCSearch.0040312D                  ; |Operation = "open"
004010AE  PUSH DWORD PTR SS:[EBP+8]               ; |hWnd
004010B1  CALL <JMP.&shell32.ShellExecuteA>       ; \ShellExecuteA
004010B6  LEAVE
004010B7  RETN 4
004010BA  PUSH 3EC                                ; /ButtonID = 3EC (1004.)
004010BF  PUSH DWORD PTR SS:[EBP+8]               ; |hWnd
004010C2  CALL <JMP.&user32.IsDlgButtonChecked>   ; \IsDlgButtonChecked
004010C7  CMP EAX,1
004010CA  JNZ SHORT FCSearch.004010E7
004010CC  PUSH 3                                  ; /IsShown = 3
004010CE  PUSH 0                                  ; |DefDir = NULL
004010D0  PUSH FCSearch.004030EE                  ; |Parameters =
"http://www.hackatak.org/forumcrack/search.php?search_keywords="
004010D5  PUSH FCSearch.00403132                  ; |FileName = "firefox"
004010DA  PUSH FCSearch.0040312D                  ; |Operation = "open"
004010DF  PUSH DWORD PTR SS:[EBP+8]               ; |hWnd
004010E2  CALL <JMP.&shell32.ShellExecuteA>       ; \ShellExecuteA
004010E7  LEAVE
004010E8  RETN 4
```

En 4010D5, le ShellExecute va lancer firefox alors qu'il doit lancer Internet explorer.

En 4010D0, on pushe le mauvais paramètre puisque le bon se trouve en [403354] (voir à l'adresse 40109E). On fait les modifications suivantes :

```
004010CE  PUSH 0
004010D0  JMP FCSearch.00401322
004010D5  PUSH FCSearch.0040313A                  ; |FileName = "iexplore"
004010DA  PUSH FCSearch.0040312D                  ; |Operation = "open"
004010DF  PUSH DWORD PTR SS:[EBP+8]               ; |hWnd
004010E2  CALL <JMP.&shell32.ShellExecuteA>       ; \ShellExecuteA
004010E7  LEAVE
004010E8  RETN 4
```

On pointe en 4010D5 vers le nom «iexplore». Pour le remplacement du PUSH 4030EE, il nous manque un octet ! On le remplace donc par un JMP 401322 et on ajoute ceci :

```
00401322  PUSH DWORD PTR DS:[403354]
00401328  JMP FCSearch.004010D5
```

Voilà, le travail est terminé. Il ne me reste plus qu'à remercier Neitsa pour son reversme très intéressant. Je salue et je remercie également tous les crackers, coders, unpackers et autres bêtes étranges qui me connaissent et qui m'en apprennent tant !

Lundi 13 décembre 2004 - BeatriX