Etude du logiciel CAMOUFLAGE

Free File Camouflage

v1.25

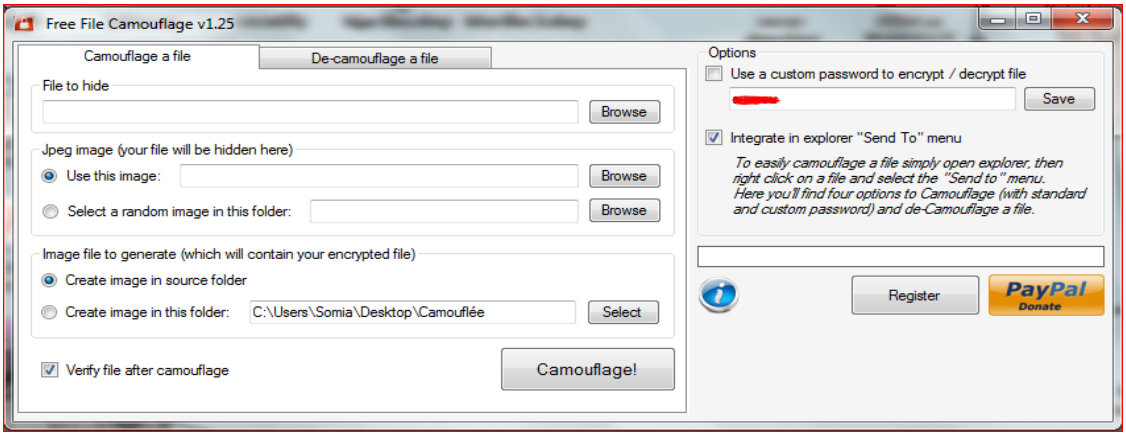
CHAHER Somia

M2 - SSI -

Université de Lorraine _ Metz

Sommaire

Présentation du logiciel



Camouflage est un logiciel qui permet de cacher et ainsi protéger tous types de fichier dans une image "jpeg".

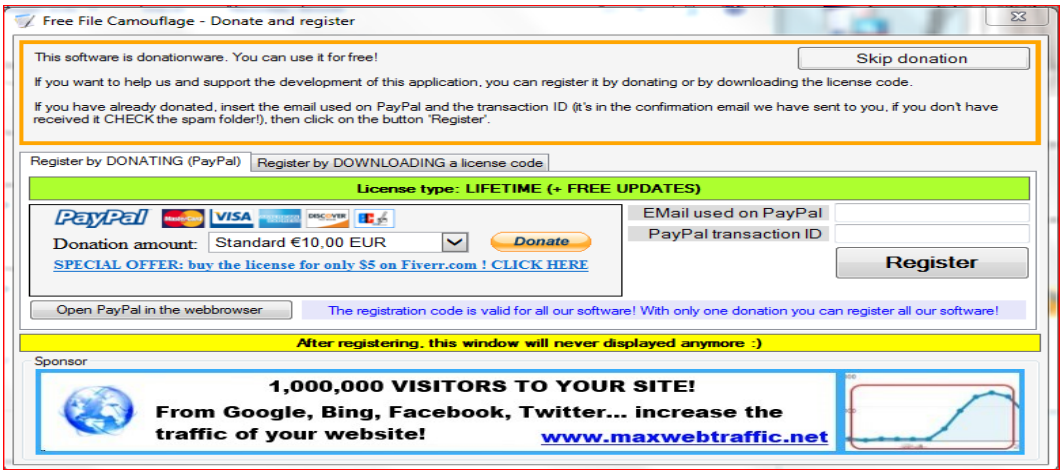
Il permet aussi de récupérer le fichier cacher à partir de l'image.

Selon la page officiel du logiciel www.myportablesoftware.com, tous les fichier sont chiffrés avec AES et caché à l'intérieur d'une image.

Le logiciel est également utile pour envoyer des programmes ".exe" par mail. Car actuellement, de nombreux fournisseurs de messagerie ne permettent pas cela !

Le site officiel assure que tous les logiciels qui proposent dans leur site sont fait par eux et sont tous originaux. Le site assure aussi que les logiciels ne contiennent ni spyware, ni adware, ni virus !

Le logiciel est complètement gratuit et peut être utilisé partout (maison, travail, école), toutefois le site fait appel à des donations, pour les gens qui veulent les aider et soutenir le développement des applications étant donné que le site propose une panoplie de logiciels gratuit.



Le logiciel peut être utilisé sous les environnements suivants : Windows XP, VISTA, 7, 8 (32 & 64 bit)

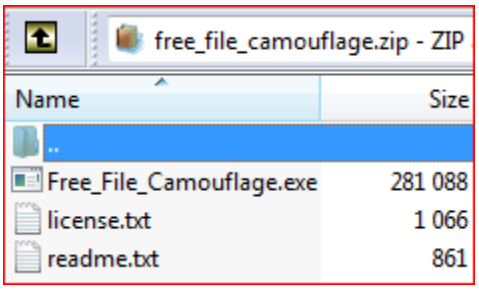
Néanmoins, on a essayé de l'installé sur une machine ubuntu (Linux), on détaillera cela par la suite.

Installation du logiciel "Camouflage"

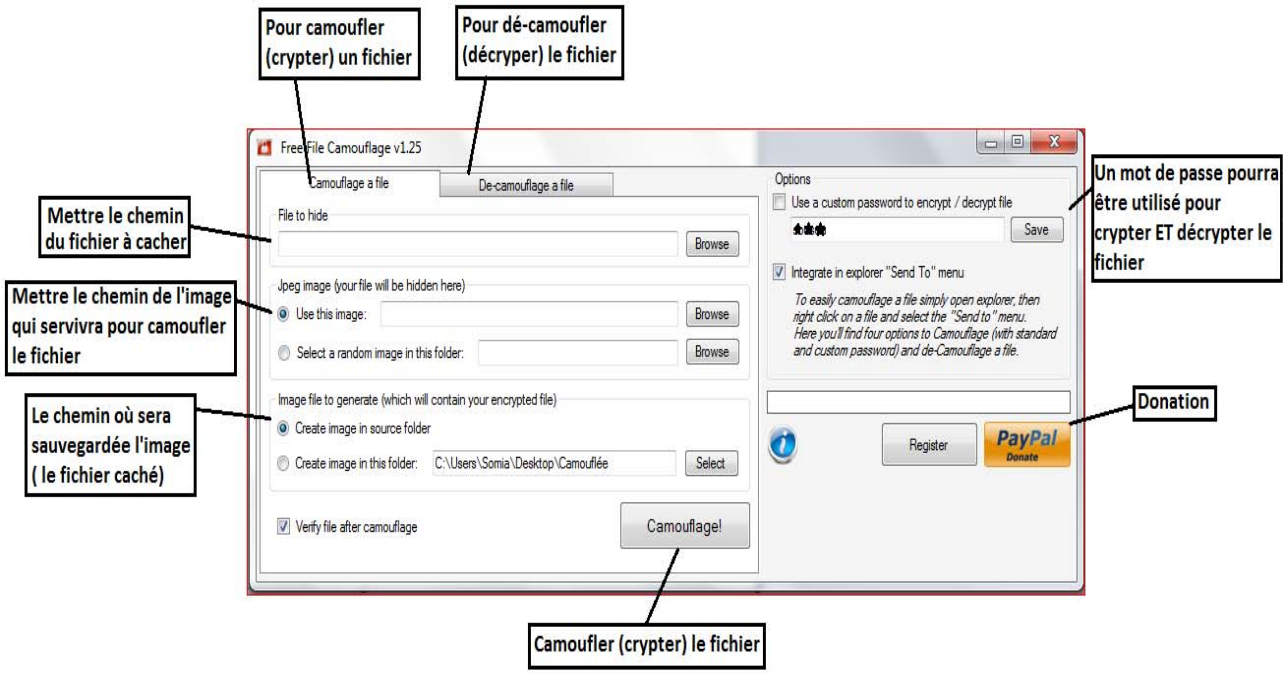
Sous Windows

Le logiciel doit être installé sous les environnements suivants : Windows XP, VISTA, 7, 8 (32 & 64 bit).

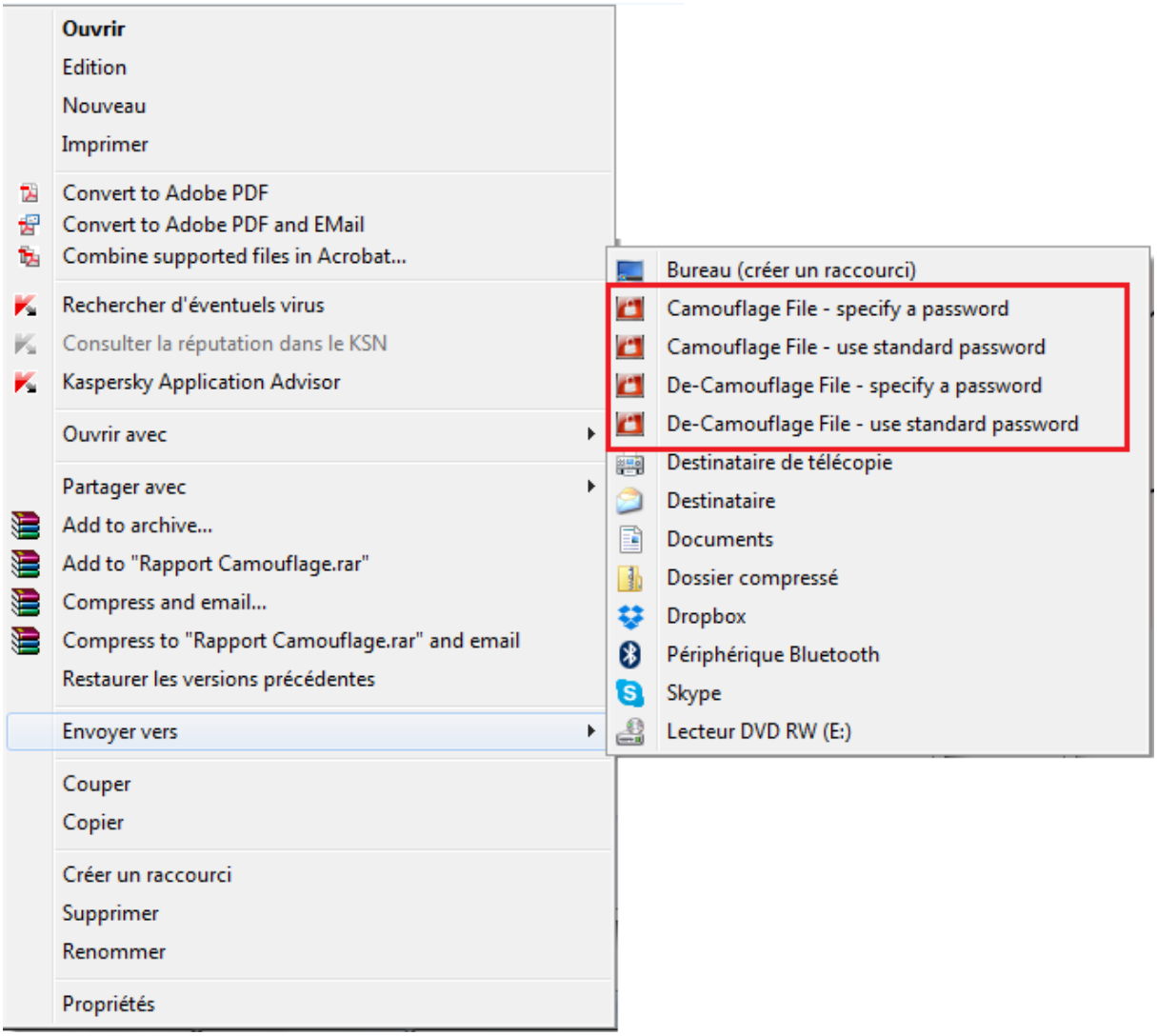
Il suffit de télécharger le logiciel du site www.myportablesoftware.com, il sera téléchargé sous format zip 'free_file_camouflage.zip', il faudra ensuite le dé-zipper. Une fois dé-zipper, un répertoire sera crée avec comme nom 'free_file_camouflage' contenant à l'intérieur le logiciel 'Free_File_Camouflage.exe' ainsi que deux fichier texte : licence et readme



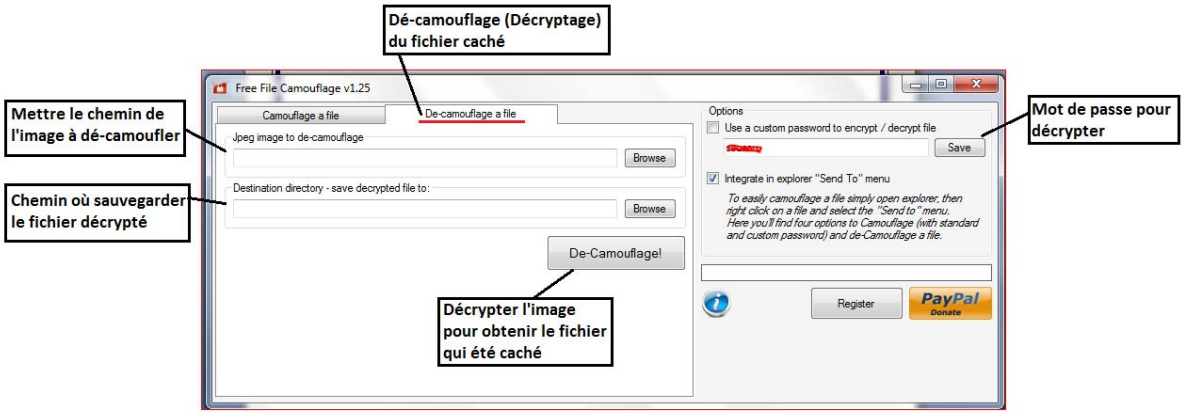
Pour le lancer, cliquer sur Free_File_Camouflage.exe



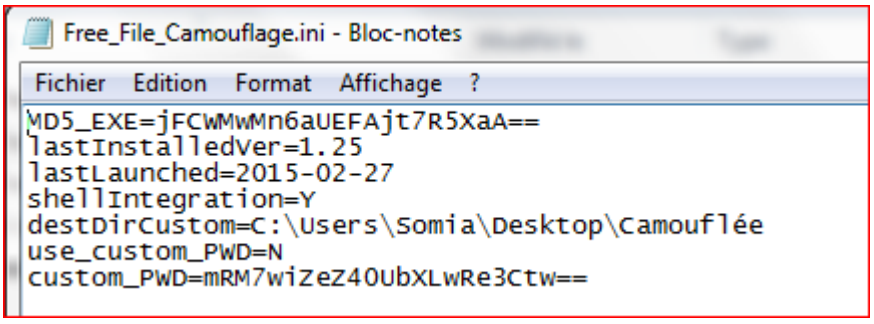
On remarque aussi qu'après le lancement du logiciel et en cliquant droit sur n'importe quel fichier sur l'ordinateur ensuite sur "Envoyer vers", on trouve la possibilité de camoufler un fichier avec un mot de passe spécifique, camoufler un fichier avec le mot de passe standard, dé-camoufler le fichier avec un mot de passe spécifique et dé-camoufler le fichier avec le mot de passe standard, comme il est montré dans l'image ci-dessous.



Pour décrypter ou dé-camoufler un fichier est aussi simple que le cryptage ; il suffit d'introduire le chemin de l'image contenant le fichier caché et le chemin d'où on veut sauvegarder le fichier une fois décrypter, introduire le mot de passe utilisé pour le cryptage de ce fichier (dans le cas d'utilisation d'un mot de passe) et finalement cliquer sur le bouton "De-Camouflage !", comme il est montré dans l'image ci dessous

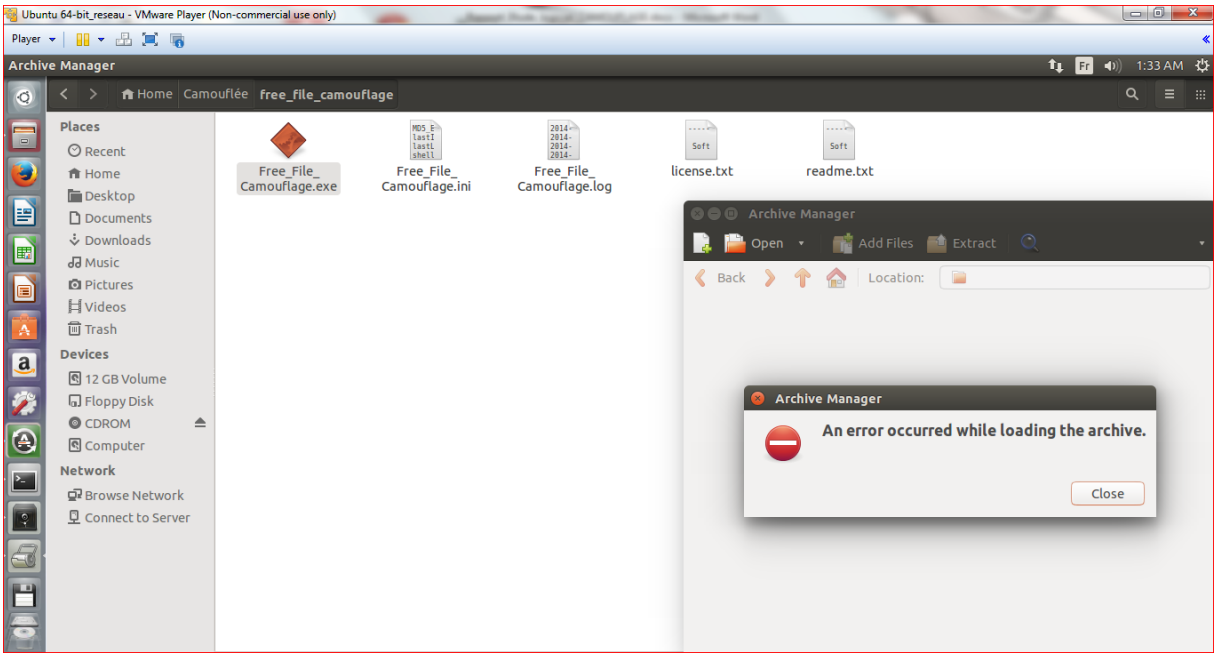


Une fois crypter un fichier, on s'aperçoit de la création d'un fichier dans le dossier "free_file_camouflage" qui est : Free_File_Camouflage.ini, contenant des informations sur la dernière utilisation du logiciel, la version du logiciel utilisé et d'autres informations.



Sous Linux

Tout simplement sous Linux, le logiciel ne fonctionne pas.



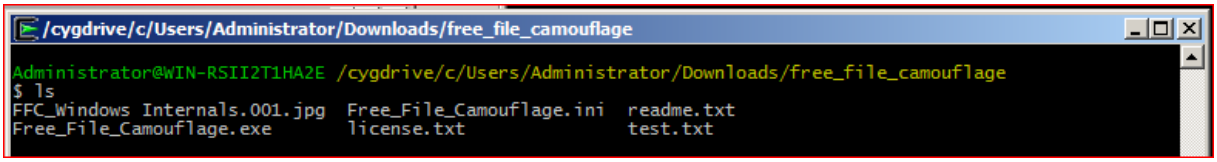
Bien entendu y a d'autre méthode pour l'installer ; il existe des outils qui font fonctionner des programmes sous linux qui sont initialement conçu pour windows et vis versa.

Etude du logiciel

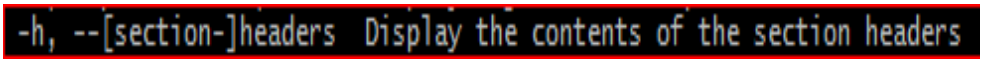
Sous Windows

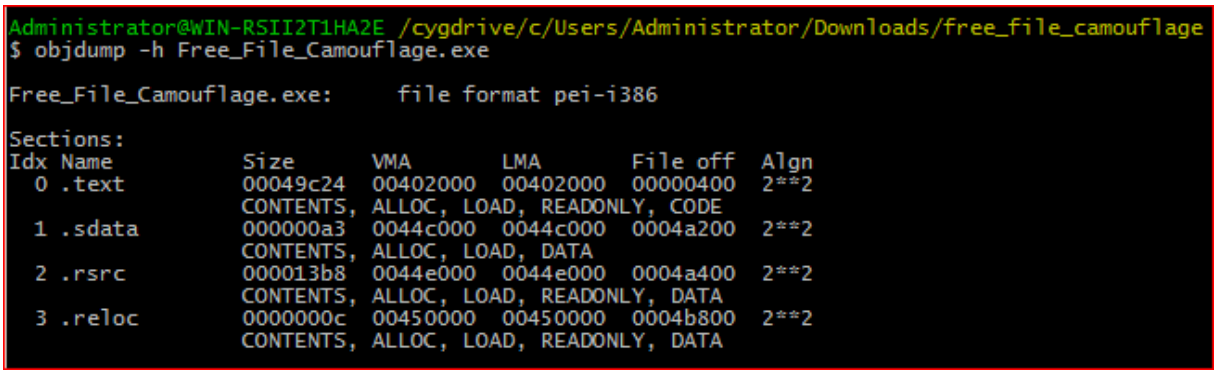
L'étude a été faite sous une machine virtuelle hébergeant un windows server 2008 :

Ouvrir une fenêtre cygwin64 terminal et allé au répertoire contenant le logiciel free_file_camouflage

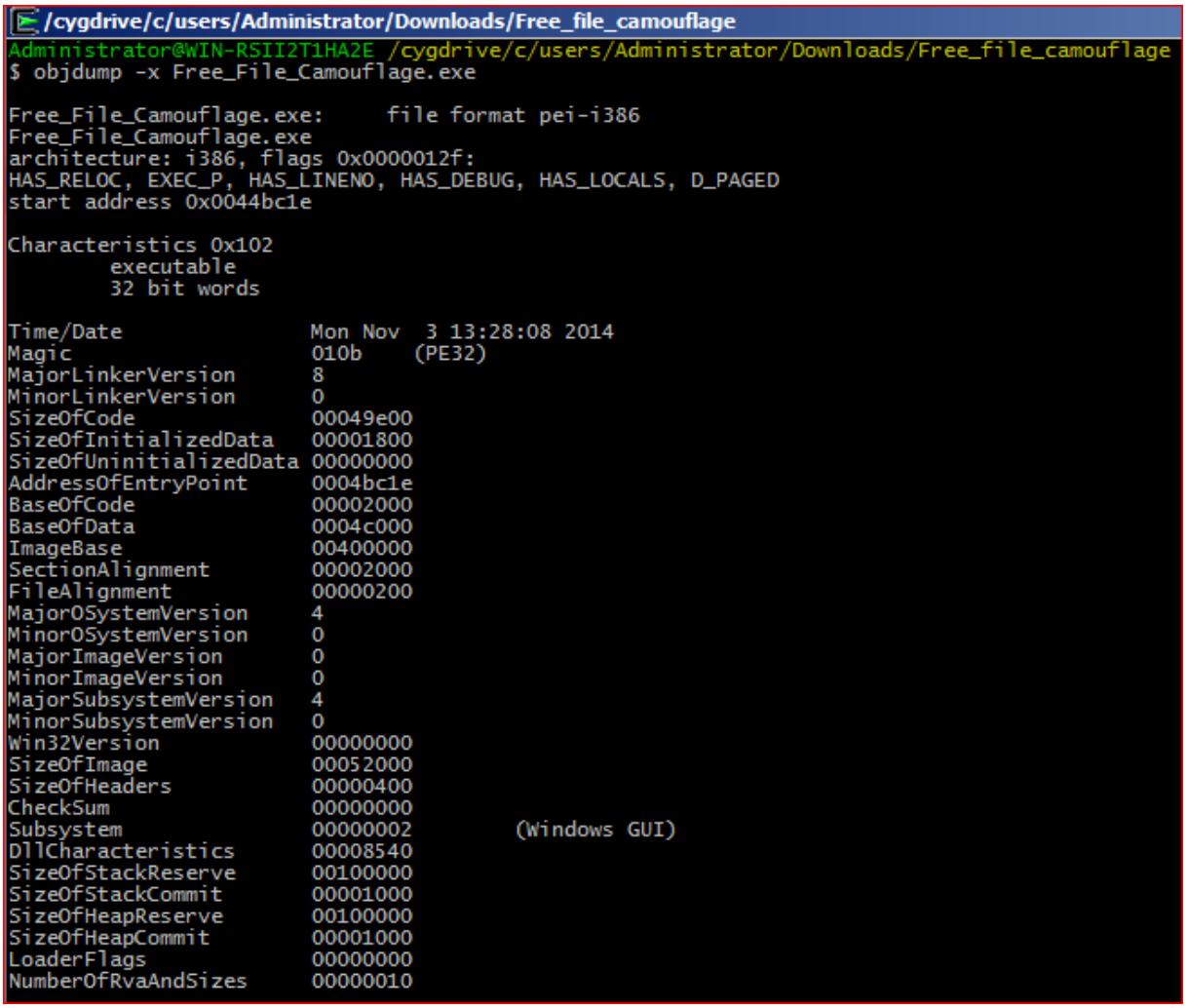


- On essai d'avoir les informations sur les sections en exécutant la commande : objdump -h Free_File_Camouflage.exe





- Trouver les DLL requise pour le programme en tapant la commande : objdump –x Free_File_Camouflage.exe

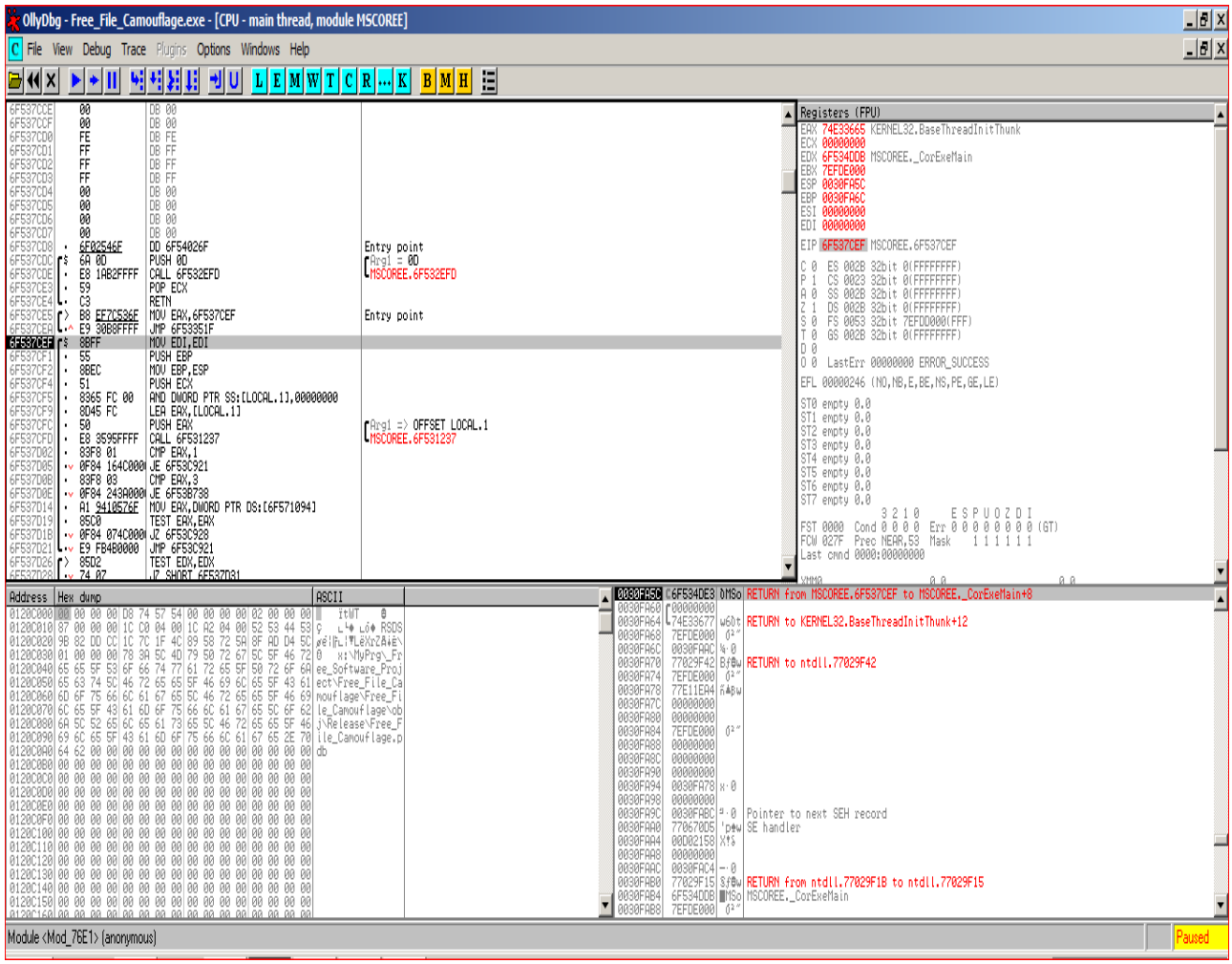

/cygdrive/c/users/Administrator/Downloads/Free_file_camouflage
Administrator@WIN-RSII2T1HA2E /cygdrive/c/users/Administrator/Downloads/Free_file_camouflage
$ objdump -x Free_File_Camouflage.exe
Free_File_Camouflage.exe: file format pei-i386
Free_File_Camouflage.exe
architecture: i386, flags 0x0000012f:
HAS_RELOC, EXEC_P, HAS_LINENO, HAS_DEBUG, HAS_LOCALS, D_PAGED
start address 0x0044bc1e
Characteristics 0x102
executable
32 bit words
Time/Date Mon Nov 3 13:28:08 2014
Magic 010b (PE32)
MajorLinkerVersion 8
MinorLinkerVersion 0
SizeOfCode 00049e00
SizeOfInitializedData 00001800
SizeOfUninitializedData 00000000
AddressOfEntryPoint 0004bc1e
BaseOfCode 00002000
BaseOfData 0004c000
ImageBase 00400000
SectionAlignment 00002000
FileAlignment 00000200
MajorOSystemVersion 4
MinorOSystemVersion 0
MajorImageVersion 0
MinorImageVersion 0
MajorSubsystemVersion 4
MinorSubsystemVersion 0
Win32Version 00000000
SizeOfImage 00052000
SizeOfHeaders 00000400
CheckSum 00000000
Subsystem 00000002 (Windows GUI)
DllCharacteristics 00008540
SizeOfStackReserve 00100000
SizeOfStackCommit 00001000
SizeOfHeapReserve 00100000
SizeOfHeapCommit 00001000
LoaderFlags 00000000
NumberOfRvaAndSizes 00000010

```
The Data Directory
Entry 0 00000000 00000000 Export Directory [.edata (or where ever we found it)]
Entry 1 0004bbd0 0000004b Import Directory [parts of .idata]
Entry 2 0004e000 000013b8 Resource Directory [.rsrc]
Entry 3 00000000 00000000 Exception Directory [.pdata]
Entry 4 00000000 00000000 Security Directory
Entry 5 00050000 0000000c Base Relocation Directory [.reloc]
Entry 6 0004c000 0000001c Debug Directory
Entry 7 00000000 00000000 Description Directory
Entry 8 00000000 00000000 Special Directory
Entry 9 00000000 00000000 Thread Storage Directory [.tls]
Entry a 00000000 00000000 Load Configuration Directory
Entry b 00000000 00000000 Bound Import Directory
Entry c 00002000 00000008 Import Address Table Directory
Entry d 00000000 00000000 Delay Import Directory
Entry e 00002008 00000048 CLR Runtime Header
Entry f 00000000 00000000 Reserved

There is an import table in .text at 0x44bbd0

The Import Tables (interpreted .text section contents)
 vma:            Hint    Time      Forward  DLL       First
                 Table   Stamp     Chain    Name      Thunk
 0004bbd0       0004bbf8 00000000 00000000 0004bc0e 00002000

        DLL Name: mscoree.dll
        vma:  Hint/Ord Member-Name Bound-To
        4bc00       0  _CorExeMain

 0004bbe4       00000000 00000000 00000000 00000000 00000000


PE File Base Relocations (interpreted .reloc section contents)

Virtual Address: 0004b000 Chunk size 12 (0xc) Number of fixups 2
        reloc    0 offset  c20 [4bc20] HIGHLOW
        reloc    1 offset    0 [4b000] ABSOLUTE

There is a debug directory in .sdata at 0x44c000

Type                Size     Rva      Offset
  2        CodeView 00000087 0004c01c 0004a21c
(format RSDS signature ccdd829b7c1c4c1f8958725a8fadd45c age 1)
```

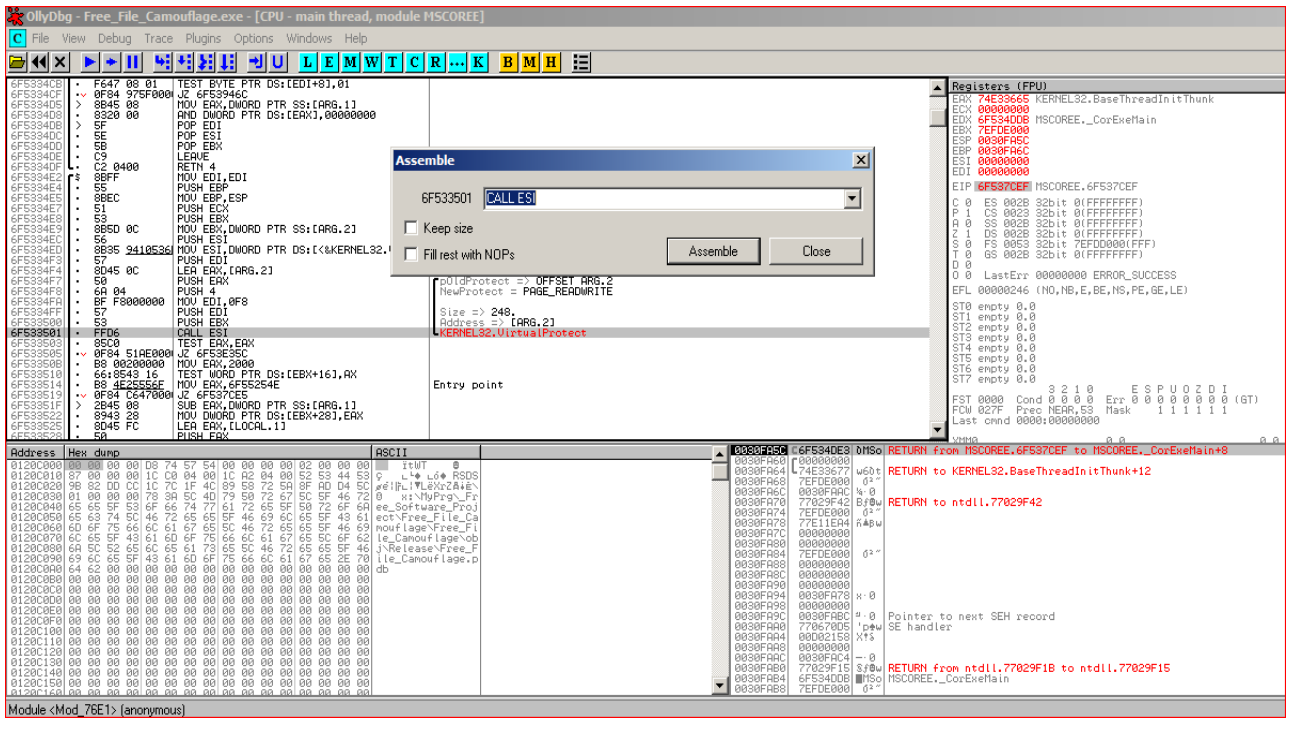
```
The .rsrc Resource Directory section:
000  Type Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 4
010   Entry: ID: 0x000003, Value: 0x80000030
030    Name Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
040     Entry: ID: 0x000002, Value: 0x80000090
090      Language Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
0a0       Entry: ID: 00000000, Value: 0x0000f0
0f0        Leaf: Addr: 0x04e508, Size: 0x000ca8, Codepage: 0
018   Entry: ID: 0x00000e, Value: 0x80000048
048    Name Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
058     Entry: ID: 0x007f00, Value: 0x800000a8
0a8      Language Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
0b8       Entry: ID: 00000000, Value: 0x000100
100        Leaf: Addr: 0x04f1b0, Size: 0x000014, Codepage: 0
020   Entry: ID: 0x000010, Value: 0x80000060
060    Name Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
070     Entry: ID: 0x000001, Value: 0x800000c0
0c0      Language Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
0d0       Entry: ID: 00000000, Value: 0x000110
110        Leaf: Addr: 0x04e130, Size: 0x0003d4, Codepage: 0
028   Entry: ID: 0x000018, Value: 0x80000078
078    Name Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
088     Entry: ID: 0x000001, Value: 0x800000d8
0d8      Language Table: Char: 0, Time: 00000000, Ver: 0/0, Num Names: 0, IDs: 1
0e8       Entry: ID: 00000000, Value: 0x000120
120        Leaf: Addr: 0x04f1c8, Size: 0x0001ea, Codepage: 0
 Resources start at 508x

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00049c24  00402000  00402000  00000400  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .sdata        000000a3  0044c000  0044c000  0004a200  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .rsrc         000013b8  0044e000  0044e000  0004a400  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .reloc        0000000c  00450000  00450000  0004b800  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
SYMBOL TABLE:
no symbols
```

On remarque que l'application utilise une DLL : mscoree.dll

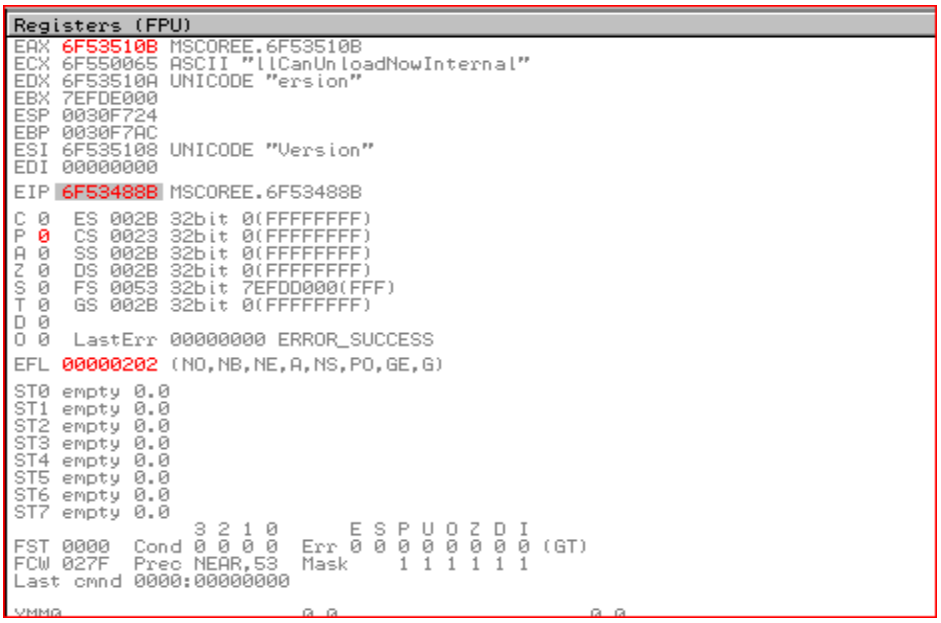
- Ouvrir le programme avec le logiciel "OllyDbg"



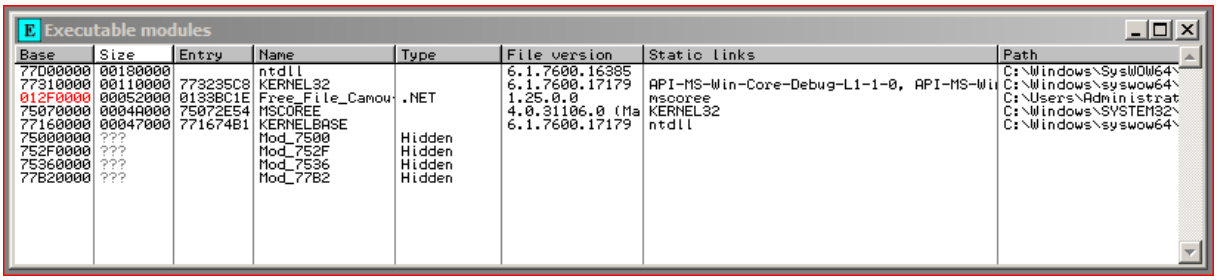
- On remarque qu'il est possible de modifier le programme en modifiant ses instructions assembleurs !



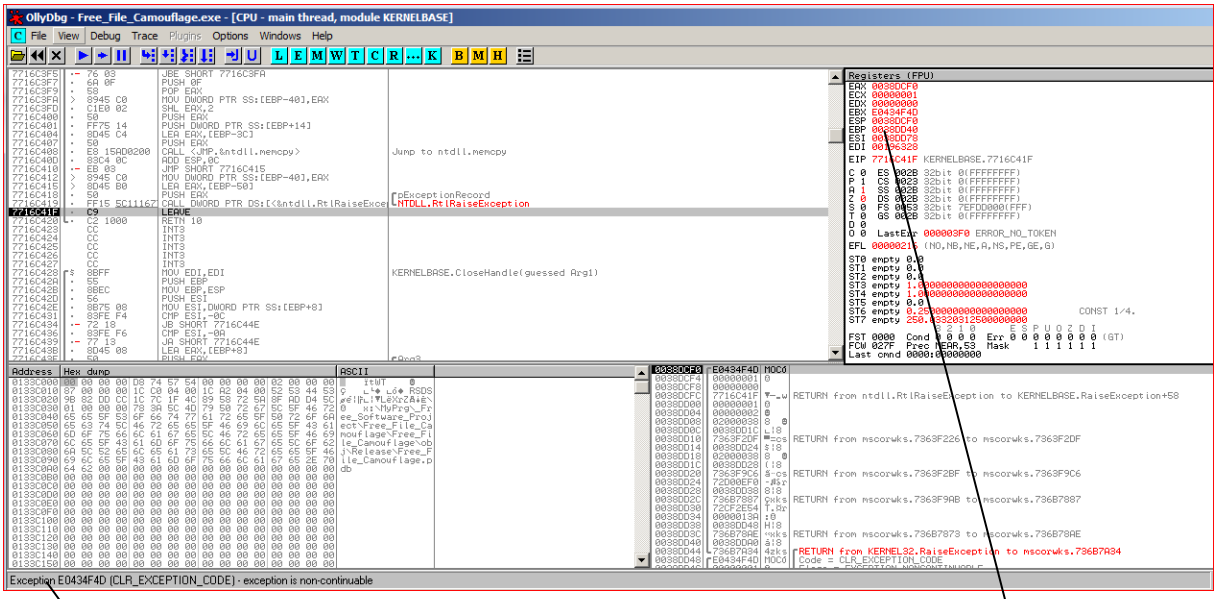
Dans cette section, on peut remarquer tous les registres ainsi que leurs contenus



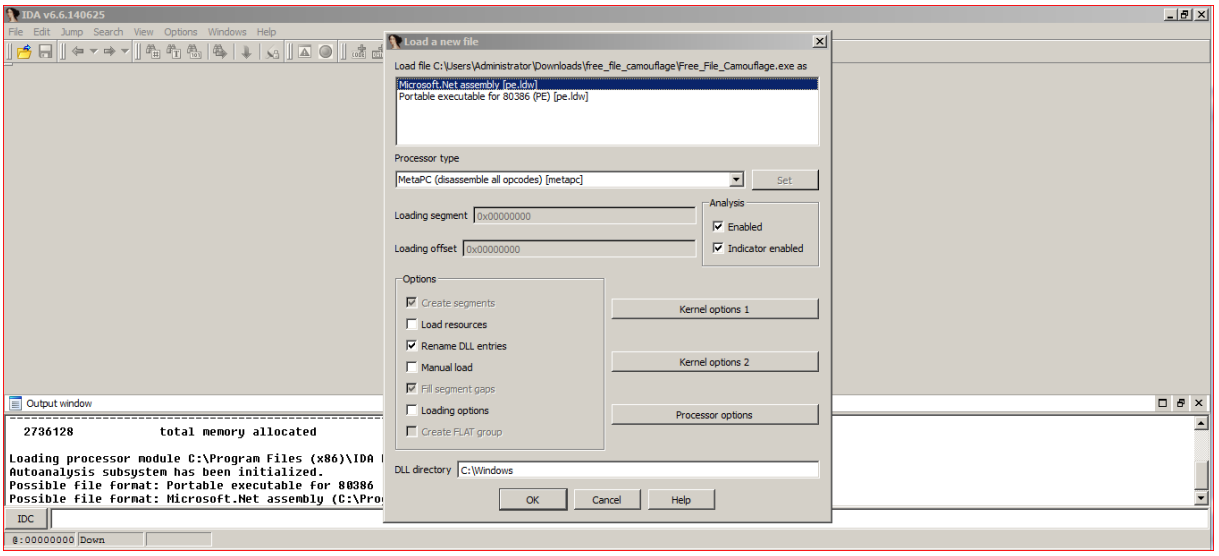
- Là, on peut voir les modules exécutables



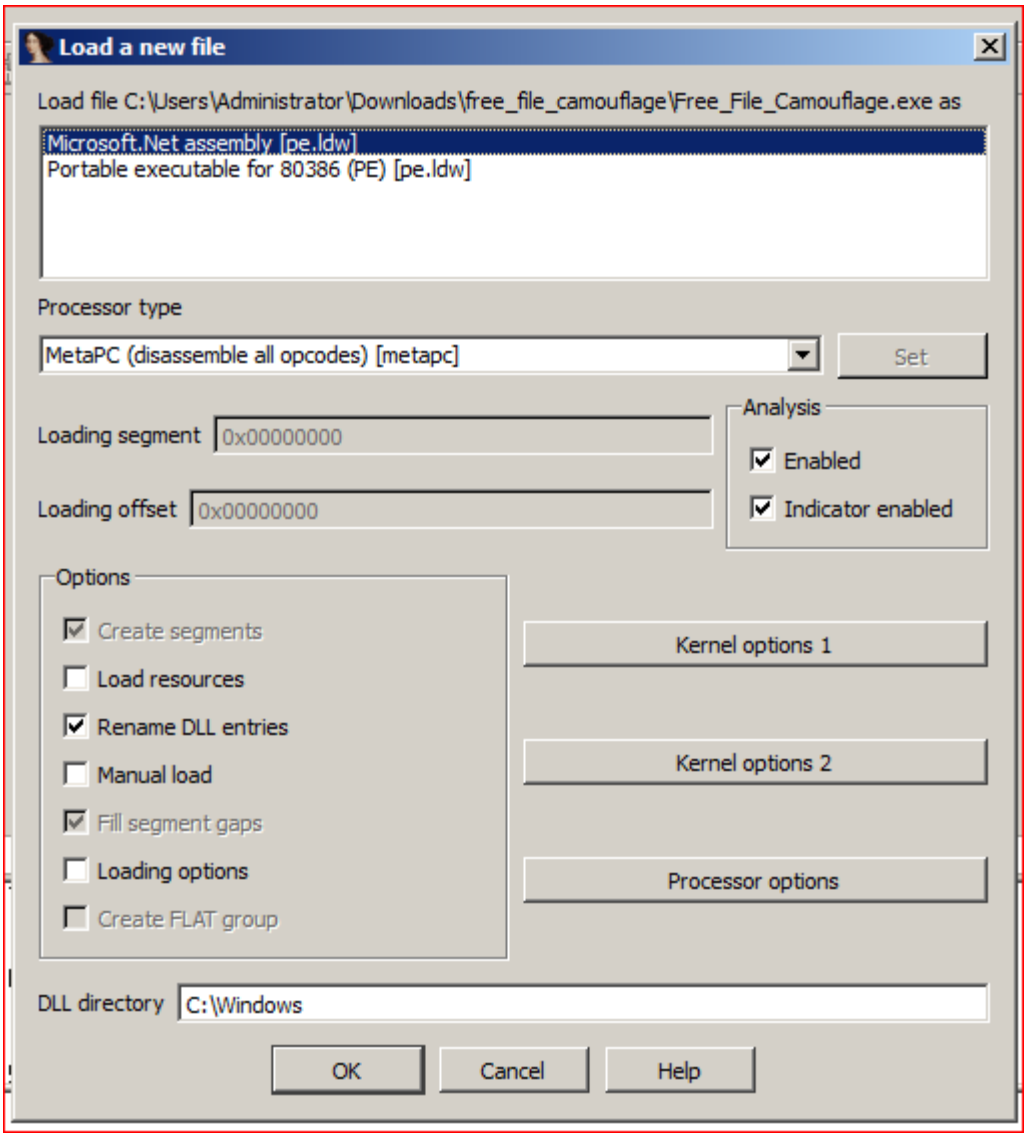
- L'exécution du programme à partir d'OllyDbg a commencé mais s'est arrêtée avec une erreur disant Exception E0434F4D et on remarque que cette valeur été dans le registre EBX !

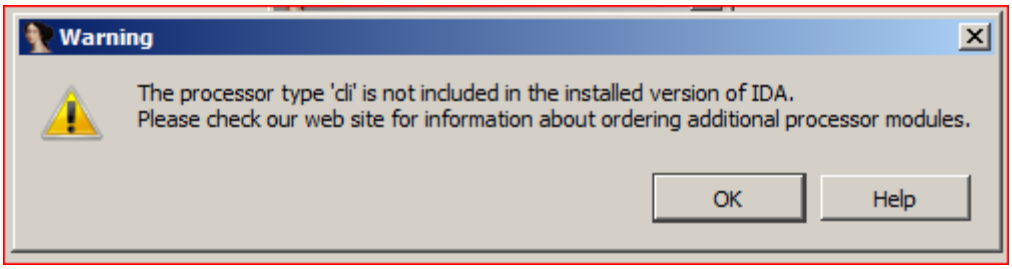


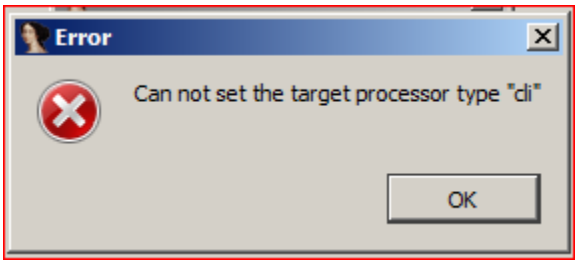
- Ouvrir le programme avec le logiciel "IDA Demo v6.6"



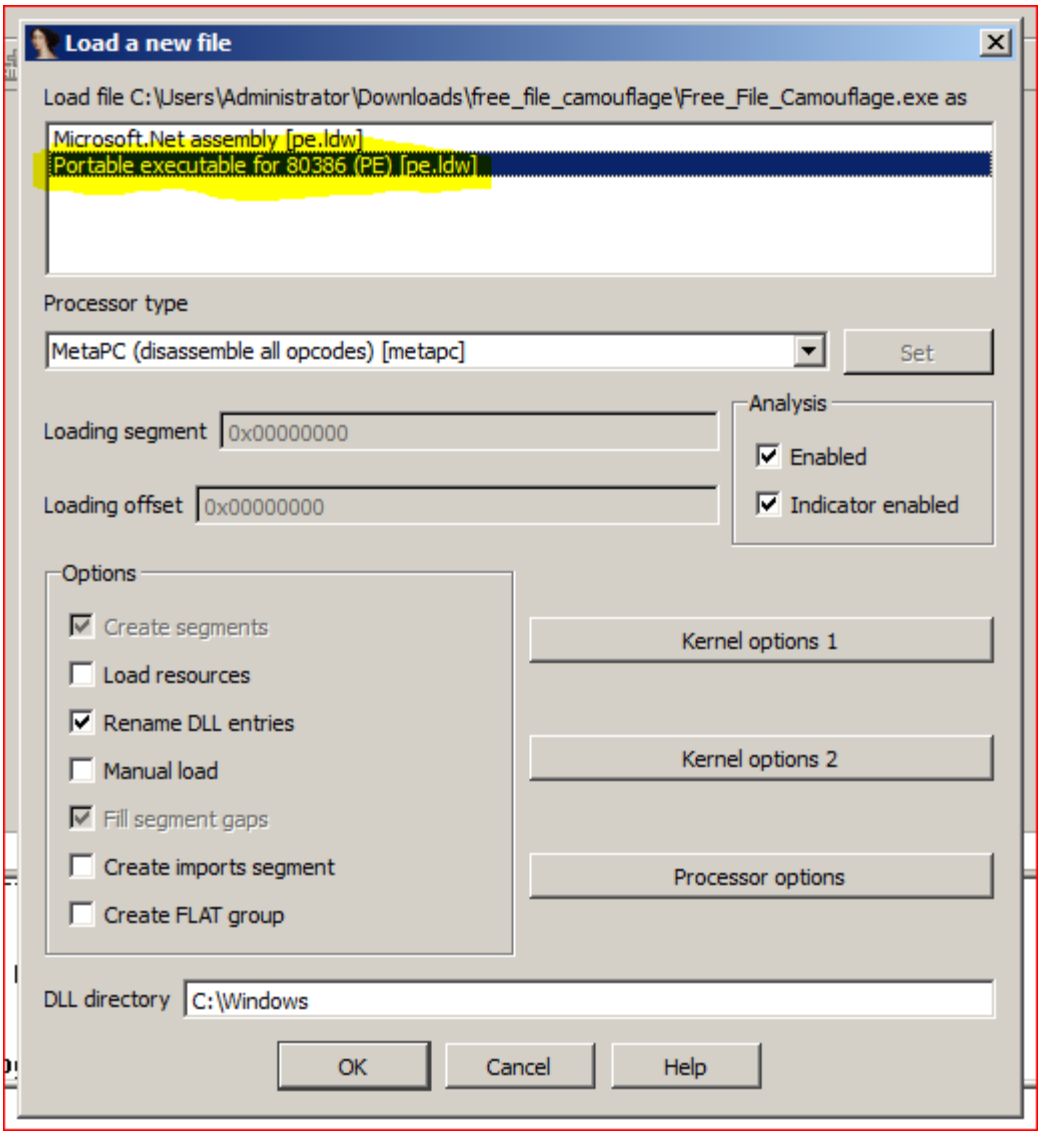
Le lancement du programme avec IDA Demo comme étant « Microsoft.Net assembly » n’a pas fonctionné, et affiche un message d’erreur comme il est mentionné ci dessous

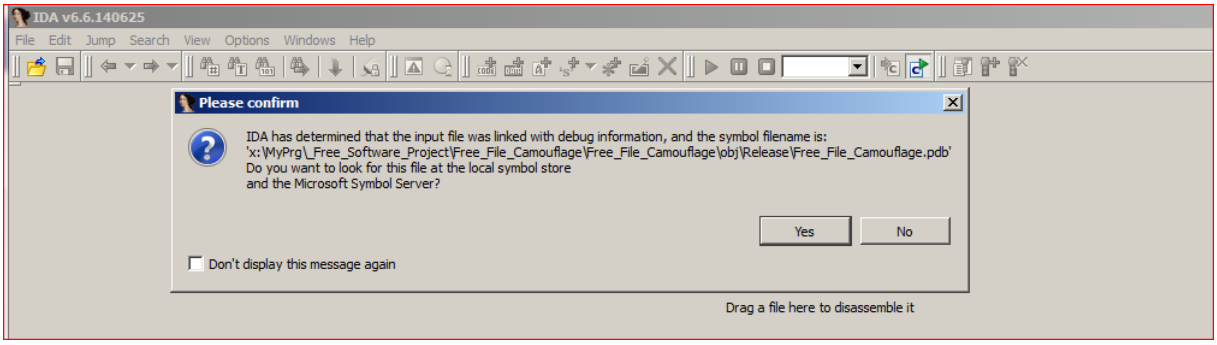


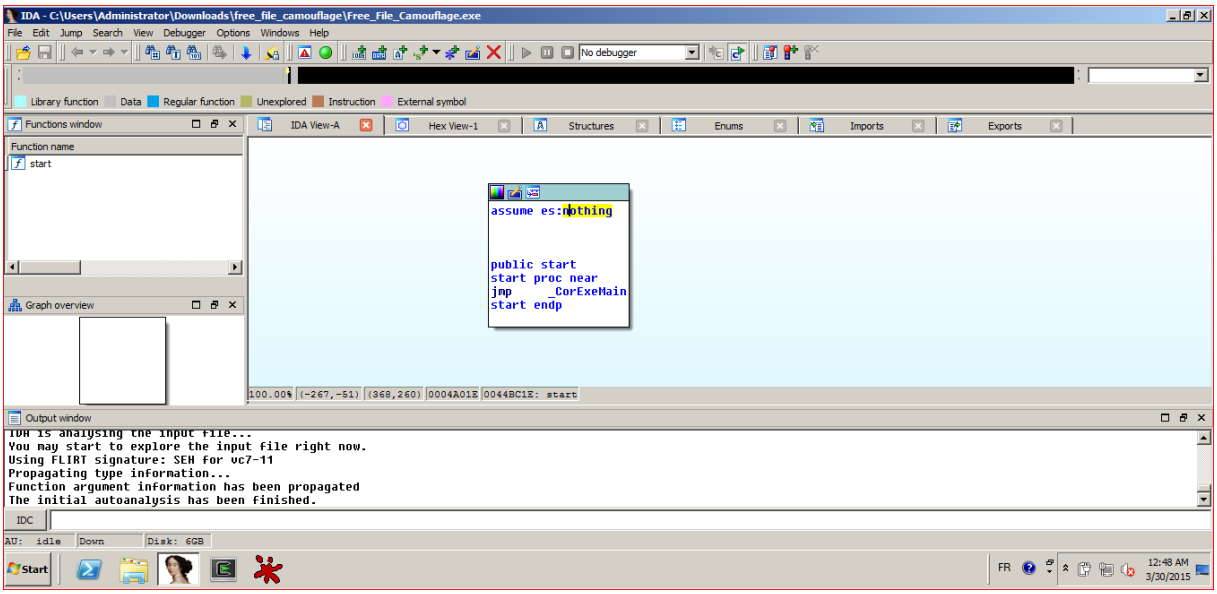


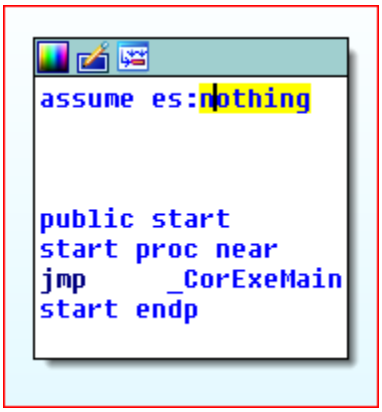


Mais en choisissant le mode "Portable executable for 80386 (PE)" comme il est montré dans l'image ci-dessous, le programme a marché

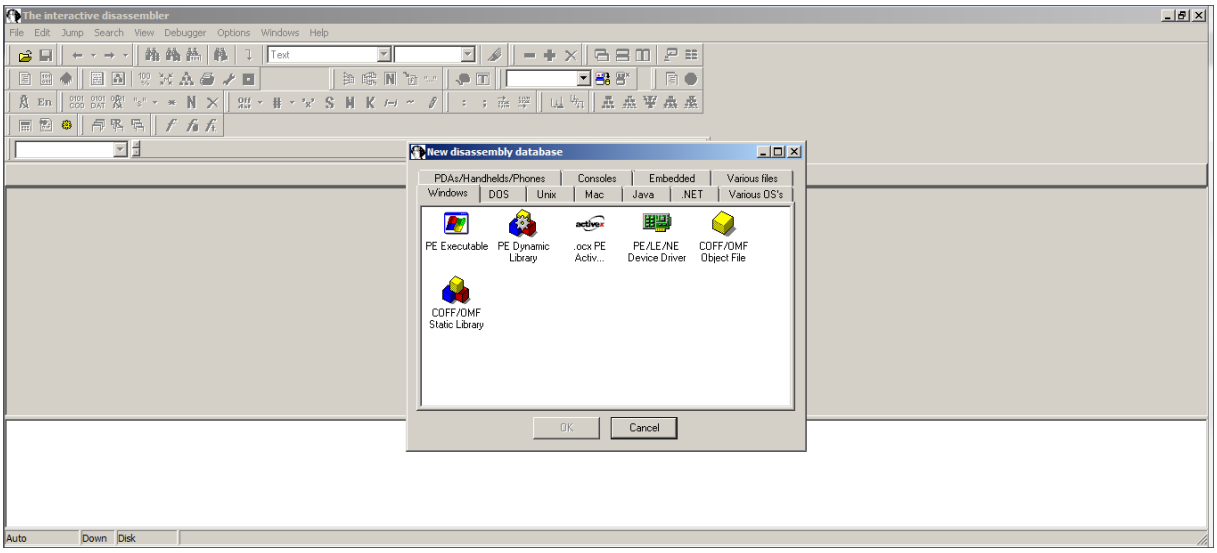

Load a new file
Load file C:\Users\Administrator\Downloads\free_file_camouflage\Free_File_Camouflage.exe as
Microsoft.Net assembly [pe.ldw]
Portable executable for 80386 (PE) [pe.ldw]
Processor type
MetaPC (disassemble all opcodes) [metapc]
Set
Loading segment
0x00000000
Loading offset
0x00000000
Analysis
Enabled
Indicator enabled
Options
Create segments
Load resources
Rename DLL entries
Manual load
Fill segment gaps
Create imports segment
Create FLAT group
Kernel options 1
Kernel options 2
Processor options
DLL directory
C:\Windows
OK
Cancel
Help

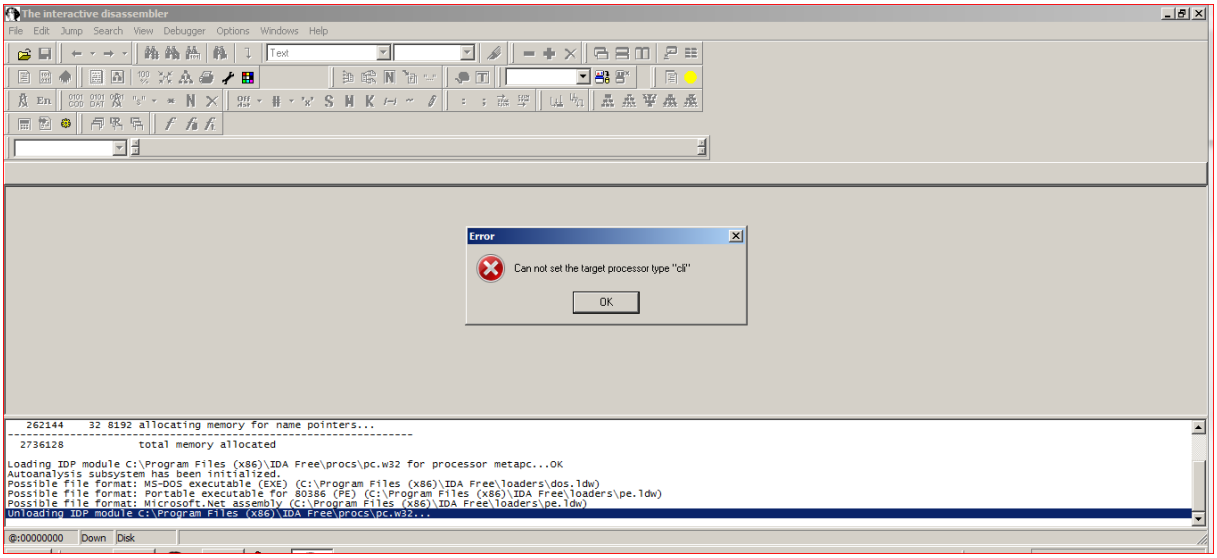

IDA v6.6.140625
File Edit Jump Search View Options Windows Help
Please confirm
IDA has determined that the input file was linked with debug information, and the symbol filename is:
'x:\MyPrg_Free_Software_Project\Free_File_Camouflage\Free_File_Camouflage\obj\Release\Free_File_Camouflage.pdb'
Do you want to look for this file at the local symbol store
and the Microsoft Symbol Server?
Yes
No
Don't display this message again
Drag a file here to disassemble it



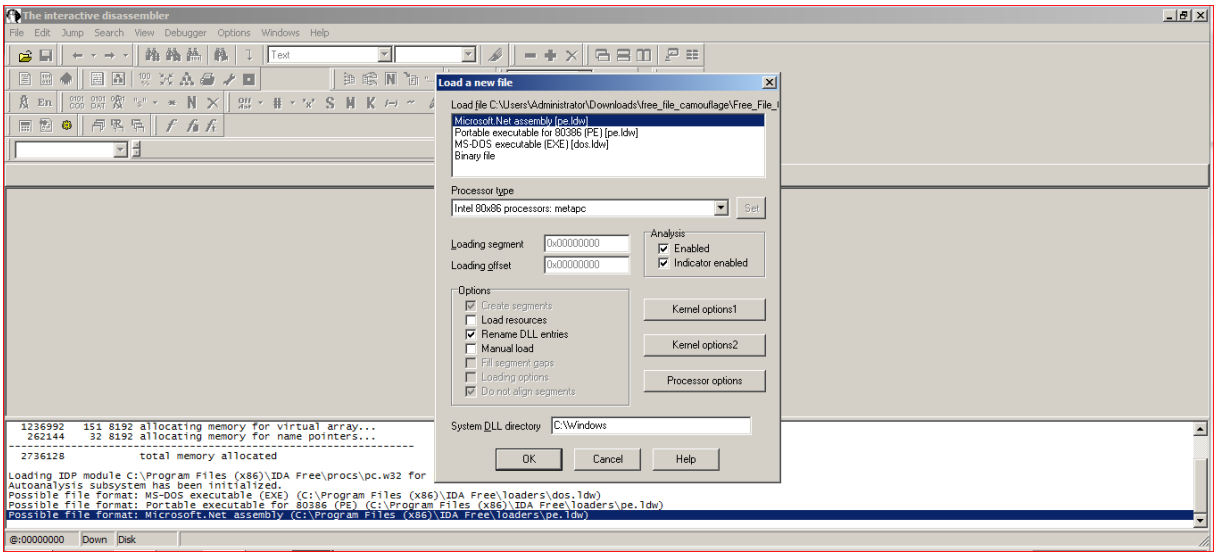


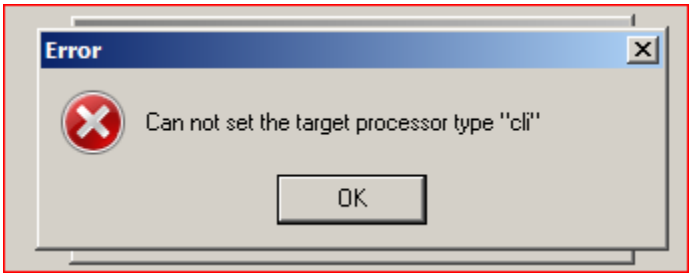
- Ouvrir le programme avec « IDA Pro Free »



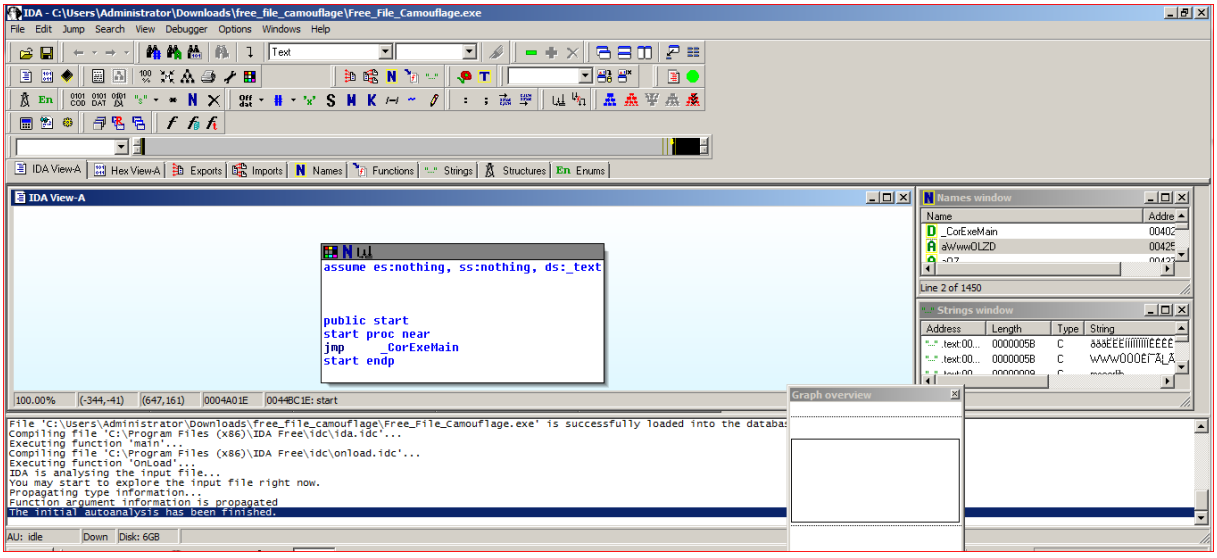


Même chose que tout à l'heure, le lancement du programme comme étant « Microsoft.Net assembly » n'a pas fonctionné, et affiche un message d'erreur comme il est mentionné ci dessous

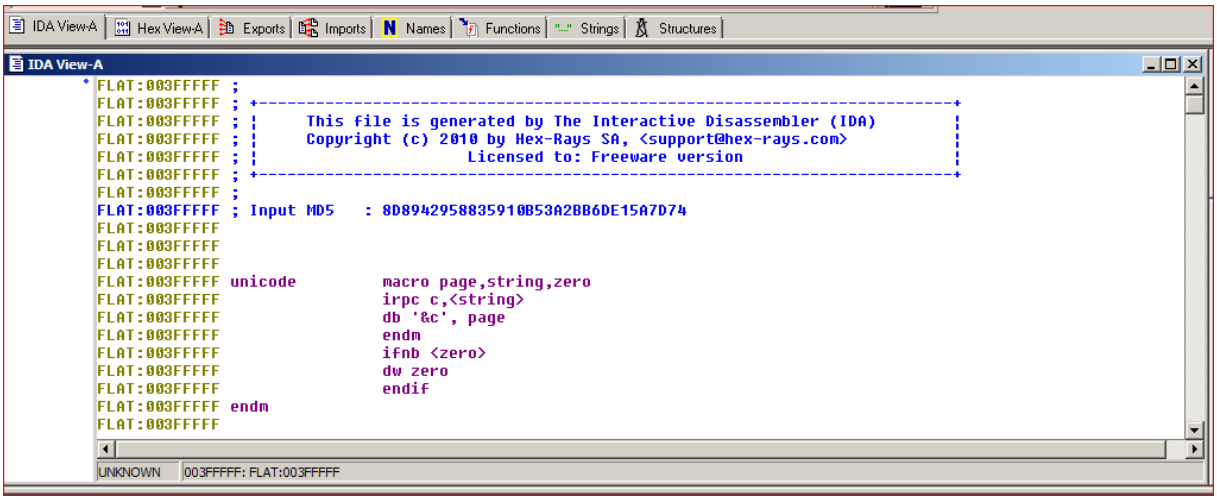




Mais en choisissant « Portable executable for 80386 », ça fonctionne



Ce que j'ai put faire à partir de là, c'est récupérer tous le code « le programme désassembler »



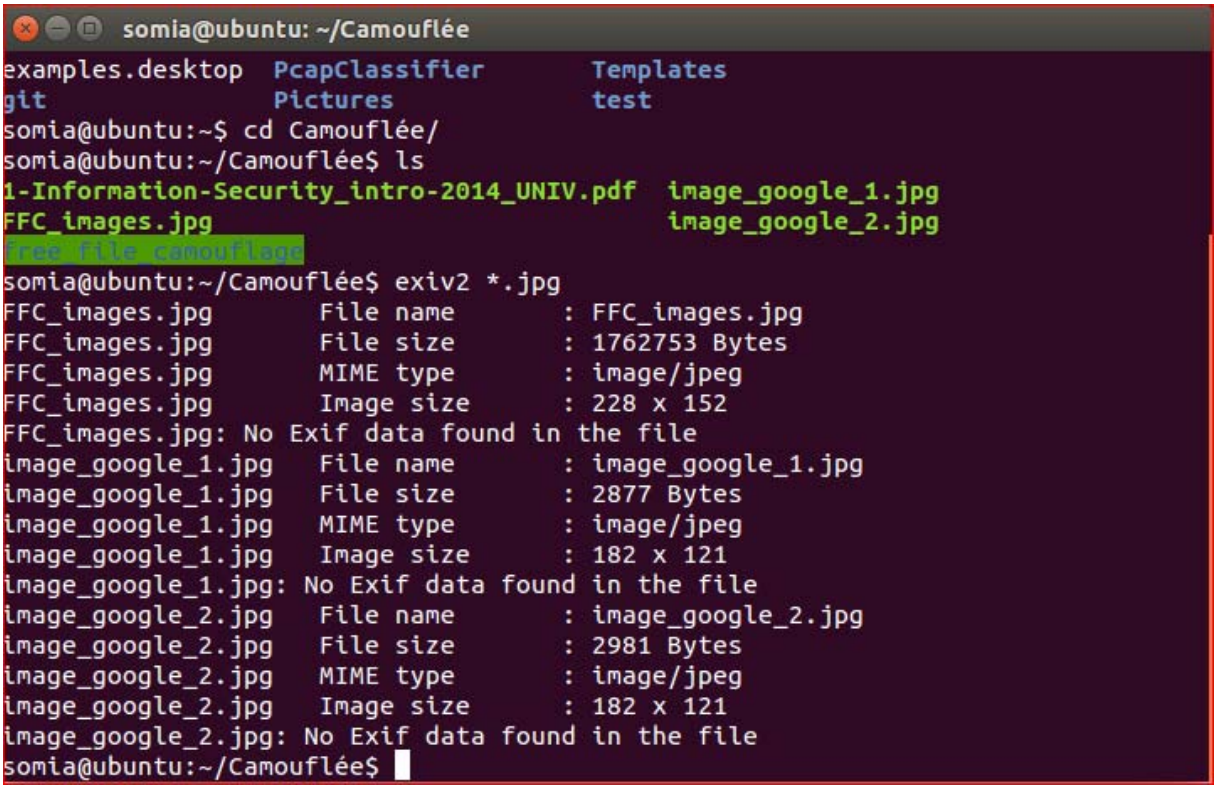
Le code est en pièce jointe nommé « IDA_programme »

Sous Linus

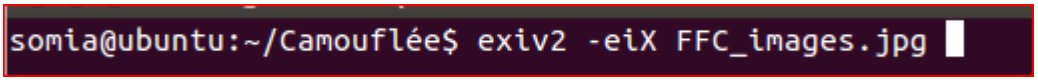
Malgré que le programme ne marche pas sous linux, ceci dit, on est resté sous linux pour effectuer les opérations suivantes essayant de comprendre au mieux le logiciel.

En premier lieu, on a camouflé un fichier .pdf sous une image s'appelant 'FFC_images.jpg', on a téléchargé du net la même image (mais sans le document caché à l'intérieur bien entendu) image_google_1.jpg et image_google_2.jpg; on a mis le tous le même répertoire

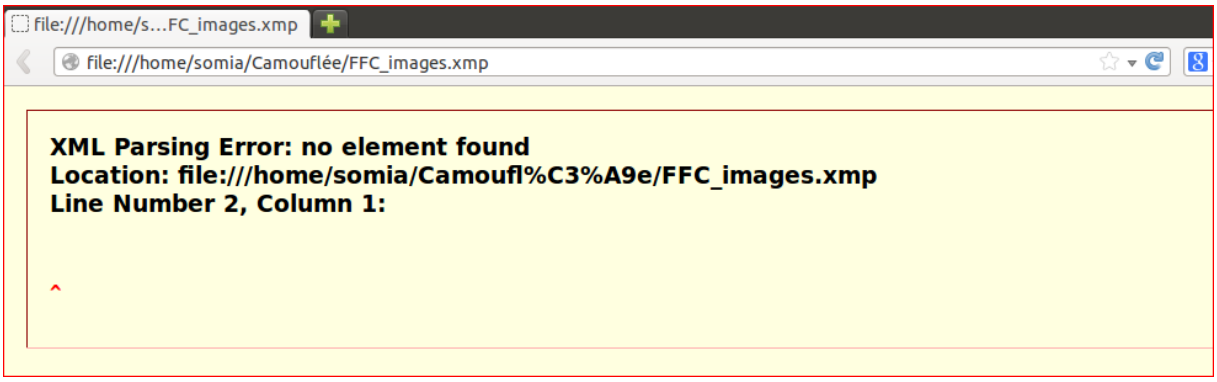
- Exiv2 *.jpg : affiche un sommaire à propos des informations Exif concernant toutes les images qui sont dans le répertoire



- Exiv2 –eiX FFC_images.xmp



Après avoir exécuté la commande, un fichier s'est créée dans le répertoire ('FFC_images.xmp'), après avoir fait un double click la dessus, ci-dessous le résultat



- File *

```
somia@ubuntu:~/Camouflée$ file *
1-Information-Security_intro-2014_UNIV.pdf: PDF document, version 1.5
FFC_images.jpg:                             JPEG image data, JFIF standard 1.01
FFC_images.xmp:                             XML document text
free_file_camouflage:                       directory
image_google_1.jpg:                         JPEG image data, JFIF standard 1.01
image_google_2.jpg:                         JPEG image data, JFIF standard 1.01
somia@ubuntu:~/Camouflée$
```

- Hexdump -C image_google_1.jpg

```
somia@ubuntu:~/Camouflée$ hexdump -C image_google_1.jpg
00000000  ff d8 ff e0 00 10 4a 46  49 46 00 01 01 00 00 01  |......JFIF......|
00000010  00 01 00 00 ff db 00 84  00 09 06 07 12 12 12 12  |................|
00000020  10 0f 14 0f 15 10 15 14  10 10 10 10 14 14 15 10  |................|
00000030  15 16 0f 15 14 16 16 14  14 14 17 18 1c 28 20 18  |.............( .|
00000040  1a 26 1c 15 14 21 3d 2d  26 29 2b 2e 2e 2e 18 1f  |.&...!=-&)+.....|
```

Concernant l'image téléchargée d'internet, après l'exécution de la commande hexdump -C, on a vu que c'est une (JFIF) ce qui signifie que le stockage de l'image est en binaire.

Idem pour l'image_google_2.jpg téléchargée elle aussi d'internet

```
somia@ubuntu:~/Camouflée$ hexdump -C image_google_2.jpg
00000000  ff d8 ff e0 00 10 4a 46  49 46 00 01 01 00 00 01  |......JFIF......|
00000010  00 01 00 00 ff db 00 84  00 09 06 06 14 0f 10 15  |................|
00000020  10 10 14 12 10 15 14 14  14 10 10 10 14 10 18 15  |................|
00000030  18 14 0f 14 10 15 15 14  10 12 17 17 1e 26 1e 17  |.............&..|
00000040  19 25 19 15 15 1f 2f 22  23 27 35 2c 2e 2c 15 20  |.%..../"#'5,.,. |
```

Concernant l'image camouflée, c'est la même chose mais on s'aperçoit que le résultat est nettement plus grand que les deux précédents

```
somia@ubuntu:~/Camouflée$ hexdump -C FFC_images.jpg | more
00000000  ff d8 ff e0 00 10 4a 46  49 46 00 01 01 00 00 01  |......JFIF......|
00000010  00 01 00 00 ff db 00 84  00 09 06 07 10 12 10 14  |................|
00000020  0f 10 10 0f 10 10 0f 10  10 0f 10 0f 0f 0f 10 10  |................|
00000030  0f 0d 15 14 14 17 16 16  14 15 15 18 1c 28 20 1a  |.............( .|
00000040  1a 25 1c 14 15 21 31 22  25 29 2d 2e 2e 2e 17 1f  |.%...!1"%)-.....|
```

J’ai essayé de trouver le mot de passe avec lequel le fichier a été crypté mais pas de résultat

```
somia@ubuntu:~/Camouflée$ hexdump -C FFC_images.jpg | grep sousou
somia@ubuntu:~/Camouflée$
somia@ubuntu:~/Camouflée$ hexdump -C FFC_images.jpg | grep password
somia@ubuntu:~/Camouflée$
somia@ubuntu:~/Camouflée$ hexdump -C FFC_images.jpg | grep PWD
00025ce0  2f 48 59 70 31 50 57 44  4c 70 68 50 69 6d 6f 54  |/HYp1PWDLphPimoT|
0003f1f0  5a 48 73 32 35 6f 44 69  56 34 63 41 71 50 57 44  |ZHs25oDiV4cAqPWD|
00066c30  74 4b 4b 69 42 39 79 50  57 44 73 48 4e 49 6f 67  |tKKiB9yPWDsHNIog|
00094160  50 6d 4d 42 76 71 37 57  35 42 50 57 44 61 56 63  |PmMBvq7W5BPWDaVc|
00174f20  4e 53 76 71 39 50 57 44  6e 6f 6c 6b 33 4b 57 43  |NSvq9PWDnolk3KWC|
somia@ubuntu:~/Camouflée$ hexdump -C FFC_images.jpg | grep pwd
00013800  70 36 66 6b 2b 63 43 6c  70 77 64 6d 39 37 59 67  |p6fk+cClpwdm97Yg|
00013f90  37 79 66 50 77 51 6c 77  74 79 47 70 77 64 72 51  |7yfPwQlwtyGpwdrQ|
00022ad0  70 77 64 43 77 33 78 4d  59 39 32 47 69 64 6e 6b  |pwdCw3xMY92Gidnk|
0002d620  70 77 64 51 62 52 34 51  73 7a 73 47 64 37 37 49  |pwdQbR4QszsGd77I|
000a31f0  45 37 2f 46 79 70 77 64  73 34 41 53 49 63 63 4f  |E7/Fypwds4ASIccO|
000bb970  70 77 64 63 77 4f 45 61  31 49 53 45 46 47 53 36  |pwdcwOEa1ISEFGS6|
somia@ubuntu:~/Camouflée$
```

- Exiftool image_google_1.jpg

Exiftool n’étant pas installé alors pour l’avoir, on a installé libimage-exiftool-perl

```
somia@ubuntu:~/Camouflée$ exiftool image_google_1.jpg
The program 'exiftool' is currently not installed. You can install it by typing:
sudo apt-get install libimage-exiftool-perl
somia@ubuntu:~/Camouflée$ sudo apt-get install libimage-exiftool-perl
[sudo] password for somia:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libimage-exiftool-perl
0 upgraded, 1 newly installed, 0 to remove and 527 not upgraded.
Need to get 2,154 kB of archives.
After this operation, 11.6 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/universe libimage-exiftool-per
l all 9.46-1 [2,154 kB]
Fetched 2,154 kB in 4s (440 kB/s)
Selecting previously unselected package libimage-exiftool-perl.
(Reading database ... 170049 files and directories currently installed.)
Preparing to unpack .../libimage-exiftool-perl_9.46-1_all.deb ...
Unpacking libimage-exiftool-perl (9.46-1) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up libimage-exiftool-perl (9.46-1) ...
somia@ubuntu:~/Camouflée$
```

```
somia@ubuntu:~/Camouflée$ exiftool image_google_1.jpg
ExifTool Version Number         : 9.46
File Name                       : image_google_1.jpg
Directory                       : .
File Size                       : 2.8 kB
File Modification Date/Time     : 2015:01:24 01:33:26-08:00
File Access Date/Time           : 2015:03:30 01:37:30-07:00
File Inode Change Date/Time     : 2015:01:24 01:38:09-08:00
File Permissions                : rwxrw-rw-
File Type                       : JPEG
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Image Width                     : 182
Image Height                    : 121
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 182x121
somia@ubuntu:~/Camouflée$
```

```
somia@ubuntu:~/Camouflée$ exiftool image_google_2.jpg
ExifTool Version Number         : 9.46
File Name                       : image_google_2.jpg
Directory                       : .
File Size                       : 2.9 kB
File Modification Date/Time     : 2015:01:24 01:33:47-08:00
File Access Date/Time           : 2015:03:30 01:37:37-07:00
File Inode Change Date/Time     : 2015:01:24 01:38:09-08:00
File Permissions                : rwxrw-rw-
File Type                       : JPEG
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Image Width                     : 182
Image Height                    : 121
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 182x121
somia@ubuntu:~/Camouflée$
```

```
somia@ubuntu:~/Camouflée$ exiftool FFC_images.jpg
ExifTool Version Number         : 9.46
File Name                       : FFC_images.jpg
Directory                       : .
File Size                       : 1721 kB
File Modification Date/Time     : 2014:10:15 02:29:08-07:00
File Access Date/Time           : 2015:03:30 01:37:24-07:00
File Inode Change Date/Time     : 2015:01:24 01:38:09-08:00
File Permissions                : rwxrw-rw-
File Type                       : JPEG
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Image Width                     : 228
Image Height                    : 152
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 228x152
somia@ubuntu:~/Camouflée$
```

Après avoir exécuté le exiftool sur les 3 images, on remarque qu'il n'y a que la taille 'File size' qui change !

- dd

```
somia@ubuntu:~/Camouflée$ dd if=FFC_images.jpg bs=1 count=1024
1024+0 records in
1024+0 records out
1024 bytes (1.0 kB) copied, 0.0126648 s, 80.9 kB/s
somia@ubuntu:~/Camouflée$
```

```
somia@ubuntu:~/Camouflée$ dd if=FFC_images.jpg skip=4048 bs=1 of=out
1758705+0 records in
1758705+0 records out
1758705 bytes (1.8 MB) copied, 7.64706 s, 230 kB/s
somia@ubuntu:~/Camouflée$
```

Après avoir exécuté cette commande, un fichier ‘out’ a été créé.

```
somia@ubuntu:~/Camouflée$ file *
1-Information-Security_intro-2014_UNIV.pdf: PDF document, version 1.5
FFC_images.jpg:                             JPEG image data, JFIF standard 1.01
FFC_images.xmp:                             XML document text
free_file_camouflage:                       directory
image_google_1.jpg:                         JPEG image data, JFIF standard 1.01
image_google_2.jpg:                         JPEG image data, JFIF standard 1.01
out:                                        ISO-8859 text, with very long lines,
 with CRLF line terminators
somia@ubuntu:~/Camouflée$
```

Après avoir fait un ‘more’ du fichier ‘ou’, on remarque que c’est un fichier encodé en base64.

J’ai essayé de le décodé mais ceci n’a pas fonctionné

```
somia@ubuntu:~/Camouflée$ sudo base64 -d out
base64: invalid input
somia@ubuntu:~/Camouflée$
```

Là on a fait un file * du répertoire contenant le logiciel

```
somia@ubuntu:~/Camouflée/free_file_camouflage$ file *
Free_File_Camouflage.exe: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for
MS Windows
Free_File_Camouflage.ini: UTF-8 Unicode text, with CRLF line terminators
Free_File_Camouflage.log: UTF-8 Unicode text, with CRLF line terminators
license.txt:              UTF-8 Unicode (with BOM) text, with CRLF line terminators
readme.txt:               UTF-8 Unicode (with BOM) text, with CRLF line terminators
somia@ubuntu:~/Camouflée/free_file_camouflage$
```

```
somia@ubuntu:~/Camouflée$ sudo base64 -d out.64 | file -
base64: invalid input
/dev/stdin: no read permission
somia@ubuntu:~/Camouflée$
```

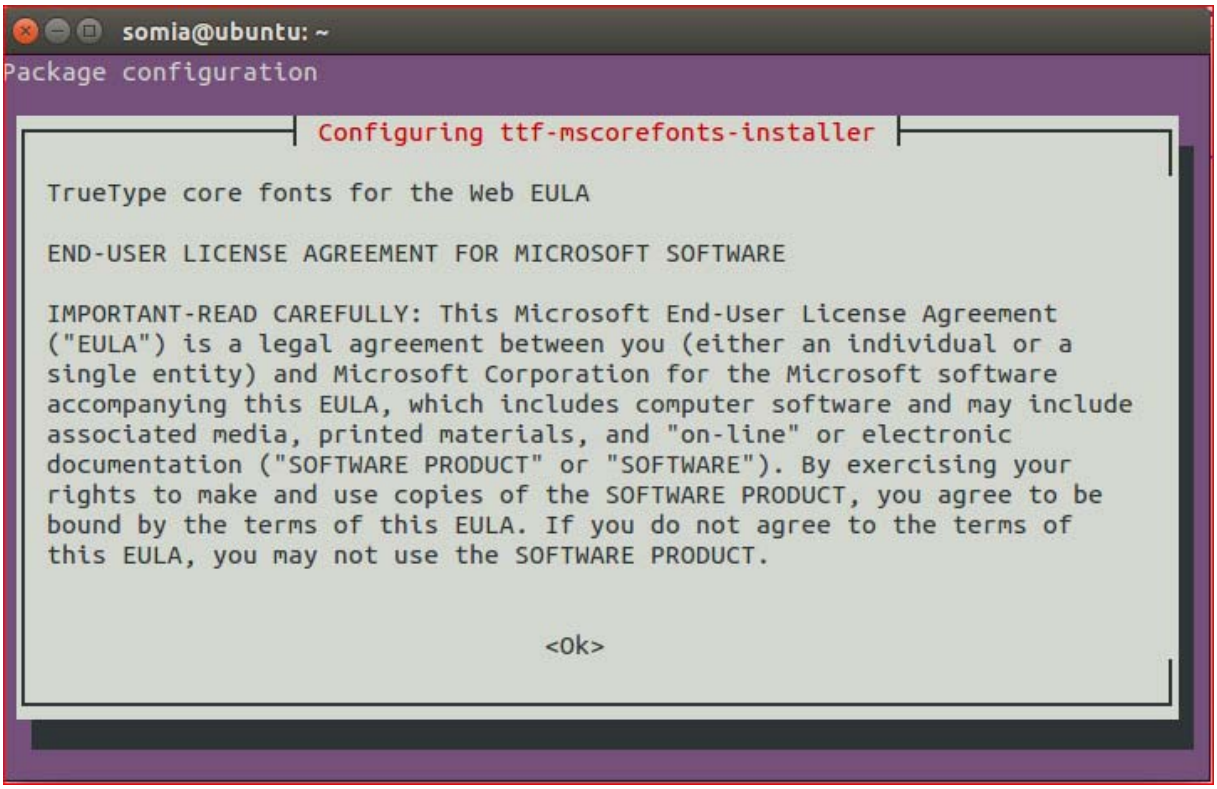
```
somia@ubuntu:~/Camouflée$ sudo base64 -d | file -f out.64
CjidVyf2GushnUDQUpcwzfhGGf7o24KQ0mxemlhDUBaGFvJ+Xr5RJulRW3srFvyjpWQXQ05Nj0ogBJCOvKhb
A1acr/iNAMAPy766xbepvdBwAXdbuc6C89K428tyj6GqZhzwoW0LtNuV4/i444pUwrJkPvfPCSvDZUIxs0Lu
hc59qQ2Ru6mux8+KcaWvaRyYslrgaIfkuh7OlvGu7maFJm8tA3+izVxlIVIPFKyjMP1sF0u3oSo/FBEdCaRY
PgoyfaYnFuVDmqnn9dM5fq9U0RvLZzbQMTmPcvROrZ1oGfVVLCmYtwEzw+HujV25CwlSm2GcYtI2F+koia7E
CjvNWACm9FcF79VImUY33y6ekX++ZZdrHkhVG8+dmDChfQPcti2/tTpkXuVFLf2Y/HhzA4AZtpQ6TQ3nXy/7
HczbaCr0p5047cYdreYhbB6BMMyZ+PGYNGNhl61HNxUabU7DkE5T/BlXeq65CZqBAa75reIek7OFtcShRg8C
m0gwxUMZjdhddJ/KpNkPPCtVa5tzZa5xEgLl0wQmbt1w5IyLikqszL41S6MdE9858x0+Rx4nPmPmRskD/His
xWET43+wU8jVz/r4WrBm1DqauPGtStVSpLW95DekZwj7qmqJekY32gRaAxllACzUV3qb/HKhO3s4qbx5OUlR
wJ/z7boEqJ2gRHAOlCs1Rpnm9jsOXyEHkH1cpYLE6/BqyKOjZW4gHSO5tsCphiXvExHGCpdEiuBe9BFpjBbx
Rf4efyuCKclWNk7U3ouH+ctYeQB1u8/Avgpj7aNorZKqPYWljxRksB/VPtzj1JzkOxDESb520tCh4tWIkyo0
Xs3FJwdk2cJMbghR6yCY9Ja7Sm4cHoZPKMvXoKI3ODitWTVOKfxLQ38HixDAqFJIgjb99Gx7Pcd/6fpykmns
9wOeIBjdjpd+4lbaeI7Q0XtDzWKH611eNYGR+9Q4CjwNJ3zFOTujlEAOLvhnQuDukXpjbgTCSD72vE02GNjL
jhY7p09gdIhdLHAobIKoy0eeuk2HB3i3U4JrjHw0YAy/OHswb8JeP/ODzO/R59G6/s7xKa2O5eYiQkmtPyd3
ycPBC8YTxe4rmp8Zu0HATiUzzNAInHO+DNTq8HCtFMfIIohxEnVQLNlUYbGMhDmzqcxLPMMe+nlHlTW0AR4l
0PIrRuIDSShd+InhATTBWVU3M489Evr1aHbN1QNS/aUzG405ty/OAM4ijCYetUAArIZSCHDCMdK55c1vum4k
ZtTV5J1cKSN0QvTU00Gqs4OZJoZn1fWY5MSXAKQIlfB2aE1zu/gD7tCD+jdUk441zvKFntkTTIvVR4Eil4gG
P9BvGZIkOW+17dGX0qXcW1nkwfgw71kAGK7Qtxkw3dHfM2inXe3FhXGCD45yvmMyj4dqQoDVNS9PmgpIoI/P
KwUhbphwsQ2wtg5r87CmQcEneppCFmpMZ2nMup+RfkIRgnrVtEVxH8vmMZv98FShSASxh/szjP/bY9zbUgry
7hWQYN+6tA5P2/i0nWfiJv1GCPWxuOqUAnT27WVeKohPkLqqxcJ4MlYEnLF4Eka5IxCZCcydOaFBhP+uz4AO
xvzmzWnxrcqxOVp+KJwPiscU9nGGEKaDIEqgtCfLrnUNwk7PbsWm5hE7jqjR5WCUdf8DKv22VqTUIBBn0e+W
AWyeglEK57IJV4ZpidD1KMrPkwc2npybYFnV7n+FXetS3gDY7L8K2RwcbyFu16hKM++Rzn9a8j7SGnPGEWcU
oUGlm3vReffCLOh+YSCGlZCOpyIQmNIhs/+CMnGv+9AjeSl7+YvXvYyFH3vbktA6AkctyplPkREyaKGaisnP
s6H/9Tr6++n+rHNcOSQFGqwwV8oSo0NCSG79VDBUV+paMtPAeuzflg3Ez+I0S/Rr9xfXwae3k2EEGsn4Xesk
MhyJICNFNZoI+GMVErLNvmPEsQesvS+lc5tH19KVFf8w97A3gPbD5Z4q+zZCh298I8nK4m7FQMlI96ZCXgsj
cd++QU/6/QO0jXsSzcY0Mz8JkFaXWaj5gSs8EAChDq01w2OVe52JpvGDyDp2VNlqzh6jITnsylL+4A3p+pC1
VYg+Ate5my89PVbI+jzMf5bvvGLtspjd1J6wn0fJE6H//0Q30AzNJBrku6gf33+ItyqkUuTh9sbaufmOCpna
dFuG2RPlmacryPbczbxKXUZOfT2JIRI7lzaBUqzcEqMevf1mFM2w7xpA1KjUStofYKjXm07JUzRWsSkvzXTx
Qdrs4eKooI4/J8wMfMznjGGZvcefchmh0rUjD7ZkSzYo02h6bhtr2bC2dreK1XSozQrvONrlj6xvCTIH22nT
NzfC38NkP5hJY4uLQWMWdtYoRDwYWOU47RWrcEhOexOHh1aaLDAxs1DezPjKHkmrvxChFBKupBTVxszx9VhV
TYYe7tbgTzFFPPKimhorhgG+6QsUv+qsg2YJjle2TbMwRXlcNVCv1irKTKAWEA1IhXgw8CTO1aPgAReLHrym
Uc/9sXxNu8hW6MW6VkFF3zTmM9vqjEx/o5wcPFRexKjysgGnfmaNXvJRkzqgWkluH+ZfGXwcopaPwokHzidt
7oldopFcghQ3Oy/aC8iQhpN1cX1MwBw1xVA7Ji0phRicNMGHK4CRjA4lzw8Aq2GVQQODJlABuPTX+jfPp7CM
' (File name too long)vyCiChh9SQ3qBSaV7YhvqsSFDqlpbDg=
```

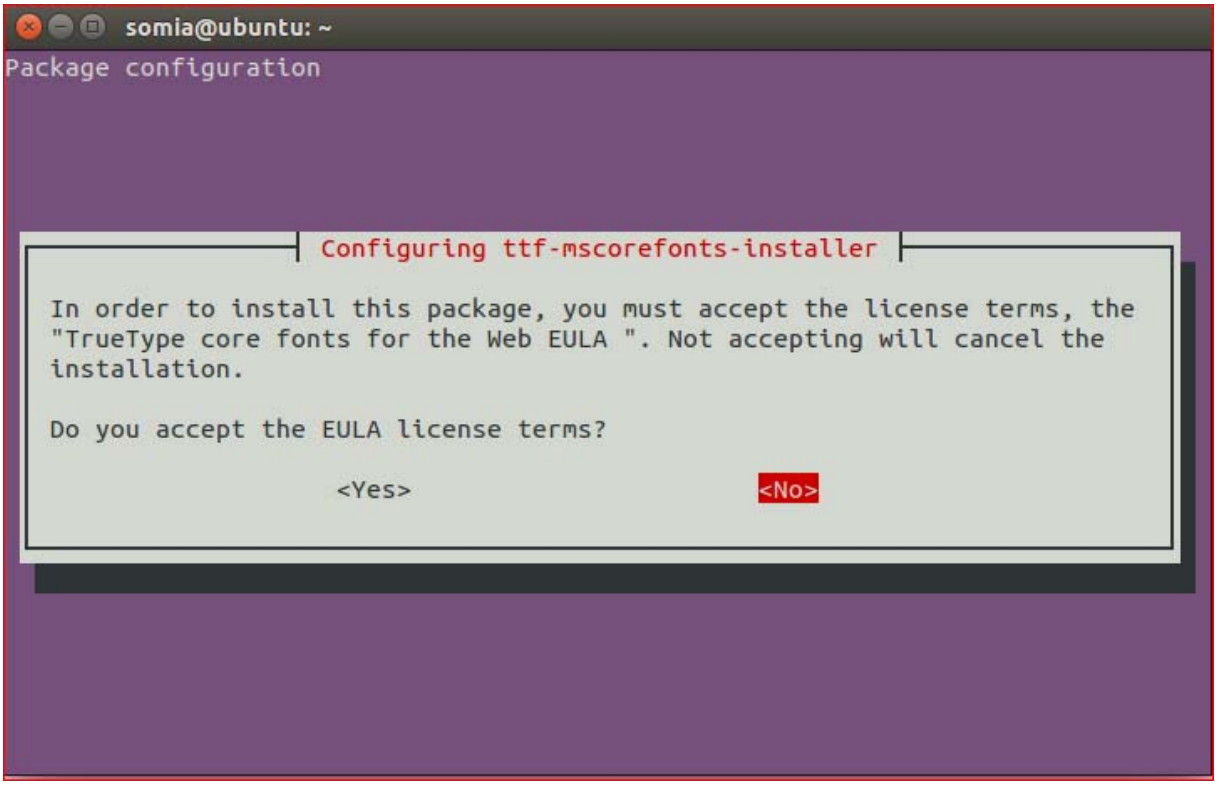
- imagemagick

```
somia@ubuntu:~/Camouflée$ cat image_google_1.jpg | convert -strip - - | md5sum
The program 'convert' can be found in the following packages:
 * imagemagick
 * graphicsmagick-imagemagick-compat
Try: sudo apt-get install <selected package>
d41d8cd98f00b204e9800998ecf8427e  -
somia@ubuntu:~/Camouflée$ sudo apt-get install imagemagickl
```

- wine ! L'outil a été téléchargé pour pouvoir tester et le logiciel camouflage.exe et aussi pour pouvoir installer 'idag.exe' et l'exécuter

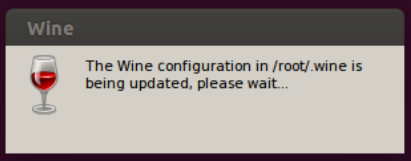
```
somia@ubuntu:~/Camouflée$ sudo apt-get install wine
```





Après la fin de l'installation de 'wine', on a essayé de lancer « wine free_File_Camouflage.exe » mais ceci n'a pas marché

```
somia@ubuntu:~/Camouflée/free_file_camouflage$ sudo wine Free_File_Camouflage.exe
wine: '/home/somia' is not owned by you, refusing to create a configuration directory there
somia@ubuntu:~/Camouflée/free_file_camouflage$ sudo su
root@ubuntu:/home/somia/Camouflée/free_file_camouflage# wine Free_File_Camouflage.exe
wine: created the configuration directory '/root/.wine'
```




```
err:msi:cabinet_copy_file failed to create L"C:\\windows\\mono\\mono-2.0\\lib\\mono\\gac\\Accessibili
ty\\2.0.0.0__b03f5f7f11d50a3a\\Accessibility.dll" (error 3)
err:msi:extract_cabinet_stream FDICopy failed
err:msi:ACTION_InstallFiles Failed to extract cabinet: L"#image.cab"
err:msi:ITERATE_Actions Execution halted, action L"InstallFiles" returned 1603
err:appwizcpl:install_file MsiInstallProduct failed: 1603
fixme:storage:create_storagefile Storage share mode not implemented.
err:mscoree:LoadLibraryShim error reading registry key for installroot
err:mscoree:LoadLibraryShim error reading registry key for installroot
err:mscoree:LoadLibraryShim error reading registry key for installroot
err:mscoree:LoadLibraryShim error reading registry key for installroot
fixme:iphlpapi:NotifyAddrChange (Handle 0x114e2b8, overlapped 0x114e2d0): stub
fixme:storage:create_storagefile Storage share mode not implemented.
err:mscoree:LoadLibraryShim error reading registry key for installroot
err:mscoree:LoadLibraryShim error reading registry key for installroot
err:mscoree:LoadLibraryShim error reading registry key for installroot
err:mscoree:LoadLibraryShim error reading registry key for installroot
err:msi:cabinet_copy_file failed to create L"C:\\windows\\syswow64\\gecko\\2.21\\wine_gecko\\dictiona
ries\\en-US.dic" (error 3)
err:msi:extract_cabinet_stream FDICopy failed
err:msi:ACTION_InstallFiles Failed to extract cabinet: L"#winegecko.cab"
err:msi:ITERATE_Actions Execution halted, action L"InstallFiles" returned 1603
err:appwizcpl:install_file MsiInstallProduct failed: 1603
Could not load wine-gecko. HTML rendering will be disabled.
wine: configuration in '/root/.wine' has been updated.
wine: Install Mono for Windows to run .NET applications.
```

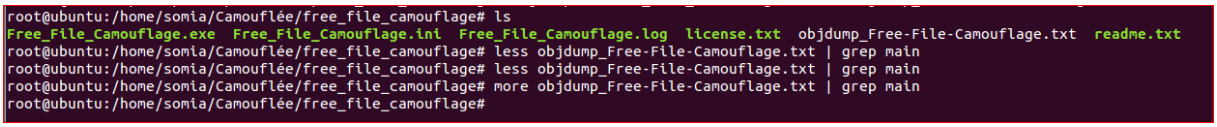
```
root@ubuntu:/home/somia/Camouflée/free_file_camouflage# wine Free_File_Camouflage.exe
wine: Install Mono for Windows to run .NET applications.
root@ubuntu:/home/somia/Camouflée/free_file_camouflage#
```

- objdump

```
root@ubuntu:/home/somia/Camouflée/free_file_camouflage# objdump -d Free_File_Camouflage.exe >> objdump_Free-File-Camouflage.txt
```

```
objdump -d Free_File_Camouflage.exe >> objdump_Free-File-Camouflage.txt
```

J'ai ensuite fait un grep du 'main' sur ce fichier, mais aucun résultat n'y apparait !



L'idée été de pouvoir trouver soit le mot de passe ou un indice pour remonter au mot de passe et ainsi pouvoir décrypter le fichier.

C'est-à-dire en trouvant où dans le programme il fait le test pour voir si le mot de passe est correct alors il va faire le décryptage ; ce test là, je voulais le changer pour que le test soit toujours vrai. C'est-à-dire avec ou sans le bon mot de passe, le fichier se décryptera toujours.

Si j'avais trouvé l'emplacement, la prochaine étape été d'installer « ghex » qui est un éditeur hexadécimal pour pouvoir modifier le test en question et permettre le décryptage quelque soit le mot de passe.

J'ai aussi fait exécuter le logiciel Free_File_Camouflage.exe tout en ayant wireshark en exécution, ainsi j'ai obtenu un fichier pcap, qui est aussi joint à ce fichier, et ce afin de pouvoir étudier le fichier pcap, pour voir lors de l'exécution du programme, est ce qu'il se connecte à internet (il s'est avéré que oui), en se connectant à internet, il fait quoi, etc.