Introduction au cracking

(explications de base)

(Les explications données dans ce tuto sont très simplifiées, afin que les plus débutants puissent comprendre facilement, ce tuto n'a pas pour but de donner un cours précis sur le cracking, mais de faire comprendre aux novices en quoi cela consiste .)

By Mr_roW

1/ Qu'est ce que le cracking

Le cracking, c'est le fait de contourner la sécurité des logiciels, exemple :

Vous avez un programme en version d'essai, il suffit de le « cracker » pour avoir une version illimitée.

Bon d'accord là je fais un peu simple, pour expliquer correctement, techniquement, cracker un programme consiste à le désassembler, et donc , d'obtenir son code source en langage Assembleur, afin de comprendre son fonctionnement et de modifier certaines données de celui ci pour , en général, contourner les sécurité, exemple, le programme demande d'entrer une clé d'activation, en modifiant certaines parties du programme, on peut modifier la condition qui teste le mot de passe entré pour que le programme accepte même un mot de passe faux.

2/ De quoi ai-je besoin pour cracker ?

Eh bien vous avez besoin de :

1/ Un desassembleur, comme : Ollydbg

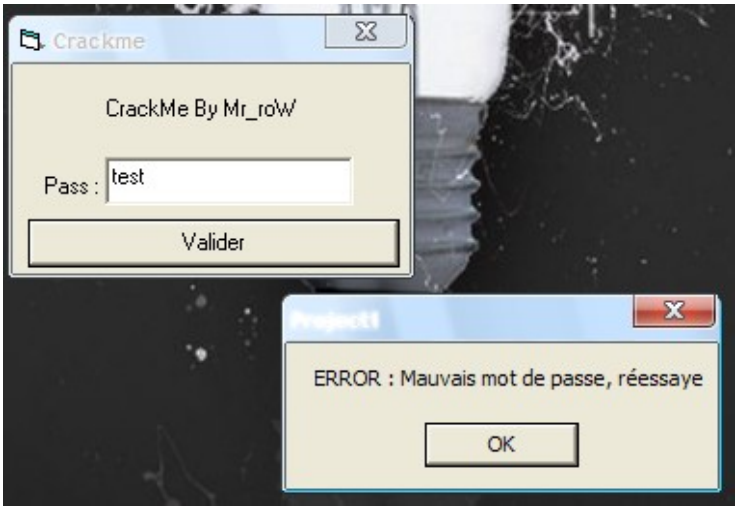
2/ Un éditeur hexadécimal, comme : EditHexa

3/ Des connaissances en Assembleur (car je ne donnerai pas de cours ici)

4/ Un cerveau performant

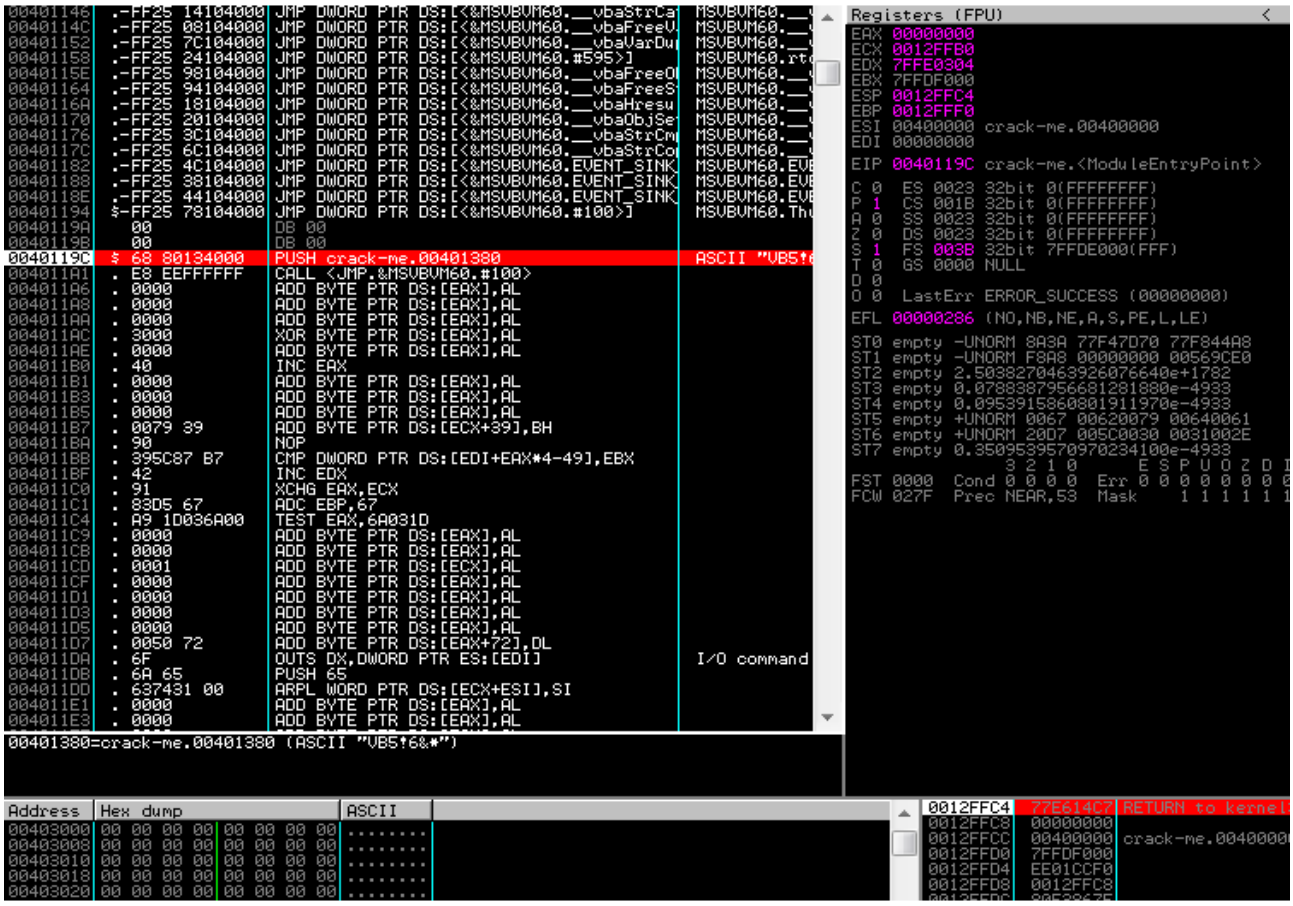
3/ Explications

Bon, pour vous expliquer cela correctement, j'ai créer un crack-me tout simple ;)



Comment faire (à part en devinant le mot de passe :p) pour réussir à contourner la sécurité ?

Eh bien c'est simple, pour cela on va désassembler notre programme à l'aide du désassembleur (dans le cas présent : Ollydbg).



Olalalalalah mais qu'est ce que c'est que tout ça ? Quel démon à pu inventer une horreur pareille ? ^^

Non , honnêtement ça parait monstrueusement compliqué a première vue mais ça ne l'est pas tant que ça ;)

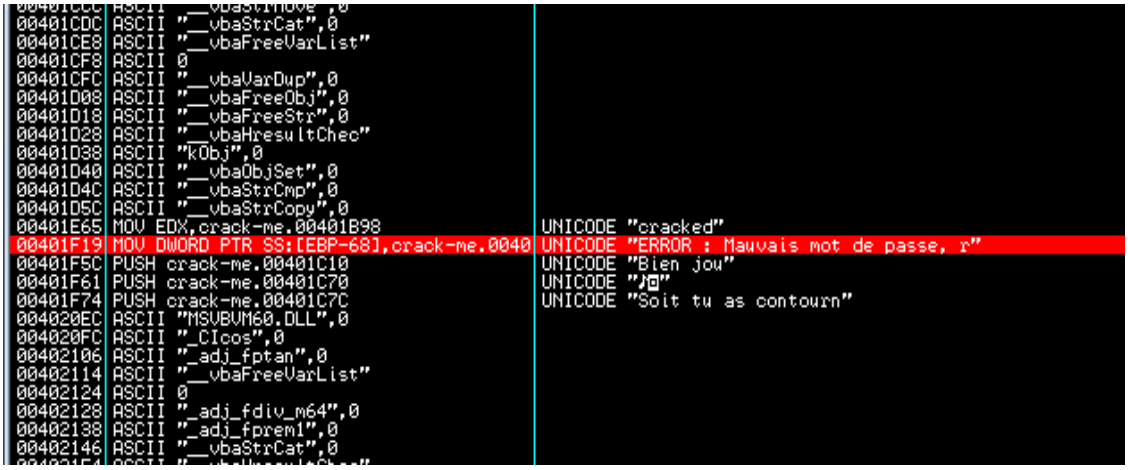
Le code source du programme est donc ce que vous voyez dans cette fenêtre en haut à gauche.

Et là un outil magique va nous venir en aide, on fait un clique droit dans le fenêtre en haut a droite, et on clique sur : search for > All referenced text strings

Ce sont les chaines de caractère du programme qui sont disponible dans leur format ASCII .

Donc une fois que l'on a cliqué là dessus, on a les chaines de caractères ASCII du programme (celles qui sont disponibles évidemment ^^)

Nous avons vu plus haut que le programme affiche un message d'erreur en cas de faux mot de passe, on peut donc se douter qu'il mettra un message de validation en cas de password valide, en cherchant un peu dans les « referenced text strings », on arrive a trouver celà

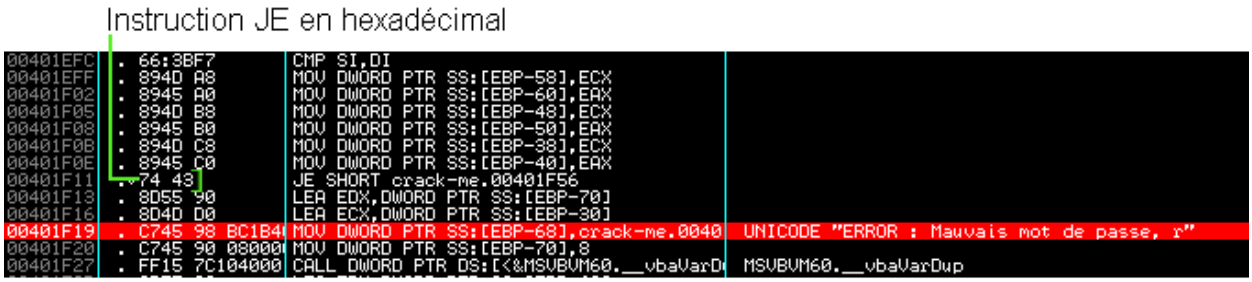


Tiens donc ^^ mais ne serai-ce pas notre message d'erreur sur la ligne que j'ai surligné en rouge ?

Et si on regarde bien en dessous, on voit un « Bien jou...» (chaine coupée, car le «é» ne passe pas au désassembleur ;))

Vous vous doutez bien que cela est notre message de validation .

Bon double cliquons sur la ligne du message d'erreur, on se retrouve dans le code source (en ASM) à l'endroit ou le message d'erreur apparait



On arrive ici, la ligne surlignée en rouge correspond au message d'erreur, vous l'aurez compris ;)

Si on regarde un peu plus haut au dessus du message d'erreur (ligne indiquée par une flèche verte)

Il y a une instruction nommée **JE** , en assembleur, l'instruction JE est un saut conditionnel, (JUMP if EQUAL), Je n'ai pas besoin de vous faire un dessin, le programme fait un saut au message « Bien joué » si le mot de passe entré par l'utilisateur est égal à la variable qui le contient

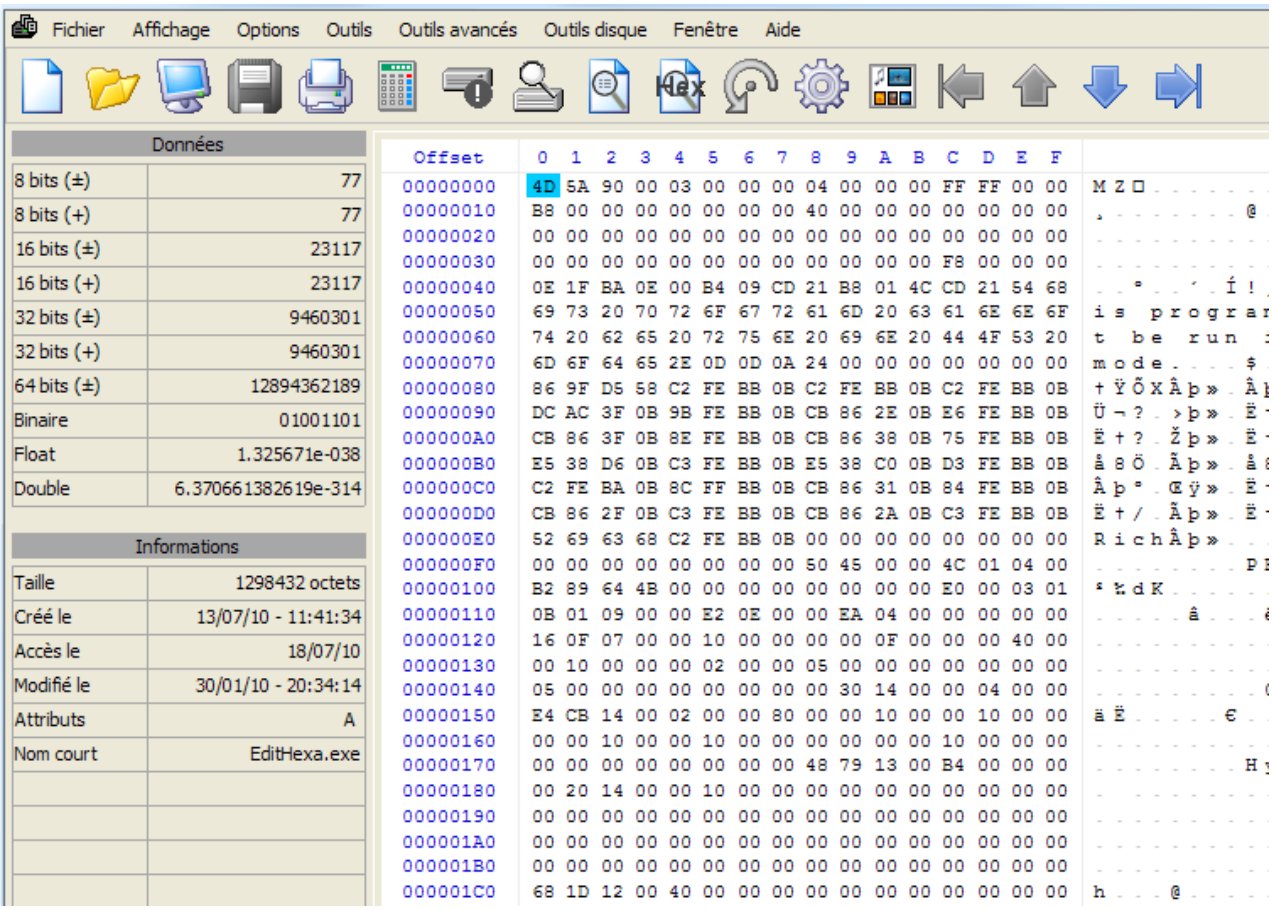
(d'ailleurs on peut le remarquer dans Ollydbg, car si on se met sur la ligne du JE et que l'on appuie sur Entrée, le désassembleur nous place à l'endroit du code où le message « Bien joué » apparait)

Pour vous renseigner sur les instruction principales en assembleur,

jettez un oeil ici : http://www.commentcamarche.net/contents/asm/liste.php3

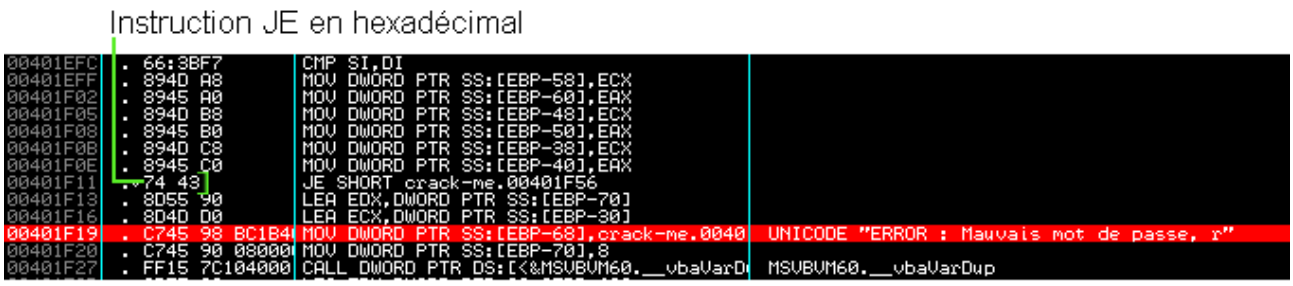
Il nous suffit donc de modifier cette condition dans le programme, l'instruction JE en hexadécimal indiquée sur le screen est **74 43** , il nous suffit donc de modifier le 74 en 75 à l'aide de l'éditeur hexadécimal, pourquoi 75 ? car 75 est un «JNE» c'est a dire une instruction qui fait le contraire de JE, il fait un saut au message de réussite «Bien Joué», tant que le mot de passe est faux dans le champ texte ^^ , il n'interdira donc pas l'accès ,

On ouvre donc notre crack-me avec l'éditeur hexadécimal :



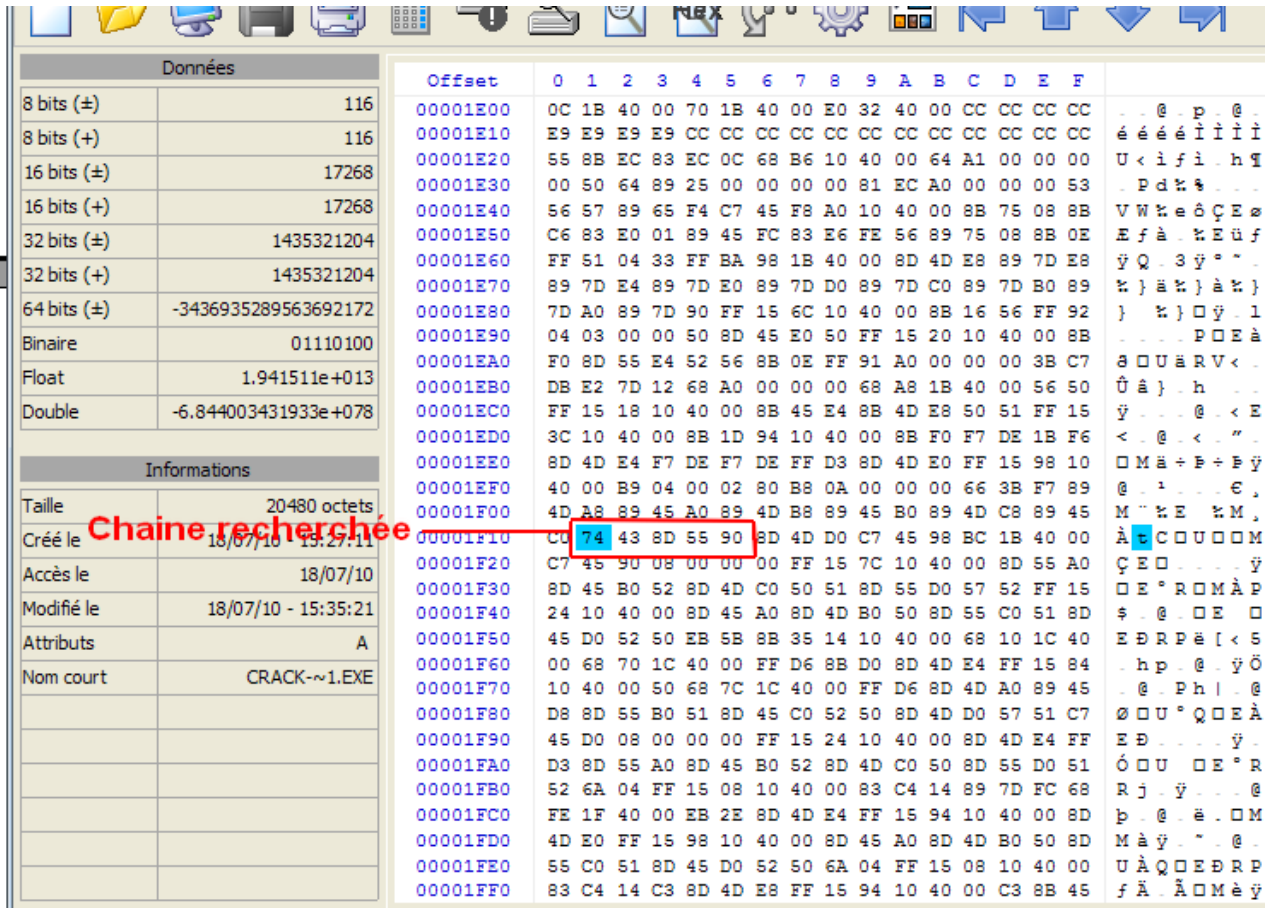
Eh oui, tout ce que vous voyez là, c'est notre programme sous forme de chaines hexadécimales, **(avec les caractères ASCII disponibles a droite ^^)**

Donc, pour retrouver notre JE, il faut retourner à OllyDbg , et regarder la chaine exacte de l'instruction (avec la chaine précédente et la chaine suivante)



On voit donc que si on prends la chaine du JE avec la chaine lui succédant, la chaine hexadécimale est : **74438d5590**

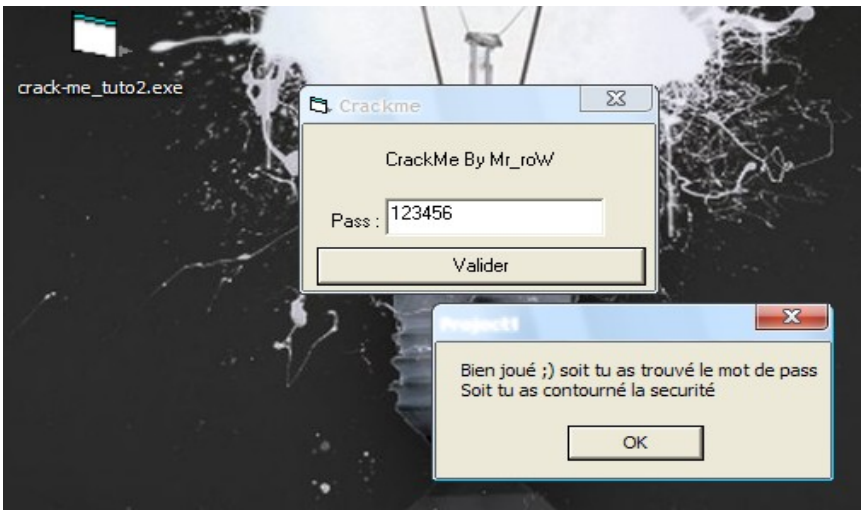
Pourquoi prendre la chaine suivant notre instruction ? Hahaaa ? Car on peut facilement retrouver plusieurs fois la même occurrence dans le programme, donc afin de ne pas se retrouver avec une vingtaine de 74 43 a trier pour trouver le bon ^^ on prends une chaine plus grande, comme ça, moins de chances de la retrouver plusieurs fois :) **(Généralement on prends aussi un morceau de la chaine précédant l'instruction, je ne l'ai pas fait das mon exemple, mais je le précise ;))**



Nous n'avons plus qu'a modifier notre 74 43 en 75 43 comme je l'ai mentionné plus haut, on sauvegarde ensuite notre fichier (de préférence sans écraser notre fichier d'origine, donc sauvegardez sous un autre nom)

Ensuite, on lance le crack-me modifié, on entre un mot de passe bidon, exemple : 123456

et on valide ;)



Voilà, c'était une brève introduction au cracking pour ceux qui se demandaient en quoi ça consistait.

Évidemment, pour de vrais programme cela n'est pas aussi rapide, il faut un certain temps avant de trouver LA ou LES sécurités a contourner. Si après cette introduction le cracking vous intérresse, je vous conseille ce site afin de suivre des cours bien construits, bien expliqués et qui vous aideront à avancer à grands pas dans le monde de la sécurité logicielle.

Lien du site : http://www.deezdynasty.xdir.org/tutorials.html

Cordialement, Mr_Row ...