Tutorial Crackme1 de Coronetix

Introduction

Bienvenue dans ce petit tutorial ceci est ma première contribution pour DéfisFC. Le challenge proposé par Coronetix n'étant pas compliqué (il se trouve que c'est lui aussi, son premier challenge) il n'y a pas vraiment de mérite à en tirer. Ceci dit quand on est débutant, il est tout a fait possible de bloquer dessus, simplement parcequ'on appréhende pas le code correctement. Grâce a une méthode courte de pas à pas, vous aurez les bons réflexes, et vous y verrez plus clair à travrs l'analyse d'un binaire.

Ca commence

Lancez le programme de Coronetix, appuyez sur **E** pour enregistrer le programme, entrez un mot de passe bidon (par exemple 1234). Un message d'erreur ! Retenez le bien, ça va nous servir pour notre partie de pêche !

Commençons par nous assurer que cet executable n'est pas compressé par un packer.

- Lancez l'éxécutable dans pied (soit en drag & droppant, soit en allant le chercher dans votre disque grâce au bouton [...]

--> Ok ! Pas de problème, pied nous indique qu'il a été codé avec Dev-C++.

Maintenant, nous allons charger l'exe dans OllyDebugger (soit avec un drag & drop, soit via le menu « File > Open » du debugger.

- On commence sur un PUSH, la vue du code ne montre rien de bien impressionnant, on commence notre pêche: souvenez vous du message d'erreur, il est quelque part dans toutes ces lignes, pour le trouver on va utiliser Olly. Click droit sur la fenetre du debugger, « search for » puis choisissez « all referenced strings ».

--> On obtient une grande liste, parcourez-là des yeux pour trouver notre message d'erreur précédemment rencontré.

```
Text strings referenced in crackme:.text, item 53
 Address=00401627
 Disassembly=MOV DWORD PTR SS:[ESP],crackme.00401490
 Text string=ASCII 0A,"Erreur de "
```

La voilà ! Double-cliquez dessus pour aller dans le code.
On se retrouve sur un MOV, l'endroit précis ou le message d'erreur est montrée dans le programme.

Remontez un petit peu...que voyez vous en 004015B2 ? Il s'agit sûrement du bon message à

obtenir ! Pour récapituler on sait ou se trouve le bon et le mauvais message.

Il y a quelque part dans le code une condition, une vérification qui teste si le serial entré est le même que celui attendu..continuons.

- Remontez un peu juste au dessus du bon message, en 4015A4. On voit en rouge marqué SCANF. C'est le nom de la fonction en langage de programmation C, qui lit ce que l'utilisateur a entré.

Suivi d'un CMP DWORD PTR SS:[EBP-8, 2B07C30]. Qu'est ce que cela signifie me direz vous ? Et bien il s'agit de l'opération de comparaison, entre ce que l'utilisateur a entré (moi c'etait 1234) et une valeur.

Constatez qu'immédiatement après cette opération il y a un JNZ qui fait sauter le programme vers le message d'erreur si les valeurs comparés ne sont PAS identiques.

Que faire ?

On a notre valeur de comparaison (2B07C30), nos messages et on sait que si ce que l'on entre n'est pas identique à notre valeur on aura faux. Cette valeur est une traduction hexadécimale du véritable serial. Prenez la calculatrice Windows en mode scientifique, cliquez sur le bouton « Hex », puis entrez la valeur dans la calculatrice. Maintenant cliquez sur « Dec », vous obtenez 45120560 ,le serial ! Pour vous assurer de sa validité, fermez Olly, lancez le programme de Coronetix et entrez la valeur traduite pour vous par la calculatrice. C'est bon, c'est fini !

Outroduction

Nous avons eu de la chance, cette fois ci le serial était apparent, et une partie de serial phishing nous a bien renseigné. Mefiez-vous ! Ce ne sera pas souvent le cas, et parfois il y aura des fonctions pour désactiver votre debugger, voir crypter le contenu du programme. Il existe beaucoup plus court pour tutorer ce challenge, mais le but était surtout de visualiser une démarche de réflexion et de recherche.

Remerciements

FRET et FC, DefisFC, Coronetix pour son challenge de débutant et vous, lecteur.