Cours de Cracking

(10ième Partie)

Mon objectif : craquer winRescue95 version 8.01

1/ Les logiciels utiles pour ce cours

-> Le programme à craquer : **winRescue95**
-> Un désassembleur : **W32dasm 8.93**
-> Un éditeur hexadécimal : **Winhex 10.2**

2/ Les protections

-> Limitation dans le temps (30 jours)
-> Registration par code
-> Certaines options bridées
-> Un Nagscreen au lancement + 9 secondes d'attente (pas une protection c'est juste chiant)

3/ Installation

Commencez par installer Winrescue dans le répertoire de votre choix sur votre disque dur.
Cela fait, au boulot !! (faut arrêter de glander de temps en temps...).

4/ Relevé des messages d'erreurs

On note d'abord les messages d'erreurs :
On lance Rescue95 on mais un code au hasard, et là un message nous informe :
"WARNING - Incorrect Key Entered".

Sachant que le logiciel est limité dans le temps on change la date en 2002 et on relance le programme.
Ainsi, lorsque l'on veut accéder à certaines options un message nous dit *"Trial Period Expired"*.

De plus certaines options sont en temps normal bridées cela nous donne un message du genre :
"RegPack Disabled. Please register."

5/ Désassemblage et craquage

Voici la liste des actions que l'on va exécuter avant de commencer à craquer le logiciel :

-> Faire une copie de Rescue95.exe et renommez la 1.exe.
-> Lancez Wdasm32 et désassemblez Rescue95.exe dans w32dasm 8.93 .

On constate qu'il n'y a pas de réference au menu (Menu Ref) ni aux boîtes de dialogues (DLG Ref). Par contre on constate qu'il

y a quelque chose en "String Refs". On recherche donc dans la liste des "string data references" s'il n'y a rien qui pourrait nous intéresser. C'est cool ! On trouve tous les messages d'erreurs !! On pourrait s'amuser à enlever toutes les protections une après l'autre si on le voulait ... Mais autant essayer de s'enregistrer directement cela a le même effet.

-> On clique donc dans la liste des "string data references" sur *"WARNING - Incorrect Key Entered"*.
Et cela nous donne quelque chose comme cela :

---------- Listing de désassemblage de Rescue95.exe

```
 * Possible StringData Ref from Code Obj -> "msR3I8aUi9y2E84L"  (=> msR3I8aUi9y2E84L c'est le bon
code)
 :004681BC    B89824600               mov eax, 00468290
 :004681C1    E8EEBDF9FF              call 00403FB4
 :004681C6    85C0                    test eax, eax
 :004681C8    7420                    je 004681EA  => si mauvaise réponse, aller à 004681EA

 * Possible StringData Ref from Code Obj -> "Registration Key Accepted"
 :004681CA    B8AC824600              mov eax, 004682AC
 :004681CF    E8104D4DFF              call 0043CEE4
 :004681D4    A1B0EA4700              mov eax, dword ptr [0047EAB0]
 :004681D9    8B00                    mov eax, dword ptr [eax]
 :004681DB    8B80F0010000            mov eax, dword ptr [eax+000001F0]
 :004681E1    C7400C64000000          mov eax, dword ptr [eax+0C], 00000064
 :004681E8    EB16                    jmp 00468200

 * Referenced by a (U)nconditional or (C)onditional Jump at Address :
 :004681C8 (C)  (=>le (C) veut dire que c'est un saut conditionnel en 004681C8 qui nous a amené ici)

 * Possible StringData Ref from Code Obj -> "WARNING - Incorrect Key Entered"
```

-> Arrivé à *"Registration Key Accepted"* il faut enlever le "je " qui fait effectuer un saut jusqu'à *"Warning Incorrect Key Entered"*.

Pour cela on peut tout simplement enlever la ligne 004681C8 et remplacer 7420 par 9090
Rappel : 90 = nop = No Operation

Donc maintenant, quand le programme arrive à la ligne 004681C6 même si le code est mauvais il va directement sur "Registration Key Accepted " puisqu'il n'y a plus de saut conditionnel !

-> On peut éditer maintenant 1.exe. On va à l'offset 675C8 (car cela correspond à la ligne 004681C8) et on remplace 7420 par 9090.

-> On relance 1.exe, on entre n'importe quel code et **BRAVO , RESCUE95 EST CRACKE !!!**

Winrescue98 a exactement le même schéma de protection, sauf que le serial n'est pas le même. Pour conclure on peut simplement dire que ce logiciel est vraiment super simple à cracker, et que la protection est vraiment simpliste puisqu'il n'y a même pas une vérification du code ...

[interlude de Smeita...]
Bon, la c'est vraiment le cas le plus simple qu'il existe !! Ya pas plus facile !!
Disons que c'etait juste pour vous rememorer le principe de base :)

Le prochain cours de cracking s'occupe de Unreal, le celebre Doom-Like ! Vous verrez que c'est aussi facile que WinRescue :)

Notez que tout ces cours de cracking à partir de celui-ci ne sont pas la pour vous apprendre de nouveau truc, mais simplement vous montrer leur mise en pratique.... Inutile donc de vous etonnez et de trouver ca trop simple :) On le sait que c'est hyper simple !! Le plus dur reste a venir, et si on insiste sur les bases, c'est pour etre sur que dans le MemenTo 3 on puisse accelerer un peu plus.... Allez, si vous voulez un truc un tout petit peu plus "dur", (ou tout petit peu moins facile...) allez directement au cours de cracking n° 12. il s'agit de mIRC 5.5...
[...Fin d'interlude...]

6/ Supplément de pifoman

AU 04/01/2005 sur le web je n'ai pas trouvé la version du proramme correspondant au cours de smeita. Je n'ai trouvé que la version 10.08.25 du 30 juillet 2004. C'est celle que j'ai mis en ligne dans la rubrique de téléchargement. Je me suis donc penché sur ce cours en détail pour voir si le mécanisme de protection proposé par smeita était toujours d'actualité sur cette nouvelle version. C'est le cas avec une petite nuance dans le code.

---------- Listing de désassemblage de Rescue95.exe (version 10.08.25) ----------

```
:004C3A84    E8A317F4FF              call 0040522C
:004C3A89    85C0                    test eax, eax
:004C3A8B    742D                    je 004C3ABA     => si mauvaise réponse, aller à 004681EA
:004C3A8D    33D2                    xor edx, edx
:004C3A8F    8B8304030000            mov eax, dword ptr [ebx+00000304]
:004C3A95    E8CAB4F8FF              call 0044EF64

* Possible StringData Ref from Code Obj ->"Registration Key Accepted"
:004C3A9A    B82C3C4C00              mov eax, 004C3C2C
:004C3A9F    E86CEDF7FF              call 00442810
:004C3AA4    A1788B5200              mov eax, dword ptr [00528B78]
:004C3AA9    8B00                    mov eax, dword ptr [eax]
:004C3AAB    8B8020030000            mov eax, dword ptr [eax+00000320]
:004C3AB1    C7400C64000000          mov [eax+0C], 00000064
:004C3AB8    EB16                    jmp 004C3AD0

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:004C3A8B(C)

* Possible StringData Ref from Code Obj ->"WARNING - Incorrect Key Entered"
:004C3ABA    B86C3C4C00              mov eax, 004C3C6C
```

-> Au dessus du message *"Registration Key Accepted"* on voit un je 004C3ABA .qui saute vers l'adresse 004C3ABA. Il faut donc annuler ce saut pour que le programme continue vers "Registration Key Accepted".

Pour cela on peut tout simplement remplacer à l'adresse 004C3A8B le code hexadécimal 742D par 7400. Cela revient à remplacer l'instruction assembleur je 004C3ABA par je 004C3A8D.Ici au lieu de nopper comme smeita l'avait fait dans l'exemple précédent on annule simplement le saut en sautant juste après l'instruction qui suit l'adresse 004C3A8B.

Donc maintenant, quand le programme arrive à la ligne 004C3A8B même si le code est mauvais il continue vers "Registration Key Accepted" puisqu'il n'y a plus de saut conditionnel !

-> On peut éditer maintenant 1.exe dans winhex.exe.On va à l'offset C2E8B (regardez la barre de statut de w32dasm quand vous sélectionnez la ligne d'adresse 004C3A8B là où il y a marqué @offset). Pour cela dans winhex.exe on fait ALT G -> C2E8B et on remplace 742D par 7400.

-> On relance le 1.exe, on entre n'importe quel code par exmple pifoman/123456 et **BRAVO , RESCUE95 EST CRACKE !!!**

Nombre de visites depuis le 15/02/2003