

Tutorial #11: CrackMe en Visual Basic (VB)

Outils:

- Le [CrackMe](#) de 38 kb
- Un désassembleur VB: [SmartCheck 6.2](#)
- PEiD 0.94 pour l'analyse
- Patcher pour faire le patch

Observation du CrackMe:

- Nag Screen au démarrage
- Bad Smiley et "UNREGISTERED"
- Demande d'un **serial** valide
- Badboy apparait dans une boite de dialogue

Objectifs:

- Patcher pour forcer enregistrement
- Trouver un serial Valide pour son nom
- Faire un patch

J'ai choisi ce programme car il est écrit en Visual Basic, mais aussi parce qu'il a déjà été cracké.

On commence par ouvrir le programme (ce n'est pas vraiment un CrackMe mais un utilitaire transformé en CrackMe) dans PEiD, et on voit que celui-ci est écrit en **Microsoft Visual Basic 5.0 / 6.0**. Il n'est donc pas compressé mais écrit dans un langage différent. Cela va nous compliquer un peu les choses mais vous verrez, ce n'est pas si difficile que ça en a l'air. En effet, on a à notre disposition un outils très pratique: il s'agit de *SmartCheck*, de la même société de *SoftIce* (Numega), conçu spécialement pour debugger des programmes en VB.

1 - Patcher le programme pour forcer l'enregistrement

On ouvre le prog dans Olly après avoir observe son comportement. On sait que le nag de démarrage n'apparait plus lorsqu'on s'enregistre. Il va donc falloir trouver un moyen de patcher l'enregistrement.

Dans les string references, on faire une recherche de texte sur le message d'erreur. Par conséquent, Click droit > Search for > all reference text strings. On en voit beaucoup et apparemment rien de bien intéressant!

Pour ça, on remonte, et on se positionne sur la première ligne de la fenêtre des text strings, et on clique droit > Search for Text. On décoche *Case Sensitive* (prend en compte les majuscules et minuscules), et on coche *Entire Scope* (regarde dans l'intégralité des strings).

Le message d'erreur nous dit "Sorry, better luck next time". On va donc taper "sorry" et voir ce que ca donne. On fait OK et la on tombe sur un truc qui ne nous intéresse pas trop. On fait CTRL + L pour passer a la référence suivante de ce qu'on a tape. AAAH! la voila notre SDR! on double-clique dessus pour se retrouver ici:

```
0040A101 > \8D95 54FFFFFF LEA EDX,DWORD PTR SS:[EBP-AC]
0040A107 . 8D4D 94 LEA ECX,DWORD PTR SS:[EBP-6C]
0040A10A . C785 5CFFFFFF EC42400>MOV DWORD PTR SS:[EBP-A4],CConvert.004042EC ; UNICODE
"Sorry, Better Luck Next Time!"
0040A114 . C785 54FFFFFF 08000000>MOV DWORD PTR SS:[EBP-AC],8
0040A11E . FF15 94E24000 CALL DWORD PTR DS:[<&MSVBVM50.__vbaVarDup>] ;
MSVBVM50.__vbaVarDup
0040A124 . 8D85 64FFFFFF LEA EAX,DWORD PTR SS:[EBP-9C]
0040A12A . 8D8D 74FFFFFF LEA ECX,DWORD PTR SS:[EBP-8C]
0040A130 . 50 PUSH EAX
0040A131 . 8D55 84 LEA EDX,DWORD PTR SS:[EBP-7C]
0040A134 . 51 PUSH ECX
0040A135 . 52 PUSH EDX
0040A136 . 8D45 94 LEA EAX,DWORD PTR SS:[EBP-6C]
0040A139 . 6A 00 PUSH 0
0040A13B . 50 PUSH EAX
0040A13C . FF15 D0E14000 CALL DWORD PTR DS:[<&MSVBVM50.#595>] ; MSVBVM50.rtcMsgBox
```

Pour voir comment on est arrivés ici, il suffit de remonter le code et on voit dans Olly que c'est par un saut a l'adresse 409FBE que l'on est arrivés ici.

```
00409F8E . FFD3 CALL EBX ; <&MSVBVM50.__vbaFreeVarList>
00409F90 . 83C4 0C ADD ESP,0C
00409F93 . BF 04000280 MOV EDI,80020004
00409F98 . 66:85F6 TEST SI,SI
00409F9B . BE 0A000000 MOV ESI,0A
00409FA0 . 89BD 6CFFFFFF MOV DWORD PTR SS:[EBP-94],EDI
00409FA6 . 89B5 64FFFFFF MOV DWORD PTR SS:[EBP-9C],ESI
00409FAC . 89BD 7CFFFFFF MOV DWORD PTR SS:[EBP-84],EDI
00409FB2 . 89B5 74FFFFFF MOV DWORD PTR SS:[EBP-8C],ESI
00409FB8 . 897D 8C MOV DWORD PTR SS:[EBP-74],EDI
00409FBB . 8975 84 MOV DWORD PTR SS:[EBP-7C],ESI
00409FBE . 0F84 3D010000 JE CConvert.0040A101 <-- Saute vers BadBoy si égal: on va patcher ici.
00409FC4 . 8D95 54FFFFFF LEA EDX,DWORD PTR SS:[EBP-AC]
00409FCA . 8D4D 94 LEA ECX,DWORD PTR SS:[EBP-6C]
00409FCD . C785 5CFFFFFF 6042400>MOV DWORD PTR SS:[EBP-A4],CConvert.00404260 ; UNICODE
"Nice Work! You're Now Registered!"
00409FD7 . C785 54FFFFFF 08000000>MOV DWORD PTR SS:[EBP-AC],8
00409FE1 . FF15 94E24000 CALL DWORD PTR DS:[<&MSVBVM50.__vbaVarDup>] ;
MSVBVM50.__vbaVarDup
```

Lorsqu'on s'y rend, on aperçoit le GoodBoy en 409FCD! On sait donc déjà ou l'on ne veut pas tomber et la ou on veut aller! On change donc dans Olly le JE en JNE (dans ce cas on ne sautera vers le BadBoy que si le code entre est le bon... donc il y a peu de chances!). Double-cliquez sur la ligne du JE et changez-le manuellement.

On enregistre les changements en renommant le prog sous un autre nom, et le tour est joué: on va pouvoir s'enregistrer avec n'importe quel code et n'importe quel nom!

A savoir, il y a plusieurs façons de cracker ce programme: en effet les infos de registration sont contenues dans les deux fichiers cconvert.ccc et cconvert.\$\$\$\$. On aurait donc pu travailler sur les keyfiles par exemple.

2 - Faire un patch pour le programme

On va maintenant apprendre a faire un patch pour le prog de façon très simple. Ce patch va comme au dessus nous permettre de nous enregistrer avec n'importe quel serial.

Pour ça, téléchargez par exemple **PE Patch Engine**. Ouvrez-le et observez un peu.

Pour faire le patch:

- Dans Original file, ouvrez le prog original.
- Dans Patched file, ouvrez la copie que l'on vient de sauvegarder (celle qui est patchée).

- Si vous le désirez, vous pouvez mettre l'icône du prog a patcher en la pompant grâce a [ResHacker](#) (ce que j'ai fait, en ajoutant pour le plaisir un patch graphique avec mon nom dans la barre principale: pour ceux qui veulent savoir comment faire, contactez moi sur le forum).

Cliquez sur "Generate", et votre patch est fait!

3 - Trouver un serial valide pour son nom (SmartCheck)

Préambule: les fonctions VB sont différentes des API Windows habituelles. Voici un aperçu des fonctions les plus utilisées:

- `__vbavartsteq` <-- string equivalence
- `__vbastrcmp` <-- string comparison
- `__vbastrcomp.` <-- string compare

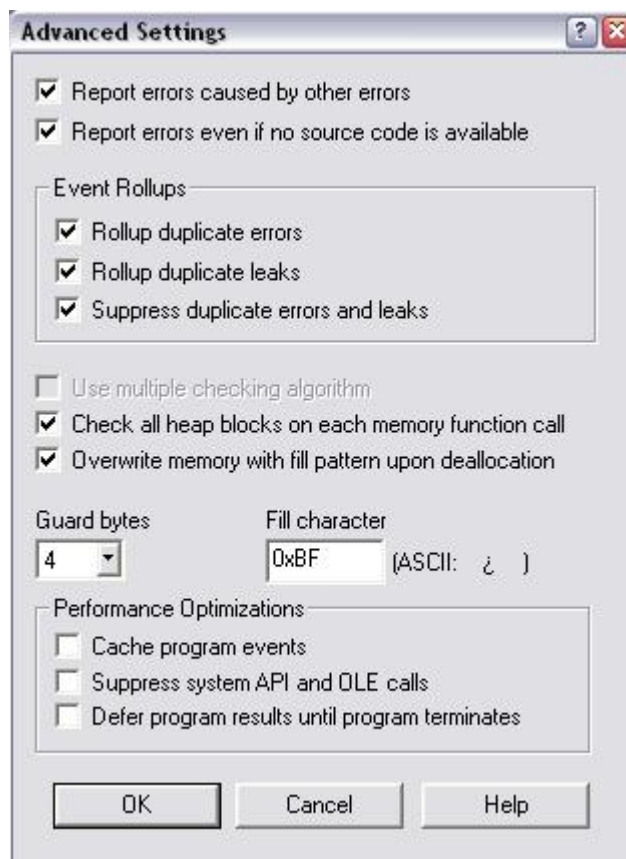
Ce sont ces dernières qu'il va falloir chercher dans 90 à 95% des cas.

Il va nous falloir configurer SmartCheck une fois qu'on l'a installé car la configuration par défaut ne nous permet pas de faire grand chose. Suivez les instructions :

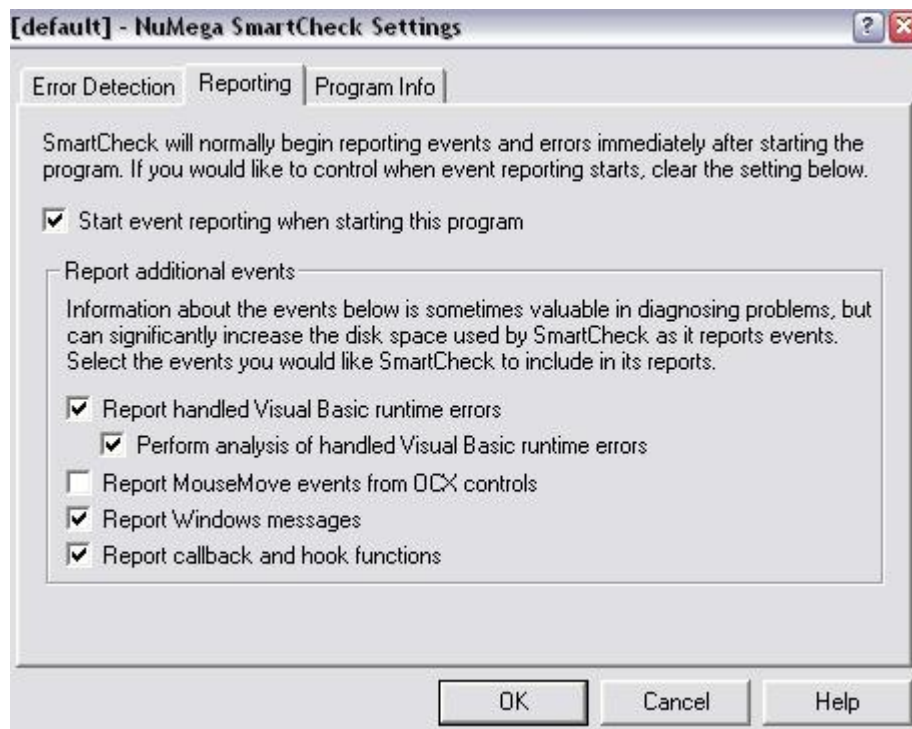
Ouvrez SmartCheck, ouvrez cconvert.exe dedans puis avant de toucher quoi que ce soit, cliquez sur *Program > Settings*.



Dans l'onglet *Advanced* cochez les cases suivantes.



Dans l'onglet *Reporting*, cochez les suivantes.



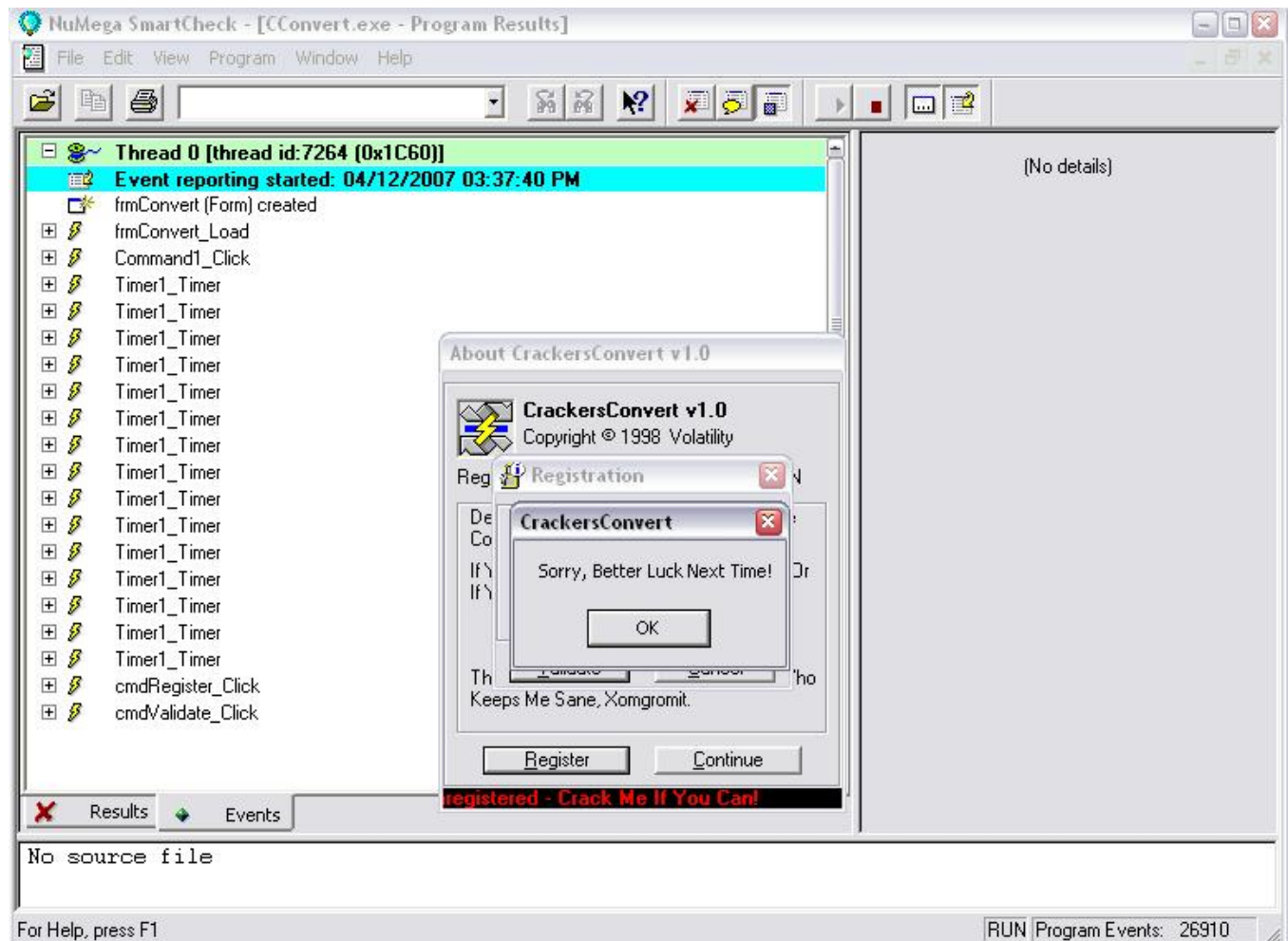
Surtout faites bien attention de **ne pas cocher la case** "Report MouseMove events from OCX controls".

SmartCheck est maintenant configuré. *Vous trouverez des tutoriaux sur SmartCheck en cherchant bien en ligne, je ne rentrerai pas ici dans les détails et les explications des fenêtres du programme. Voyez par exemple le tutoriel de [Falcon](#) qui se trouve toujours sur son site.*

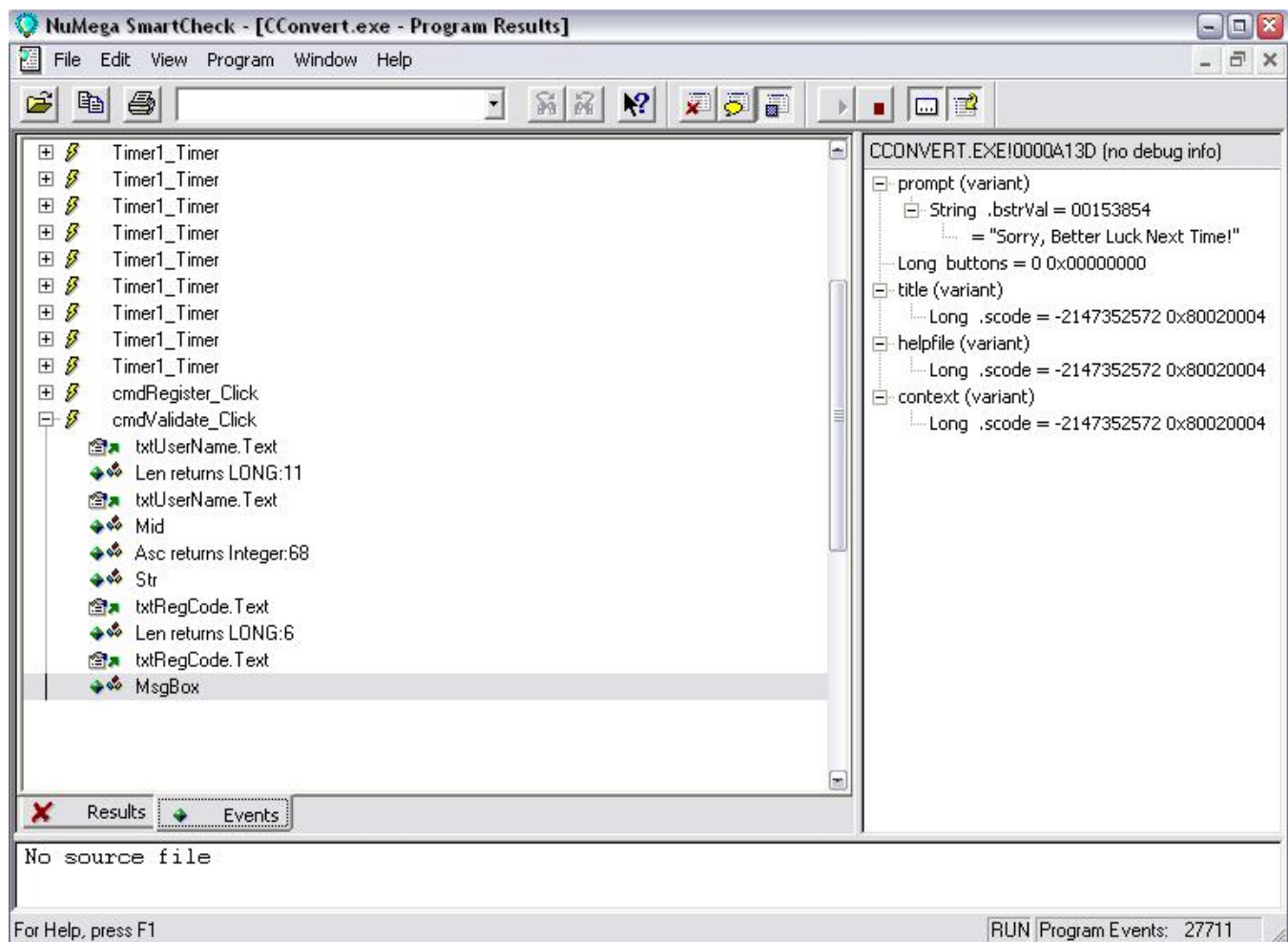
Je vais partir du principe que vous avez déjà observé le comportement du prog avant de commencer cette partie.


A présent, lancez le debugging du prog en cliquant sur le bouton flèche du haut de la barre de menu. La, le prog se lance et on aperçoit le nag. Positionnez-vous sur l'onglet *Events* en bas a gauche (selon la config de SmartCheck, on a un comportement différent. Ne vous en faites pas s'il se comporte de manière différente chez vous, le résultat est le même!). Il se peut que vous y soyez déjà.

Cliquez sur OK dans le nag, puis allez dans la section About du programme. Notez qu'au fur et a mesure, des fonctions s'affichent dans la fenêtre principale de SmartCheck. Rentrez des informations bogus, pour moi **DeezDynasty** et **123456**.



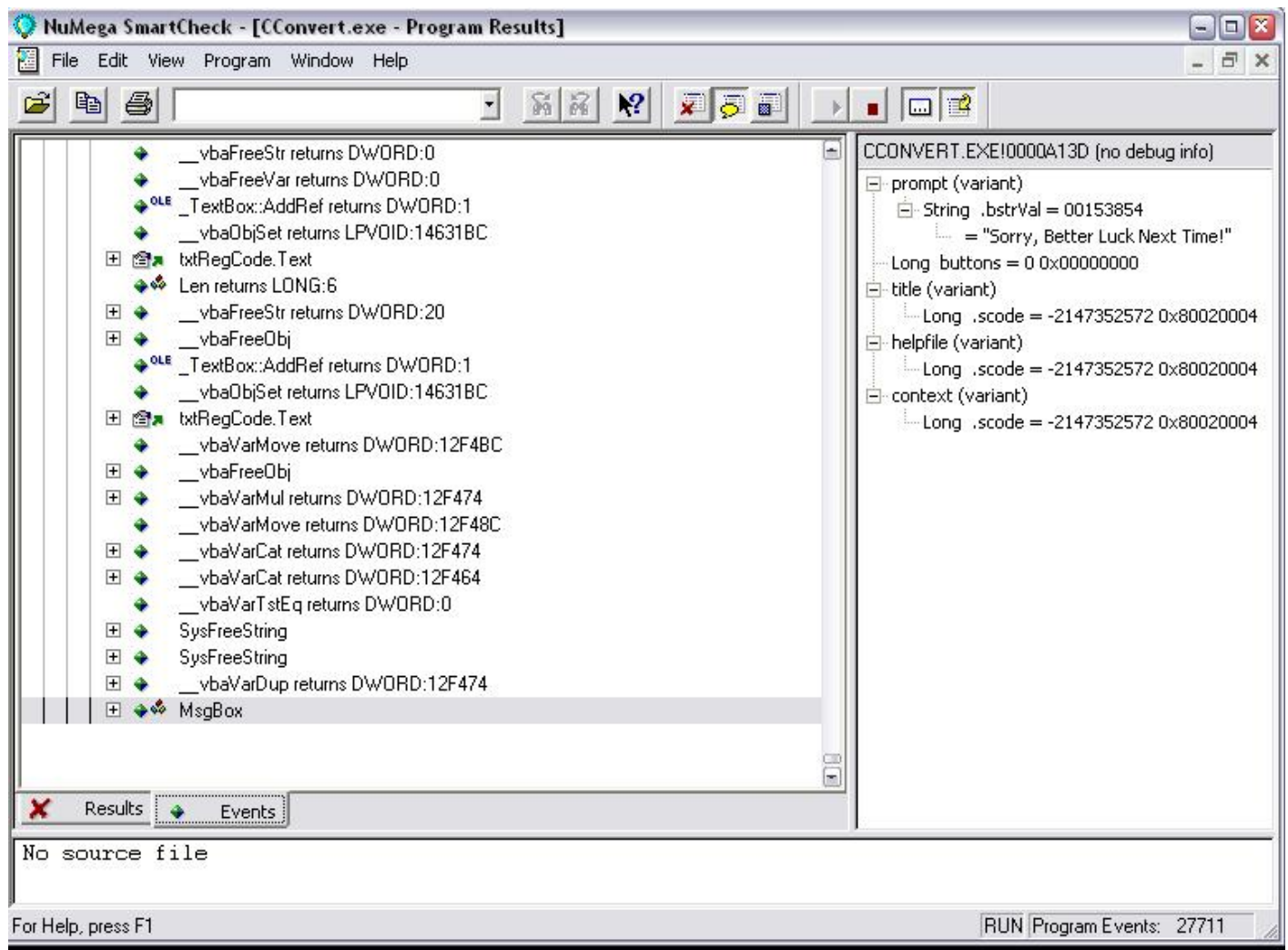
Revenez dans SmartCheck en cliquant dessus. Étendez l'arborescence de la dernière commande apparue, **cmdValidate_Click**. Cette commande sert a repérer les clics que l'ont fait dans Cconvert.



On va se positionner sur la ligne MsgBox, qui correspond a l'affichage de la boite de message. Maintenant on va cliquer sur **Show All Events** ----> .

Attention, il est important avant de cliquer sur Show All Events de se positionner a un endroit stratégique, pour avoir une vue la plus claire possible des événements, et pour s'éviter de chercher pendant trop longtemps. Le serial fishing avec SmartCheck consiste beaucoup en l'observation des fonctions.

On se retrouve ici:



Ici, l'observation n'est pas très longue... on voit tout de suite la fonction **__vbaVarTstEq** juste au-dessus de notre ligne. En se positionnant dessus, on va pouvoir visualiser notre serial: en effet, c'est la que la string du vrai serial est comparée à la string du faux que nous avons rentré. Dans mon cas, le serial est **REG-1360-CODE**.

Après observation des fonctions dans SmartCheck, on peut voir que le serial est toujours de la forme **REG-quatre chiffres-CODE**. On peut faire un keygen après plus d'observation, pour savoir comment sont calculés nos 4 chiffres du milieu. Ceci dit je ne rentrerai pas ici dans les détails.

Voilà, félicitations!

Ceci étant mon premier tutoriel sur le VB (et le dernier!), il se peut que quelques éléments ne soient pas clairs. Dans ce cas n'hésitez pas à me faire part de vos hésitations et incompréhensions sur le forum.