The Italian CRaCKiNG Encyclopedia vol. 5

CRaCKiNG VB5 Programs

Scritto da

---==: ALoR :==---
---==> Proud member of NEURO ZONE 2 <==---

Gennaio 1999

Target = **

(*=Lamer, **=Novizio, ***=Apprendista, ****=Esperto, *****=CrackMaster)

==========

Table of context :

==========

1.0 DISCLAIMER

Every reference to facts, things or persons (virtual, real or esoteric) are purely casual and involuntary. Any trademark nominated here is registered or copyright of their respective owners. The author of this manual does not assume any responsability on the content or the use of informations retriven from here; he makes no guarantee of correctness, accuracy, reliability, safety or performance. This manual is provided "AS IS" without warranty of any kind. You alone are fully responsible for determining if this page is safe for use in your environment and for everything you are doing with it!

Solita palla lo so....

2.0 I TOOLS NECESSARI

Per crackare un programma scritto in VB5 sono necessari:

* SmartCheck 5.0 or higher

* Soft-Ice 3.2 or higher (opzionale)

tutto qui...
Vedremo in seguito come i maghi della Numega siano riusciti a fare un prg. che rende la vita di noi crackers assai facile.

3.0 IL LINGUAGGIO VB

Per cominciare diamo una rapida occhiata a come e' strutturato un VB5 prg.
Visual Basic produce eseguibili interpretati e non veri e propri compilati come as esempio Visual C++.
Un linguaggio interpretato quindi non puo' essere distribuito senza le apposite

DLL di runtime (circa 3 Mb per la ver 5) che servono appunto ad interpretare
le richieste del programma e a passarle al sistema operativo.
Il vero lavoro e' dunque compiuto dalle DLL di runtime e non dal prg.
Infatti decompilando un prg fatto in VB non si cava un ragno dal buco !!!
Ci ritroveremo di fronte a migliaia di chiamate a DLL che non ci dicono nulla
di interessante (comparazione d stringhe... ecc...) su quello che avviene all
interno del programma.
Le DLL principalmente interessate sono:

    vbrun300.dll - VB3 (16-bit)
    vb40016.dll - VB4 (16-bit)
    vb40032.dll - VB4 (32-bit)
    msvbvm50.dll - VB5 (32-bit)

Quindi se volessimo usare Soft-ice (ma perche' ??? ora c'e' SmartCheck !!) con
un prg scritto in VB dovremo aggiungere al file winice.dat le dll sopracitate
come export per i simboli. Cosi' potremmo piazzare breakpoint sulle funzioni
di Visual Basic.

 4.0 STRINGHE IN VB5

Piazzare un breakpoint su Hmemcpy in un VB5prg e' da suicidio !!!
Istantaneamente ci ritroveremo a navigare in quell'inferno che il nostro amico
Bill Gate$ ha soprannominato simpaticamente: msvbvm50.dll. Perdendo cosi' di
vista il nostro obbiettivo: il confronto tra stringhe.
Dobbiamo poi tenere conto che VB5 salva le stringhe in formato Wide, cioe' con
uno spazio (0x20) tra una lettera e l'altra.

Stringa ordinaria: ALOR (0x41 0x4C 0x4F 0x52).
Wide Character Format: A L O R (0x41 0x20 0x4C 0x20 0x4F 0x20 0x52).

In questo caso puo' essere utile il breakpoint su MultiByteToWideChar.
Mettendo questo BP in Soft-ice ci ritroveremo nel solito msvbvm50.dll e nei
registri EAX e EBX possiamo notare la lunghezza della stringa inserita. Se non
e' quella corretta provare con Ctrl+D... fino alla lunghezza corretta.
A questo con un po' di fortuna ci ritroveremo nel data segment la nostra
stringa e li intorno da qualche parte quella giusta.

 5.0 PRE-SMARTCHECK ERA

Prima della venuta di nel nostro salvatore... (smartcheck) si viveva in un
epoca oscura... dove i programmi VB erano come fumo negli occhi per i "reverse
engineers"...
ok. basta stronzate...
Come accennato prima i programmi VB sono interpretati e quindi il loro lavoro
e' svolto dalle DLL di runtime. Quindi e' inutile arrovellarsi il cervello per
cercare di trovare la procedura di check all'interno del programma.
Cerchiamola direttamente nelle DLL !! E siccome non credo che quelli che hanno
codato VB si siano divertiti a fare 150 funzioni diverse per comparare le
stringhe... non sara' poi cosi' difficile trovarla, visto che tutti i prg
usano la STESSA funzione.
Stessa funzione vuol dire stesso indirizzo (o al massimo due) per ogni prg.
Quindi crackare questo tipo di prg vorra' dire piazzare il solito breakpoint
nel solito posto OGNI volta.

*** VB4 checking (WideChar) routine ***

```
* Referenced by a CALL at Address:
|:0F73CD83
|
:0F79B348 56          push esi
:0F79B349 57          push edi
:0F79B34A 8B7C2410    mov edi, dword ptr [esp+10]
:0F79B34E 8B74240C    mov esi, dword ptr [esp+0C]
:0F79B352 8B4C2414    mov ecx, dword ptr [esp+14]
:0F79B356 33C0        xor eax, eax
:0F79B358 F3          repz                        <--- qui compara le
:0F79B359 66A7        cmpsw                       <--- stringhe WideChar
:0F79B35B 7405        je 0F79B362
:0F79B35D 1BC0        sbb eax, eax
:0F79B35F 83D8FF      sbb eax, FFFFFFFF
```

```
* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0F79B35B(C)
|
:0F79B362 5F          pop edi
:0F79B363 5E          pop esi
:0F79B364 C20C00      ret 000C


*** VB5 checking (WideChar) routine ***

* Referenced by a CALL at Address:
|:0F0035A6
|
:0F00D9EA 56         push esi
:0F00D9EB 57         push edi
:0F00D9EC 8B7C2410   mov edi, dword ptr [esp+10]
:0F00D9F0 8B74240C   mov esi, dword ptr [esp+0C]
:0F00D9F4 8B4C2414   mov ecx, dword ptr [esp+14]
:0F00D9F8 33C0       xor eax, eax
:0F00D9FA F3         repz                        <--- qui compara le
:0F00D9FB 66A7       cmpsw                       <--- stringhe WideChar
:0F00D9FD 7405       je 0F00DA04
:0F00D9FF 1BC0       sbb eax, eax
:0F00DA01 83D8FF     sbb eax, FFFFFFFF

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0F00D9FD(C)
|
:0F00DA04 5F         pop edi
:0F00DA05 5E         pop esi
:0F00DA06 C20C00     ret 000C
```

Ma guarda un po'... sembrano uguali... anzi SONO UGUALI !!!
Quindi non solo una procedura per tutti i VB5 ma anche per i VB4 !!!

Per ulteriori informazioni vi rimando alla guida "How to crack all visual basic programs" di razzia della +HCU.

6.0 ORA C'E' SMARTCHECK

Trovare la giusta stringa non sembrava tanto difficile... fino a quando e' arrivato SmartCheck. Ora e' addirittura una cazzata !!!
Come dicono fravia+ e +HCU "Like watching a movie!" :)
Analizziamo allora il nostro nuovo "strumento" di reverse enjeneering.

SmartCheck e' un debugger run-time per Visual Basic che permette di avere info su errori del programma, parametri passati alle funzioni e un dettagliato elenco degli eventi del programma.
Siccome quelli della Numega non hanno confezionato un programma esplicitamente per i cracker, innanzitutto dovremo settarlo a dovere.
Quindi una volta caricato l'eseguibile, andiamo sul menu Program | Settings... checkare tutte la caselle in "Reporting" tranne quella del MouseMove, in "Error Detecrion" | Advanced, report error even if... e NON suppress system...
Salvate queste impostazioni come di default e chiudete la finestra di dialogo.
And now watch at the movie!!! :)))

Eseguendo un programma con SmartCheck potremo notare sulla parte sinistra tutti (e dico TUTTI) gli eventi del programma. Dal form.load ai text.keypress ai label.caption alle funzioni di formattazione (lenght(), trim(), hex(), ecc) alle funzioni di calcolo, insomma quando dico tutto... intendo anche il numero di scarpe del programmatore... :)))
Sulla parte destra della schermata si possono invece controllare i valori dei parametri passati alle funzioni e quelli ritornati.
Per di piu' sempre nello stesso pannello ci dice anche l'indirizzo in cui e' chiamata la funzione in esame.

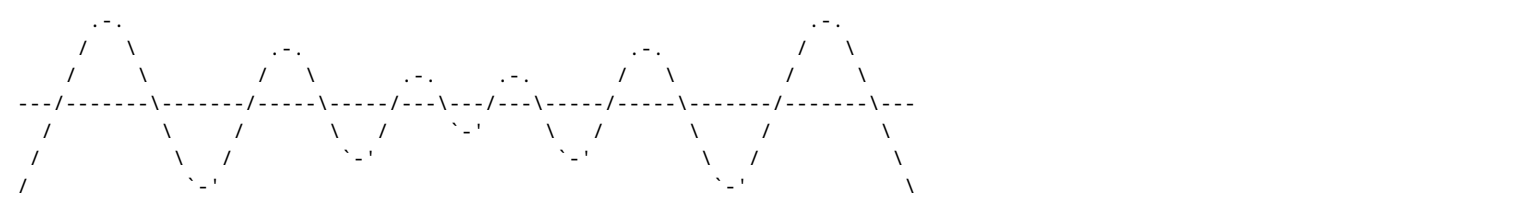
Cosa possiamo chiedere di piu' ?? Sappiamo dove e cosa sta facendo il nostro prg da "analizzare"... Io direi che in questa maniera il cracking dei VB5 prg e' davvero fin troppo facile e senza quell'ingrediente che spinge un cracker a portare a termine il suo lavoro: LA SFIDA !!

Il primo prg che ho crackato con SmartCheck e' stato cosi gentile da dirmi lui il codice giusto.... e' bastato guardare i valori passati dall'elemento cod.text (che fantasia...) e cioe' il mio codice e quello giusto... e il gioco era fatto.... :))

7.0 CONCLUSIONI

Il prossimo numero probabilmente trattera' le istruzioni assembler. Tanto ne abbiamo parlato in questi volumi, a volte dandole per scontate, ma non le abbiamo mai analizzate in profondita'. Da qui nascera' il sesto volume.
Se avete suggerimenti per eventuali vol successivi, contattatemi pure via mail all'indirizzo: ALoR@thepentagon.com

"If you give a man a crack he'll be hungry again
tomorrow, but if you teach him how to crack, he'll
never be hungry again"
+ORC

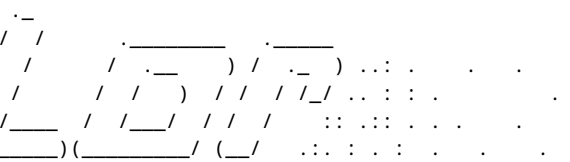


Greetings to: LordKasKo (my cracking teacher...)

The other NEURO ZONE 2 members (10t8or, DK2DEnd, LordKasKo, MaPHas, Ob1, XXXX, ZenGa)

All the Cracker on the NET from whom I have learned something.

==========



--==> ALoR <==---------- - - -
ICQ: 10666678 In IRC: WhiteFly

e-mail: ALoR@thepentagon.com www: http://ALoR.home.ml.org
http://ALoR.cjb.net

==========