# COMO CRACKEAR UN PROGRAMA PROTEGIDO CON ARMADILLO

## (LECCIÓN ONCE)

## ¿Qué cuernos es ARMADILLO?

Es una protección comercial que ya lleva varias versiones, la del programa que nos ocupa es la versión 1.83.

COMO NOS DAMOS CUENTA QUE ENTRE TANTAS PROTECCIONES QUE HAY ES ARMADILLO LA QUE USA EL PROGRAMA VICTIMA?

Bueno como primer paso, para ver algo con el SOFTICE tendremos que sortear las protecciones ANTI-SOFTICE que tiene el armadillo, estas son tres tipos de protección, se pueden desbloquear a mano, pero es muy laborioso y si tenemos la ultima versión del FROGS-ICE pasara todas las protecciones perfectamente y sin que aparezcan pantallas azules ni nada, como si no estuviera el SOFTICE.
Les recomiendo abiertamente bajar la última versión del FROGSICE pues ha mejorado muchísimo e inclusive se puede utilizar el SYMBOL LOADER que en versiones anteriores no se podía.
Pues entonces cargo el FROGSICE 1.085b que hasta hoy creo que es la ultima versión que hay y como por default aparece desactivado entro al menú y pongo ENABLE SOFTICE y listo.
Ahora ejecuto el programa RELAX 1.1 que esta protegido con ARMADILLO 1.83 y arranca perfectamente sin enterarse de que el SOFTICE esta allí.
BUENA POR EL FROGSICE.
Ahora aparece una ventana de que el programa va a ser utilizado por siete días y después vencerá y una opción REGISTER donde entro y veo que me pide un numero de serie y una clave.
Pongo narvaja y 989898 como siempre y me aparece una messagebox que dice que no que no que no.
Bah que esta mal la clave.
Voy al SOFTICE y allí pongo un BPX MESSAGEBOXA y vuelvo a al ventana y pongo OK y PUFFF dentro del SOFTICE.
Ahora, si nos habíamos tomado la molestia de mirar el archivo RELAX con el Wdasm vemos que aparecen STRING REFERENCES y ninguna como la que nos dice que la clave no es buena, lo mismo que no hay STRINGS que digan nada de VENCIDO o algo de eso.
Hmmm, pensemos
Volvamos al softice después de haber caído allí por el messageboxa y aprieto F12 para que aparezca el mensaje de error y hago clic en ACEPTAR y de nuevo en el SOFICE, lo primero que vemos es que el nombre el archivo es ARM... lo cual nos da la idea de que aquí hay algo raro ya que el archivo RELAX es el único que se puede ejecutar en el directorio del programa y no hay otro por allí, y sin embargo aquí se realiza la verificación de la clave en un archivo ARM .... que encima si busco un poco mas arriba puedo ver un CALL, un TEST AL,AL y un salto que pasa por encima del messageboxa, y si invierto el salto me dice que La clave es correcta que la va a GUARDAR, hmmm, y cuando arranca de nuevo el programa esta de nuevo desregistrado.

Bueno, donde esta el BENDITO archivo ese ARM..., voy a INICIO-BUSCAR y allí pongo arm* y entre otras cosas me aparece un ARM seguido de un numero .TMP por ejemplo ARM1025.TMP.
Bueno entonces el protector crea un archivo temporal donde ocurre la registración, entonces si quitamos la protección del ARMADILLO, que pasara con el programa RELAX?
CHA CHAN CHA CHAN…
Les digo que hay una forma de quitar el armadillo MANUAMENTE pero al existir un programa que lo hace perfectamente y funciona bien, para que nos vamos a matar.
Ese programa es el ARMADILLO KILLER 1, y se consigue en las mejores casa del ramo, hasta la versión 1.83 del armadillo la hace PURE.
Lo abro y me pide que ponga el ejecutable protegido con ARMADILLO, lo busco al RELAX y lo abro, espero un poquito y me dice donde lo quiero guardar, lo guarda.
Ahora lo busco y lo ejecuto, SORPRESA, no aparece mas la ventana que dice que es una versión de prueba ni la pantalla para registrarse ni nada, ni tampoco la protección antisoftice , ni la de que el programa te retaba cuando adelantabas el reloj, nada quedo el programa pelado y funcionando y parece que bien.
Adelanto un mes el reloj y sigue funcionando, todo parece estar bien, y el programa crackeado, por ahora cumplimos con nuestro objetivo, si al pasar el tiempo surge alguna novedad, (LO QUE NO CREO PUES TODA LA PROTECCIÓN ESTABA EN EL ARMADILLO) seguiremos con el tema, por ahora.

PROGRAMA CRACKEADO.

**Ricardo Narvaja**

# UNA NUEVA HERRAMIENTA QUE NSO PUEDE AHORRAR TIEMPO Y ESFUERZO EN MUCHAS TAREAS AUXILIARES DEL CRACKER

## (LECCION DOCE)

## PUPE

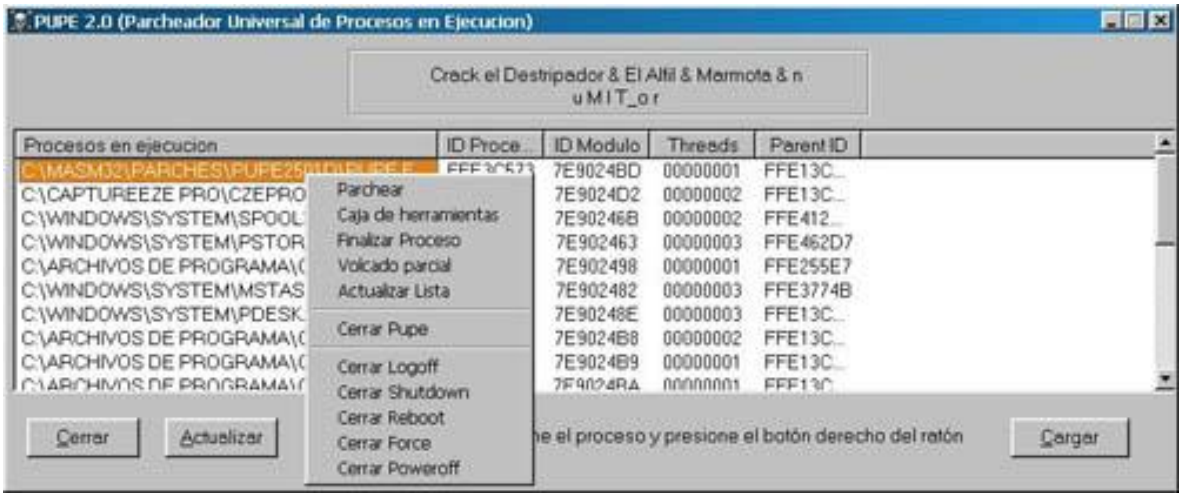
Como ya avise en mails anteriores vamos a comenzar a estudiar esta excelente herramienta auxiliar llamada PUPE que fue escrita por crackers a diferencia de muchas de las otras herramientas que podemos encontrar, además está totalmente en CASTELLANO ya que fue hecha por crackers Hispanos, se puede bajar de la pagina de uno de sus autores CRACK EL DESTRIPADOR, en.
http://teleline.terra.es/personal/guillet/

Sección Herramientas.
Esta herramienta en sus primeras versiones fue un parcheador de la memoria pero le fueron agregando mas utilidades y prometen agregarle mas todavía con lo que esta herramienta puede llegar a convertirse en una gran ayuda para nosotros, voy a tratar de describir un poco algunas utilidades que pueden llegar a ser muy importantes y que nos pueden ahorrar tiempo y esfuerzo en nuestras tareas de crackear.

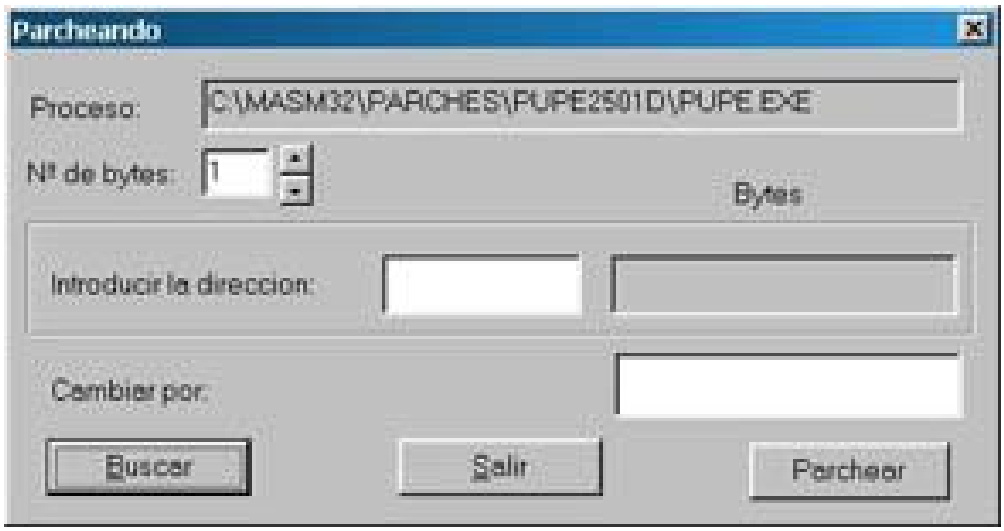
La pantalla que nos aparece apenas abrimos en PUPE es la siguiente:


Allí podemos ver los procesos que están ejecutándose en este momento en la computadora, una gran ventaja que tiene PUPE es que no es reconocido por la mayoría de los descompresores (YO DIRIA QUE POR NINGUNO) por lo que pasa completamente desapercibido, inclusive, uno puede arrancar el programa a crackear primero y una vez que ya arranco y no realiza mas verificaciones, arrancar el PUPE después con lo que su identificación es prácticamente imposible.

Esto de la identificación del programa es muy importante sobre todo con las últimas versiones de compresores como el ASPACK 2000 y 2001, que cuando arranca el programa tienen triple protección ANTISOFTICE (AUNQUE CON EL ULTIMO FROGSICE SON EVITADAS) pero también desactivan el REGMON, FILEMON y cualquier parcheador que existe hasta hoy como nuestro RISC PROCESS PATCHER que no funciona con esos compresores, además como la descompresión y que salga el archivo funcionando es complicadísima, para esos casos puede utilizarse el PUPE para parchear en memoria siendo que es el único que no es detectado por esos compresores.
Una vez que arranca nuestro programa víctima, va a aparecer en la lista de procesos que vimos anteriormente, allí seleccionamos el nombre del proceso sobre el cual vamos a trabajar y hacemos clic derecho y allí nos aparecen varias opciones.

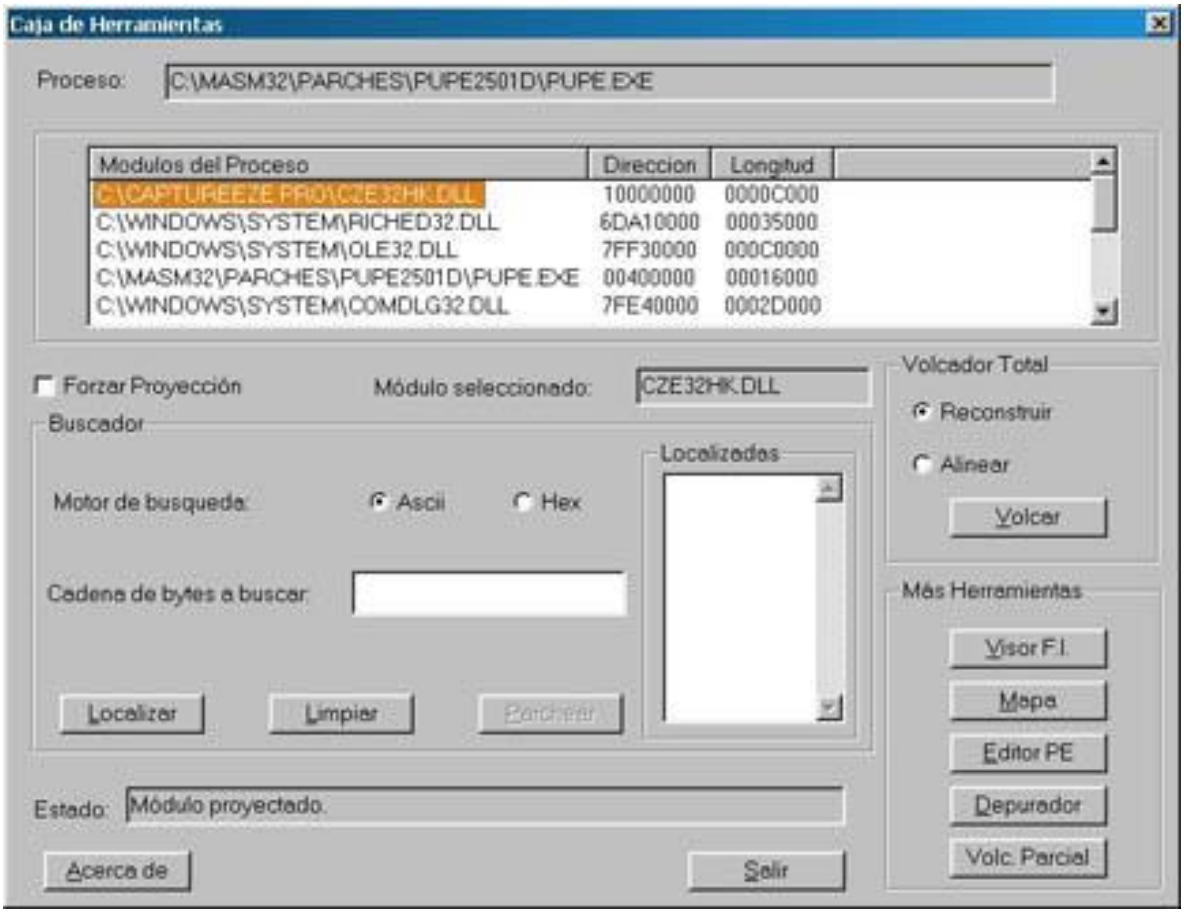



Si queremos utilizar el PUPE como parcheador solamente, elegimos entre estas opciones que aparecen, PARCHEAR y en una ventana de dialogo como la que vemos a continuación, colocamos la dirección que averiguamos con el SOFTICE o el WDASM que necesitamos parchear, cuantos Bits deseamos parchear y el valor de estos bits y listo, serán reemplazados en memoria inmediatamente…



Esta función de parcheador en memoria directamente, puede evitarnos mucho tiempo ya que si vimos algún salto sospechoso en el WDASM y antes de intentar con el SOFTICE poniendo BPX y todo eso aquí podemos probar directamente lo que queremos cambiar y ver si funciona, sin tanta historia, muchas veces necesitamos verificar varios saltos y esto nos puede ahorrar muchísimo tiempo, además que como dijimos en algunos compresores como ASPACK 2000/ 2001 hasta hoy es la única herramienta capaz de hacerlo.

La segunda es la opción CAJA DE HERRAMIENTAS. Si elegimos esta opción vemos lo siguiente:




Bueno aquí ya entramos a trabajar con nuestro programa directamente, aparece una lista de todos los módulos que utiliza el programa, y aunque casi siempre elegiremos el ejecutable, el programa nos da la opción de trabajar con cualquiera de los módulos que utiliza el mismo (MUY BUENA OPCION EN CIERTOS CASOS).
Elegimos el nombre del ejecutable o modulo sobre el cual vamos a trabajar y lo ingresamos haciendo clic derecho hasta que aparece el nombre en donde dice MODULO SELECCIONADO.
Ahora tenemos aquí varias herramientas que pueden ser de mucho interés. Hasta ahora para descomprimir un ejecutable era muchas veces bastante complicado según el compresor que sea, aquí podemos obtener un ejecutable descomprimido que aunque puede no funcionar,

nos permite rápidamente poder visualizar con el WDASM las STRING REFERENCES sin tener que perder horas descomprimiendo a mano.
Vemos a la derecha la opción VOLCADOR TOTAL y tenemos dos alternativas, podemos probar cual de las dos es la que funcionara pero si es un compresor moderno, usaremos la segunda opción, podemos probar pues tarda un segundo en realizar la tarea.
Una vez que tenemos el ejecutable nos queda solamente verificar con el PROCDUMP (O CON EL MISMO PUPE SEGÚN VEREMOS MAS ADELANTE) si hay que cambiar la protección ANTIDESENSAMBLADO de C000040 u otro valor a E0000020 como vimos en las lecciones de descompresión y listo, un ejecutable para poder ver con el WDASM en menos que canta un gallo, y sin rompernos mucho la cabeza.
Esta opción de VOLCADOR TOTAL funciona con CUALQUIER COMPRESOR QUE EXISTE y el resultado es un archivo para ser analizado con el WDASM, muchas veces el ejecutable que descomprimimos asi funciona perfectamente, otras hay que trabajar un poco para que funcione, pero, es muy útil listar en el WDASM las STRING REFERENCES de un programa comprimido en segundos, y además poder parchearlo aquí mismo.
Si cuando seleccionamos el módulo a utilizar nos dice que es muy largo entonces hay que colocar la tilde en FORZAR PROYECCIÓN para poder trabajar sobre el mismo.

Otra opción muy importante en la caja de herramientas es la de MOTOR DE BÚSQUEDA, la cual permite buscar en la memoria cadenas de texto, valores hexa o cadenas de valores hexa, esto nos puede ayudar en el crackeo en ciertas ocasiones:
Pongámosle un programa que tiene quince ejecuciones o que vence en 30 días y que cuando comienza nos dice cuantos días nos quedan o cuantas veces nos quedan para utilizar, también podemos usarlo en juegos en los cuales tenemos cierta cantidad de vidas o municiones o avioncitos que se yo , pongámosle cualquier caso de esos es lo mismo.
Pongamos el caso de que nos queden quince usos de un programa, buscamos el valor hexa de quince o sea F, colocamos que busque F dentro del programa, por supuesto nos saldrán muchísimos resultados, ahora copiamos los valores de memoria que nos salen y utilizamos una vez más el programa para que nos diga que quedan 14 usos, ahora copiamos los resultados para catorce (E) y vemos que ya las posiciones de memoria que salen en ambas búsquedas son menores y seguimos el mismo proceso hasta que descartamos y nos queda solo la posición de memoria donde el programa guarda los días, o las balas que utiliza el jueguito, o las veces que nos quedan utilizar, etc.
Una vez que ya sabemos la posición de memoria donde se guarda ese dato, podemos interrumpir el jueguito y cuando se nos están acabando las balas o nos queda una sola vida, utilizamos la opción parchear y colocamos el valor de la cantidad de balas que queremos o de vidas, o de días, etc, y lo modificamos desde el PUPE también.
Podemos también encontrar cadenas de texto y reemplazar desde aquí directamente, con el parcheador, las posibilidades son ilimitadas.

Después otra utilidad muy importante de la caja de herramientas es el VISOR DE FUNCIONES IMPORTADAS, lo que en el WDASM se conoce como IMP FN, ya que muchos programas al descomprimirse muestran las SRTINGS REFERENCES pero no salen la lista de Funciones Importadas, aquí podemos ver cuáles son las que utiliza el mismo sin hacernos muchos problemas.
La sección MAPA nos permite realizar un volcado a un archivo de la sección de memoria que queremos para analizarlo con tranquilidad.

La sección EDITOR PE nos permite hacer el mismo trabajo que realizamos con el Editor Pe del PROCDUMP (O SEA CAMBIAR LA PROTECCIÓN ANTIDESENSAMBLADO DESDE AQUÍ MISMO SIN ABRIR OTRO PROGRAMA) pero agregándole la posibilidad de volcar las secciones que queremos a un archivo, (ESTO PUEDE SER MUY UTIL YA LO VAMOS A VER)
En la sección DEPURADOR podemos hacer un desensamblado del programa directamente desde la memoria, MUUUY UTIL, en HEXA o en lenguaje ensamblador como el WDASM y verlo aquí o volcarlo a un archivo, además que aquí también tenemos botones para acceder al parcheador y al buscador en la memoria, además tenemos la opción de realizar un volcado parcial de alguna parte que queramos, como vemos es una linda herramienta que comenzamos a ver hoy y que en las sucesivas lecciones nos va a ayudar muchas veces en el trabajo de crackeo, y vamos a ver ejemplos prácticos de cómo utilizarla y aprovecharla en nuestro beneficio ya que para eso fue hecha por crackers.

**Ricardo Narvaja**

# COMO CRACKEAR EL MORPHINK 99 trial
# UN PROGRAMA PROTEGIDO CON VBOX

## (LECCION TRECE: Primera Parte)

Un amigo y listero me pidió que lo ayudara con el MORPHINK 99 Trial (GRANDE VESPERS) que es un programa que vence a los 15 días de utilizarlo, y me dijo que estaba muy protegido, lo cual es totalmente cierto, es CASI una fortaleza, salvo un pequeño detalle que abrió la puerta a la desprotección.
Ya cuando lo abrí con el WDASM empezaron las sorpresas, no hay STRING REFERENCES, no está hecho en VISUAL BASIC ni en DELPHI, si le aplico un identificador para ver si esta comprimido, como el LANGUAGE 2000 o el FILE ANALIZER no identifica nada.
Sin embargo el programa se desensambla con el WDASM no es como esos programas comprimidos en los que no aparece nada y termina rápidamente no aquí trabaja y desensambla bien.
Bueno aquí comienza la desesperación pero a no asustarse hasta un novato como nosotros puede si insiste encontrarle la vuelta al programa y una vuelta fácil de entender, ya que para mí los tutoriales sobre el tema son bastante complejos, entonces busco la solución fácil (SI LA HAY), y aquí la hay.
Bueno las STRING REFERENCES no nos dicen nada y las IMPORTED FUNTIONS me dicen VBOXP401, que no me dice mucho salvo que creo haber visto por ahí algún tutorial sobre VBOX pero no me acuerdo donde.
Ah ya lo encontré, busco en los tutoriales que tengo y veo que es un sistema de protección comercial (COMO EL ARMADILLO) pero este se basa en unos DLL que copia en la carpeta WINDOWS/SYSTEM que hacen el trabajo sucio de contar los días, de poner esa pantalla molesta que dice los días que faltan para que venza, (SON QUINCE EN TOTAL) y la protección ANTISOFTICE.
Allí hay tutoriales con sesudas explicaciones de cómo parchear en memoria esta porquería, y como descomprimirlo, pero a decir verdad a mi no me sirvió ninguna, no sé si será porque no es igual para todos los archivos la protección Vbox pero después de descomprimir el archivo según esos tutoriales no me funciona a pesar de haber arreglado el ENTRY POINT y todo eso, por ahí me equivoque yo, casi seguro, pero ya que halle otra forma más fácil por lo menos para este programa, bueno utilizare esta, y dejare la otra para los bochos de la descompresión.
La protección ANTISOFTICE tenemos que esquivarla y en esto agradezco nuevamente a VESPERS que me paso el datito

Existe una utilidad para limpiar esta protección comercial que se llama
VBOX Unwrapper [uCF] by MAK & EinZtein in 2000

Pero para mi hay malas noticias
This Unwrapper will only run on Windows 2000!!!!

O sea que solo funciona en WINDOWS 2000, ay a quien se le ocurre hacer una herramienta solo para WIN2000 y que no funciona en WIN98, y es así la probé y se cuelga en WIN 98.

Bueno lo que si tenemos es una gran herramienta como es el nuevo PUPE que fue programado por crackers y es evidente que es de gran utilidad, por lo menos para mí, grande MARMOTA, CRACK EL DESTRIPADOR Y COMPANIA sin PUPE no se que habría hecho.
Arranco el programa y espero que aparezca la ventana que dice cuantos días me quedan, (QUE HORROR), antes de poner TRY o cualquier otra cosa arranco el PUPE, y allí elijo el proceso del MORPHINK y allí entro a la CAJA DE HERRAMIENTAS, y elijo el modulo que según los tutoriales es autor de todo el desastre
VBOXT401.DLL y elijo VOLCADO TOTAL y alinear y lo guardo.
Lo abro con el WDASM y me aparece perfecto con FUNCIONES IMPORTADAS y STRING REFERENCES, lo que no sería posible si lo buscara en el rígido y lo abriera con el WDASM ya que esta comprimido.
Para saltear la protección ANTI SOFTICE veo la parte siguiente

```
* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:07006F44(C)

:07006F4B     6A00                 push 00000000

* Possible Reference to Dialog: DialogID_00D1, CONTROL_ID:00FF, ""
:07006F4D     6AFF                 push FFFFFFFF
:07006F4F     683B9D0000           push 00009D3B
:07006F54     8D8D34FEFFFF         lea ecx, dword ptr [ebp+FFFFFE34]
:07006F5A     E8E1220100           call 07019240
:07006F5F     8945BC               mov dword ptr [ebp-44], eax
:07006F62     8D458C               lea eax, dword ptr [ebp-74]
```

Ese CALL me devuelve un cero si no está el SOFTICE y me devuelve un uno cuando está presente así que hay que poner un BPMB 7006F5F justo cuando después de pasar el CALL y allí poner **r eax=0** entonces pasa la protección.
Miramos las STRING REFERENCES del dll maldito.
Allí en las STRING REFERENCES veo algo que me llama la atención, dos cadenas **UNLIMITED TRIAL** y **TRIAL EXPIRED**, hmmm sospechoso y hay un salto en **70117a2** que me tira a **TRIAL EXPIRED** y si no salta va a **UNLIMITED TRIAL**, mmm que lindo esta esto.

```
0701179f   test al,10
070117a1   push ebx
070117A2  je  70117be     (SI SALTA VA A TRIAL EXPIRED)

Posible Reference to String Resource "UNLIMITED TRIAL"

070117a4   push 00001478
```

O sea que testea Al con 10 y allí decide si seguir el camino TRIAL EXPIRED o UNLIMITED TRIAL.

Abro el SYMBOL LOADER y allí una vez que cargo el ejecutable y para cuando se inicia le pongo un **BPMB 701179f x** , un poquito antes del salto.
Pongo BREAKPOINTS DE EJECUCIÓN en memoria en vez de BPX porque el programa detecta si pones BPX y te dice que tenés un debugger activo y no funciona.
Una vez que para allí, ya que compara AL con 10 y según eso salta o no y ya que AL no es igual a 10 lo que va a hacer que vaya por la zona de TRIAL EXPIRED CHICO MALO, entonces con

R AL=10

Y ahora en vez de saltar es como si hubiéramos invertido el salto, va a UNLIMITED TRIAL, saco todos los BREAKS con BC*
Y corro el programa con x
Y la primera vez la pantalla que dice los días que faltan aparece pero ya no dice los días esa zona aparece negra, paso la protección ANTI DEBUGGER como vimos antes, cierro el programa y lo vuelvo a abrir y ya no aparece más la NAG asquerosa esa y arranca directamente el MORPHINK 99 sin ningún problema. Adelanto el reloj una año y sigue arrancando perfectamente.

## (LECCION TRECE: Segunda Parte)

Como desproteger completamente al MORPHINK.

Una vez que reseteé la máquina el programa se volvió a desregistrar. La verdad que el programa me haya engañado ya me había subido la presión, así que decidí no parar hasta no limpiarlo completamente y dejarlo como un archivo suelto sin necesidad de dlls Vbox ni nada.

Lo primero que me llamo la atención en todo éste proceso es lo siguiente, porque a mí el MORPHINK no me detecta el SOFTICE y a mi amigo VESPERS si, dado que los dos tenemos la misma versión del programa bajada del mismo lugar y el mismo softice 4.05 ? Porque a él si esta el SOFTICE cargado le aparece el mensaje de error y no le funcionaba el MORPHINK y a mí no me aparece.
La verdad que yo me confundí con respecto a esto, como hice tantas cosas no sabía si exactamente era algo que había hecho yo sin querer o porque sucedía.
Cuando el programa se desregistro, tuve una segunda oportunidad de ver bien que pasaba y la situación en mi maquina después de muchísimas pruebas es la siguiente:

Si arranco el programa con el SOFTICE cargado, me arranca SIEMPRE el MORPHINK y no lo detecta.
Si pongo BPXs ahí si recién me sale el cartelito de error.
Mientras me mantenga poniendo BPMs y BPR el programa arranca normalmente.
Al revés si utilizo el FROGSICE me sale el mensaje de error.
Lo mismo que si en el CALL que puse en la primera parte de la lección para saltear el SOFTICE pongo r eax=0 me sale el mensaje de error y si dejo el valor de EAX=22 que me aparece solo, arranca normalmente el programa.

Ahora cual es la diferencia entre mi softice y el de VESPERS?

Después caí en la cuenta que yo había modificado el ejecutable del SOFTICE con un ocultador llamado BACKDOOR KEEPER que MODIFICA y OCULTA según el autor, algunos puntos débiles que el FROGSICE no puede tapar.
Por eso el programa a mi no me detecta el Softice directamente y a VESPERS si, de cualquier manera a la persona que le sirva la forma de ocultar el SOFTICE que puse en la primera parte que la use, funciona para el que no modifico el SOFTICE con BACKDOOR KEEPER.
Ahora el BACKDOOR KEEPER es la primera vez que me tapa la detección del SOFTICE ya que siempre lo había podido hacer con el FROGSICE, y bueno ahí está la diferencia ya que el BACKDOOR KEEPER MODIFICA el ejecutable del SOFTICE para SIEMPRE y como yo no me acordaba que lo había hecho, no sabía que pasaba, igual solo sirve para algunos pocos casos (COMO ESTE POR EJEMPLO)
Bueno al grano:
Lo primero hice arrancar el programa MORPHINK hasta que pase la pantalla de PRUEBA y llegue a arrancar completamente.

Ahora probé que si el programa esta arrancado y vuelvo a arrancarlo sin haberlo cerrado antes, esta segunda vez arranca sin poner la pantalla de TRY o sea de probar, arranca directamente.
Fijándome en las funciones importadas del archivo VBOXP410.dll encuentro
BPX GETPROCESSADDRESS

entro en el SOFTICE y pongo
BPX GETPROCESSADDRESS

siempre con el programa abierto y trabajando sobre una segunda apertura simultanea del mismo, o sea siempre dejo minimizado el programa ya abierto y trabajo ahora como si no estuviera abierto, ejecutándolo por segunda vez.
Una vez que empieza a arrancar después de un rato para, me fijo abajo a la derecha que esta parada corresponda al proceso MORPHINK y una vez que aparece allí MORPHINK (OJO NO EN LA LINEA VERDE SINO BIEN ABAJO A LA DERECHA), entonces borro todo con BC*
y vuelvo con F12 hasta que aparezco en VBOXP410.
Allí veo que el modulo VBOXP410 se ejecuta en direcciones superiores a 5.000.000, lo que puedo ver también con el WDASM, y el programa MORPHINK se ejecuta en direcciones apenas superiores a 400.000, esto lo comprobé también cuando al tener arrancado del todo el programa veo que al poner un BPX hmemcpy y tocar cualquier cosa y entrar en el SOFTICE el programa tiene direcciones apenas superiores a cuatrocientos mil, mientras que el VBOX se ejecuta por encima de los cinco millones.
Entonces volviendo al punto en que estoy en el softice dentro del VBOXP410 supongo que en algún momento el archivo este va a dejar de trabajar y pasar el control al programa MORPHINK.
Por eso de aquí hago como en la lección de descompresión manual y tecleo a ojo sin calcular valores

BPR 400000 500000 r if (eip>=400000) && (eip<=500000)

Para que el programa pare en la primera sentencia del MORPHINK que debe estar a ojo entre esos valores 400000 y 500000. Pongo a correr el programa y para en

43E1C0 que es nuestro nuevo ENTRY POINT.

Ahora borro todo con BC*
Y hago un loop infinito allí para que el programa quede trabajando siempre en la misma sentencia

A 43E1C0 JMP 43E1C0

Y vuelvo con X a WINDOWS. Ahora lo única diferencia con el método que usamos en la lección de descompresión manual es que aquí no funciona el PROCDUMP, así que yo abrí el PUPE y elegí el proceso MORPHINK y el ejecutable Morphink.exe e hice un volcado total con la opción ALINEAR activada.
Lo guarde en el escritorio con el nombre MOCO.exe.
Ahora con el ULTRAEDIT modifico los valores que tuve que cambiar para lograr el LOOP INFINITO o sea busco
EB FE EC 6A FF 68 D0

Y vuelvo a colocar como era originalmente en vez de **EB FE , LOS VALORES 55 8B**, que copie antes de modificar para hacer el loop infinito. Una vez que hicimos esto solo nos queda modificar el ENTRY POINT al nuevo valor, y eso lo podemos hacer COM PUPE o con PROCDUMP, o sea si uso PROCDUMP por ejemplo el nuevo ENTRY POINT debe ser 003e1c0 ya que le reste los 400000 de la image base.

Ahora tengo el archivo moco sobre el escritorio, cierro todas las instancias de MORPHINK, reseteo la máquina para que no me vuelva a engañar lo ejecuto allí mismo en el escritorio y ABRE PERFECTAMENTE Y FUNCIONA, otra protección que pudimos vencer no sin bastante trabajo.
Gracias a VESPERS que cuando yo me equivoque me ayudo, la verdad que ya es un cracker a todo trapo y estoy contento de que aunque sea yo con mis lecciones haya colaborado un poquito con algo de todo lo que aprendió.
Otra cosa que quiero destacar es la utilidad del PUPE, la verdad una muy linda herramienta, de inapreciable valor.

**Ricardo Narvaja**

# UNA LECCIÓN PARA RELAJARSE Y CRACKEAR ALGO FÁCIL

## (LECCION CATORCE)

### *BIKERS LOG 3.5*

Ya que las ultimas lecciones fueron bastante difíciles y dado que llegan las fiestas vamos a crackear algo sencillo, a pesar de que es un programa nuevo, en su última versión la 3.5 , para que recuperemos la fe en que todos los programas no son armatostes complicados en DELPHI, con protecciones VBOX, ARMADILLO, ASPACK y todas esas porquerías que cuestan un PERU crackear.

Este es un programa refácil de crackear que nos hará recordar los buenos viejos tiempos de cracks fáciles, tiene protección ANTI-SOFTICE pero con el Frogsice se pasa, es refácil, sencillón, para las fiestas, con un pan dulce y una sidra en la mano, y la cabeza tranquila sin rompérsela por nada.

Voy a subir el BIKERS LOG 3.5 al FREEDRIVE que es un programa para los que les gusta andar en bicicleta y se pueden anotar todos los datos que necesitan los ciclistas obviamente, como yo no soy ciclista, no lo uso, pero es un placer crackear algo así.

Además si alguien quiere experimentar con algo sencillo de crackear les recomiendo que no lean el tutorial y se larguen sin miedo, es fácil.

Lo único un poquito más complicado es que esta comprimido y no se ve nada con el WDASM pero si lo agarran con el PUPE y cuando ya arrancó el programa le hacen un VOLCADO TOTAL, con la opción ALINEAR van a poder desensamblarlo con el WDASM y ver las STRING REFERENCES.

Bueno vamos a UNLOCK y ahí ponemos mi clave preferida 989898 y una vez que la escribimos, entramos al SOFTICE con CTRL+D y tecleamos BPX HMEMCPY y con CTRL+D volvemos a la ventana para escribir la clave, esperamos unos segundos para ver si no vuelve a entrar solo en el SOFTICE antes de apretar el BOTON OK de la ventana de REGISTRO (YA SABEMOS QUE SI ENTRA SOLO AL SOFTICE ANTES DE APRETAR OK TENEMOS QUE SALIR Y CERRAR LOS PROGRAMAS QUE ESTAN FUNCIONANDO CON CTRL+ALT+DEL y FINALIZAR TAREA MENOS EL EXPLORER Y EL BIKERS LOG OBVIO)

Una vez que apretamos OK entra en el SOFTICE por la acción del BPX HMEMCPY, aprieto F12 hasta volver al ejecutable BIKER (son siete veces que hay que apretar F12) y una vez que veo BIKER en al línea verde del SOFTICE empiezo con F10 a recorrer el ejecutable saliendo de esos RET y cuando vemos alguna sentencia que transfiere algún valor a algún registro (EAX, EDX, ECX o alguno de esos, miramos con D EAX o el registro que sea y seguimos traceando y mirando hasta que llegamos a

6de41e y 6de421 donde se transfiere a EDX y EAX mi clave falsa y la clave buena, si hacemos justo después de ejecutar esas dos instrucciones D EAX y D EDX, vamos a ver la clave verdadera que es 977-0956....., se las dejo para que la averigüen ustedes, ja ja.

**Ricardo Narvaja**

# UN NUEVO WINOMEGA (5.15.13)

## (LECCION QUINCE)

A veces cuando sale una nueva versión de un programa que ya crackee es lindo ver los cambios que se han realizado en la protección que uno ya conoce y comparar si ha mejorado, en ese sentido.
Este es el caso del nuevo WINOMEGA 5.15 (el anterior era 5.12) que se puede bajar de www.winomega.com

Una vez desinstalada la versión anterior para que no tome registros de antes, procedemos a mirar la nueva, parece bastante similar aunque tiene una decoración NAVIDEÑA por las fiestas y un poquito mejorada la interfase, ahora vamos a lo nuestro.
En la versión anterior el ejecutable lo descomprimimos con PROCDUMP utilizando WWPACK32 II y ahora vemos con el WDASM que esta comprimido también, probemos con el mismo descompresor a ver que pasa y VOILLAAA, funcionaaaaa.
Hasta aquí todo igual, pero no es todo igual, reemplazo el ejecutable descomprimido y arranca el programa, protección ANTISOFTICE no tiene, esperaremos la próxima versión para esto, así que vamos a AYUDA - REGISTRARSE y ponemos cualquier numero por ejemplo 99999999999 y nos sale PIN INCORRECTO, que parecido que es hasta aquí..
Bueno arranco el SOFTICE y pongo BOX MESSAGEBOXA y pongo un numero falso y no para allí o sea que no usa MESSAGEBOXA, busco en el WDASM la STRING PIN INCORRECTO y no aparece, mmmm, esto se comienza a complicar, pongo BPX HMEMCPY y trato de volver al ejecutable a ver que pasa, da mil vueltas y no se llega a nada, es un despiole, parece que en algo mejoro, han cerrado algunas puertas un poco mejor.
Sería muy piola decir fui aquí y hice esto, pero la verdad es que tuve que intentar bastante para abrir esta puerta esta vez, prueba que te prueba, a veces los crackers ganamos por cansancio o nos ganan a veces por cansancio a nosotros.
Bueno después de muchos intentos infructuosos, me fui a abrir con el DEDE para ver si el ejecutable descomprimido estaba hecho en DELPHI, dada la cantidad de vueltas que da el programa, y si, está escrito en DELPHI y el DEDE lo abrió perfectamente.
Bueno, sabemos que con el DEDE solo podemos trabajar con ejecutables descomprimidos si no, no funciona, así que puesto el ejecutable descomprimido en el lugar del original, lo abrió perfectamente.
Bueno una vez allí vamos a PROCEDURES y allí buscamos algo y allí esta REGISTERFRM o sea algo que tenga que ver con REGISTRO.
Aclaremos que creo que aquí sin el DEDE nos volveríamos japoneses o chinos buscando entre vueltas y vueltas de DELPHI, así que su ayuda es invaluable.
Bueno haciendo clic en REGISTERFRM-EVENTS aparecen los EVENTOS que hay que mirar, hay varios botones y un poco por intuición y otro poco por mirar un poquito los otros y no encontrar nada interesante, me incline por el primero que es el OK BUTTON, descarte el HELP button ya que no creo que un botón de help te registre y en el FORM BUTTON no hay comparaciones y saltos interesantes, además el REGBUTTON parece empezar donde termina el OK BUTTON o sea que primero parece ejecutarse el OKBUTTON.
Bueno la cosa estaba entre el OKBUTTON Y EL REGBUTTON ya que el OKBUTTON está primero lo vamos a estudiar primero.

Después de probar salto por salto con el SOFTICE a ver qué pasaba (NO SON TANTOS SON CINCO O SEIS), di en el clavo con el salto de 5E136A, ESE CUANDO CAMBIE EL VALOR DE **BL** PARA INVERTIR EL SALTO ME DIJO QUE ESTABA REGISTRADO, al reiniciar el programa me desregistro de nuevo así que debe haber al igual que en la versión anterior un CALL antes del salto que es llamado a comparar la clave y lo hace antes del salto 5e136a y lo llama a ese CALL cuando comienza el programa, ese CALL es
5e1326 CALL 662020
y si lo vemos con el WDASM podemos ver que viene de otro lugar en REFERENCED BY A CALL AT
64f43e CALL 662020 (CUANDO ARRANCA EL PROGRAMA)
Así que ahora solo queda poner un BPX en 64f43e para que pare cuando arranque invertir ese salto, y queda registrado para siempre, se pueden realizar los cambios con el ULTRA EDIT y NOPEAR el salto que hay después de 64f43e así siempre te acepta tu registro y no te desregistra.
Parece fácil una vez ya hecho pero el hecho de no tener referencias claras en las STRINGS y de no ser ventanas fácilmente localizables, hacen que sea un poco mas difícil, encontrar los saltos, otra cosa que se puede ver que si seguimos el salto 5e136a y vamos a la parte de CHICO MALO enseguida haciendo F10 varias veces encontramos cual es el CALL de la famosa ventana de PIN INCORRECTO, ya que en DELPHI las ventanas son casi siempre CALLS enteros y esta lo es.

**Ricardo Narvaja**

# COMPRESORES MALDITOS

## (LECCION DIECISEIS)

## Vamos a crackear el programa WEBCLICKER 1.0

La verdad esta historia de los compresores a veces lo saca a uno de quicio, y este es el caso, vamos a abreviar, esto esta comprimido todavía no sé con qué, pero no se puede descomprimir, por lo menos con las herramientas clásicas, el PROCDUMP cuando lo lee se cuelga, el WDASM también el PUPE lo vuelca pero el archivo resultante es una porquería que no se puede abrir con nada, una verdadera basura.
Este tema de los nuevos compresores que existen por ahí es bastante molesto, ya tuve experiencia con el ASPROTECT que es más difícil que este que vamos a ver hoy día y todavía no le encuentro la vuelta de cómo descomprimirlo, a este tampoco, pero eso no quiere decir que no lo vamos a poder crackear, lo haremos usando la única herramienta que nos permite este compresor que es el SOFTICE, aunque también tiene trampa anti-debugger, si modificamos el ejecutable del SOFTICE con el BACKDOOR KEEPER y usamos el nuevo FROGSICE , podemos hacer algunas cosas, otras no, pero igual lo haremos.
Esto es como crackear con una mano atada y sin ver pero lo más importante en este crack es ser creativos y usar un camino que inventemos aunque no sea el más recorrido en los tutoriales de crack, por suerte este compresor permite parchear en memoria con el RISC PROCESS PATCHER, cosa que el ASPROTECT no permite, así que tiene esa pequeña puertita abierta, donde nos vamos a tratar de meter para crackearlo, la verdad es que fue difícil, muy difícil, tarde más de una semana en hacerlo, aunque aquí parezca breve ya que muchos intentos frustrados yo no los transcribo y parece fácil entonces.
Bueno manos a la porquería esta, jaja, chiste.
Probemos que nos deja hacer y que no con el SOFTICE.
Arrancar el programa: SI SE PUEDE siempre que tengas el SOFTICE modificado con el BACKDOOR KEEPER y el nuevo FROGSICE, arranca perfectamente.
Con el SYMBOL LOADER: arranca el programa y no para cuando se inicia, o sea que no sirve para nada arrancarlo de aquí, no para porque en las características de las secciones que leí con el PE BUILDER (EL UNICO QUE SE DIGNO A ABRIRSE), me dice que todas las secciones son C0000040 lo que hace que no pueda ser desensamblado ni pare cuando arranca con el SYMBOL LOADER, si le cambio a E0000020, el programa para con el SYMBOL LOADER pero no arranca ya que se testea a si mismo y al verse modificado sale un mensaje de error.
Si le modifico a E0000020 también se desensambla con el WDASM pero no aparece nada interesante ya que esta comprimido y no aparecen STRIGS ni IMP FN ni nada.
Es un acorazado, por donde lo podemos atacar?
Si pongo un BPX antes de que arranque el programa vamos a ver que pasa:
Si es MESSAGEBOXA me sale DEBUGGER DETECTED y no arranca, paciencia amigo ya lo vamos a destripar, probemos con BPX GETVERSION es un BPX que usan la mayoría de los programas cuando se inician, ya que así saben la versión de WINDOWS que tenés en tu maquina y ajustan el programa a eso.

Entonces BPX GETVERSION, para en el SOTICE (ACLARACIÓN SI EN LA ESQUINA INFERIOR DERECHA QUE ES DONDE APARECE EL NOMBRE DEL PROCESO QUE LLAMA A LA FUNCION APARECE EXPLORER EN VEZ DE WEBCLIKER, HAGO X HASTA QUE APAREZCA ALLI WEBCLICKER no en la línea verde sino en el ángulo inferior derecho de la pantalla) pongo BC* y X a ver si arranca, ARRANCA, bueno ya tenemos una forma de entrar, vamos despacito.
Probemos buscar el punto de inicio del programa descomprimido para ver si pasa:
Pero antes tenemos que volver al ejecutable, tenemos que borrar el BPX que habíamos puesto sino va a aparecer DEBUGGER DETECTED, y para volver si hacemos F12 no sirve, a mi se me cuelga así que tengo apretada la tecla F10 hasta que en la línea verde aparezca el nombre del EJECUTABLE WEBCLICKER y ahí suelto.
Según los datos que obtuve con el PEBUILDER y aplicando el método de descompresión manual que vimos en la LECCIÓN 6, escribo
BPR 401000 473000 R if (eip>=401000) && (eip<=473000)

y a ver que pasa X.
PASA Y PARA EN EL INICIO.
4720BC PUSH EBP
les digo para que ahorren tiempo que si aplicamos aquí el método de cambiar a un loop infinito con
A (ENTER)
y escribo JMP EIP o JMP 4720BC para que quede en loop infinito y dumpearlo, el PROCDUMP se cuelga cuando lo querés grabar y el PUPE lo que sale no sirve para nada es un chorizo que ni se puede abrir con el PEBUILDER como el ejecutable original.
Todos los otros DUMPEADORES que probé lo mismo lo que sale es una bazofia.
Bueno por lo menos estamos en el inicio del programa, no podemos ver el listado muerto, no tenemos STRING REFERENCES, ni IMP FN, pero a no llorar, parece que llegamos aquí en condiciones lamentables pero hagamos un repaso de los pertrechos que tenemos:
1) Estamos en el arranque del programa, y podemos partir de aquí usar el SOFTICE a voluntad ya que no chequea mas nada ni hay mas protecciones ANTI DEBUGGER a partir de aquí.
2) Si encontramos aquí algún salto o algo importante que REGISTRE el programa lo podemos modificar con el RISC PROCESS PATCHER y a cobrar.

Bueno algo tenemos ahora veamos la línea de razonamiento que seguí para crackear esto.
Mire el programa cuando arranca y vi en la ventana del mismo por varios lugares la palabra SHAREWARE, y además sabemos que el programa se registra ONLINE así que debe haber algún lugar que testee si estas REGISTRADO o no y según eso haga aparecer la palabra SHAREWARE en pantalla si no estas REGISTRADO y si estas REGISTRADO no tiene que aparecer.
Eso vamos a buscar la decisión de que aparezca la palabra SHAREWARE en la ventana o que no aparezca, como eso tiene que ver con que si estas REGISTRADO o no puede ser algo que nos lleve a la solución (O NO).
Allí en el punto de INICIO del programa uso la sentencia que tiene el SOFTICE para buscar CADENAS DE TEXTO
S 0 L FFFFFFFF 'SHAREWARE'

ASI CON MAYUSCULAS como aparece en la ventana.
La sentencia de buscar texto es así y si quieren buscar otra vez ya que así sale solo la primera SHAREWARE que encuentra tiene que volver a escribir
S XXXXXX L FFFFFFFF 'SHAREWARE'

donde XXXXXX es la dirección de memoria justo después de donde apareció la primera vez.
Ya que aquí apareció en 470b8b (EN MI CASO ESTO PUEDE VARIAR EN CADA COMPU), si quisiera encontrar la segunda vez que aparece tendría que escribir.

S 470b9b L FFFFFFFF 'SHAREWARE'

y además tener en cuenta que el SOFTICE aquí acepta como comillas no las comillas dobles estas " , sino las comillas simples estas ' .

Bueno apareció en 470b8b, entonces pongamos un BPR allí para ver cuando el programa la usa a esa palabra. Primero borro los BP anteriores con BC* y escribo.

BPR 470b8b 470b9b RW

cosa que cuando el programa se decida a utilizar la palabra SHAREWARE pare allí el SOFTICE,y estaremos en la zona de CHICO MALO, y tendremos que buscar la decisión en una comparación y un salto anterior para llegar a la ZONA DE CHICO BUENO o REGISTRADOS.y que evite que use la palabra SHAREWARE.

Bueno para en 402855 REPZ MOVSD
que es la sentencia que esta copiando la palabra SHAREWARE para escribirla en pantalla.
Veo que unas sentencias más abajo hay un RET lo cual quiere decir que estoy dentro de un CALL y ese RET es la FINALIZACION del CALL , entonces para salir del CALL y volver a la sentencia subsiguiente después del mismo, traceo con F10 hasta el RET y lo traceo también y salgo a

403d40
El CALL que está justo antes en
403d3b CALL 402828 que era adentro de donde yo estaba y donde el programa trabaja la palabra SHAREWARE es para mí la ZONA de CHICO MALO, tengo que tratar de saltear ese CALL de alguna forma para que no caiga allí. Pruebo poniendo BPX en los saltos que están arriba del CALL y reiniciando todo para que caiga allí pero no funcionan, entonces como veo otro RET me doy cuenta que estoy en otro pequeño CALL todavía y hago lo mismo para salir haciendo F10 hasta el RET y caigo en la sentencia posterior a:

470960 CALL 403d0c

O sea que este CALL también es la ZONA DE CHICO MALO ya que dentro de el esta el otro CALL 402828 que salí primero, o sea que si caigo en 470960 voy a parar a la palabra SHAREWARE, miro arriba del CALL y encuentro algo interesante bastante arriba:

en 470815 jz 47094b
es un salto que si no salta, al seguir traceando con f10 veo que sigue línea por línea hasta 470946 jmp 4709e4 que saltea el CALL maldito, y si el salto en 470815 salta, al esquivar el JMP ya que cae después del mismo, nos lleva directo al CALL SHAREWARE, quiere decir que ese salto no debería saltar para que no caiga en SHAREWARE, probemos.
Pongo un BPX en 470815 y reinicio todo (TENGO QUE HACER TODOS LOS PASOS QUE HICE ANTES PARA QUE PARE SI NO PASA DE LARGO), bueno cuando paro allí, veo que el SOFTICE me avisa que va a saltar y mandarme a SHAREWARE, invierto el salto con
r eip=47081b
y luego hago X y PROGRAMA REGISTRADO, oh que maravilla no solo esquivo SHAREWARE sino que agarro por el camino de CHICO BUENO directo, igual queda un detalle que es que a veces aparece el cartel ese blanco diciendo que nos tenemos que registrar a pesar que en el PROGRAMA nos dice que ya estamos registrados, eso se arregla fácil veamos que compara antes del salto.

MOV eax, [474e08]

mueve a eax el contenido de 474e08 que es 47451c
o sea que eax vale ahora 47451c
y después compara el primer byte de el contenido de 47451 con cero si es cero nos tira a los chanchos y si es uno estamos registrados, aquí lo que ocurre es que este es un flag que le dice al programa si esta registrado o no, y más adelante lo debe testear para ver si sale el cartel blanco ese que aparece cada cuatro o cinco veces que arrancas el programa, entonces para evitar problemas, cambio la sentencia

cmp byte ptr [eax],00

a
mov byte ptr [eax],01

con lo que pasa el flag a ser uno y nunca más aparece el cartel blanco ese y para el programa estamos perfectamente registrados.
Luego hacemos el cargador con el RISC PROCESS PATCHER cuyo script sería el siguiente:

**; WEBCLICKER**

**F=Webclicker.exe: ; PROCESS TO PATCH**
**O=crack_the_webclicker2.exe: ; LOADER TO CREATE**
**P=470812/80,38,00/c6,00,01: ;**
**P=470815/0f,84/eb,04: ;**

**$ ;end of script**

En la primera línea P cambio la sentencia comparar a MOVER como vimos antes y en la segunda para no nopear tantos bites directamente cambio el salto por un JMP 47081b, sino tendría que poner muchos NOPs, lo hago al parche lo copio en la carpeta del programa, lo ejecuto, y arraaaaancaaaa perfectamente registrado.
En resumen le hicimos morder el polvo a una dificilísima protección sin siquiera descomprimir el ejecutable, jaja, pero fue muuuy duro realmente.
El programa Webclicker estará en el freedrive, para bajarlo.

**Ricardo Narvaja**

**ALGO SUUUPER FACIL**
**EL CRACK DE LA ACTUALIZACION DE VIRUS DEL NORTON**

**(LECCION DIECISIETE)**

Bueno como ustedes sabrán si no se les cuento, nuestro amigo NORTON nos dejaba actualizar un año por medio de su LIVE UPDATE las definiciones de virus, y cuando se te vencía ese año (QUE PARECE QUE DURA UNOS MESES SOLAMENTE), podías seguir bajando las actualizaciones de la pagina de SYMANTEC
http://www.symantec.com/avcenter/cgi-bin/navsarc.cgi
o
http://www.sarc.com/avcenter/cgi-bin/navsarc.cgi

y no había problema, bajabas el archivito que tiene un nombre así 0825i32.exe que quiere decir que es la actualización del 25 del ocho o sea de agosto, bueno todo era hermoso, hasta que se le ocurrió, limitar también el archivo que podes bajar, o sea que una vez que lo ejecutas, testea si te venció ese año y si ya paso la fecha no te deja actualizarlo y te pide que pagues una suma para poder actualizar los virus.

Bueno les comento que crackear ese archivito es muy sencillo, desde hace dos meses todos los archivitos de actualización tienen la misma protección y en el mismo lugar, o sea que al día de hoy 10 de enero del 2001, lo que explico en esta lección es válido, si llegara a cambiar de lugar la protección, el método es muy sencillo y seguro va a ser el mismo aunque en otro lugar, pero hasta hoy parece que va a seguir igual.

Si ejecuto el archivo de definiciones de virus cuando ya vencieron las actualizaciones me sale un cartel que dice:
YOUR VIRUS SUSCRPICIONS CANNOT BE UPDATED

YOUR SUSCRIPCION HAS EXPIRED…. y un par de pavadas mas

Abrimos el archivo con el WDASM y vemos las STRINGS REFERENCES y allí aparece la cadena de texto YOUR VIRUS SUSCRIPCION.... parece que NORTON no tenía ganas de molestarse mucho, hacemos doble clic en esa STRING REFERENCE y vemos que corresponde a un solo lugar 4042af (hasta hoy) , y que viene de un
REFERENCED BY A CALL AT 401dff

Vamos a 401dff y vemos un poquito más arriba que hay dos saltos, probamos ambos si queremos con el SOFTICE, pero es el primero de los dos el que funciona bien esquivando el CALL que nos lleva al cartelito maldito.

**401df4 75 1C jne 401e12**

aquí si salta pasa por encima del CALL así que reemplazamos 75 por EB para hacer que salte siempre, pero no lo vamos a hacer con el ULTRA EDIT porque si no cada archivo que bajemos vamos a tener que editarlo, vamos a hacer un cargador con el RISC PROCESS PATCHER para cargar el archivo de actualización, como el RISC busca el nombre del

ARCHIVO lo único que vamos a tener que hacer cuando bajemos uno, es cambiarle el nombre a NORTON.exe y listo.
El SCRIPT es el siguiente.

**; NORTON UPDATE**

**F=NORTON.exe: ; PROCESS TO PATCH**
**O=PARCHEARNORTON.exe: ; LOADER TO CREATE**
**P=401df4/75/eb: ;**

**$ ;end of script**

Con esto el parche cargara cualquier archivo de actualización llamado NORTON.EXE (ACORDARSE DE RENOMBRARLO ) y buscara el salto en 401df4 y alli reemplazara el 75 por un EB convirtiéndolo en un JMP para que siempre esquive el CALL.
FACILISIMO
Probamos el cargador y FUNCIONA, actualiza los virus perfectamente.
Si más adelante el cargador alguna vez no funciona es:
1) Porque no renombraron el archivo de actualización a NORTON.EXE
2) Porque no pusieron ambos el parche y el NORTON.EXE en la misma carpeta
3) Porque la protección cambio de lugar y el salto esta en otro lado, de cualquier manera se ve que es muy fácil hallar donde está ya que el archivo no está comprimido ni protegido, por lo tanto solamente mirándolo con el WDASM, sin siquiera ser necesario usar el SOFTICE pudimos hacer el PARCHE.

Esta LECCIÓN fue como un oasis de descanso después de la 16 que fue dificilísima, no?

**Ricardo Narvaja**

**SIGAMOS CON LA ONDA LIGHT**
**CRACK DEL ATAMA 1.8**

**(LECCION DIECIOCHO)**

Bueno vamos a seguir la onda light con cracks fáciles ya que el verano y el calor están haciendo estragos en nuestras neuronas para complicarnos mucho. Este programa es supuestamente sobre HOMEOPATIA de lo cual yo no sé nada ni tampoco de medicina, si esta ciencia sirve o no , yo no puedo opinar, debería preguntar alguien más versado sobre el tema como el famoso Doctor Theumann que de esto sabe, jaja, pero bueno, igual nos va a servir para practicar un poco .
Supuestamente el programa dice según la dolencia cual es el remedio que hay que usar, es como un inventario sobre Homeopatía, y bueno quizás a alguien le sirva, se baja de

http://www.abouthomeopathy.com/

Instalamos el programa y abrimos el ejecutable ATAMA.exe con el WDASM y vemos en FN IMP (FUNCIONES IMPORTADAS) la mención de MSVBVM6.0 lo cual nos dice que está hecho en VISUAL BASIC, igual resultado podríamos haber obtenido abriendo el ejecutable con algún identificador. Probamos abrir el ejecutable con el SMART CHECK y no funciona, parece estar protegido contra el SMART pero no nos hagamos problema.
Cerramos entonces el WDASM y abrimos el otro WDASM el que esta modificado para VISUAL BASIC según vimos en la lección sobre VISUAL y allí lo abrimos y lo desensambla perfectamente y aparecen las STRING REFERENCES lo que indica que no está comprimido.
Si ejecutamos el programa vemos que se limita luego de algunos usos y al arrancar aparece una ventana donde haciendo clic en REGISTRATION aparece donde ingresar la clave.
Si ingreso un número cualquiera me sale el siguiente cartelito "INVALID REGISTRATION NUMBER"
Vamos al WDASM y nos fijamos en las STRING REFERENCES y ALLI ESTAAAA, hacemos click encima de INVALID REGISTRATION NUMBER y el WDASM nos marca un solo lugar 43D1CA y allí vemos la mención al cartel podrido y mas arriba vemos que viene de dos lugares:

**REFERENCED BY A UNCONDITIONAL O CONDITIONAL JUMP AT ADDRESS**

***43D043 y 43D050***

Que parecen ser dos comparaciones una después de otra, y en cualquiera que salta a esta zona de chico malo nos aparece el cartelito maldito, habrá que evitar el cartelito maldito.
Vamos a ver en el WDASM que hay en 43d043 , En GOTO - GOTO CODE LOCATION, ponemos 43DO43 y nos muestra

**43D043 0F8481010000 JE 0043D1CA**

Aquí vemos el salto crítico y dos renglones más abajo hay otro que es el de 43d050 que nos tira a chico malo también.

Si arranco el programa y voy a la ventana de registro y pongo cualquier numero y antes de ACEPTAR entro en el SOFTICE y pongo BPX HMEMCPY y hago clic en REGISTER, rompe en el SOFTICE pero si quiero volver al ejecutable con F12 hago como treinta veces F12 y nunca aparezco en el ejecutable, me aparece el cartelito de INVALID, como entro en el ejecutable para poner los BPX en los puntos que ya hallamos (43D043 y 43D050)?
Vuelvo a poner el BPX HMEMCPY y cuando rompe en el SOFTICE hago F12 hasta que llego al archivo de VISUAL BASIC MSVBVM60 y a partir de allí tengo apretada F10 hasta que me aparece en la línea verde ATAMA, que es cuando llegamos al ejecutable, ufff.

Una vez allí borro los BPX viejos con BC* y pongo BPX 43D043 y BPX 43D050 y hago X, parara en 43D043, si no para porque se habían pasado de esa sentencia y vuelven a poner cualquier numero en la ventana de registro y ahora si parara allí en 43D043.
Vemos que el SOFTICE indica que en el salto 43D043 va a saltar a la ZONA DE CHICO MALO (JUMP) o sea que podemos hacer **r eip=43D056** así salteamos los dos saltos que nos llevan a CHICO MALO, hacemos X y vemos que nos sale el cartelito **DEMO HAS BEEN CONVERTED TO SHAREWARE** o sea que el programa dejo de ser una versión de prueba, que bien.

Vemos que si arrancamos de nuevo el programa aparece la ventanita de REGISTRACIÓN pero ya no cuenta las veces que faltan para que caduque si salteamos la ventana de REGISTRACIÓN arranca el programa normalmente, o sea que funciona y no vence, aunque queda pendiente eliminar la ventana molesta del inicio, lo cual dejaré para que ustedes practiquen, el que pueda hacerlo, escríbame a mi email un pequeño informe de cómo lo hizo y pasara a engrosar la nueva sección de la pagina WEB del curso, llamado **LOS TUTORIALES DE LOS LISTEROS**, el primero que me lo mande y la solución sea correcta , su tutorial va a ser el que va a aparecer ya que no podemos poner a todos.

Aquí está el parche hecho en el RISC PROCESS PATCHER para que no caduque más, (LA ELIMINACIÓN DE LA VENTANA NO ESTA AQUÍ EN ESTE PARCHE)

```
;ATAMA

F=AtamA.exe:            ; PROCESS TO PATCH

O=crack_the_AtamA.exe:  ; LOADER TO CREATE
P=43d043/0f,84/EB,11:   ;

$ ;end of script
```

Hasta la próxima…

**Ricardo Narvaja**

# OTRO EN VISUAL BASIC PERO MAAAS DIFÍCIL

## (LECCION DIECINUEVE)

Este programa llamado Registry Compare 1.22 se baja de

http://www.kmcsonline.com/rcompare.htm

Es un programa que permite tomar como una instantánea del registro y después de instalar un programa volver a tomar otra para ver cuáles fueron las modificaciones en el mismo, yo para eso uso el programa ART ADVANCED REGISTRY TRACER que es más detallado y te dice que entradas fueron agregadas, cuales fueron borradas y en cuales fueron cambiados algún valor.
El ART se baja de
http://www.elcomsoft.com/art.html
y el crack de ASTALAVISTA y algunos se preguntaran porque no crackeamos el ART, en vez del Registry Compare, lo que pasa es que el ART ya lo crackee y no tiene algo interesante para enseñar pero el REGISTRY TRACER nos puede enseñar dos o tres cosas, o sea vamos a usar el ART para detener el conteo de 14 veces de prueba del REGISTRY COMPARE dándole un trago de su propia medicina, o sea con el mismo método que utiliza el programa. (14 veces para probar un programa, MISERABLE. ja,ja)
Primero bajo el ART y lo instalo, bajo el crack de ASTALAVISTA y una vez que hice todo eso, tomo la primera foto del registro antes de instalar el REGISTRY COMPARE o sea en el ART voy a SCAN REGISTRY y me toma una instantánea con fecha y hora del registro.
Una vez que termino eso, instalo el REGISTRY COMPARE y una vez instalado, lo ejecuto una vez y lo cierro, supuestamente ya se crearon las entradas en el registro donde guarda la información de las veces que lo usaste, asi que vuelvo a abrir el ART y de nuevo SCAN REGISTRY y una vez que termina, marco la primera de las dos fotos que tome , para que compare a partir de allí, y en el menú elijo COMPARE FROM HERE o sea COMPARE DESDE AQUÍ.
Como ahora queremos ver las nuevas entradas que creó el programa al instalarse, entonces vamos a la pestaña ADDED que son las entradas agregadas para ver qué hay de nuevo en el REGISTRO después de haber instalado el REGISTRY COMPARE.
Vemos que en added hay tres nuevas entradas. Las tres nuevas entradas son:

**1)HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall\KMCS Registry Compare_is1**
**2)HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\MenuOrder\Start Menu\&Programs\KMCS Computer Software**
**3)HKEY_USERS\.DEFAULT\Software\VB and VBA Program Settings\KMCS\REGCOMPARE**

Bueno ahora voy a INICIO - EJECUTAR -REGEDIT y allí busco cada una de las cadenas estas y cuando están completamente abiertas y muestran los datos en la parte derecha, entonces voy al menú REGISTRO y allí EXPORTAR ARCHIVOS DEL REGISTRO y exporto un archivo por cada entrada que agrego el programa las copio en un lugar que tenga a mano.

Ejecuto un par de veces el REGISTRY COMPARE para que consuma un par de veces ahora me dice que me quedan doce veces para utilizar.
Cierro el programa REGISTRY COMPARE y ejecuto uno por uno los archivos que cree del registro para inyectar los datos como estaban originalmente, al ejecutar me dice si quiero ingresar datos al registro y pongo que si, arranco el REGISTRY COMPARE y me dice que me quedan 14 veces para probar, juuuuy, ya descubrí como volverlo para atrás, jajaja.
Esto es muy importante porque me da tiempo para descubrir el crack y si quiero lo puedo seguir utilizando así, ya que cuando llegue a que me quedan cinco veces vuelvo a ejecutar los tres archivos del registro y me vuelve la cuenta atrás a 14 de nuevo, iuuupi.
Ahora podemos trabajar tranquilos pero no voy a abundar mucho en como crackear esto, ya que para mí lo importante era que vean como se hace en programas que vencen después de varias veces y guardan la información en el registro.
Si por esas casualidades (QUE NO ES ESTE CASO), al realizar estos pasos con otro programa y restaurar el registro a los valores que tenia al momento de instalar el programa este no vuelve atrás es posible que guarde la información en un archivo, ejemplo de lo cual veremos más adelante, pero les adelanto que utilizando el FILE MON vemos que filas usa y las copiamos en otro lado y cuando usamos un par de veces el programa, volvemos a copiar las filas esas que guardamos en algún lugar encima de las del programa y podemos lograr que vuelva atrás y si utiliza una fila en C también la podemos detectar con el FILEMON y reemplazarla por una copia después de usar el programa un par de veces.

Bueno si arrancamos el ejecutable COMPARE.EXE con el SYMBOL LOADER y cuando empieza vamos traceando con F10, y no salteamos ningún CALL, si encontramos uno entramos con T y seguimos traceando, enseguida nos vamos a encontrar que estamos en el archivo de VISUAL BASIC VB40032 , ahora viene el truco de magia ponemos BPX 0f79b358 y hago correr el programa con X, cada vez que para allí hago D EDI o D ESI y me dar la clave., si llego a la ventana de registro y no me dio la clave sigo, pongo una clave falsa y acepto y una o dos veces después que paro allí me dio mi clave RCEQ70NS en formato ancho, o sea con puntitos entre las letras pero en la ventana de registro hay que escribirla normal.
COMO DESCUBRI ESTO?
Ja ja, es difícil explicar las mil vueltas que tuve que dar para llegar a esto pero ahí van varios datos para el que quiera marearse un poco.
El programa a pesar de ser de VISUAL BASIC al estar escrito en P-CODE no es mucho lo que se puede ver con el SMART CHECK ya que este avisa que con archivos en P-CODE no son muchos los datos que suministra para crackear.
Yo lo hice y me costo mucho con el método de HMEMCPY una vez que escribí la clave falsa puse aceptar y allí entre en el SOFTICE y seguí según el método que vimos en las lecciones pasadas con la diferencia que aquí en vez de d ds:esi o es:edi hay que utilizar ds:si o es:di ya que según vemos estamos en HMEMCPY de 16 bits en vez de 32 pero es igual después hacemos PAGE el valor que nos aparece la cadena 989898 (QUE ES LA QUE YO PUSE) y una vez que localizo el valor donde se encuentra empiezo a poner BPR siempre y cuando mueve la cadena a otro lugar también BPR y cuando pasa con MULTIBYTETOWIDECHAR el valor a formato ancho y descubro donde lo guarda , allí también un BPR y cada tanto hago un

S 0 L FFFFFFFF '989898"  y
S 0 L FFFFFFFF '9.8.9.8.9.8'

Y así llegue a  que en un momento (DESPUES DE BASTANTE TRABAJO) paro en 0F79B358 REPZ CMPSW
QUE ES UNA SENTENCIA DE COMPARACIÓN QUE EN D ESI y en D EDI están las claves falsa y buena en formato ancho.
Así que copie mi clave que decía en el SOFTICE **R.C.E.Q.7.0.N.S** y la escribí en la ventana para REGISTRARME todo corrido **RCEQ70NS** y voila jitanjafora, estoy REGISTRADO bastante trabajo mediante, jaja,
SI A USTEDES NO LES SIRVE MI CLAVE ES PORQUE USA UNA DIFERENTE PARA CADA COMPUTADORA PERO LA PUEDEN ENCONTRAR FÁCILMENTE COMO LOS INDIQUE.
No importa aquí mucho como hallar el crack pero si el método de detener el conteo, por medio de las fotos del registro es lo que vale, ya que el crack es cansador por lo vueltero pero con paciencia se encuentra, por eso no insistí mucho con eso y si con el método de las fotos con el ART y ahora también lo podemos hacer con el REGISTRY COMPARE.

**Ricardo Narvaja**

# COMO QUITAR EL CARTELITO MOLESTO Y REGISTRAR EL ATAMA

## (LECCION VEINTE)

Sinceramente esta semana no pensaba realizar ninguna lección dado que tuve muchísimo trabajo y no tuve tiempo, pero dado que no hubo soluciones sobre como quitar el cartelito del ATAMA de la LECCIÓN 18 vamos a quitarlo y de paso REGISTRARNOS, en una lección breve, ya que no tuve mucho tiempo esta semana.

Dado que el ATAMA no tiene trucos ANTISOFTICE lo arranco desde el SYMBOL LOADER y realizo este procedimiento:

Apenas arranca encuentra un CALL que si lo ejecutamos arranca el programa y no para mas, volvemos a arrancarlo y entramos dentro de ese CALL, y así seguimos traceando F10 y tomamos la precaución de antes de pasar por encima de un CALL poner un BPX allí cosa de que si arranca el programa, podemos reiniciar y parar allí, realizamos este procedimiento y cuando encontramos un CALL que arranca el programa y nos hace aparecer el cartelito maldito, volvemos a arrancar y como habíamos puesto un BPX vuelve a parar allí y ahora en vez de pasar por encima del CALL entro con T y sigo traceando con F10, cada CALL que encuentro antes de pasarle por encima borro todos los BREAKS anteriores con BC* y pongo un BPX en el ultimo CALL este, si paso por encima y no sale el cartelito maldito, sigo hasta el otro CALL, y allí repito el procedimiento, entrando con T en los CALLs que me hacen aparecer el cartelito y salteando los CALLS que cuando los paso por encima no paso nada.

Así me voy acercando cada vez mas al lugar donde el cartel principal de ATAMA tenga un CALL y el cartelito maldito otro CALL separado esto ocurre en:

417FA7 CALL [EAX+2A0]  (CARTEL ATAMA)

4180A9 CALL [EDI+2B0]  (CARTEL MALDITO QUE HAY QUE ELIMINAR)

Es seguro que en el programa registrado el primer cartel que dice ATAMA aparece y el segundo cartel no, por lo tanto entre los dos CALLs debe haber una comparación y un salto que evita el segundo cartel y que puede registrarnos ya que seguimos el camino del CHICO BUENO en ese salto.

La comparación y el salto están a partir de 417FCF ya que allí compara si el valor que hay en 459090 es cero y según esa comparación salta evitando el cartelito molesto.

Puedo probar a ver qué pasa si coloco a mano en 459090 un uno haciendo d 459090 y después E con lo que puedo escribir el primer  número y cambiarlo de 00 a 01, quedándome ahora el valor de 459090 en uno, ejecuto la comparación y el salto y arranca el programa sin aparecer el cartelito maldito y cuando voy a ABOUT me dice REGISTERED, jaja, pudimos con él.

Para asegurarme que en 459090 quede el valor uno que me asegura que este registrado voy a cambiar la sentencia

**417FCF CMP WORD PTR [459090],00**

O sea la sentencia que compara si estas registrado por

**417FCF MOV WORD PTR[459090],01**

Lo que va a colocar en 459090 el valor uno para que quede registrado y ahora modifico el salto que viene después para que salte siempre pongo

**417FD8 JMP 418117**

Y listo… Puedo ir al Ultra Edit, buscar la cadena y cambiarla sin miedo ya que este programa no está protegido contra esos cambios.

**66833d90904500000f853a010000** lo reemplazo por

**66c705909045000100e93a010000**

Y listo, queda registrado para siempre y no sale más el cartelito molesto. Hasta la próxima…

**Ricardo Narvaja**