aRC-FL-Cracking 004 (01/08/2003)

¡Lo logré! (No, aún no)

(por Furious Logic [aRC])

Advertencia

Antes de poner en práctica el presente documento y cualquier material asociado al mismo, sea éste naturaleza tangible o intangible usted debe estar totalmente de acuerdo con todos los términos y condiciones siguientes:

Del software

Cualquier software se proporciona tal como está, sin ninguna garantía expresa ni implícita de ningún tipo.

aRC no garantiza ni asume responsabilidad alguna en cuanto a la integridad o exactitud de cualquier información contenida en el software.

Ni los miembros ni los colaboradores ni los invitados aRC se hacen responsables por el uso que se le pueda dar al software.

Al examinar, copiar, ejecutar, instalar o utilizar el software, el lector está aceptando su total conformidad con todos los términos y condiciones enunciados.

Del documento

Al abrir este documento, el lector acepta incondicionalmente su total y exclusiva responsabilidad legal, de acuerdo a las leyes vigentes en su país, por el uso de las técnicas experimentales, educativas y/o de investigación aquí vertidas en materia de programación especializada de computadoras.

En caso de discrepar con alguno de los puntos descritos, deberá eliminar inmediatamente el presente documento y todo material asociado al mismo.

Agradecimientos

Simplemente somos usuarios felices cuando empleamos Lotus Word Pro de Lotus Corporation (TM), incluido en la suite de oficina Lotus Millenium.

A FontLab 3.00F de FontLab Developers Group por permitirnos asignar permiso completo a las fuentes true type protegidas contra copia y a Acrobat Distiller 5.0 de Adobe Systems por su excelente resultado en la creación del documento electrónico en formato PDF.

Objetivos

Aplicar el método el método de investigación enseñado en la lección anterior, en la eliminación del nag presentado en esta lección.

Acercar al lector al fin de su primer crack.

Preparar las primeras interrogantes que servirán de base para la sustanciosa lección siguiente.

Enseñar a eliminar un nag.

Teoría

Keyfile

(1) Se traduce como "archivo clave".
(2) Archivo físico que contiene información, codificada o no, acerca del registro correcto/incorrecto de un programa determinado.
(3) En ocasiones la simple presencia de un archivo tal, se asume como registro satisfactorio.

Múltiplos y súbmúltiplos

(1) Se denomina múltiplo al prefijo que simboliza un factor por el que se multiplica a la unidad a fin de indicar un valor determinado.
(2) A continuación la tabla de prefijos SI (*Internacional System of Units*, "Sistema internacional de unidades"):

NOMBRE	SIGNIFICADO		EN BYTES		
	SÍMBOLO	FACTOR			
yotta	Y	$\times10^{24}$	1 YB	=	$1^{4}000,000^{3}000,000^{2}000,000^{1}000,000$
zexa	Z	$\times10^{21}$	1 ZB	=	$1,000^{3}000,000^{2}000,000^{1}000,000$
exa	E	$\times10^{18}$	1 EB	=	$1^{3}000,000^{2}000,000^{1}000,000$
peta	T	$\times10^{15}$	1 PB	=	$1,000^{2}000,000^{1}000,000$
tera	T	$\times10^{12}$	1 TB	=	$1^{2}000,000^{1}000,000$
giga	G	$\times10^{9}$	1 GB	=	$1,000^{1}000,000$
mega	M	$\times10^{6}$	1 MB	=	$1^{1}000,000$
kilo	k	$\times10^{3}$	1 kB	=	1,000
hecto	h	$\times10^{2}$	1 hB	=	100
deka	da	$\times10^{1}$	1 daB	=	10
deci	d	$\times10^{-1}$	1 dB	=	No aplicable
centi	c	$\times10^{-2}$	1 cB	=	No aplicable
mili	m	$\times10^{-3}$	1 mB	=	No aplicable
micro	µ	$\times10^{-6}$	1 µB	=	No aplicable
nano	n	$\times10^{-9}$	1 nB	=	No aplicable
pico	p	$\times10^{-12}$	1 pB	=	No aplicable
femto	f	$\times10^{-15}$	1 fB	=	No aplicable
atto	a	$\times10^{-18}$	1 aB	=	No aplicable
zepto	z	$\times10^{-21}$	1 zB	=	No aplicable
yocto	y	$\times10^{-24}$	1 yB	=	No aplicable

Posiblemente el lector se esté preguntando por qué **1 kb** es **1,000 bytes** y no es **1,024 bytes**.
El SI ha establecido esta primera tabla como los 20 prefijos que pueden ser utilizados con cualquier unidad de medida, pero estrictamente representativos de potencias de 10. Es un error decir **1 kB** = **1,024 bytes**, pues **1 kB** es **1,000 Bytes**.

(3) Tabla de prefijos para múltiplos binarios:

NOMBRE	SIGNIFICADO		EN BYTES		
	SÍMBOLO	FACTOR			
kibi	Ki	$\times2^{10}$	1 KiB	=	1,024
mebi	Mi	$\times2^{20}$	1 MiB	=	$1^{1}048,576$
gibi	Gi	$\times2^{30}$	1 GiB	=	$1,073^{1}741,824$
tebi	Ti	$\times2^{40}$	1 TiB	=	$1^{2}099,511^{1}627,776$
pebi	Pi	$\times2^{50}$	1 PiB	=	$1,125^{2}899,906^{1}842,624$
exbi	Ei	$\times2^{60}$	1 EiB	=	$1^{3}152,921^{2}504,606^{1}846,976$

Aprobado en 1998 por la IEC (*International Electrotechnical Commision*, "Comisión internacional de electrotecnia"), como nombres y símbolos standard internacionales para utilizarse como múltiplos binarios en el campo del procesamiento y transmisión de datos.

Es decir, cuando nos referimos a **1,024 Bytes**, debiéramos escribir **1 KiB** que se pronuncia **un kibiByte**.
Por lo tanto, si **1 KB** = **1,000 Bytes** y **1 KiB** = **1,024 Bytes** entonces **1 KB ≠ 1 KiB**.
Para mayor información consulte *The NIST Reference on Constants, Units, and Uncertainty* en: http://physics.nist.gov/cuu/Units/index.html

Historia de un cracker

Los libros de biografías de grandes personajes, no necesariamente relacionados con la informática, engrandecerán nuestra ideología. Recomendamos al lector, estudiar las biografías de los personajes que arman nuestra historia universal en todas las ramas del conocimiento, manteniendo siempre nuestra mente abierta y sin ideas prejuzgadas al leerlas, para aprender de los aciertos y errores, decisiones e indecisiones, conceptos y prejuicios, avances y retrocesos, justicias e injusticias, envidias y venganzas que plasmaron las vida de los notables.

El siguiente extracto es nuestra traducción no oficial de la biografía de The Keyboard Caper (tKC), uno de los más grandes under que hayamos tenido en el underground y uno de los fundadores de Phrozen Crew. Prueba viviente de que sí se pueden conjugar al hacker, al phreaker y al cracker, en un solo individuo con total armonía. Estos eran los tiempos en que no se calificaba a los individuos simplemente por sus funciones, sino que los individuos desempeñaban diversas funciones según les eran necesarias, sin por ello dejar de lado las demás.

Invitamos y motivamos al lector a sumarse con entusiasmo contagiante, a esta fabulosa comunidad de aprendizaje tecnológico, la comunidad de los hackers/phreakers/crackers, la comunidad del underground o *The Scene*.

La rápida y sucia historia detrás del grupo
(Gracias a The Keyboard Caper por proporcionar una biografía)

Phrozen Crew, es conocido por muchas personas simplemente como PC. Es el más grande grupo de cracking en Internet, aunque no es el más antiguo. Con más de 45 miembros y más de 4,500 cracks, ningún grupo llega a acercarse a los logros de PC. Sin embargo, PC no es un simple grupo de cracking nacido un día cualquiera. PC tiene una historia y montones de experiencias. The Keyboard Caper, uno de los fundadores de Phrozen Crew es una persona realmente sorprendente.

Creció en Sudáfrica y aunque era sordo, emergió entre las jerarquías de hackers y crackers luchando a su modo para ser respetado por muchos. El hecho más curioso es que **PC empezó a partir del hacking**. tKC comenzó aprendiendo de los hackers. Aquél que hizo que se abriese paso entre ellos, llevaba el nick Blade Runner, quien le enseñó a ser hábil con las computadoras. Esto no lo influyó, sino que fue hasta que empezó a usar juegos de computadora que se topó con una protección contra copias y se dijo a sí mismo: "¿por qué no crackeo esos juegos?". Y lo que iniciaría el legado de PC, sería el momento cuando tKC crackeó su primer programa, un juego para D.O.S., que también lo llevó a bautizarse a sí mismo con el apelativo "The Keyboard Caper".

En 1991 escuchó acerca del escenario BBS. Las BBS le mostraron cuán grande era el hacking y cómo diversos cracks ya habían sido hechos por muchos crackers. Es entonces que juntó todos los cracks que pudo conseguir y aprendió de ellos los refinados propósitos del cracking. Muchos de los cracks fueron hechos por el grupo de cracking RAZOR 1911. Así que tKC se inició crackeando más juegos y utilitarios para D.O.S., mayormente de tipo shareware, descargado de las BBS. Tan pronto como hizo y grabó muchos cracks, las personas empezaron a notarlo. Fue entonces que el mismo Blade Runner, le habló de otro grupo de cracking en Sudáfrica denominado Masters of Terror (MoT). Masters of Terror no estaba enlazada por las BBS locales, debido a que parecían infundir pánico entre los usuarios. Blade Runner informó a tKC que MoT era conocido por publicar programas ilegales, claves para registrarse, etc., lo que llevó a tKC a decir: "Recuerdo cuánta gente les teme..."

Blade Runner le preguntó cómo logró vencer su propio juego. A lo que tKC respondió "cómo había hecho eso". Blade Runner le dio las pautas e información necesarias para unirse a MoT. tKC exploró las BBS inactivas y eventualmente descubrió el cuartel general de la BBS de MoT, la cual guardaba un valiosísimo secreto en ella. Bajo un nombre falso, tKC fue capaz de conectarse a ese sistema en el que se hizo amigo de los miembros de MoT. Lo que encontró fue que MoT realizaba llamadas transoceánicas, hecho que tKC consideraba imposible hasta ese momento. Y muchas de esas llamadas internacionales eran de los miembros de RAZOR 1911. En la BBS de MoT él se reunió con Midi Maniac, que era un distribuidor de RAZOR para Sudáfrica. También fue en MoT donde tKC se encontró con Aphex Twin (también conocido como Kokey). Ambos vivían en Sudáfrica. En esa misma

época, se reunió con un muy buen hacker llamado Acid Phreak, que sin saberlo, realmente era Aphex Twin. Acid Phreak preguntó: "¿Por qué no creas tu propia BSS?" y fue así que tKC trajo al mundo a la BBS Wolfenstein. Lo que era especial acerca de la BBS de tKC, fue que era el único SYSOP sordo en Sudáfrica, hecho que ocultaba de los demás. La BBS de tKC creció hasta ser la más grande en Sudáfrica, almacenando una enorme cantidad de cracks y los mejores programas shareware. Se inició con 20 a 30 llamadas diarias. tKC era conocido como Dr. Wolfen. Él no deseaba que supieran que era tKC, así que simplemente continuó como Dr. Wolfen. Varias personas que lo contactaron como Dr. Wolfen, le preguntaban cómo comunicarse con tKC, pero a pesar de todo el seguía diciendo: "¡Él es un gran amigo mío!". Su BBS llegó a ser muy famosa. Muchas personas llamaban. tKC aún no sabía qué hacer de sí mismo.

Los miembros de MoT llamaban y saboteaban la BBS de tKC hasta que, finalmente, luego de múltiples daños a su BBS, se preguntó a sí mismo: "¿Por qué yo no puedo hackear sus BBS?". Esto hizo que el sistema de MoT dejara de ser el único sistema basado en UNIX en toda Sudáfrica. Al comienzo, UNIX era difícil de hackear. Sin embargo, luego de un corto tiempo tKC aprendió UNIX y a hackear su BBS. Conectándose como tKC, causó el caos en el sistema MoT cuando envió un correo masivo a todos los integrantes de la BBS diciendo que era el primero en hackear un sistema UNIX. MoT respondió cobardemente distribuyendo un programa DEMO que decía que tKC era un mal hacker/cracker y revelando su identidad como Dr. Wolfen. Las personas preguntaban a tKC si él realmente era Dr. Wolfen y también tKC. Pero como Dr. Wolfen, solamente les dijo que tKC era un gran amigo suyo, lo que hizo que MoT desapareciera súbitamente. Acid Phreak le habló a tKC mucho acerca de MoT, pues desde la desaparición de este grupo, tKC no molestó nunca más a Acid Phreak por causa de MoT.

Acid Phreak fue también la persona que enseñó a tKC a realizar llamadas telefónicas gratuitas, incluyendo cómo realizar llamadas al exterior de Sudáfrica. Razón por la cual tKC empezó a llamar gratuitamente a Estados Unidos. También le enseñó a llamar a muchos BSS de ultramar. Posteriormente, Acid Phreak reveló que también era conocido como Aphex Twin y también como Kokey, lo que sorprendió a tKC porque Acid Phreak era el rey del phreak en Sudáfrica. Entonces, tKC se reunió con una persona apodada Snake. Snake era un pirata de Sudáfrica que subía diversos juegos a la BBS de tKC. La BBS de Snake también fue la primera y única BBS que distribuía juegos en Sudáfrica. Posteriormente, tKC se involucra con un programa llamado Vision-X. Vision-X era un programa BBS hecho por un canadiense de nick Ryec. Vision-X incrementó las llamadas a la BBS de tKC. En ese momento, Midi Maniac y Aphex Twin llamaron a la BBS de tKC descubriendo que él utilizaba Vision-X. Ellos estaban molestos porque tKC lo utilizaba para una BBS pública. Deseaban saber de dónde obtuvo el programa. tKC les dijo de dónde consiguió Vision-X y porqué deseaba usarlo, pero ellos le dijeron que lo eliminara de su BBS. Llegaron al punto de amenazarlo, exigiéndole que lo elimine de su BBS y le dieron una semana de plazo para regresar al programa que utilizaba antes. Previamente, tKC había utilizado RA. Decidió no hacerles el menor caso y mantuvo Vision-X en línea. Así, después de una semana, ellos llamaron nuevamente a la BBS de tKC y se enfadaron nuevamente; a pesar de esto, simpatizaron con tKC y supieron lo que él podría hacer. Nada sucedió porque temían que la policía arruinara a tKC por el uso ilegal de software BBS. Pero en cambio, Midi Maniac y Aphex Twin ayudaron a tKC a hacer de su BBS, la mejor BBS pública del underground con una de las mayores colecciones de cracks, herramientas hacking y utilitarios phreaking. Por muchos meses tKC crackeó diversos programas y ayudó a los demás a crackear otros programas shareware. tKC también consiguió que muchos hackers y crackers se conectaran a su BBS. Es ahora, cuando este relato nos lleva al **inicio de Phrozen Crew**.

En algún momento en 1993, Aphex Twin contacta a tKC para unirse a Phrozen Crew, cuando todavía era un grupo de hacking. Era un grupo underground elite con más de 30 miembros. tKC se unió a PC como un artista ANSI y programador de demos. En esa época, fue conocido como The Resetter. Durante ese tiempo tKC aprendió mucho más y realizó muchas más llamadas al extranjero. Hizo muchos amigos en RAZOR 1911, ACiD, Hybrid, etc. tKC también se reunió con Ryec, creador de Vision-X. Ryec le dio actualizaciones gratuitas de la nueva versión de Vision-X que era una versión registrada. tKC incluso intentó hackear Vision-X, aún mientras pensaba en que Ryec le advirtió de los daños que podría causarse, lo que aprendió por el camino difícil. Ryec le dio a tKC el código fuente en Pascal de Vision-X para mostrarle porqué los hackers y crackers no podían con Vision-X. tKC podría tener y tuvo el conocimiento para crackear Vision-X, pero mantenía una sincera amistad con Ryec que le hacía respetar el esforzado trabajo de Ryec. Súbitamente y sin ningún aviso o razón, en 1994, el grupo Phrozen Crew llegó a su fin. Por lo cual, tKC continuó crackeando programas bajo su nombre.

tKC nunca había creado un grupo. Él solo deseaba hacerse de un gran nombre por sí mismo. Sin embargo, pocos meses después, pensó que podría ser mejor si empezaba un grupo. tKC consultó con Aphex Twin si podría utilizar Phozen Crew y revivirlo como un grupo de cracking en lugar de un

grupo de hacking. **Más tarde en 1994, Phrozen Crew fue revivido por tKC**, Lucido como artista ANSI, Psylocke como programador de demos y Mutha como músico. PC hizo muchos cracks, demos, gráficos ANSI, gráficos VGA y eventualmente se convirtió en el mejor grupo de cracking en Sudáfrica, derrotando a todos los otros grupos de su país.

Aún así, tKC deseaba más y en abril de 1996 escuchó comentar bastante sobre Internet. De esta manera, empezó consiguiendo muchos cracks de UCF. Leyendo un UCF.NFO él aprendió cómo contactarse con UCF utilizando el IRC. Así que realizó su primer intento en el IRC y se reunió con miembros UCF. En ese momento tKC solo se preguntaba si podría dejar PC y unirse a UCF, pues en esa época, UCF era el grupo cracking más grande del mundo. Luego de unas pocas semanas, Psylocke llamó la atención de tKC, sugiriéndole que iniciara su propio canal IRC. Así fue creado #pc96 en el IRC, aunque no fue muy conocido por el público y tKC gastó la mayor parte del tiempo conectado al canal #cracking. tKC invirtió el tiempo ayudando y crackeando para otros grupos. tKC se unió a varios grupos de cracking para los que realizó varios cracks, no obstante aún realizaba cracks para PC, lo que no preocupaba mucho a los otros grupos. tKC consultó entonces con el líder de UCF, en ese entonces Marquis, si podría distribuir cracks de UCF en su propio site. También contactó con diversos grupos de cracking para ver si podría distribuir sus cracks en su site. La BBS de tKC aumentó enormemente de tamaño con toneladas de cracks en línea. Ello hizo muy felices a los habitantes de Sudáfrica.

tKC fue el único SYSOP que tenía 27 sites de distribución y 59 conferencias en línea, por lo que logró más de 250 llamadas telefónicas diarias. Aprendió como crackear programas para Windows utilizando programas como SoftIce y aprendiendo cómo depurar. De súbito, como la teoría de Big Bang, los usuarios del IRC giraron su atención hacia tKC. Con la ayuda de varias personas, tKC creó el canal PC en el IRC. Invitó a varios a unirse a PC. tKC llegó a decir: "Si no fuera por personas como TheBomb, ThatDude y MegaByte, PC no podría haber llegado tan lejos como lo ha hecho". Alien sugirió el uso de bots para el canal PC y después de muchos meses de arduo trabajo, **Phrozen Crew llegó a ser el más grande y el mejor grupo de cracking de Internet y del mundo**.

Phrozen Crew es el nombre que será conocido como el mejor por un largo, largo tiempo, si no es que por siempre. Y esto es lo que tKC deseaba, ser conocido como alguien importante.

Poco después, tKC conoció a una señorita, que últimamente lo obligó a dejar PC y el IRC. Ella le enseñó a tener una vida y a pesar de que piensa que tKC ya no está relacionado directamente con nosotros, él aún continua apareciendo en el canal PC del IRC.

Actualmente, Phrozen Crew es conocido como PC98 y usted puede contactarlo en el canal #PC98 del IRC Efnet.

"Nosotros SIEMPRE conseguimos lo que deseamos".

10 de junio de 1998

¡Lo logré!

Después de la inspiradora historia de The Keyboard Caper (tKC) y llenos de esos incontenibles deseos de: (1) Ser los mejores comparados; y de (2) Ayudar permanentemente a los demás, continuaremos con el crack a medio terminar que dejamos en la lección anterior.

Quedaron tres inconvenientes por resolver:

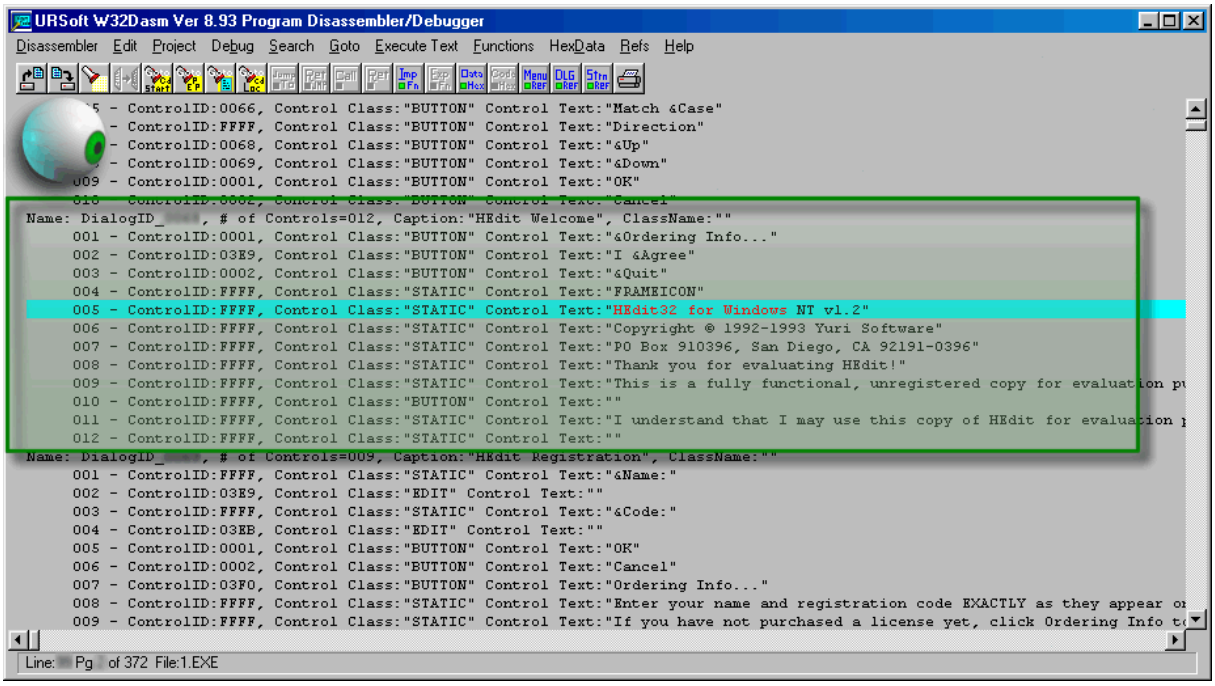
1. El nag sigue fastidiando.
2. Al volver a cargar el programa e ir a **Help, About** dice: **Unregistered copy...** (¿?)
3. Cuando se ingresa un nombre y código que sí son válidos, el programa dice que el código no es correcto.

Nos encargaremos del 1ro. y 2do. dejando el tercero pendiente. Consideramos innecesario repetir las herramientas que deben acompañar a todo under como usted, en sus prácticas de cracking.

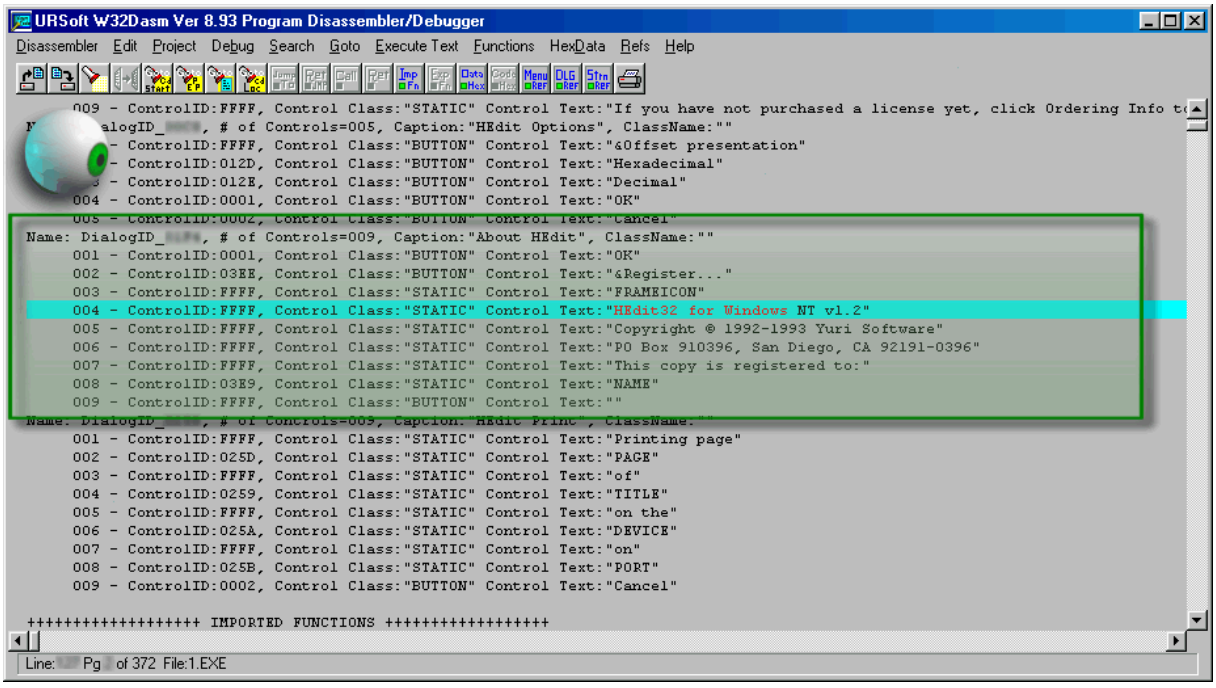
1. De acuerdo a las notas que bosquejamos en la clase anterior, en el punto 7 de la sección Tiempo de cracking, frecuentemente, solo era necesario anotar las primeras palabras de un nag. Es decir:

   HEdit32 for Windows

2. Desensamblamos 1.exe en W32Dasm (W32). Si no recuerda, revise la lección anterior, pHC-Cracking 003 Tiempo de cracking.
3. Dentro de W32 seleccione el menú **Search**, opción **Find text**. En el diálogo que aparece escribimos **HEdit32 for Windows** sin preocuparnos por las mayúsculas o minúsculas y aceptamos.
4. Mucho ojo a la primera coincidencia:



Y también a la segunda ocurrencia cuando presionamos **Find next**:



Por lo pronto, estas 2 ocurrencias se encuentran en una sección que W32 denomina:

```
+++++++++++++++++ DIALOG INFORMATION ++++++++++++++++++
```

Podemos encontrarlo varias líneas arriba en la pantalla de W32. Precisamente se trata de la definición del contenido de los diversos cuadros de diálogo que serán empleados por el programa, posiblemente a través de la **API DialogBoxParam** de la librería **user32.dll**, pero eso no debe importarnos por ahora. Lo que sí debe importarnos y debemos aprender, son las secciones que

hemos enmarcado con un recuadro de color en ambas coincidencias. Tienen una cabecera con un formato común:

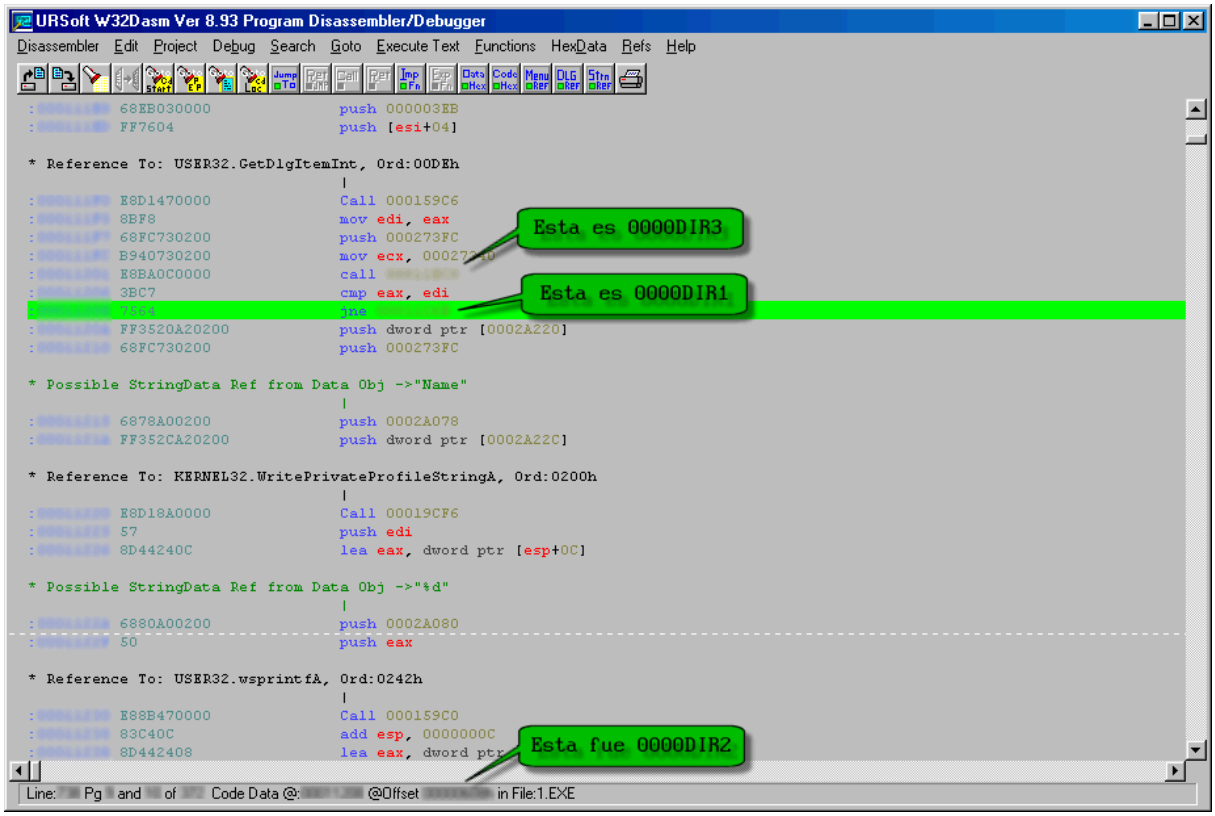
```
Name: DialogID_nnnn, # of Controls=012, Caption:"HEdit Welcome", ClassName:""
```

En donde **nnnn** representa un número y **DialogID_nnnn** el nombre que W32 le ha asignado para identificarlo. Su título es **HEdit Welcome** como podemos verificar también en el propio nag. Contiene **12** controles, todos de tipo etiquetas de texto. No se le asignado ningún nombre de clase. De haberse hecho así, estaría disponible para su reutilización, pero esto es asunto de programación ensamblador que lo invitamos a investigar.

5. Volvamos a seleccionar el menú **Search**, opción **Find text**. Solo que esta vez escribiremos el nombre del diálogo que estamos buscando, que para propósitos de la presente lección denominaremos **DialogID_NUM1**. Cuando aceptemos, encontraremos **10** ocurrencias que podríamos analizar una por una, pero eso tomaría algo más de tiempo y la especialidad del cracking, como toda especialidad del underground, requiere de sentido común, deducción e intuición.
6. Sabemos, por la lección anterior, que en **0000DIR1** existe una bifurcación que genera el mensaje de código de registro erróneo. Un par de líneas antes de ella, se encuentra un **call** que podría ser un procedimiento verificador del serial. Este orden en la codificación, nos referimos a la combinación **call–comparación–bifurcación**, se emplea con diferentes propósitos:

   a. Verificar el nombre y código al momento de registrarse.
   b. Determinar si las funciones restringidas se pueden activar.
   c. Condicionar la presentación de un nag.
   d. Activar o desactivar un temporizador de autoverificación cada **n** segundos durante la ejecución de un programa.
   e. Crear/eliminar un archivo físico de registro/licencia.

7. Muy bien, ahora apliquemos lo recién aprendido. Suponemos que, a menos que usted tenga el cerebro de un antediluviano, estará anotando ordenamente todos sus nuevos conocimientos, ¿o acaso espera que le dicten?. Estamos en **0000DIR1**:



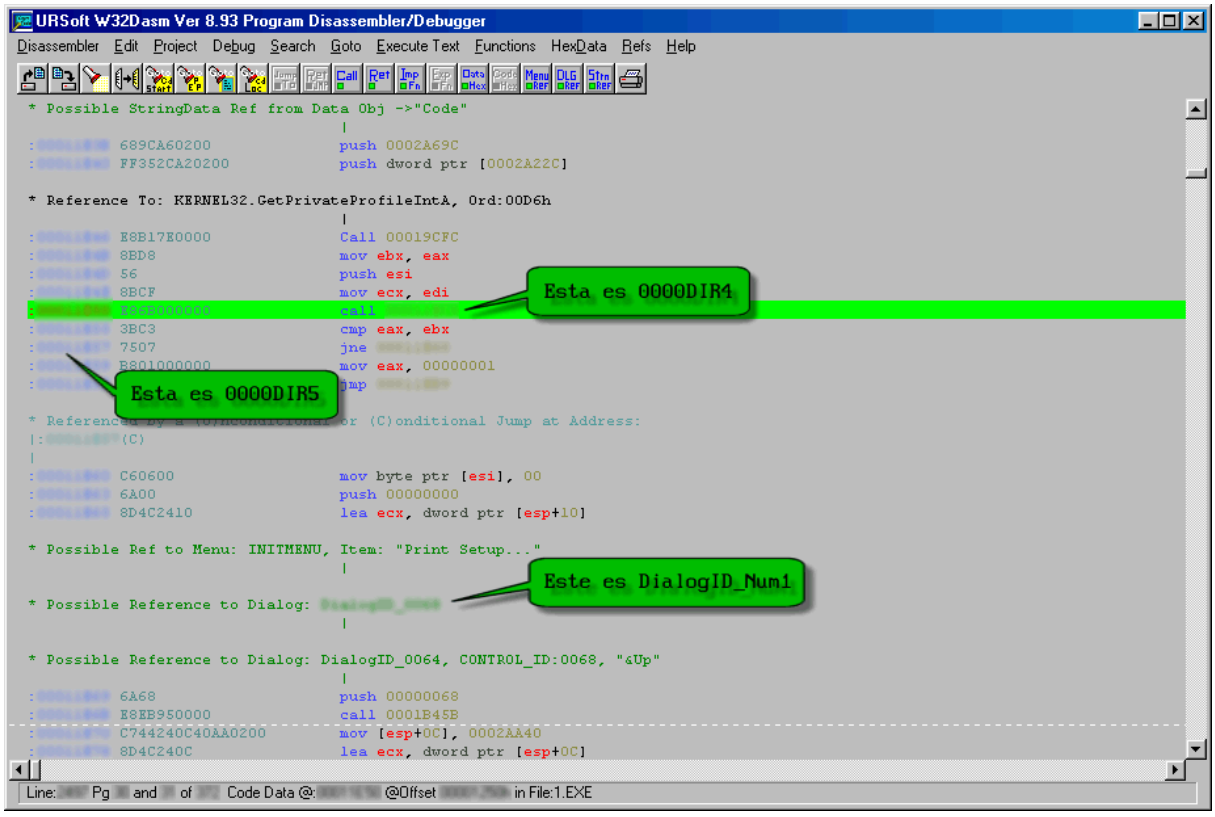
Un par de líneas antes de esta bifurcación, se encuentra el famoso **call** hacia la dirección **0000DIR3** que suponemos es el procedimiento verificador. Anotamos esta dirección.

8. Situamos el cursor de W32 (la barra verde claro) en ese call y presionamos la tecla → (Flecha derecha) una sola vez, para ir hacia donde apunta el **call**.
9. Del punto 6, ¿qué podemos inferir? Inferimos que los **call** de ese tipo pueden ser llamados repetidas veces para realizar su trabajo. Esta es la razón por la que vemos:

   ```
   * Referenced by a CALL at Addresses:
   |:0000DIR3   , :0000DIR4
   ```

   Se realizan 2 llamadas a este procedimiento. Sabemos que la llamada en **0000DIR3** es para el mensaje de error. Anotemos la dirección **0000DIR4** para determinar dónde y por qué es llamado este **call**.
10. En seguida, menú **Goto**, opción **Goto Code Location** (o también **SHIFT+F12**), escribimos nuestra **0000DIR4** y aceptamos. Analicemos lo que tenemos en pantalla. ¡Descúbralo por sí mismo antes de pasar al punto 11! ¡Usted puede hacerlo!
11. Recordando el punto 5, habían **10** llamadas al diálogo **DialogID_Num1** (nuestro nag) y vemos, unas pocas líneas debajo de nuestra **0000DIR4**, que ésta es exactamente una de aquellas ocurrencias:

    ```
    * Possible Reference to Dialog: DialogID_Num1
    ```



12. Como seguramente habrá notado, los bytes a editar serían los de la bifurcación posterior al **call** de la **0000DIR4**. Tomemos nuestra tabla de bifurcaciones de la lección anterior y encontremos el complemento del **Jxxxx** a modificar. Anotemos su respectivo código hexadecimal.
13. Es momento de ir a Hiew para editar. Con la lección anterior ya aprendimos a hacerlo. En TC situamos el cursor en **hedit.exe** y presionamos el botón de Hiew que ya tenemos creado. Recordemos que en TC tenemos, tanto el directorio propio del programa como el directorio **Hedit 1.2** creado dentro de **\cracking\parciales**. El **hedit.exe** que vamos a editar es el que se encuentra en su propio directorio.
14. En la pantalla de Hiew presionamos **F4** (**Mode**), seleccionamos **Decode** y aceptamos.
15. Deberíamos tener en pantalla una lista muerta. Pulsamos **F5** (**Goto**), escribimos **.0000DIR5**, **ENTER**. El cursor se desplazará al byte indicado, entonces **F3** (**Edit**), escribimos el código hexadecimal que obtuvimos en el paso 12, **F9** (**Update**) para actualizar los cambios.
16. Salimos de Hiew presionando **ESC** o **F10** (**Quit**).
17. Probamos el **hedit.exe** recién editado y si hemos seguido los pasos detalladamente, entonces ya no veremos el molesto nag, sino que el programa cargará directamente. Incluso podemos

registrarnos con cualquier nombre y número de serie, salimos del programa, volvemos a cargarlo, menú **Help**, opción **About** y allí aparece aún nuestro nombre.

18. Efectivamente, ahora usted puede decir: ¡Lo logré!

Alguno de los lectores más avanzados, se estará preguntando por qué no hemos utilizado un depurador. Muchos programas pueden ser crackeados empleando exclusivamente un desensamblador. Existen otros que requieren de un depurador como complemento. Tome cualquier página para descargar shareware y pruebe sus conocimientos del desensamblador W32. Si algún programa presenta complicaciones, anótelo en su lista negra y siga con otro. Cuando transcurran unos días, podría revisar su lista negra y retomar alguno de los programas que dejó pendiente.

Resulta igual de importante, recomendar al lector que no se acostumbre a coleccionar cientos de herramientas para crackear. El número de herramientas a utilizar es inversamente proporcional a la cantidad de conocimiento teórico que usted tenga. Es decir, cuánto más aprenda leyendo y practicando, notará que necesita menos herramientas

(No, aún no)

De los 3 inconvenientes, aún queda por resolver el tercero: Si intentamos registrarnos con un nombre y número de serie válidos, aparecerá el mensaje de error.

En segundo lugar, el proceso de modificación de bytes debe ser automático. Aunque usted podría utilizar un generador de patchs, lo ideal sería que codifique su propio programa base para crackear y tal vez hasta podría crear un generador de patchs para los menos afortunados, pero

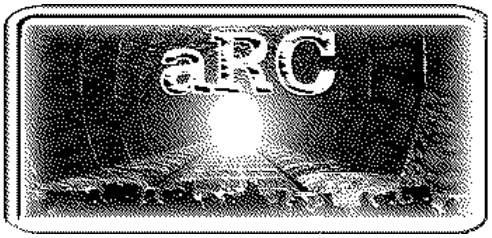
Estamos por llegar a la cima. En la lección 5 usted resolverá el primer pendiente y en la lección 6, tendrá los algoritmos para la generación de su propio patch en diversos lenguajes a escoger.

Derechos de autor

El autor puede ser contactado en:

IRC CyberArmy /server -m irc.cyberarmy.com: #arc3000, #arc
IRC DalNet: #arc3000
Email: furiouslogic@eml.cc

Furious Logic
Cofounder of



aDVANCED rESEARCH cOMMUNITY
http://arc3000.web1000.com
Lista: http://beam.to/arc3000
IRC CyberArmy: #arc3000, #arc
IRC DalNET: #arc3000

"Porque buscamos la libertad que sólo en el conocimiento podemos encontrar"