LECCIÓN 9: LA AVENTURA ÉPICA DE HALLAR UN SERIAL VÁLIDO Y KEYGEN PARA QUEST II 16 GREY

INTRODUCCIÓN- Una apacible mañana de enero...:

Nuestro entrañable amigo, maestro y mentor X-Grimator (servidor de ustedes) recibió un correo de un chico llamado Glojes con muchas ganas de aprender y que hacía muchas preguntas al bueno del autor de este texto. Así que, poco a poco, a base de contestarle cosas, le fue devolviendo la ilusión de escribir un tuto y ver que hay gente interesada en el Zen Palm Cracking, ya que por un momento, X-Gri casi casi había tirado la toalla. Así que ni corto ni perezoso, se hizo con las herramientas que ya comentamos en otros tutoriales anteriores y se dispuso a destripar una aventura gráfica llamada Quest II para nuestra amada Palm.

De acuerdo, de acuerdo, prometo no contar historias aburridas en este tutorial e ir directo al grano, hermano...

1- Saliendo de nuestra aldea hobbit:

Como estoy releyendo "El Señor de los Anillos" se me escapan expresiones como esta, ya veis...

En primer lugar vamos a sacar un serial válido y luego ya veremos un generador de llaves con más calma para esta milonga. Así que cargamos el X-Master y el programita objetivo en nuestro Emulador (¡¡si no os enteráis, leed los tutoriales anteriores!!. Quizá empecemos con programas para PalmOs 5, así que procurad entender bien todo lo que hasta ahora explicamos) y una vez instalados arrancamos el Southdebugger y entramos en el X-Master del emulador, con lo que el South se nos activará y nos permitirá introducir Systraps en la ventana correspondiente. Después de probar con varias, X-Grimatorito descubrió que la única útil para este caso es SysTrapDlkGetSyncInfo. Pues bien, volvemos al Emulador, nos vamos al Quest II y al cargar una aventura se nos activa el icono del menú (donde pongo "pulsa aquí")

y tal y como vemos, nos aparecen varios botones donde tocaremos el último, el de "Register". Creo que hasta ahí lo vamos entendiendo, ¿verdad?. Pues bien, una vez hecho esto nos salta el Soutdebugger (le damos a F5, que es "go" y continuamos sin más, ya que no es éste el momento que nos interesa analizar el SysTrap) y entonces nos encontramos con la siguiente ventanita, en donde vemos nuestro nombre de conexión HotSync, un código de registro que de momento no nos sirve de nada y un sitio donde meter el serial, que admite un máximo de 5 números (como habréis observado en otros tutoriales míos, este tipo de cosas deben siempre ser observadas por un cracker avezado, pues cualquier tipo de pista puede resultarnos luego de una gran ayuda en un momento dado. A veces, creedme, las mayores obviedades os pueden sacar de un apuro de horas y horas traceando y mareándoos con tanto código raro):

Pues vamos a meter un fake serial cualquiera, pongamos que el "12345", aunque no soy muy partidario de poner números correlativos, como sabéis, a fin de que nos os confundan luego cualesquiera cadenas de texto (es más fácil que un trozo de código que ponga "12345" nos conduzca a error que otro que ponga "25468". Pero por esta vez no nos va a importar, como veréis, y tendremos suerte).

Le damos a OK, y entramos de cabeza en nuestro amado Southdebugger, directitos en la SysTrap. Os estaréis preguntando por qué he usado esta SysTrap y no las típicas de FrmAlert y por qué ni he mencionado el PrcExplorer y esas cosas que suelo usar. Pues la respuesta es fácil. Si observáis, estas ventanas están hechas con dibujos. Si las buscáis en el PrcExplorer, sencillamente no encontraréis nada de nada, os llevará a confusión y desesperanza y eso es lo que menos le interesa a un cracker ¿verdad?. Haced la prueba y veréis, no aparece nada interesante ni digno de atención, pues no es una llamada a la API de PalmOs sino un dibujo hecho por el programador combinado con trozos de programación en texto. Además, los únicos bmp que vamos a encontrar con el PrcExplorer y que se refieran a "Register" poco nos van a decir. Eso sí, si lo que deseáis es traducir el juego a otro idioma o cambiar los dibujos, entonces eso es otro cantar, caballeros :o)

Bueno, estamos en el South como os dije y nos encontramos con este código (lo traspongo en listado muerto obtenido con Prc2bin como vimos en otras entregas):

```
000064d4  4e4fa2a9                  sysTrapDlkGetSyncInfo; la systrap que nos interesa. ¡No confundir con la anterior,
                                                           ésta es después de meter fake!.
000064d8  4a2effd6                  TST.B   -42(A6); en A6- $42 tenemos la primera letra del HotSync.
000064dc  4fef0018                  LEA     24(A7),A7
000064e0  6610                      BNE     L780; salta ya que D0 no es igual a la primera letra de HotSync

000064f2  486effb4          L780    PEA     -76(A6); Esta instrucción no se ejecuta. El debugger lee la siguiente como
                                                     la primera de la L780. El motivo, no lo sé...
000064f6  486effd6                  PEA     -42(A6);
000064fa  4ebaea54                  JSR     L617; si me salto el proceso con F7 veo:

                                                  CMP.W #$8 (A6) [=#12345 ! $3039], D0 ← el 12345 es el fake, y
                                                  D0=142h (que es 322 decimal, el serial para un HotSync que sea X-
                                                  Grimator)
```

Como sabéis, para avanzar cada instrucción debemos pulsar F8. Para esta instrucción:

```
00005118  b06e0008                  CMP.W   8(A6),D0; que es justo la anterior de dos líneas arriba
```

supe dónde estaba en el listado muerto buscando la instrucción hexa que marca la ventana "dissasembler" del debugger en el code0002.bin, con la opción "Buscar" (en donde puse "b06e0008"). Esto os lo comento ya que no sé por qué extraño arcano, en realidad no ejecuta la subrutina de la L617, pero así es.

Por lo tanto, yo veo claro que el debugger me compara mi fake (12345) con lo que ponga D0, que como vemos en la ventana de registro, nos dice nuestro serial correcto (miradlo en decimal, ya que si lo introducimos en hexa, pues no funcionaría, claro...). Por lo tanto, 322 es el serial correcto para "X-Grimator".

2- Keygeneando:

Ahora viene un paso más adelante, queridos niños míos, y vamos a procurar hacer un keygen para este juego tan ameno. Para ello no nos quedará más remedio que meternos de lleno en la instrucción

que salta a la subrutina L617 (aunque vimos que en realidad no es así) a ver qué demonios ocurre ahí dentro. Así que una vez que lleguemos a la siguiente línea del debugger:

```
000064fa  4ebaea54              JSR     L617
```

en vez pulsar F7 para ejecutar todo el proceso de golpe, pulsaríamos F8 para ir paso a paso. Fijaos (y esto podría ser una prueba si se tercia más adelante) que estamos en una parte del código llamada "Bastardo" (en el debugger, en rojo, encima justo de la línea de código que se está traceando). Si esto no es una broma del programador que se cree que sabe español y todo, entonces es un insulto. Ya sea por haches o por bes, ese cerebro-ameba se merece que le saquemos un keygen. Hay que ser serio programando, señores, un poco de profesionalidad por el amor de Dios... ya me encargaré de poner alguna alusión a eso en mi keygen je, je.

Así las cosas, el código que nos topamos de aquí hasta que calcula el número en cuestión es (os lo pongo comentado paso a paso):

```
00005070  4e56fff4      L624    LINK    A6,#-12
00005074  48e71e00              MOVEM.LD3-D6,-(A7);  típicas instrucciones al inicio de rutina
00005078  7a00                  MOVEQ  #0,D5; D5=0 y D3=nuestro fake
0000507a  486efff4              PEA     -12(A6); carga algo en memoria
0000507e  2f2e0008              MOVE.L 8(A6),-(A7); #252638 $3DADE, que es mi nombre HotSync como
                                                    comprobaré en 4ef0
00005082  4ebafe58              JSR     L611; ejecutamos de golpe la subrutina con F7
00005086  486efff4              PEA     -12(A6)
0000508a  4e4fa0c7              sysTrapStrLen; D0 con la longitud de mi HotSync
0000508e  3c00                  MOVE.WD0,D6;  y D0=D6, que pasará a contener la longitud de mi HotSync
00005090  7600                  MOVEQ  #0,D3; D3=0
00005092  4fef000c              LEA     12(A7),A7
00005096  6026                  BRA     L626


000050be  b646          L626    CMP.W  D6,D3; compara D6 con D3
000050c0  65d6                  BCS     L625; Salta si la bandera de acarreo está activa (salta)


00005098  3805          L625    MOVE.WD5,D4; D5 es la parte del serial válido ya calculada. D4=D5
0000509a  3003                  MOVE.WD3,D0; D0=D3
0000509c  e648                  LSR.W   #3,D0; D0 mueve tres bytes a la derecha
0000509e  7205                  MOVEQ  #5,D1; D1=5
000050a0  c2c4                  MULU.WD4,D1; D4*D1 y lo guarda en D1
000050a2  7400                  MOVEQ  #0,D2; D2=0
000050a4  3403                  MOVE.WD3,D2; D2=D3
000050a6  41efff4                 LEA     -12(A6),A0
000050aa  14302800                MOVE.B0(A0,D2.L),D2; letra del HotSync analizada (en hexa) a D2
000050ae  4882                  EXT.W   D2
000050b0  d441                  ADD.W   D1,D2; D2=D1+D2
000050b2  e16a                  LSL.W   D0,D2; D2 mueve a la izquierda el número de bytes que tenga D0
000050b4  c4fc001a                MULU.W #26!$1a,D2; multiplicación por 26 decimal(o 1A) hexa y guarda en D2
000050b8  d444                  ADD.W   D4,D2; le suma D4 a D2 y lo guarda en D2
000050ba  3a02                  MOVE.W D2,D5; D2 lo pasa a D5 y es el serial calculado
000050bc  5243                  ADDQ.W #1,D3; suma uno a D3
000050be  b646          L626      CMP.WD6,D3; siendo D6 la longitud del HotSync
000050c0  65d6                    BCS     L625; salta si el flag de acarreo está activado. Deja de hacerlo al analizar
                                                  todas las letras del HotSync
000050c2  3005                    MOVE.W          D5,D0; D0 tiene el serial bueno aquí
000050c4  4cdf0078                 MOVEM.L         (A7)+,D3-D6
000050c8  4e5e                    UNLK    A6
```

y esta pieza de código que viene a continuación ya la vimos, que ya no nos interesa para hacer el keygen.

```
00005118  b06e0008              CMP.W8(A6),D0;
0000511c  57c0                  SEQ     D0;  8(A6) tiene el fake serial y D0 tiene el buen serial.
0000511e  4400                  NEG.B   D0
```

Ahora sólo debéis coger vuestro lenguaje de programación favorito y plasmar todo esto en la forma que más cómoda os resulte y con los efectos visuales que más os apetezcan. Aunque no lo he comprobado, tiene todas las trazas de ser esa la única parte de código que se necesita para activar el buen serial. De nuevo, los zombi-programadores nos lo han puesto muy fácil ¿a que sí?.

3- En la memoria del tiempo:

Para rematar y a modo de despedida, quiero mandar un fuerte abrazo a mis alumnos Glojes y Sunevil y al canal #codex y #disidents del IRC-Hispano, que son majísimos y siempre estarán dispuestos a ayudar a newbies, como el bueno de S-P-A-R-K. Un saludo también a ManiacPc de Kut, ex-tripulante de un barco pirata de traductores que hace años tuve :o)

Y una mención muy muy especial a Karpoff y a Craaack el Destripador, cuyas páginas acaban de ser cerradas después de muchos años enseñando a otros y archivando miles de tutoriales. Sin ellos nunca habría aprendido nada, ni yo ni cientos de crackers. Son los últimos de mi época y han tenido que dejar esto por diversas causas. La scene hispana ha perdido a dos mitos y a ellos les dedico estas letras:

"los viejos lobos de mar van poco a poco reuniéndose con las sirenas en el fondo de los mares. Supongo que es el tributo a pagar por haber llegado a la gloria de ser un maestro. La memoria perdurará en los newbies que lean todos los tutos que escribisteis y recopilasteis. Un maestro cracker queda tocado por la mano de la sabiduría y jamás muere".

En algún lugar de la EU, 15/01/03
"Pensad, pensad... siempre pensad"