# LECCIÓN 5: ANEXO A LA LECCIÓN 4
# (DUDAS RESUELTAS)

¡¡Hola de nuevo pequeños gamberros!!. Como os medio prometí en la anterior entrega, vamos a proceder a aclarar algunas cosas que nos quedaron en el tintero y que nos resultan realmente interesantes en este crack.
Para poneros al día, nos habían quedado por ver tres puntos:

1- **Comprobación del trial**. Bueno, al final decidí no prestar demasiada atención a esto por dos motivos. El primero, porque con el programa registrado, el trial no me preocupa; y el segundo porque prefiero buscar un programa basado sólo en esa protección y mirarlo con más calma. Sin embargo, a cambio, os voy a dar una pequeña clase de ASM de Motorola para ir profundizando en esto.
2- **Analizar lo del número mágico**. Pues la verdad es que no es más que eso, un número mágico. Si lo usas, te registra por una sola vez. En cuanto vuelves a iniciar el programa ya no te vale. Podríamos hacer un cambio en la comparación, pero teniendo el serial correcto, poco nos importa usar el "sucedáneo". Por lo tanto, creo que es una pérdida de tiempo fijarse más en eso. Como dice el título de un LP de los Sex Pistols "Flogging a dead horse" es decir, por mucho que azotes a un caballo que ya está muerto, la carrera ya no la ganas. A otra cosa, butterfly.
3- **El keygen**. Esto es lo realmente importante. Remitiéndonos al punto uno, vamos a ver cómo el programa halla nuestro ID al partir de nuestro HotSync y cómo luego calcula nuestro serial. ¿Os hace más ilusión que haya capturas de pantalla?... bueeeeeeeeeeno, os meteré alguuuuuuuuuna. Aquí vamos a usar una nueva herramienta llamada SouthDebugger, debugger para Palm que os viene protegido con una nag. Está hecho en java, así que ya sabéis que hacer si lo que deseáis es probar el producto con todas sus posiblilidades ;o).

**NOTA IMPORTANTE:** Como sé que se me va a olvidar, os apunto algo que llamó poderosamente mi atención en una web de Ingeniería Inversa de Palms que ya no existe. Era de un italiano llamado Quequero y ponía lo siguiente "Se il listato che vi dissasembla..." ejem, bueno, mejor en español..."Si el listado que desensambláis es ilegible o no corresponde, provad a cancelar el primer byte del .prc y reintentar el desensamblado...¡debería funcionar!". Con lo cual me quedé muy contento porque ya sé que hay protecciones anti-desensamblador para Palm y que es tan sencillo romperlas como cancelar el primer byte del ejecutable. Personalmente no me encontré ningún caso, pero me ha parecido un apunte importante.

## 1- Escribo estas líneas desde...:

Lo que parecía un soleado día de vacaciones en yate poco a poco se convirtió en una pesadilla de tormenta. El crack se presentaba risueño y fácil y no considerábamos complicado hallarlo, sin embargo, el keygen no salía, las líneas se complicaban... el cielo ennegrecido dio paso a rayos sesgantes y a truenos que sonaban como gritos de gigantes condenados. El barco zorzobraba, nuestra moral se hundía, el Zen Palm se nos atascaba. Cuando quisimos darnos cuenta, éramos náufragos de nuestro conocimiento.

Bueno, si mal no recordáis, habíamos dado bastantes vueltas hasta encontrar un punto de ataque correcto para nuestro programa. Primero, la llamada a la Alert, luego mirar el HotSync y al final decidimos tirarnos de cabeza al momento en que se capturan los strings que introducimos como S/N (para neófitos o newbies, esto significa serial number. Para nada significa Sólo Newbies, así que podéis estar tranquilos... :o) ).
Nos aparecía nuestro nombre, una id y debajo un espacio para poner el número, algo así como esto:

En donde veis que se calcula un User ID al partir del nombre de conexión HotSync con el que la Palm se pone de acuerdo con el Pc. En fin, pues como en el anterior tutorial no acabábamos de ver con qué comparaba nuestro fake serial, el pobre X-Grimator, en su torpeza, empezó a pensar que el serial se calculaba al leer el HotSync y que se activaba un flag que luego se comparaba con el fake serial (que a su vez acababa por convertirse en un flag también, llegando al final a una mera comprobación de flags). Ni por asomo. Al final, las cosas son más fáciles para los zombi-programadores. Este razonamiento era erróneo por la sencilla razón de que en ningún lado existía la conversión del fake serial a flag. De esto, que era evidente, me di cuenta después de un rato trasteando la llamada a HotSync que hace el programa. De todos modos, os comento cómo convierte el programa nuestro nombre a User ID y así aprendemos un poco de ASM de Motorola como os prometí.
Lo primero que encontramos al hacer un ATB en "DlkGetSyncInfo" (en la forma que os expliqué en el otro tuto) es esto: (**nota**: lo que vaya detrás del punto y coma son comentarios que no lee el programa. Os he ordenado el código para que leáis los saltos de corrido)

```
00005f78  4e4fa2a9                  sysTrapDlkGetSyncInfo ; llamada a API donde ponemos el ATB en el debugger
00005f7c  3600                      MOVE.W  D0,D3
00005f7e  4fef0018                  LEA     24(A7),A7; a7+18(h) tiene mi HotSync. El (h) significa hexadecimal.
00005f82  663c                      BNE     L609; no salta
00005f84  486effd6                  PEA     -42(A6); a6-2a tiene mi HotSync. PEA pushea en la pila algún buffer.
00005f88  4e4fa0c7                  sysTrapStrLen; calcula la longitud de mi HotSync y lo mete en hexa en D0
00005f8c  4a40                      TST.W   D0; compara D0 con cero.
00005f8e  584f                      ADDQ.W  #4,A7; corrige la pila sumándole 4 bytes
00005f90  672e                      BEQ     L609; no salta ya que D0 no es cero (Branch if equal)
00005f92  486effd6                  PEA     -42(A6); A6-2a el HotSync
00005f96  3f3c1b5b                  MOVE.W  #7003!$1b5b,-(A7); field 7003: Name (usar para ello PrcEdit)
00005f9a  2f0a                      MOVE.L  A2,-(A7)
00005f9c  4ebafaac                  JSR     L563

00005a4a  4e560000          L563    LINK    A6,#0 --→SETFIELDFROMSTR (pinta en pantalla el HotSync)
00005a4e  2f0b                      MOVE.L  A3,-(A7)
00005a50  2f0a                      MOVE.L  A2,-(A7)
00005a52  2f2e000e                  MOVE.L  14(A6),-(A7)
00005a56  4e4fa0c7                  sysTrapStrLen
00005a5a  5240                      ADDQ.W  #1,D0
00005a5c  7200                      MOVEQ   #0,D1
00005a5e  3200                      MOVE.W  D0,D1
00005a60  584f                      ADDQ.W  #4,A7
00005a62  2f01                      MOVE.L  D1,-(A7)
00005a64  4e4fa01e                  sysTrapMemHandleNew
00005a68  2448                      MOVEA.L A0,A2
00005a6a  2f0a                      MOVE.L  A2,-(A7)
00005a6c  4e4fa021                  sysTrapMemHandleLock
00005a70  2648                      MOVEA.L A0,A3
00005a72  2f2e000e                  MOVE.L  14(A6),-(A7)
00005a76  2f0b                      MOVE.L  A3,-(A7)
00005a78  4e4fa0c5                  sysTrapStrCopy
00005a7c  2f0a                      MOVE.L  A2,-(A7)
00005a7e  4e4fa022                  sysTrapMemHandleUnlock
00005a82  3f2e000c                  MOVE.W  12(A6),-(A7)
00005a86  2f2e0008                  MOVE.L  8(A6),-(A7)
00005a8a  4e4fa180                  sysTrapFrmGetObjectIndex
00005a8e  5c4f                      ADDQ.W  #6,A7
00005a90  3f00                      MOVE.W  D0,-(A7)
00005a92  2f2e0008                  MOVE.L  8(A6),-(A7)
00005a96  4e4fa183                  sysTrapFrmGetObjectPtr
00005a9a  2648                      MOVEA.L A0,A3
00005a9c  2f0a                      MOVE.L  A2,-(A7)
00005a9e  2f0b                      MOVE.L  A3,-(A7)
00005aa0  4e4fa158                  sysTrapFldSetTextHandle
00005aa4  4fef0022                  LEA     34(A7),A7
00005aa8  245f                      MOVEA.L (A7)+,A2
00005aaa  265f                      MOVEA.L (A7)+,A3
00005aac  4e5e                      UNLK    A6
00005aae  4e75                      RTS
00005ab0  93                                DC.B    #147
00005ab1  5365744669656c645465              DC.B    'SetFieldTextFromStr'
00005abb  787446726f6d537472

00005fa0  486effc6                  PEA     -58(A6); carga una variable
00005fa4  486effd6                  PEA     -42(A6); pushea mi HotSync
00005fa8  4ebafd62                  JSR     L586
```

```
00005d0c  4e560000            L586      LINK     A6,#0 --→MAPHOTSYNCNAMETOPID (convierte HotSync en UserID)
00005d10  48e71830                      MOVEM.L  D3/D4/A2/A3,-(A7); al inicio de subrutinas, para recuperar luego los valores
00005d14  266e000c                      MOVEA.L  12(A6),A3; mueve a A3, A6+c
00005d18  7600                          MOVEQ    #0,D3; inicia a cero D3 y D4 (move quick)
00005d1a  7800                          MOVEQ    #0,D4
00005d1c  4227                          CLR.B    -(A7); borra un byte de la pila
00005d1e  4878000f                      PEA      $000f.W
00005d22  2f0b                          MOVE.L   A3,-(A7)
00005d24  4e4fa027                      sysTrapMemSet; pone un rango de memoria en un puntero dinámico a un valor específico
00005d28  246e0008                      MOVEA.L  8(A6),A2; mete mi HotSync en A2
00005d2c  4fef000a                      LEA      10(A7),A7
00005d30  6008                          BRA      L588; salto obligatorio
00005d32  1012                L587      MOVE.B   (A2),D0; mueve la primera letra de mi HotSync a D0 en ascii hexadecimal
00005d34  4880                          EXT.W    D0; extiende D0 en un word
00005d36  d840                          ADD.W    D0,D4; suma los valores en ascii de las letras del HotSync que va sacando
00005d38  528a                          ADDQ.L   #1,A2; saca una letra de HotSync
00005d3a  4a12                L588      TST.B    (A2); testea el primer byte de mi HotSync para ver si es cero
00005d3c  66f4                          BNE      L587;al vaciar el HotSync de A2, finaliza el loop.
00005d3e  246e0008                      MOVEA.L  8(A6),A2; HotSync pasa a A2
00005d42  7601                          MOVEQ    #1,D3; inicia D3 a valor uno (ver valores de bytes, no datos).
00005d44  6044                          BRA      L593; salto obligatorio
00005d46  4a12                L589      TST.B    (A2); compara el primer byte que quede en el HotSync con cero
00005d48  6716                          BEQ      L590; si es cero, salta
00005d4a  101a                          MOVE.B   (A2)+,D0; el primer byte que tenga A2 lo mete en D0
00005d4c  4880                          EXT.W    D0; extiende el tamaño de D0
00005d4e  48c0                          EXT.L    D0
00005d50  81fc0010                      DIVS.W   #16!$10,D0; divide entre 10(h) el word de D0
00005d54  4840                          SWAP     D0; invierte el orden de los bytes (así, 500013 pasa a 130005)
00005d56  41edffee                      LEA      -18(A5),A0; carga A5-12, que es "9876543210453721"
00005d5a  16f00000                      MOVE.B   0(A0,D0.W),(A3)+
00005d5e  6008                          BRA      L591; salto obligatorio
00005d60  41edffee            L590      LEA      -18(A5),A0; a5-12 es "9876543210453721"
00005d64  16f03000                      MOVE.B   0(A0,D3.W),(A3)+
00005d68  3044                L591      MOVEA.W  D4,A0; pasa a A0 la suma de los valores ascii de mi HotSync
00005d6a  2008                          MOVE.L   A0,D0; de A0 a D0
00005d6c  81fc000a                      DIVS.W   #10!$a,D0; divide D0 entre A(h)
00005d70  4840                          SWAP     D0; inivierte los bytes
00005d72  06400030                      ADDI.W   #48!$30,D0; añade 30(h) a D0
00005d76  16c0                          MOVE.B   D0,(A3)+
00005d78  48c4                          EXT.L    D4
00005d7a  89fc000a                      DIVS.W   #10!$a,D4; divide D4 entre A(h)
00005d7e  0c430002                      CMPI.W   #2,D3; compara D3 con 2
00005d82  6604                          BNE      L592; de no ser igual, nos salta
00005d84  16fc0020                      MOVE.B   #32!$20,(A3)+
00005d88  5243                L592      ADDQ.W   #1,D3; suma 1 a D3
00005d8a  0c430004            L593      CMPI.W   #4,D3; compara D3 con 4
00005d8e  6fb6                          BLE      L589; acaba el loop cuando D3=5
00005d90  4cdf0c18                      MOVEM.L  (A7)+,D3/D4/A2/A3
00005d94  4e5e                          UNLK     A6
00005d96  4e75                          RTS
00005d98  93                            DC.B     #147
00005d99  4d6170486f7453796e63          DC.B     'MapHotSyncNameToPID'
00005da3  4e616d65546f504944
00005dac  0000                          DC.W     #0


00005fac  486effc6                      PEA      -58(A6); A6-3A es el número calculado al partir del HotSync
00005fb0  3f3c1b61                      MOVE.W   #7009!$1b61,-(A7); Field 7009: UserID
00005fb4  2f0a                          MOVE.L   A2,-(A7)
00005fb6  4ebafa92                      JSR      L563; SETFIELDTEXTFROMSTR, para pintar el PID en la pantalla
00005fba  4fef001c                      LEA      28(A7),A7; carga en pila mi UserId
00005fbe  6012                          BRA      L610; salto obligatorio
00005fc0  486effd6            L609      PEA      -42(A6)
00005fc4  3f3c1b61                      MOVE.W   #7009!$1b61,-(A7)
00005fc8  2f0a                          MOVE.L   A2,-(A7)
00005fca  4ebafa7e                      JSR      L563
00005fce  4fef000a                      LEA      10(A7),A7
00005fd2  261f                L610      MOVE.L   (A7)+,D3
00005fd4  245f                          MOVEA.L  (A7)+,A2
00005fd6  4e5e                          UNLK     A6
00005fd8  4e75                          RTS
00006048  3f3c1b5c                      MOVE.W   #7004!$1b5c,-(A7); Field 7004: Serial
0000604c  2f0a                          MOVE.L   A2,-(A7)
0000604e  4e4fa180                      sysTrapFrmGetObjectIndex
00006052  5c4f                          ADDQ.W   #6,A7; suma 6 para corregir la pila
```

```
00006054  3f00                    MOVE.W D0,-(A7)
00006056  2f0a                    MOVE.L  A2,-(A7)
00006058  4e4fa179                sysTrapFrmSetFocus
0000605c  4fef000a                LEA     10(A7),A7
00006060  2f0a            L614      MOVE.L          A2,-(A7)
00006062  4e4fa193                sysTrapFrmDoDialog; muestra el formulario en pantalla para introducir ya el serial
00006066  3a00                    MOVE.W D0,D5
00006068  0c451b5d                CMPI.W  #7005!$1b5d,D5
0000606c  584f                    ADDQ.W #4,A7
0000606e  660000f4                BNE     L617
00006072  2f0b                    MOVE.L  A3,-(A7)
00006074  2f0a                    MOVE.L  A2,-(A7)
00006076  4ebafda2                JSR     L599
```

Pues después de leer todo esto pasito a pasito desesperadamente, queridos niños, y tras tirarse horas y horas traceando con el debugger, el alma cándida de X-Grimator ha descubierto que ¡¡no vale de nada!!, no es más que una engañifa, un engañabobos, una trampa para elefantes...¡¡será asqueroso el programador!!... nos ha tomado el pelo haciéndonos ver cómo calcula el UserID para luego hallar el número sin tener para nada en cuenta todo este proceso. Grrrrrrrrrrr... >:o(

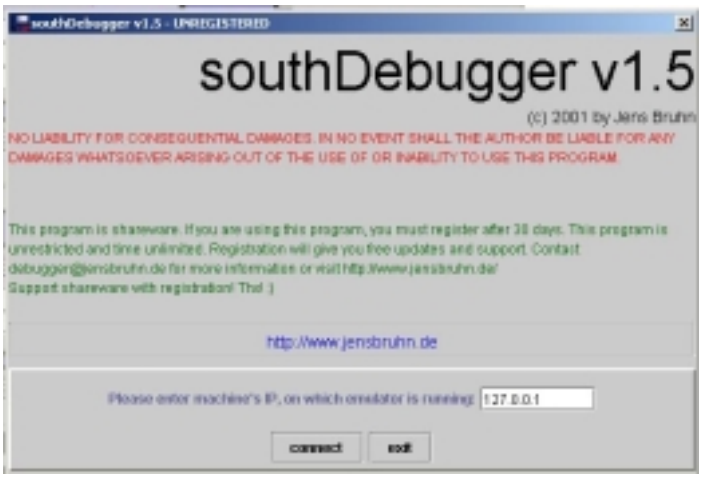
**2- En esta isla hay un mono que es amigo mío y me quita los piojos:**

"Querido diario, tras un naufragio de cinco días entre líneas y líneas de código, me he hecho un amigo muy simpático que se llama Amelio. Estoy empezando a perder la razón y apenas me alimento de la leche de un coco que he conseguido pelar" (pelar cocos... ya empezamos con expresiones capciosas XD ).

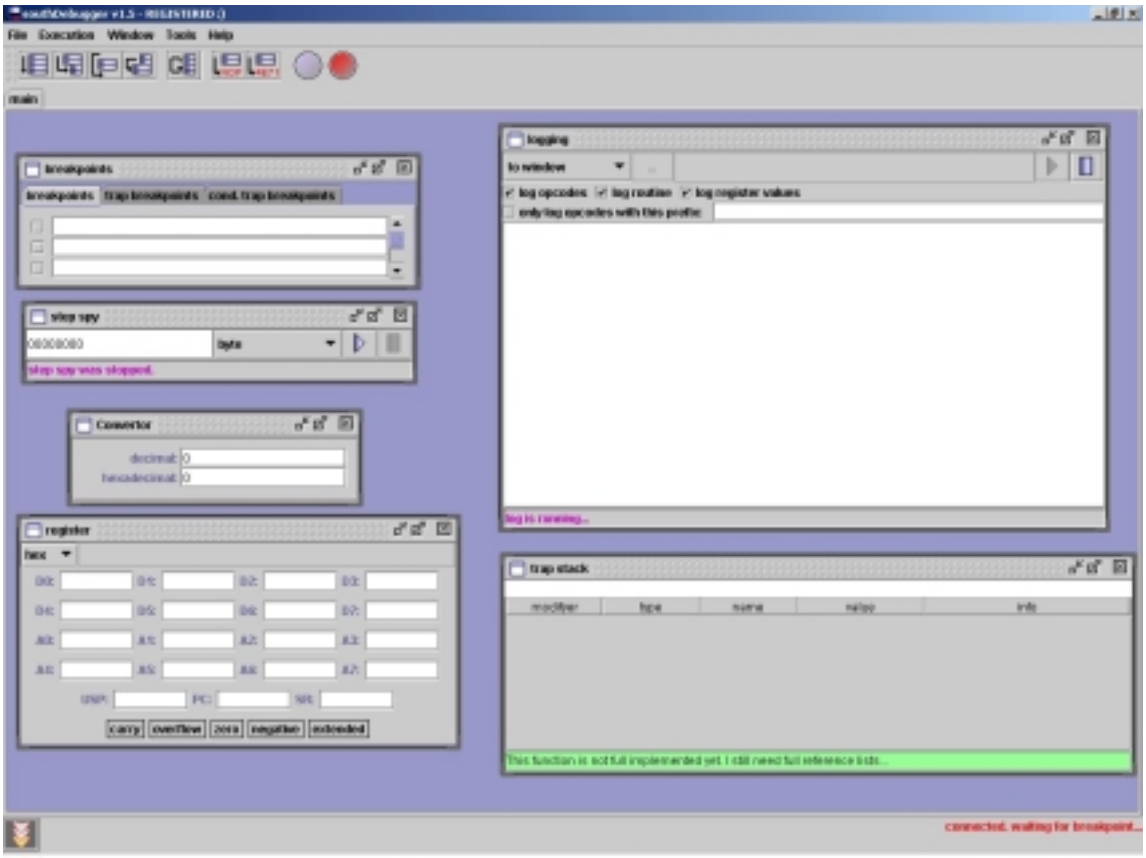
En fin, como esto no es propiamente un tutorial sino que es un anexo, no quiero extenderme demasiado. Hemos visto cómo calcula el UserID al partir de nuestro HotSync, y he aprovechado para comentaros las cosas más importantes que se van realizando a fin de que tengáis una idea algo más avanzada sobre ASM de Motorola. Ahora, volvemos al grano y sacamos el numerito. Para ello, volvemos al tutorial anterior a la subrutina llamada EnteredKeyValid (esta vez dejamos a la virgen en paz que bastante tuvo, je, je...). De ahí vamos traceando hasta

```
00005df4  b6aa0016                CMP.L   22(A2),D3
00005df8  6204                    BHI     L597--→ CON PASS CORRECTA NO SALTA
00005dfa  7001                    MOVEQ   #1,D0
00005dfc  6002                    BRA     L598
00005dfe  7000            L597    MOVEQ #0,D0--→ PONIENDO A #1 SE PARCHEA
```

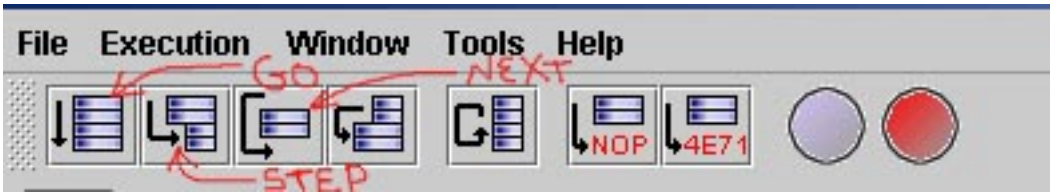
donde a estas alturas, a cualquiera de vosotros no se le escapa que lo interesante está en 5df4, por lo que tendremos que ver qué comparación hace. Para ello vamos a usar una nueva herramienta, el SouthDebugger. Es un programita en java que una vez lanzado nos aparece esto:



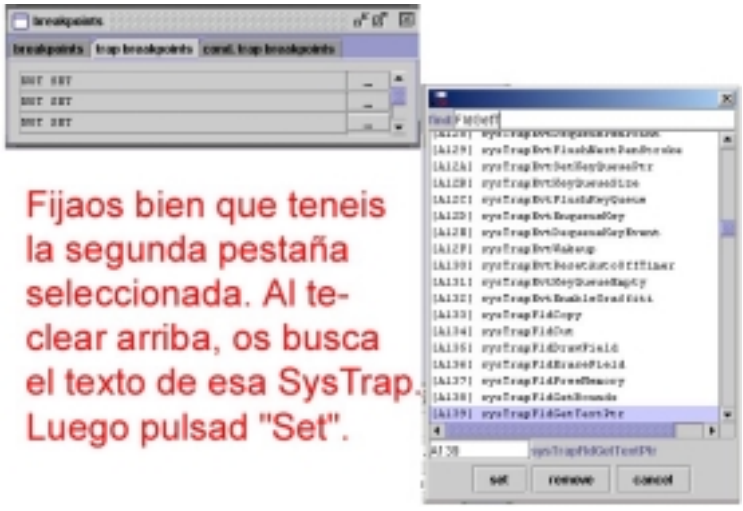

le damos a connect y entramos de lleno en el debugger, donde de forma feliz nos topamos de bruces con una pantalla que debería tener activadas las ventanas que veis, y que tenéis en el menú Window. Os recomiendo activar las mismas que yo, ya que son muy cómodas:



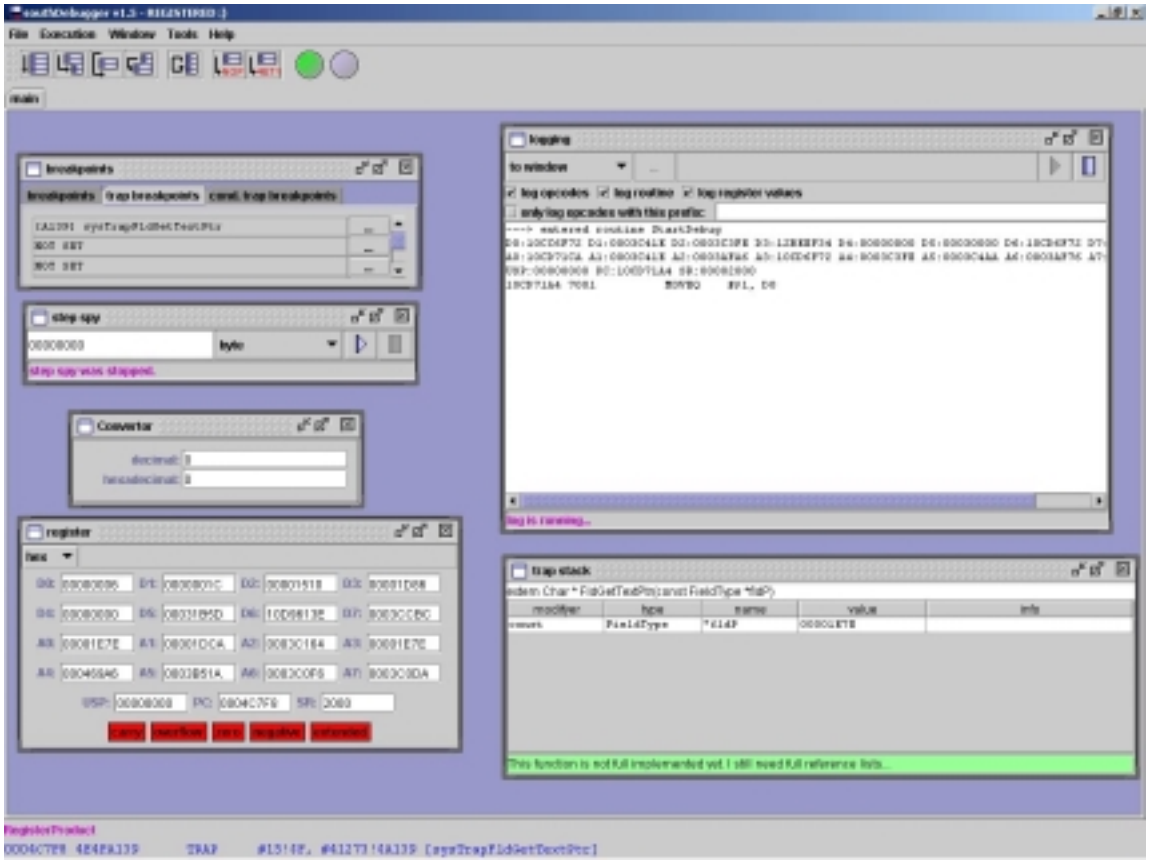
como observáis, las ventanas interesantes son Converter (convierte valores hexa en decimal y viceversa), step spy, la de registros, la de datos (trap stack) y una de loggin por si queréis loggear paso a paso de lo que vais haciendo. En la ventana de registros tengo marcado datos hexa, pero podéis seleccionar decimales. La forma de ejecutar es la siguiente:



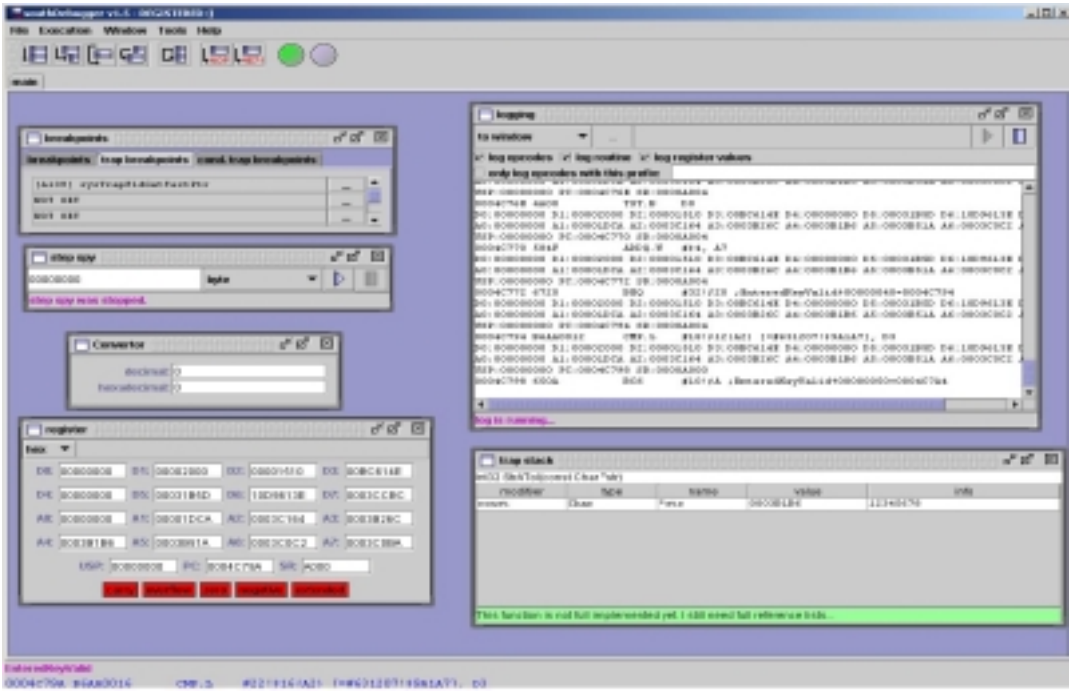

donde Go lanza el debugger, Step permite ejecutar una instrucción y Next ejecuta de golpe toda una llamada sin entrar en ella. Los breakpoints los elegimos en la ventana de breakpoints tecleando arriba el que nos interesa, así:



Bueno, no dije que para abrir esa ventana debéis pulsar el botón con los tres puntos. Ponemos un atb en “FldGetTextPtr”, le damos a Go y aparecemos de este modo tras meter un serial basura en la ventana de Palm:



aquí hay que destacar que la línea siguiente a ejecutar es la que vemos debajo de todo en color azul, y que los valores de los registros están en la ventana de registros. ¡Dios, me repito más que el ajo!... le vamos dando a F8 para ejecutar paso a paso y a F7 para ejecutar llamadas de golpe y de este modo llegamos a nuestra querida línea de comparación:



pues bien , si os fijáis, (aparte de que nuestro fake 12345678 está en la ventana de datos trap stack), debajo de todo, en azul, pone

**CMP.L #22!$16(A2)[=#631207!$9A1A7],D3**

¿y qué pensáis que es 631207, que en hexa es 9A1A7?... jeje, pues si, eso... y si vemos en la ventana de registros, en D3 tenemos el valor hexa de mi fake serial, que es 12345678 (podéis elegir el modo decimal para verlo más claro en esa misma ventana de datos). Si el número fake que metemos es de 6 o menos dígitos, la comparación se hará justo encima de ésta que estamos viendo, pero se hace sobre igual número. Y luego viene el salto BHI que ya comentamos en el anterior tuto y que lee el flag Zero que ya sale activo de la aplicación Virginal esa que nos quita el sueño. BHI significa que salta cuando (tanto C como Z) estén desactivadas.
Pues la risa de esto es que usemos el HotSync que usemos ¡¡el numerito nos vale para todos por igual!!. Así que llego a la conclusión de que el UserID no es más que basura para despistar.

**3- Camarón que se duerme se lo lleva la corriente:**

Pues así es. Nuestro náufrago será rescatado en breve por una patrulla gracias a que a base de pensar decidió investigar por su cuenta en vez quedarse de brazos cruzados, descubriendo que a unos metros había un complejo hotelero que organizaba vacaciones en esa isla (juas).

He de reconocer que me despistó lo del UserID pero todavía sigo preguntándome que fin tiene gastar líneas y líneas de código enrevesado para calcular un número que luego ¡¡no vale de nada!!. Hay gente tan sui generis por ahí en adelante... (eso de “sui generis” es una expresión de Humanidades, no de Informática, pero en algo tenía que valerme mi carrera ¿no?).
Bueno acabo aquí el anexo al cuarto capítulo de Zen Palm (Diario secreto de X-Grimator), bastante cansado ya porque son unas horas de la noche que en fin...
Confío que esto haya matizado el tuto anterior y pula por fin los flecos que quedaban de este programa. Ahora sí que no queda nada de nada que mirar ya.

Un abrazo muy especial desde aquí a Ziritione, Sunevil, S-P-A-R-K, DDiego y los canales #disidents y #codex del IRC-Hispano por acogerme como miembro cuando tuve problemas en #crackers. Por supuesto no olvido por nada en el mundo a eSn-mIn, quizá el cracker que más sabe de esta zona (S-P-A-R-K vive en otro lado más lejano, pero le va a la zaga...) y el único sensato y de buen corazón. Tendríamos que aprender tanto de él.

Ahora me voy a dormir mucho que ya casi amanece XD.

X-Grimator, Europa 08/07/02
“Pensad, pensad, siempre pensad”

PD: Mis tutos los tenéis en www.disidents.org, sección cracking/documentos, bajo el nombre de Zen Palm (Diario Secreto de X-Grimator).