Info sobre Ensambladores :

Introducción

Bueno, este tutorial está dedicado a los que no tienen ningún conocimiento de lo que es crackear o Ingeniería Inversa. Este tutorial lo hago de acuerdo a la manera en que yo entendí las cosas cuando recién empezaba, a ver si les resulta más fácil aprender así.

T e m a s :

1)¿Qué necesitamos?

2)Herramientas

3)Crackear y los sistemas numéricos

4)Muy básico sobre ensamblador

5)Familiarizándonos con las herramientas

6)Nuestro primer crack

1) ¿Qué necesitamos?

Para esto de la Ingeniería Inversa se necesitan fundamentalmente las siguientes cosas:

1)Si quieren hacerse las cosas más fáciles para empezar, algún conocimiento de ensamblador, éste es un lenguaje de programación de bajo nivel. Cuanto más sepan mejor y más rápido crackearán (para empezar, les recomiendo que lean el tutorial de Pr@fEsOr X sobre ensamblador que está muy bueno)

2)Cuando se llega a cierto nivel, a veces la ayuda de otro/s cracker/s más experto/s

3)Ganas de aprender. No es leer los tutoriales sin entender nada y hacer los cracks sin saber lo que hicimos, es estudiar y comprender el funcionamiento del programa.

4)Herramientas

2) Herramientas

1) Depurador o debugger: es un programa que permite detenerte en cada paso que realizan otros programas, creo que no expliqué bien. Un programa no es más que un conjunto de instrucciones, haz esto, haz lo otro. Entonces cuando se está ejecutando un programa y lo "depuras", por ejemplo hay una instrucción que dice: Muestra un cartel con las letras de color verde, te detienes en esa instrucción para poder cambiarla, parchearla, etc (el depurador te muestra todo en lenguaje ensamblador). El depurador más famoso que usan casi todos los crackers es el Soft Ice (de ahora en adelante le llamaremos SICE) y se pueden hacer cosas que ni siquiera te has imaginado con él.

Otros depuradores: tr (ahora está el TRW2000 que está bastante bueno), SICE (que lo utilizan casi todos los crackers), Debug(lo tienen todas las PC, sólo abre el DOS, escribe Debug aprieta enter y ya estás en el, no lo recomiendo).

2) Desensamblador o dissasembler: este programa convierte el ejecutable de otros programas hechos en otros lenguajes a lenguaje ensamblador. A diferencia del depurador no lo hace cuando el programa se está ejecutando, si no que cuando nosotros queramos (nosotros abrimos el ejecutable con el desensamblador y estudiamos el código todo el tiempo que queramos, a esto se le llama listado muerto).

Desensambladores: IDA pro (DOS 32 bits), W32Dasm(el que la mayoría usa y mi preferido), Sourcer 7, y muchos más.

3) Editor Hexadecimal: es un programa que te permite editar archivos de forma hexadecimal y así cambiar bytes en él, les enseñaré a convertir de decimal a hexadecimal y binario.

Editores hexadecimales: Hacker's View (DOS 32 bits), Ultraedit, y Hex Workshop, mi preferido.

Creo que esto es todo ya que pienso que al principio no necesitarán herramientas como el Frogs Ice(programa que se usa con los ejecutables que tienen métodos de detectar cuando el SICE está corriendo, y se usa para "esconderlo".

Estas herramientas las pueden encontrar en la página de http://www.crackstore.com/, http://w3.to/protools, y si falta alguna fíjate en otros tutoriales o en las páginas de TNT cracking team, kUT, Wkt, en la de karpoff, por supuesto en la de mi nuevo grupo, K-FoR y otras que hay millones de links.

3) Crackear y los sistemas numéricos

Supongo que ya tienen todo, si lo tienen comencemos lo teórico de verdad. Por favor lean algo sobre ensamblador, es verdad, al principio todos estamos ansiosos de empezar a crackear ahora mismo y sin saber absolutamente nada ensamblador ni de ninguna de las herramientas

pero lamentablemente eso no es posible:-(.Hay que tener paciencia para llegar a ser un buen cracker.

¿Qué es crackear?

Crackear, o Ingeniería Inversa (Reverse Engineering, como dicen los ingleses) se basa en reventar protecciones de software o hardware con fines intelectuales, personales, u otros, pero nunca lucrativos. De esta manera, el programa que estemos crackeando funcionará para lo que nosotros queramos y no para lo que fue diseñado.

¿Para qué crackear?

La sociedad en que vivimos da asco. Hay una enorme discriminación social. En esta sociedad la mayoría de la población está sometida al control de los que tienen dinero. El Cracking, es una forma de distribuir la riqueza entre la población. Si ves un programa en Internet que te gusta y no tienes una cuenta un un banco internacional ni nada de ello, para que esperar para tenerlo, crackéalo y publica el crack en Internet y estarás ayudando a mucha gente que vive a merced de los que son "importantes"(o sea, tienen cuentas en bancos internacionales, mucho dinero, etc).

Por supuesto que el Cracking también es para los programadores. Así podremos enseñarles que sus programas no son perfectos y tienen muchos errores tontos que podrían corregirlos. Y también enseñarles a no ser perezosos y crear buenas protecciones.

Empezaremos por aprender lo que es un sistema numérico.

Nosotros contamos el el sistema numérico de base decimal (la base de este sistema es el número 10). Esto quiere decir que el primer número formado por más de una cifra (0 1 2 3 4 5 6 7 8 9 10)de este sistema es el 10. Pero para las computadoras, este sistema no es conveniente, ya que la información se maneja codificada en forma de bits verdaderos y falsos 1=verdadero 0=falso. Esto lleva a la utilización de dos sistemas en la informática, binario (de base dos) y hexadecimal (de base dieciséis).

Convertir números de binarios a decimales:

En ensamblador nos encontramos con la necesidad de convertir números de binarios utilizados por la PC, a decimales que son los que nosotros entendemos.

Mira la siguiente tabla y puedes ir guiándote: Decimal Binario

1 1

2 10

3 11

4 100

5 101

6 110

7 111

8 1000

9 1001

10 1010

11 1011

12 1100

13 1101

14 1110

15 1111

Como se muestra en la tabla 1+1 en binario=10. Esto es muy fácil, ya verás. Los números de binarios a decimales se pasan así:

Binario: 11011

Decimal:

1 + 2 + 0 + 8 + 16

todo esto = 27 en decimal

Nota: Hice este dibujo para que se entendiera mejor. En realidad, en la computación, el carácter ^ se utiliza para potenciar y el carácter *se utiliza para multiplicar.

Supongo que todos entienden ese dibujo. Si lo miran bien, verán que los números que están resaltados en azul, forman el número binario.

Convertir números de decimales a binarios:

Hay varias formas de hacerlo pero esta es la más fácil:

Ejemplo, el número 25 en decimal.

25/2 =12 y el resto es 1

12/2 = 6 y el resto es 0

6/2 = 3 y el resto es 0

3/2 = 1 y el resto es 1

1/2 = 0 y el resto es 1

Después tomamos el número de abajo hacia arriba y tenemos el número en binario 11001.

Números hexadecimales:

Puedes ir guiándote con la siguiente tabla:

Decimal Hexadecimal

1 1

2 2

3 3

4 4

5 5

6 6

7 7

8 8

9 9

10 A

11 B

12 C

13 D

14 E

15 F

16 10

17 11

18 12

19 13

20 14

21 15

22 1A

Como ven en esta tabla, en el sistema hexadecimal los números que están formados por números de una sola cifra son: 0 1 2 3 4 5 6 7 8 9 A B C D E F. Es exactamente igual que contar en decimal pero nada más que el 9 sería el 16 (no sé si entienden).

Para convertir de números hexadecimales a binarios se debe agrupar en grupos de cuatro bits empezando de derecha a izquierda. Después, en el último grupo, se rellenan los espacios en blanco con ceros.

Tomaremos como ejemplo el número binario 1101011

separamos en grupos de 4 bits,

110 1011

rellenamos los espacios con ceros,

0110 1011

Después tomamos cada grupo y lo transformamos a base 10(pueden fijarse en la tabla)

0110= 6 1011= 11

Pero como 611 no es correcto en el sistema hexadecimal sustituimos el 11 por su valor correspondiente en hexadecimal y obtenemos 6Bh (la h se pone para indicar que el número está en el sistema hexadecimal)

Para convertir un número de hexadecimal a binario solo es necesario invertir estos pasos.

SUPONGO que alguna vez tuvieron una clase de matemáticas y saben sumar, restar y potenciar, no, era un chiste:-) ya que si no saben, están muertos (y les sugiero que tomen clases de matemáticas)

4) Muy básico sobre ensamblador

Si ya leyeron algo sobre ensamblador, supongo que ya saben esto, pero lo escribo para los que son vagos y no se quisieron molestar en dejar este tutorial y leer otro para luego volver a este. El ensamblador o ASM es un lenguaje de programación de bajo nivel. Esto quiere decir que, a diferencia de C o C++ (lenguajes de alto nivel), mientras más programemos o más programa creemos, menos control tendremos sobre éste.

Este lenguaje les va a ayudar mucho ya que es fácil de aprender y necesitan saber de ensamblador para saber crackear.

Registros

Empezaremos por aprender (los que no leyeron sobre assembler) que son los registros. AX, BX, CX, DX (no son sólo estos, hay más pero por ahora pienso que es suficiente). Estas "palabras"

son registros de datos. Éstos contienen cálculos y otras cosas que necesita un programa para ejecutarse. Es como que guardan datos temporalmente en la memoria. Los registros de 16 bits son estos (AX, BX, CX y DX) pero si trabajamos en 32 bits, o sea Windows 95, se les agrega una E delante entonces los registros de 32 bits serían así: EAX, EBX, ECX, EDX...

Si ven algo como AL y AH, no se preocupen, es que los registros se separan (BX se separa BL y BH...) Cada registro tiene un valor que se le asigna y se le modifica con algunas instrucciones en ensamblador.

Aquí tienen los principales saltos condicionales e incondicionales del ensamblador:
Hexadecimal: ASM: Significa:

75 o 0F85 jne salta si no es igual

74 o 0F84 je salta si es igual

EB jmp salta directamente a.....

90 nop ningún funcionamiento (No OPeration)

77 o 0F87 ja salta si es superior

0F86 jna salta si no superior

0F83 jae salta si es superior o igual

0F82 jnae salta si no está sobre o igual

0F82 jb salta si está debajo

0F83 jnb salta si no está debajo

0F86 jbe salta si está debajo o igual

0F87 jnbe salta si no está debajo o igual

0F8F jg salta si es mayor

0F8E jng salta si no es mayor

0F8D jge salta si es mayor o igual

0F8C jnge salta si no es mayor o igual

0F8C jl salta si es menor

0F8D jnl salta si no es menor

0F8E jle saltan si es menor o igual

0F8F jnle saltan si no es menor o igual

Cuando vean algún cmp por ejemplo cmp eax, ebx, quiere decir que compara, cmp proviene de "comparar". Compara el valor de los registros para luego haber un salto condicional o incondicional como 75 o 0F84 (condicionales) o un EB (incondicional).

Veamos un poco de esto en un ejecutable cualquiera de mi computadora: :00402BB1 6A28 push 00000028<-----Mete 28 en la pila

:00402BB3 50 push eax<----------Mete EAX en la pila.

:00402BB4 8BCD mov ecx, ebp<------EBP y ECX ahora valen lo mismo

:00402BB6 FF523C call 00436E04<-----Llama a la instrucción ubicada en la dirección 00436E04

:00402BB9 83F828 cmp eax, 00000057<-Compara el valor de EAX con 28, en ASCII, la W

:00402BBC 0F8593010000 jne 00402D55<------Si no son equivalentes nos vamos a la dirección 00402d55

:00402BC2 8B442440 mov eax, dword ptr [esp+40]<----Son iguales, continuamos con esta operación

:00402BC6 83F828 cmp eax, 00000061<--Compara EAX con 61, en ASCII la a.

(Nota importante: en Cracking desde cero para súper newbies 2 hablaremos de la pila y del código ASCII).Antes de ver un salto condicional o incondicional, verán alguna instrucción que diga cmp xx-xx o test-xx-xx (dónde xx es un registro), estás instrucciones provienen de comparar los valores de los registros para determinar un posterior salto condicional o incondicional.

5) Familiarizándonos con las herramientas

Para crackear bien y más rápido, es necesario familiarizarse con las herramientas. Así aprenderemos a utilizarlas bien. Si bajaste un trial que expira dentro de un día y tu no has aprendido a usar bien esa herramienta, tómate tu tiempo, la crackeas, y podrás usarla todo el tiempo necesario. Pero cuando la sepas utilizar la compras, ya que habíamos dicho que el Cracking nunca se practicaba con fines lucrativos.

W32Dasm: W32Dasm es un desensamblador, o sea, convierte el código de un programa hecho en otros lenguajes como C, C++, Delphi, Visual Basic, Pascal, etc, al lenguaje de ensamblador.

Editor hexadecimal: el Hex Workshop y el ultraedit son muy fáciles de usar y no necesitarán leer ningún tutorial para aprender a usarlos (y que yo sepa no hay ninguno). El Hacker's View (HIEW) es más complicado ya que es de DOS y hay una cantidad de teclas para memorizarse, F1, F2, F7....y muchas más. Recomiendo que si eligieron el HIEW, se lean algún tutorial sobre él. Mi preferido es el Hex Workshop pero es mi preferido y eso no quiere decir que los esté obligando a tener Hex Workshop, encuentren el editor hexadecimal que más les guste.

Soft Ice: este programa si es muy difícil llegar a conocerlo todo. No podrán hacer absolutamente nada si no leyeron algún tutorial sobre él. Háganme caso, yo al principio odiaba al SICE, con esa estúpida ventanita de DOS, pero ahora que aprendí a usarlo, me parece que es una maravilla. Cuando lo instales el SICE (o Soft Ice, como le quieran llamar) deberás pasar por las siguientes configuraciones:

1) Número de serie: si bajan el SICE y viene en un zip, posiblemente venga con un archivo en formato txt llamado sn, ahí estará nuestro número de serie. Si no viene con un archivo llamado sn, puede venir con un archivo, llamado léame, leer o sino un archivo nfo también puede estar el serial en estos archivos.

2) Tarjeta de video: el SICE detectará tu tarjeta de video cuando oprimas el botón test. Como en mi caso (no detectó mi Diamond Viper 770), si no es así, selecciona la tarjeta Standar Display Adapter VGA.

3) Mouse: en mi caso, cuando instalé el SICE por primera vez, movía el mouse y no me andaba, si no que me aparecían en la parte de arriba los carteles que dicen copy, paste, copy&paste, display, un-assemble, what y previous. Lo peor era que no se podía seleccionar nada en los carteles, hacían como intermitencia cada vez que movía el mouse. Si esto les pasa, en la instalación, pongan cómo que están usando Microsoft Intelli Mouse y a ver si les anda.

4) Modificación de tu AUTOEXEC.BAT: si no sabes lo que es el AUTOEXEC.BAT puedes mandarme un mail y te lo diré. Cuando instalas el SICE, te dan la opción de crear un autoexec.ice, modificar tu AUTOEXEC.BAT o no hacer ningún cambio. Te recomiendo que no elijas la opción de no hacer ningún cambio si quieres que el SICE funcione:-) Ahora si, de verdad, te recomiendo que dejes que el programa de instalación modifique tu AUTOEXEC.BAT por que si no cada vez que quieras ejecutar el SICE tendrás que salir a DOS y cargar el archivo que el programa de instalación creó (en este caso autoexec.ice). La otra cosa que puedes hacer es crear un bat pero cada vez que quieras ejecutar SICE tendrás que salir a DOS para cargar tu bat. Como ustedes son newbies, no creo que sepan lo que son los bats, como se crea uno, configurarlos, ni nada de eso, así que mi dirección de coreo electrónico está abierta para preguntas y sugerencias, cualquier cosa mándenme un mail a dek_oin@hotmail.com .

5)Cuando esté instalado lanzar Symbol Loader: deberás lanzar el "symbol loader", ir al menú edit, luego a SoftICE initialization settings. en donde dice "Initialization string" debes poner X;wl;wr;wd7; code on;, y en donde dice "History Buffer size (KB)" debes poner 512.

6)Modificación de winice.dat: con cualquier editor de texto debes abrir el archivo "Winice.dat". Verás que hay unas cosas que dicen más o menos así, ;EXP= , en todos los ;EXP=, deberás quitarle el ; que está delante para que quede EXP=. Si no entiendes bien o piensas que estás haciendo todo mal, mándame un mail que te doy el Winice.dat ya modificado.

7) Deben reiniciar el equipo para obtener una actualización completa de los cambios.

Recomiendo que si no leyeron nada sobre el SICE y lo tienen instalado y configurado nunca presionen ctrl+d ya que con estas teclas se abre el SICE. Les digo esto para que no les pase lo que me pasaba a mi, que estaba escribiendo rápido, iba a poner una d mayúscula y al apretar shift el dedo se me corría y apretaba ctrl. Después tenía que reiniciar el equipo porque no

sabía cómo salir del SICE (se sale con ctrl+d, como se entra) y así se me perdían los trabajos que estaba haciendo!!!:-(. No hablo más porque no quiero acordarme!!!

6) Nuestro primer crack

Por fin!!!!. Si nunca hicieron un crack y este es el primero, felicitaciones, podrían convertirse en la nueva generación de crackers si ponen ganas y empeño.

Detalles del programa víctima: Víctima: Toggle Disk Space v 2.0

URL: www.toggle.com

Descripción: Optimizar el rendimiento del disco duro

Tamaño del ejecutable: 805 kb.

Tipo de Protección: Número de serie

Método de ataque Desensamblar y parchear

Objetivo Simular estar registrados

Dificultad Newbie - Aficionado - Avanzado - Experto - Élite

Herramientas W32Dasm y editor hexadecimal

El programa víctima, tenía que ser uno de los programas de Toggle (firma que pone a nuestra disposición muchos productos de shareware para bajar gratis y para registrarnos antes de 30 días porque si no el programa deja de funcionar, estos programas son facilísimos de crackear!!!). Recuerden que crackeamos solo para extender el período de prueba de un programa y para observar al máximo sus capacidades, reitero, nunca se crackea con fines lucrativos, si te gusta el programa CÓMPRALO!!!. El Toggle Disk Space es un programa que nos permite ver como está funcionando el disco duro y corregir errores en él. Si no estamos registrados, tenemos algunas limitaciones, como por ejemplo, no podemos usar la opción de speed, etc. Lo que más odio son esas estúpidas nag-screens que aparecen cuando abrimos el ejecutable, pidiéndonos que nos registremos y dándonos opciones. El Toggle Disk Space caduca después de 30 días de concurrida la instalación si no nos registramos.

Esta protección es fácil. Pero no tan fácil como otras. Si quieren aprender a hacer las cosas no hagan todo a merced de lo que dice esto. Préstenle atención a las explicaciones y también razonen por ustedes mismos (es decir, pienso que si esto pasa lo otro tendría que pasar, etc).

Muy bien, empezaremos con nuestro programilla. Ejecutémoslo y veamos que aparece una nag screen que nos da opciones para lo que queremos hacer.

La nag:

Hmm, esa nag no me gusta para nada. Hagan clic en "Enter your registration code, hagan clic en next y aparecerá algo así:

Pongan un nombre cualquiera como por ejemplo deK_Oin en donde dice company pueden no poner nada ya que dice que es opcional y luego escriban un serial como 1234567890. Hagan clic en next y...

Nooooo!!!!!!!!!!. Parece que nuestro serial 1234567890 no funcionó. ¿Les gusta la nag y el cartel que dice que nuestro serial no es el correcto?. Pues a mi no así que vamos a eliminar todo eso.

Comencemos:

Ejecutemos el W32Dasm, vayamos al menú Dissasembler\Open file to Dissasemble y abran el ejecutable toggleDISKSPACE.exe.

Primero miren este esquema para saber para que sirven los botones de W32Dasm:

Ven el botón que está al lado de imprimir, si, el que dice referencias de cadenas, hagan clic en el. Este nos mostrará muchas cosas. Recuerdan el mensaje que nos decía que no habíamos introducido un serial correcto, era "The registration information that you have entered is not valid. Please comfirm that you have entered the infomation correctly"

Bueno, busquemos este mensaje en las referencias de cadenas (o String References) y.... aha!! aquí está, casi al final.

Haz doble clic en él para ver cuantas referencias hay sobre él pero solo hay una. Ho te preocupes por si el mensaje no está completo, es lo mismo. Cuando hagas doble clic en él, el W32Dasm nos llevará a un lugar del código así:

:004028C9 8BCD mov ecx, ebp

:004028CB E820F9FFFF call 004021F0

:004028D0 EB0E jmp 004028E0

* Referenced by a (U)nconditional or (C)onditional Jump at Addresses:

|:004025DF(C), :00402622(C), :00402855(C)

|

:004028D2 6A00 push 00000000

:004028D4 6A00 push 00000000

* Possible StringData Ref from Data Obj ->"The registration information you "

->"have entered is not valid. Please "

->"confirm that you have entered "

->"the information correctly."

:004028D6 685C714500 push 0045715C

Si ven un poco más arriba de "The registration...", verán que hay un texto que nos dice que el mensaje de error fue llamado por 3 saltos condicionales o incondicionales ubicados en las direcciones 004025DF, 00402622, y 00402855.

Razonen... si el mensaje de error fue llamado desde 3 saltos que se ubican en 3 diferentes direcciones, ¿que pasaría si los nopeáramos (con 90 que significa sin operación) para que el programa no ejecute ninguno de esos saltos (son los que indican que el serial que introducimos es erróneo) cuando escribimos nuestro serial y pulsamos en aceptar?. Muy bien!!!!(supongo que ya pensaste y que pensaste la respuesta correcta) el programa nos aceptará cualquier serial.

Para parchear los 3 saltos hagan lo siguiente:

Hagan clic en el menú Goto\go to code location y escriban la dirección del salto. Empezaremos con el primes salto 004025DF, haz clic en OK. Aquí está, hmm salto condicional, :004025D8 85C0 test eax, eax

:004025DA 0F94C1 sete cl

:004025DD 84C9 test cl, cl

:004025DF 0F85ED020000 jne 004028D2

* Possible StringData Ref from Data Obj ->"26893278"

|

:004025E5 BE38724500 mov esi, 00457238

Cambiémoslo por 909090909090 para que el programa no haga nada con el serial que metemos, solo llevarnos al preciado mensaje de "GRACIAS POR REGISTRARSE".

Si se fijan bien en la parte inferior de la pantalla verán una cosa así:

Ven donde dice Offset. El offset es la dirección real de memoria y lo que introduciremos en el editor hexadecimal. NO DEBES PONER LOS CEROS, SON PARA QUE LOS ENTIENDA EL DESENSAMBLADOR, NI LA h QUE SIGNIFICA QUE ESTÁ EN HEXADECIMAL.

Abre el Hex Workshop (yo explicaré todo como si estuvieras usando el Hex Workshop ya que ese es el que yo uso) haz clic en file\open y abre el ejecutable toggleDISKSPACE.exe. Luego haz clic en edit\goto y escribe la dirección del offset, o sea 19DF. Ahí verás el salto condicional que habíamos visto en el W32Dasm 0F85ED020000 y lo cambiaremos por 909090909090. Sólo sitúa al cursor entremedio de la primera cifra y la anterior a ésta y escribe los 90. No cierres el Hex Workshop!!!, recuerda que hay 3 saltos que parchear y este es solo el primero.

Vuelve al W32Dasm busca de nuevo el mensaje "The registration information...." y haz doble clic en él. Estamos en el mismo lugar que antes. Solo mira un poco más arriba del mensaje de error y observa:

:004028C9 8BCD mov ecx, ebp

:004028CB E820F9FFFF call 004021F0

:004028D0 EB0E jmp 004028E0

* Referenced by a (U)nconditional or (C)onditional Jump at Addresses:

|:004025DF(C), :00402622(C), :00402855(C)

|

:004028D2 6A00 push 00000000

:004028D4 6A00 push 00000000

* Possible StringData Ref from Data Obj ->"The registration information you "

->"have entered is not valid. Please "

->"confirm that you have entered "

->"the information correctly."

:004028D6 685C714500 push 0045715C

Ese es el segundo salto que tendremos que parchear. Haz igual que el primero, edit\goto 00402622 a ver... otro salto condicional. Mira el offset, e insértalo en el hex workshop y cambia el salto por 90909090...

Vuelve al W32Dasm. Solo nos queda un salto por parchear. Edit\goto 00402855(dirección del tercer salto). Otro salto condicional. Insertemos el offset en el hex workshop y cambiemos el salto por 9090 (el salto es 757B).

Guarda los cambios desde el Hex Workshop y ejecuta toggleDISKSPACE.exe. Todo normal, como antes, selecciona la opción de "enter registration code" y escribe un nombre como deK_Oin en company puedes no poner nada y en Reg. code puedes poner cualquier cosa. Probemos ....cha cha cha channnnn.....

Felicidades!!!!!!!!!!!!!!!!!!!!! Has hecho tu primer crack. Todos esperamos que no sea el último. Cuando crackeen algo después de hacerlo, escriban un tutorial, ya que los conocimientos hay que compartirlos si no se te pudren en la cabeza!!!. Si hay algo que odio, son las faltas de ortografía (palabra grave terminada en vocal:-).