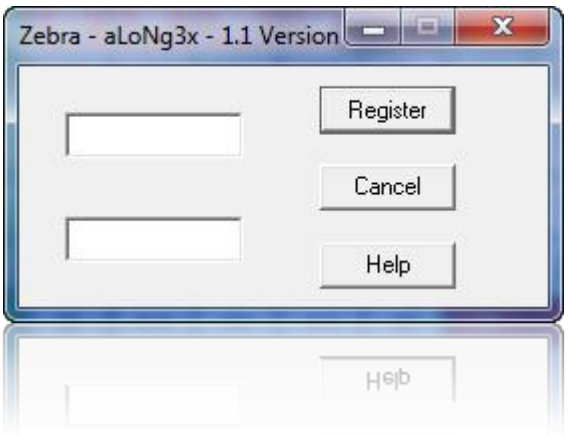
Solución al CrackMe Zebra 1.1 de aLoNg3x

Repasando trigonometría



By deurus
12/10/2014

ÍNDICE

Equipo utilizado:

S.O: Windows 7 x32 /Windows 8 x64

Depurador: Ollydbg 1.10 (32bits) con plugins

Analizador: PEiD 0.95

1. Introducción

Hoy tenemos aquí un Crackme de los que te hacen temblar las conexiones neuronales. Estamos acostumbrados al típico serial asociado a un nombre y a veces nos sorprenden.

El Crackme data del año 2000, está realizado por **aLoNg3x** y lo tenéis colgado en **crackmes.de**. En crackmes.de también disponéis de una **solución** muy elegante realizada por **cronos**, pero que no acaba de saciar nuestro afán de descubrir todas las soluciones posibles.

2. El algoritmo

Abrimos el Crackme con **Olly** y enseguida encontramos la rutina de comprobación junto con los mensajes de éxito y error. Os dejo la rutina comentada como siempre.

```
PUSH EBP
MOV EBP,ESP
SUB ESP,68
PUSH [ARG.1]                            ; /x1
CALL <JMP.&CRTDLL.atof>                 ; \atof
FST QWORD PTR SS:[EBP-18]
SUB ESP,8
FSTP QWORD PTR SS:[ESP]
CALL <JMP.&CRTDLL.floor>
FSTP QWORD PTR SS:[EBP-8]
PUSH [ARG.2]                            ; /x2
CALL <JMP.&CRTDLL.atof>                 ; \atof
FST QWORD PTR SS:[EBP-28]
SUB ESP,8
FSTP QWORD PTR SS:[ESP]
CALL <JMP.&CRTDLL.floor>
ADD ESP,18
FST QWORD PTR SS:[EBP-10]
FMUL QWORD PTR SS:[EBP-8]
FLDZ
FCOMPP                                  ;  floor(x1)*floor(x2)=0 ???
FSTSW AX                                ;  <<Store status word
SAHF                                    ;  <<Store AH into FLAGS
JNZ SHORT Zebrone.00401398              ;  Si salta todo OK
XOR EAX,EAX
JMP Zebrone.0040142E                    ;  Bad boy
FLD QWORD PTR SS:[EBP-8]                ;  <<Floating point load
FCOMP QWORD PTR SS:[EBP-10]             ;  x1 = x2 ???
FSTSW AX                                ;  <<Store status word
SAHF                                    ;  <<Store AH into FLAGS
JNZ SHORT Zebrone.004013AA              ;  Si salta todo OK
XOR EAX,EAX
```

```
XOR EAX,EAX
JMP Zebrone.0040142E                    ;  Bad boy
FLD QWORD PTR SS:[EBP-8]                ;  <<Floating point load
FSTP QWORD PTR SS:[EBP-38]
FLD1                                    ;  Carga 1 en el stack
FST QWORD PTR SS:[EBP-40]               ;  <<Floating point store
FCOMP QWORD PTR SS:[EBP-38]             ;  x1 > 1 ???
FSTSW AX                                ;  <<Store status word
SAHF                                    ;  <<Store AH into FLAGS
JA SHORT Zebrone.004013EA               ;  Si salta bad boy
FILD QWORD PTR DS:[403038]              ;  <<Load integer>> 2540BE400 = 10^10
FST QWORD PTR SS:[EBP-48]               ;  <<Floating point store
FCOMP QWORD PTR SS:[EBP-38]             ;  x1 < 10^10 ???
FSTSW AX                                ;  <<Store status word
SAHF                                    ;  <<Store AH into FLAGS
JB SHORT Zebrone.004013EA               ;  Si salta bad boy
FLD QWORD PTR SS:[EBP-10]               ;  <<Floating point load
FSTP QWORD PTR SS:[EBP-50]              ;  <<Store and pop
FLD QWORD PTR SS:[EBP-40]               ;  <<Floating point load
FCOMP QWORD PTR SS:[EBP-50]             ;  x2 > 1 ???
FSTSW AX                                ;  <<Store status word
SAHF                                    ;  <<Store AH into FLAGS
JA SHORT Zebrone.004013EA               ;  Si salta bad boy
FLD QWORD PTR SS:[EBP-48]               ;  <<Floating point load>> carga 10^10
FCOMP QWORD PTR SS:[EBP-50]             ;  x2 < 10^10 ???
FSTSW AX                                ;  <<Store status word
SAHF                                    ;  <<Store AH into FLAGS
JNB SHORT Zebrone.004013EE              ;  Salta si menor
XOR EAX,EAX
JMP SHORT Zebrone.0040142E              ;  Bad boy
FLD QWORD PTR SS:[EBP-8]                ;  <<Floating point load>> carga x1
FSIN                                    ;  Sin(x1)
FSTP QWORD PTR SS:[EBP-58]              ;  <<Store and pop
FLD QWORD PTR SS:[EBP-10]               ;  <<Floating point load>> carga x2
FSIN                                    ;  Sin(x2)
FSTP QWORD PTR SS:[EBP-60]              ;  <<Store and pop
FLD QWORD PTR SS:[EBP-58]               ;  <<Floating point load
FMUL QWORD PTR SS:[EBP-60]              ;  Sin(x1) * Sin(x2)
FILD QWORD PTR DS:[403030]              ;  <<Load integer>> 2386F26FC10000 = 10^16
FMULP ST(1),ST                          ;  10^16 * (Sin(x1) * Sin(x2))
SUB ESP,8
FSTP QWORD PTR SS:[ESP]                 ;  <<Store and pop
CALL <JMP.&CRTDLL.floor>
ADD ESP,8
FSTP QWORD PTR SS:[EBP-68]

FLDZ                                      ;  <<Load 0.0 onto stack
FCOMP QWORD PTR SS:[EBP-68]               ;  10^16 * (Sin(x1) * Sin(x2)) = 0 ???
FSTSW AX
SAHF                                      ;  <<Store AH into FLAGS
JNZ SHORT Zebrone.0040142C                ;  Si NO salta todo OK
XOR EAX,EAX
INC EAX
JMP SHORT Zebrone.0040142E
XOR EAX,EAX
LEAVE
RETN
```

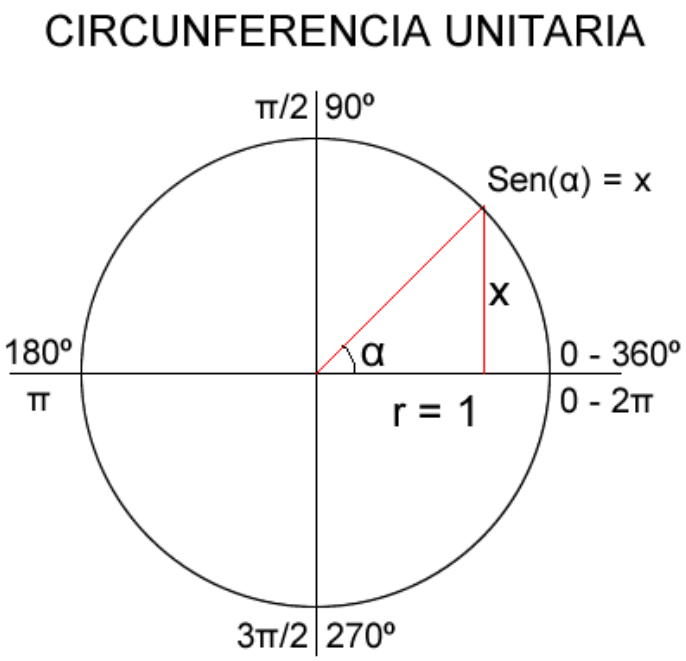
La primera dificultad que podemos encontrar es que utiliza **instrucciones FPU** y **coma flotante**, ya que si no tenemos la vista entrenada nos puede resultar un engorro. Superado esto, la rutina de comprobación se puede resumir así:

- x1 * x2 != 0
- x1 != x2
- x1 > 1 y < 10^10
- x2 > 1 y < 10^10
- Floor[10^16 * sin(x1) * sin(x2)] = 0

A priori no parece que tenga mucha dificultad, pero vamos a analizarlo más concienzudamente. **Necesitamos que la parte entera del resultado de la multiplicación sea 0**, algo que parece sencillo, pero fíjate que la **constante 10^16** nos **obliga** a su vez, a que el **resultado del seno sea muy pequeño**, cosa que como comprobareis limita mucho los resultados satisfactorios.

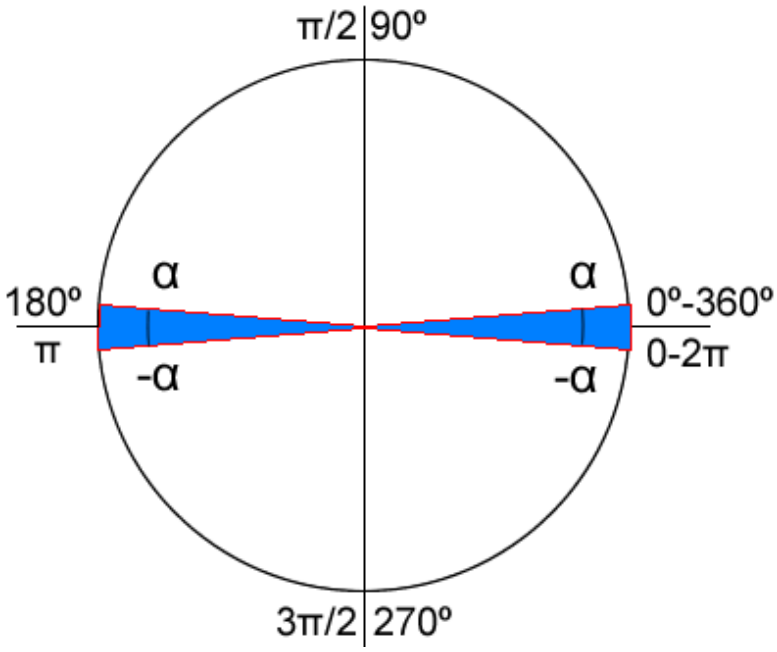
3. Repasando trigonometría

Cuando nos enseñó nuestro profesor la función del seno nos hizo el siguiente dibujo:



Partiendo de la **circunferencia unitaria**, podemos concluir que el **seno de alpha es igual a la altura x**. Como lo que nos interesa a nosotros es que el seno sea muy pequeño, en realidad estamos buscando que la x sea lo más pequeña posible. Llegamos entonces a la conclusión de que las soluciones para enteros entre 1 y 10^10 van a ser muy reducidas. Además nos percatamos que el ángulo alpha va a tener que

estar muy próximo a 0º – 360 (0 – 2π) y a 180º (π). En el siguiente gráfico queda claro el estrecho margen en el que nos movemos.



Si habéis leído la **solución de cronos** ahora le encontrareis algo más de sentido a por que él utilizó **fracciones continuas** de **π** y cogió como resultado los numeradores más cercanos a 10^10, en su caso 245850922 y 411557987.

4. Análisis operacional

Vamos a analizar un ejemplo operacional.

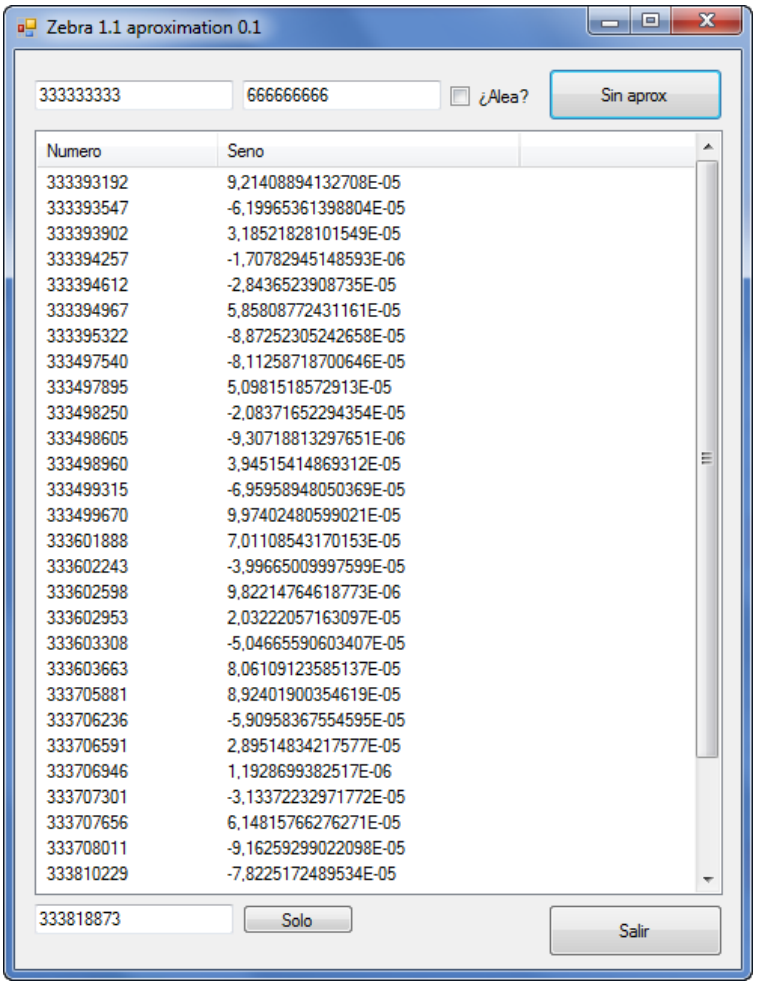
```
sin(245850922) = 6,11806538300111631427121098629 72e-9
sin(411557987) = 2,53671605196367647964898977344 8e-9

sin(245850922)*sin(411557987) = 1,55197946640222300158826053658 08e-17

10^16 * 1,55197946640222300158826053658 08e-17 = 0,15519794664022230015882605365808

Floor(0,15519794664022230015882605365808) = 0
```

Como veis, el exponente negativo (^-17) debe ser mayor que el positivo (^16) para tener éxito.

5. Fuerza bruta

Lo que vamos a hacer a continuación es buscar todos los senos con exponente negativo ^-8 o ^-9 de enteros entre 1 y 10^10, y vamos a cruzar los resultados para determinar todos los resultados válidos.

Preparamos el programa y le dejamos trabajar. En principio vamos a filtrar todos los resultados que tengan exponente negativo y luego ya aislaremos los que nos interesan. Esto lo hago por curiosidad.



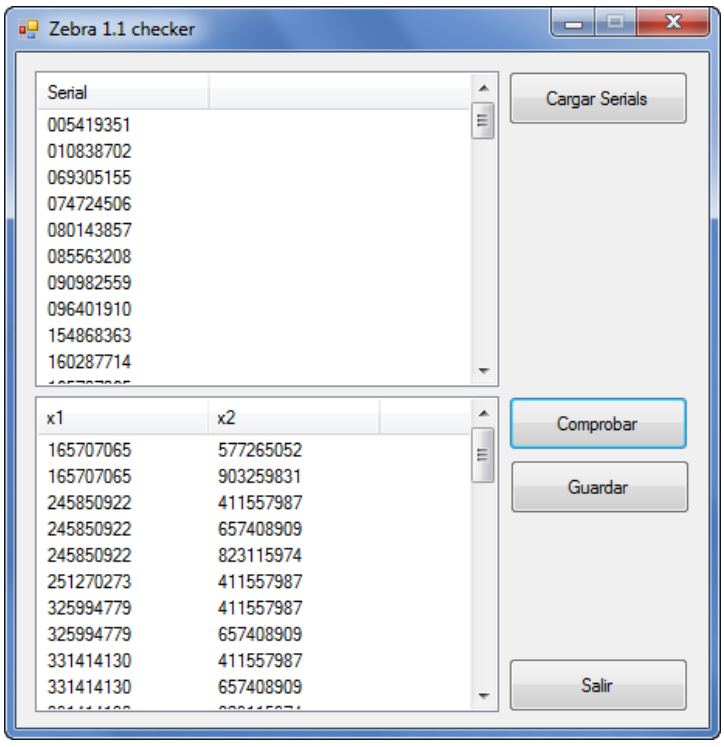
La fuerza bruta nos arroja **63663 resultados** con exponente negativo entre ^-5 y ^-9, de los cuales solamente nos quedamos con **65**, que son los comprendidos a exponentes de entre ^-8 y ^-9. Los números mágicos son los siguientes:

5419351	235012220	**411557987**	646570207	**823115974**
10838702	240431571	416977338	651989558	828535325
69305155	**245850922**	422396689	**657408909**	833954676
74724506	251270273	480863142	662828260	892421129
80143857	256689624	486282493	668247611	897840480
85563208	315156077	491701844	726714064	**903259831**
90982559	320575428	497121195	732133415	908679182
96401910	325994779	502540546	737552766	914098533
154868363	331414130	566426350	742972117	977984337
160287714	336833481	571845701	748391468	983403688
165707065	342252832	577265052	753810819	988823039
171126416	400719285	582684403	812277272	994242390
176545767	406138636	588103754	817696623	999661741

Los rojos son exponentes ^-9, el resto ^-8.

La mayoría de estos números solo valen con ciertas combinaciones, de hecho, ninguno vale para todos. Esto se debe, a parte del propio exponente, a que hay senos positivos y negativos y para hacer válido a un seno negativo hay que combinarlo con otro negativo. Esto último se debe únicamente a la interpretación que hace el Crackme.

Finalmente cruzamos los resultados y obtenemos **44** combinaciones de seriales válidos que si obviamos repeticiones se reducen a la mitad.



Combinaciones válidas:

165707065 - 577265052	406138636 - 411557987	411557987 - 823115974	737552766 - 657408909
165707065 - 903259831	411557987 - 245850922	577265052 - 165707065	737552766 - 823115974
245850922 - 411557987	411557987 - 251270273	657408909 - 245850922	742972117 - 411557987
245850922 - 657408909	411557987 - 325994779	657408909 - 325994779	742972117 - 657408909
245850922 - 823115974	411557987 - 331414130	657408909 - 331414130	817696623 - 411557987
251270273 - 411557987	411557987 - 406138636	657408909 - 411557987	823115974 - 245850922
325994779 - 411557987	411557987 - 657408909	657408909 - 737552766	823115974 - 331414130
325994779 - 657408909	411557987 - 662828260	657408909 - 742972117	823115974 - 411557987
331414130 - 411557987	411557987 - 737552766	657408909 - 823115974	823115974 - 657408909
331414130 - 657408909	411557987 - 742972117	662828260 - 411557987	823115974 - 737552766
331414130 - 823115974	411557987 - 817696623	737552766 - 411557987	903259831 - 165707065

6. Conclusiones

Tanto trabajo para al final obtener 22 combinaciones posibles. Finalmente comentar que si aLoNg3x no habría puesto el límite en 10^10, habría soluciones infinitas.

7. Enlaces

- Crackme
- Archivos del proyecto (programas, logs...)
- Solución de cronos
- Instrucciones FPU
- Fracción continua o cual es la mejor aproximación (Gaussianos)
- Circunferencia unitaria (Video)
- Función Seno
- Función Floor

8. Crackeando Crackmes by deurus

- https://mega.co.nz/#F!88BRwYoT!O0TzTSZYCdczKLOrfrOyGw
- Lolabits.es/blogcracking (Clave: **blogcrackhack**)