INTRODUCCION AL CRACKING CON OLLYDBG PARTE 2

Luego de haber visto a grandes trazos, la ubicación y las principales partes del OLLYDBG, debemos aprender el sistema de numeración utilizado y cual es el concepto de stack aunque sea para tener una idea, luego profundizaremos.

SISTEMAS NUMERICOS

Los tres sistemas numéricos que más se utilizan son el binario el decimal y el hexadecimal.

El concepto básico que deben tener de ellos es el siguiente:

BINARIO: Se representa los números con dos caracteres el 0 y 1 por eso se llama BINARIO.

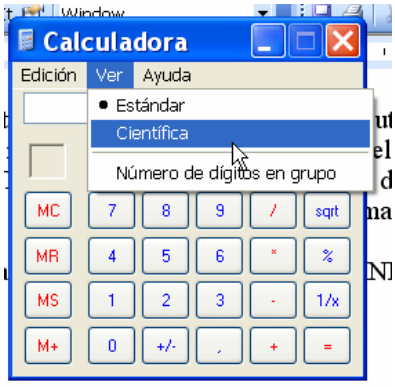
DECIMAL: Se representa todos los números con 10 caracteres (del 0 al 9) por eso se llama decimal.

HEXADECIMAL: Se representa todos los números con caracteres del 0 al F (del 0 al 9, mas A, B, C, D, E y F, o sea serian 16 caracteres en total).

Normalmente a partir de aquí cuando diga un número y no diga a que sistema de numeración pertenece es porque es HEXADECIMAL que es el que utiliza OLLYDBG, si son decimales o binarios aclarare expresamente.

Existen formulas matemáticas que no utilizaremos aquí, para convertir números de un sistema al otro, que no son muy simpáticas, pero llegado el momento, un cracker realmente usa la CALCULADORA DE WINDOWS, que es lo mas rápido y directo y no va a ponerse a hacer cuentas de potencias, sumas etc para convertir un numero de un sistema a otro.

Abramos la CALCULADORA DE WINDOWS y preparémosla

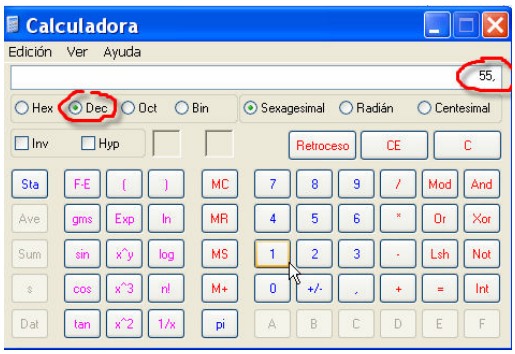


Allí vemos en el menú VER como se puede cambiar a CIENTIFICA.

Allí vemos que por DEFAULT arranca en DECIMAL, y al lado tiene la opción de cambiar a HEXADECIMAL (HEX), OCTAL (OCT) y BINARIO (BIN).

El Octal que representa la numeración con 8 caracteres no es muy usado en cracking pero la calculadora trae dicha opción incluida, si se llegara a necesitar.

Pues para pasar un número de un sistema a otro es muy sencillo, pongo la calculadora en el sistema del número que quiero cambiar, por ejemplo si quiero cambiar 55 de DECIMAL a HEXA, pongo la calculadora en DECIMAL y tipeo 55.



Ahora cambio la calculadora a HEXA y automáticamente me lo convierte a ese sistema de numeración

Ahí esta 55 decimal equivale a 37.

Resalte las letras A,B,C,D,E,F para ver que al pasar la calculadora a HEXA se nos habilita la posibilidad de teclear las mismas, que estaban deshabilitadas en modo DECIMAL.

Creo que esta es la forma mas practica de manejarse con los sistemas de numeración, y poder pasar valores de uno a otro sin grandes complicaciones.

NUMEROS NEGATIVOS en HEXADECIMAL

Esto es lo mas duro de entender por lejos tratemos de ir despacio.
En el sistema de numeración hexadecimal, como podemos representar los números negativos, ya que no se puede poner el signo menos delante como hacemos en la tradicional numeración decimal ?
Como hago para representar en formato hexadecimal -1 por ejemplo ?

Pues aquí viene el problema y espero que se entienda.

Solo tenemos la posibilidad de escribir en hexadecimal desde 00000000 hasta FFFFFFFF, como representaríamos los números negativos?

Pues bien a un genio se le ocurrió que en vez de representar desde 00000000 hasta FFFFFFFF todos números positivos, usaríamos la mitad para los positivos y la otra mitad para los negativos.

Los números positivos van entonces desde 00000000 hasta 7FFFFFFF y los negativos desde 80000000 hasta FFFFFFFF.

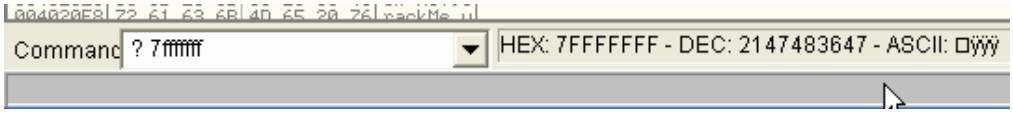
POSITIVOS

000000000 es igual a 0 decimal
000000001 es igual a 1 decimal
……..
……..
7FFFFFFF es igual a 2147483647 decimal (que seria el máximo positivo)

NEGATIVOS

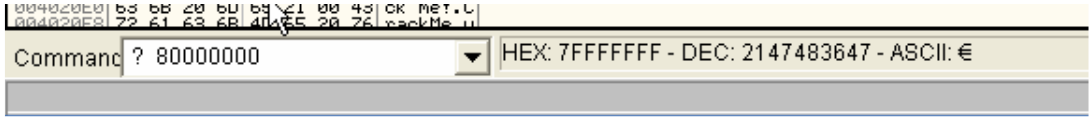
FFFFFFFF seria el -1 decimal
FFFFFFFE seria el -2 decimal
……
……
80000000 seria -2147483648 decimal (que sería el máximo negativo)

Podemos probar averiguar en la Command Bar el valor de 7FFFFFFF para ello usamos el signo de interrogación y a continuación el valor que deseamos pasar a decimal,



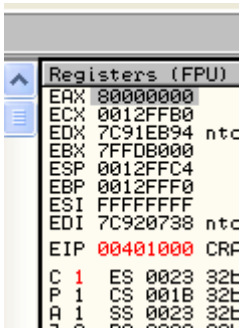
Vemos a la derecha que nos da el valor DECIMAL correspondiente, que es 2147483647 sin problemas.

Ahora cuando deseamos averiguar el valor de 80000000 que es negativo, vemos que no nos lo muestra sigue dando el resultado para 7FFFFFFF (esto es un bug de la Command Bar), así que como podemos hallar su valor en OLLYDBG ?

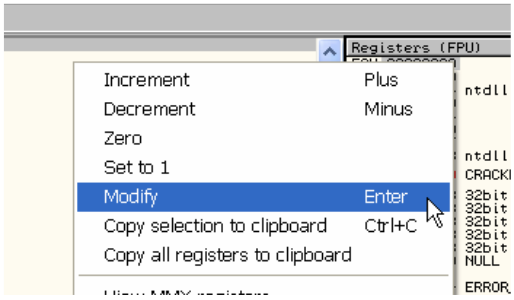


Con este pequeño truquito.

Vamos a los registros y marcamos EAX



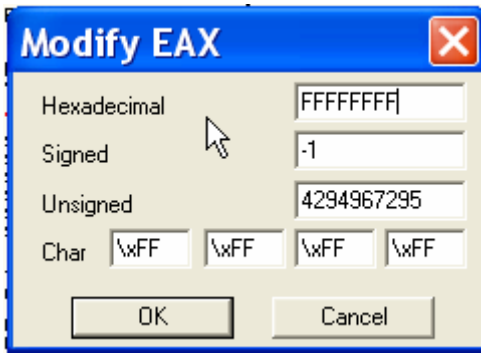
Luego hacemos CLICK DERECHO-MODIFY



Nos aparece una ventana en la que podemos colocarle a EAX el valor que queremos, así que aprovechamos y usamos dicha ventana para hacer las conversiones, en el primer renglón tipeamos el valor HEXADECIMAL que queremos convertir y en el segundo renglón nos aparcera el resultado en DECIMAL.

En este caso vemos que 80000000 corresponde al valor -214783648 decimal.

Si averiguo el valor de FFFFFFFF allí vale -1 decimal.



Por lo tanto en la ventana de modificar un registro podemos averiguar el valor de números negativos perfectamente, luego para salir podemos CANCELAR así no realizamos ningún cambio.

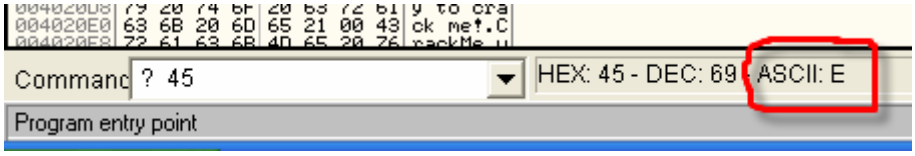
CARACTERES ASCII

Uno de los temas que debemos conocer también es la forma en que nuestro sistema escribe datos en la pantalla, para eso asigna a cada carácter un valor hexadecimal, de forma que puede interpretar los mismos como si fueran letras, números símbolos etc.

De la teoría de ASM de Caos Reptante le copiamos la tablita jeje allí vemos a continuación el valor decimal, en la segunda columna el valor hexadecimal y en la tercera el carácter o sea por ejemplo si quiero escribir un espacio en OLLY, tengo que usar el 20 o 32 decimal, cualquier carácter que necesitemos, sea letra o numero podemos verlo en esta tablita.

Dec.	Hex.	Carac	Dec.	Hex.	Carac	Dec.	Hex.	Caract
32	20	esp	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F		127	7F	□

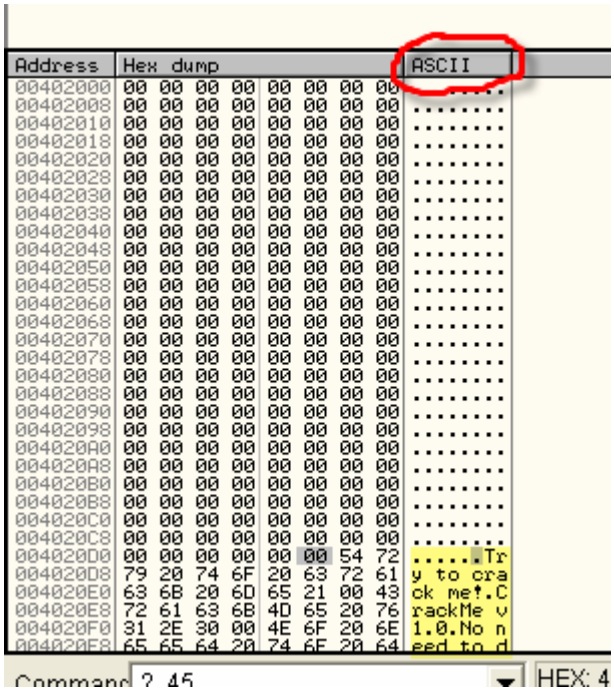
Por lo demás la command bar cuando averiguamos el valor de un numero hexadecimal, nos proporciona también el carácter ASCII correspondiente si tuviera, veamos un ejemplo tipeemos en la command bar.



Vemos que 45 corresponde a la letra E mayúscula, si en la tabla anterior buscamos 45 en la columna del medio que corresponde a hexa vemos que es la letra E

69	45	E

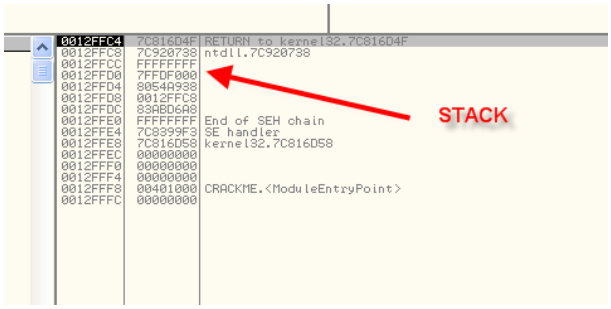
Por lo demas en la ventana del DUMP del OLLYDBG, tenemos una columna que muestra los caracteres ASCII, si miramos allí mismo en el crackme de CRUEHEAD la ventana del DUMP.



Vemos que al lado de la columna que representa los valores hexadecimales, esta la columna ASCII, donde podemos ver resaltadas algunas cadenas de texto compuestas por combinaciones apropiadas de caracteres ASCII.

QUE ES EL STACK O PILA

El stack o pila es una zona de la memoria, en la cual se van guardando datos que mas adelante deben ser recuperados.
El nombre PILA es porque asemeja un mazo o pila de cartas o barajas que se encuentran en una mesa.
En dicho mazo, si agregas una nueva carta solo podes hacerlo arriba de la pila y si quiero sacar una será la de más arriba de la pila de cartas.
Esa es la característica principal del stack es como un mazo de cartas, la carta que agregas a la pila ira arriba, y será la primera que salga, cuando quites una.
Ya veremos mas adelante en la explicación de las instrucciones la forma de modificar o agregar y quitar cartas en nuestro mazo, o sea nuestro querido STACK que como recordamos del tute anterior esta representado en la parte inferior derecha del OLLYDBG.



Bueno creo que ya tienen bastante para quemarse un rato mas la cabeza nos vemos en la parte 3 donde explicaremos que son los registros y los flags y para que sirven.

Hasta la parte 3
Ricardo Narvaja
08 de noviembre de 2005