

V

Algo básico de Hex

Es muy probable que más de uno se asuste al principio al oír hablar sobre edición hexadecimal ¿Qué es eso? ¿Cómo se hace? ¿Para qué sirve? Aunque la práctica pueda llegar a ser muy complicada, la teoría es bastante sencilla. Realmente podría resumir esto en un par de frases, pero me interesa bastante que dominéis, al menos, los conceptos.

•Numeración hexadecimal:

Cuando hablamos de “hexadecimal” estamos hablando simplemente de un sistema numérico. Nosotros, el sistema numérico que manejamos en la escuela, trabajo y, en definitiva, en nuestra vida cotidiana, es el sistema decimal. Este se llama decimal porque consta de 10 caracteres: 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9.

Sencillo, ¿Verdad? Pues ahora viene la pregunta del millón: ¿Por qué el otro sistema se llama “hexadecimal”? Exacto, porque tiene 16 caracteres. Es tan simple como eso. Este sistema tiene los siguientes caracteres: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. Después de la F seguiría con el 10, 11, 12, 13 y así sucesivamente hasta llegar al 1F, que después pasaría al 20, 21, 22...

Os dejo los números hexadecimales hasta el 100. De esa forma podéis ver mejor cómo se van sucediendo y lo comprenderéis sin problemas:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, 100.

Ese es el sistema hexadecimal, sin más vuelta de hoja, se trata de un sistema de numeración como el decimal, pero tiene 6 caracteres o valores más que el otro.

•Conversión de valores

Una cosa que muy a menudo tendréis que hacer, no sólo a la hora de editar el rom hexadecimalmente, sino también para muchas otras cosas más sencillas, será pasar un valor decimal a hexadecimal o viceversa. Es bastante sencillo hacerlo manualmente, pero resultaría muy tedioso, por lo que nos centraremos en aprender a hacerlo con la calculadora de Windows. Rápido y eficaz. En caso de que no uses Windows os formes parte de alguna asociación declarada enemiga de las calculadoras de Windows, busca en internet algún

conversor, hay muchos en la red.

Abrimos la calculadora desde el inicio o la buscamos, en su defecto, en "C:\WINDOWS\System32", donde la aplicación lleva por nombre "calc.exe". Conociendo tu gusto por las matemáticas, probablemente sea la primera vez que la abras o que la tengas en modo sencillo. Pero nosotros la necesitamos científica, así que vamos a *Ver>Científica* y ya podremos usarla para convertir valores. Si os fijáis, a la izquierda estará marcada la opción "Dec" y a lado tendréis la opción "Hex". Esas son las dos opciones que nos interesan.

Para convertir un valor decimal en hexadecimal, nos aseguramos de que esté marcada la opción "Dec" e introducimos el valor. Seguidamente se selecciona la opción "Hex" y el número se convertirá a hexadecimal. Si por el contrario quieres convertir un valor hexadecimal a decimal, seguiremos el mismo proceso pero esta vez marcaremos la opción "Hex" y una vez introducido el valor, seleccionamos "Dec".

·Bytes

Los bytes son las unidades de información que os vais a encontrar. Un byte presentará una estructura "XX", simbólicamente. Ese "XX" es un valor hexadecimal desde el "00" hasta el "FF". El juego está compuesto por muchísimos bytes que le darán forma. Esto se puede entender mejor con un ejemplo. Una letra no dice nada, pero si juntas varias letras formarás una palabra. Y si juntas varias palabras harás frases que sí tendrán un sentido. Un byte sólo, por lo general, no va a decir nada, pero si juntas varios bytes formarás esa "frase" que tendrá sentido para el juego.

El juego, por tanto, hará lo que le digan los bytes. El ejemplo más claros son los scripts. Un script de XSE no es un verdadero script, sino una traducción más entendible. Los comandos de los scripts son valores hexadecimales. Por ejemplo, el comando "lock" no existe como tal. El comando real es "6A", pero XSE lo transforma en "lock" para que sea más fácil entenderlo y modificarlo. Y eso hace con todos los comandos.

·Offset y pointers

Un offset es la dirección en la que se encuentra un determinado byte. Sirve para determinar su posición, para poder localizarlo. Imagina que todos esos bytes que hay en el ROM son casas. Cada casa tiene una dirección, es decir, tiene una calle y un número. Esa dirección sirve para saber dónde está la casa. El pointer es lo mismo, es la dirección de un byte. Es un valor hexadecimal. Por ejemplo, el byte número 24, estará en el offset 17. Esto se debe a que, como hemos dicho, es un valor hexadecimal y 24 en hexadecimal es 18. Sin embargo se encuentra en el offset 17 y no en el 18. Sencillamente porque se empieza a contar desde 0 y no desde 1.

El pointer, por otro lado, se encarga de llamar a un determinado offset. Cuando en un script ponemos "call 0x(offset)" estamos abandonando el offset en el que estamos para seguir leyendo en otro que estamos indicando. Los pointers, como la mayor parte de la información, estarán permutados ¿Pero cómo se permuta un offset? Sencillamente dándole la vuelta a los bytes. Por ejemplo:

El offset 088200F4 pasaría a ser F4008208. Es decir, 088200F4 > F4008208.

·Edición hexadecimal

Sabiendo todo esto, podemos decir que la edición hexadecimal es simplemente editar el valor de cada byte para que en vez de decir una cosa, diga otra. Según lo que digan esos bytes pasará una cosa u otra en el juego. Así funcionan los scripts. Cuando interactúas con una persona hay un pointer que apunta y nos lleva al offset donde se encuentra nuestro script. Depende de los bytes que pongamos, entonces, el script hará una cosa u otra.

Sé que puede parecer muy complicado. De hecho, puede ser muy complicado. Pero todas las herramientas de ROM hacking lo que hacen es editar los valores de los bytes y cambiar pointers para que pasen otras cosas en vez de las originales.

Recomiendo leer muy tranquilamente éste tutorial y tratar de entenderlo todo lo mejor posible. Léelo dos o tres veces si es necesario, pero intenta que te quede tan claro como sea posible. Luego continúa con el manual, ya que según vayas avanzando te será más sencillo comprender esto. Cuando creas que es el momento adecuado, vuelve a leer éste tema y todo se entenderá muchísimo mejor.

“Algo básico de Hex”

Manual redactado por **Javi4315**.

Queda estrictamente prohibida su distribución.