Virus Indetectables el método MEEPA
by MazarD

Introducción

El mejor modo de hacer indetectable un troyano es modificando la firma de modo que el código siga haciendo lo mismo de forma un poco distinta, esto sin saber ensamblador puede resultar algo complicado, con otros métodos como el método rit y el método que voy a explicar no necesitan de apenas conocimientos sobre ensamblador y son pura mécanica. La ventaja que aporta este método respecto al rit de hackxcrack es que no representa ningún problema que la firma esté en una parte encriptada del código, o en alguna parte en la que introducir código directamente represente un problema para el funcionamiento del programa. Además a diferencia del rit al no tener que interpretar la instrucción correspondiente a cierto punto de la firma se puede automatizar fácilmente.

El método MEEPA

La mayoría de antivirus, como el kaspersky cuando se le pide que analice la memoria no está realmente buscando firmas en ella sino que mira los programas en ejecución y analiza en disco sus ejecutables y librerias cargadas.
Entonces que pasaría si en disco no existe la firma pero en memoria si?
Que el virus no seria detectado y funcionaria perfectamente.
Por lo tanto lo que haremos será:
1-Modificaremos un byte de la firma en disco para que el antivirus no detecte el troyano.
2-Crearemos espacio en el exe para introducir nuestro código.
3-Cambiaremos el punto de entrada del exe para que inicialmente se ejecute nuestro código.
4-Desde nuestro código cambiaremos EN MEMORIA el byte que habíamos modificado por su valor original.
5-Saltaremos desde nuestro código al punto de entrada real del programa.

Herramientas necesarias

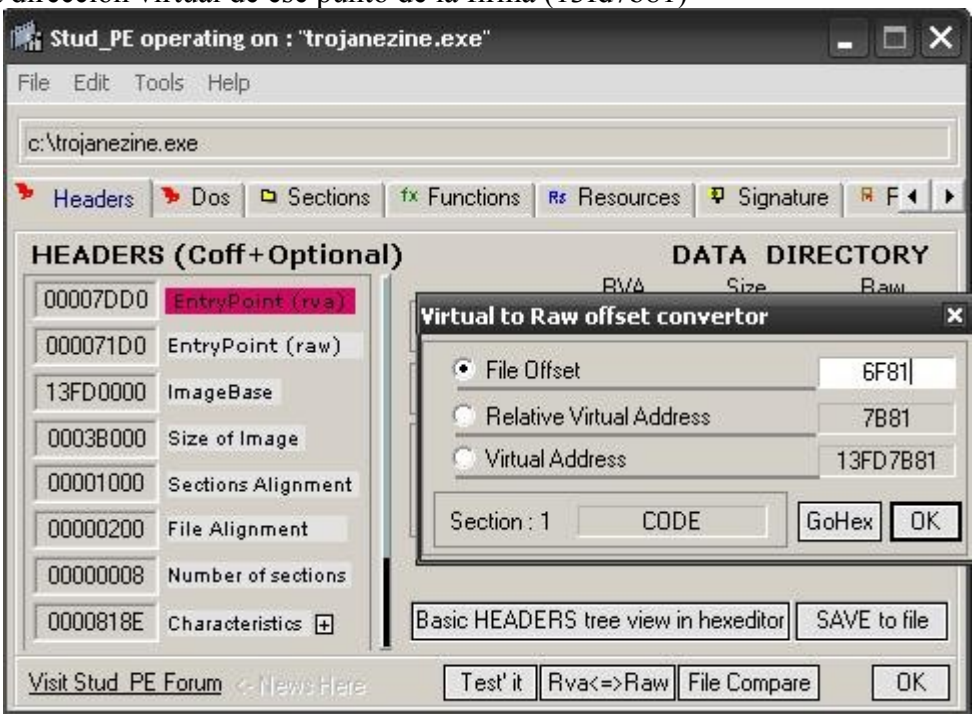
-El troyano que vamos a modificar.
Será el server del nuclearrat 1.0 configurado para que conecte a 127.0.0.1 y se instale en c:\windows\trojanezine.
-El studpe para modificar la cabecera pe32 del exe.
-Un editor hexadecimal.
-Partimos de la base que la firma de kaspersky para este troyano es desde 6f81 hasta aproximadamente 7147.

Junto con la ezine tienes el troyano para testear y el studpe, están los dos encriptados con contraseña ya que el nuclearrat lo detectarán todos los antivirus y el stud_pe puede que algunos lo detecten como hack tool. La contraseña para descomprimir el paquete es "método meepa"
Editores hexadecimales hay miles por internet, yo aconsejo hdd hex editor, no está nada mal y tiene una versión gratuita.

Recolectando información

PE32 significa portable executable y fue diseñado por Microsoft para tener el mismo formato de archivos ejecutables para todos los windows. Nosotros no necesitamos conocer mucho sobre él, lo que nos interesa es modificarlo para poder añadir código al ejecutable y cambiar el punto de entrada de este.

Con el troyano cargado en el studpe vemos toda la información de la cabecera pe32 de nuestro troyano. Cogemos el imagebase(13fd0000) y lo sumamos al entrypoint rva(7dd0) nos dará la dirección absoluta una vez cargado en memoria del punto de entrada del programa (13fd7dd0)
Ahora le damos a rva<=>raw e introducimos en file offset el inicio de la firma (6f81), debajo nos devolverá la dirección virtual de ese punto de la firma (13fd7b81)

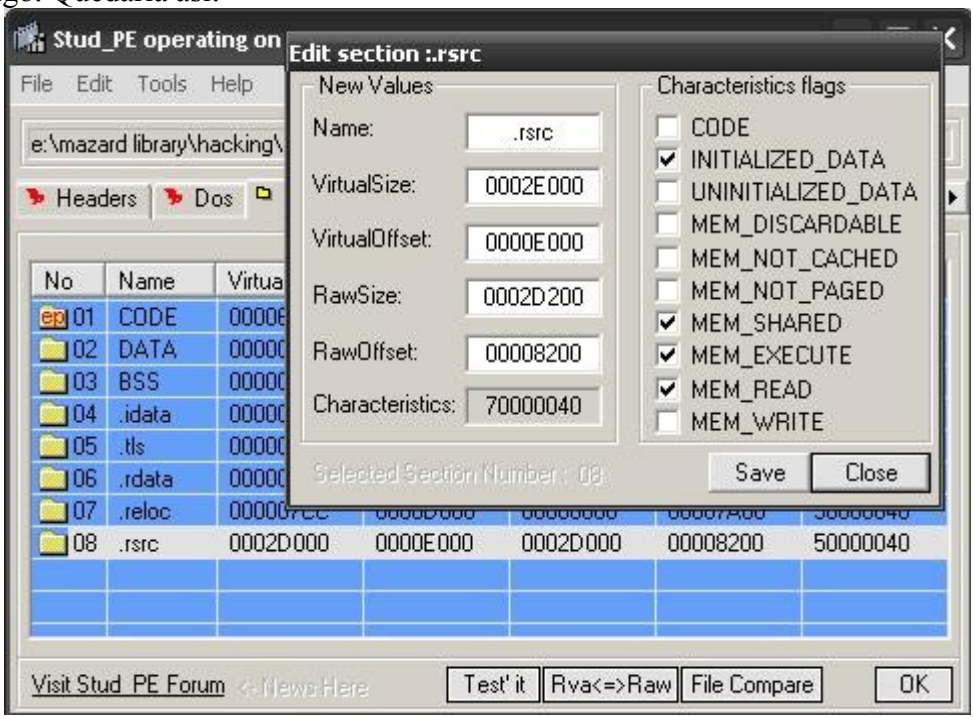


Mas adelante veremos para que queremos esta información.

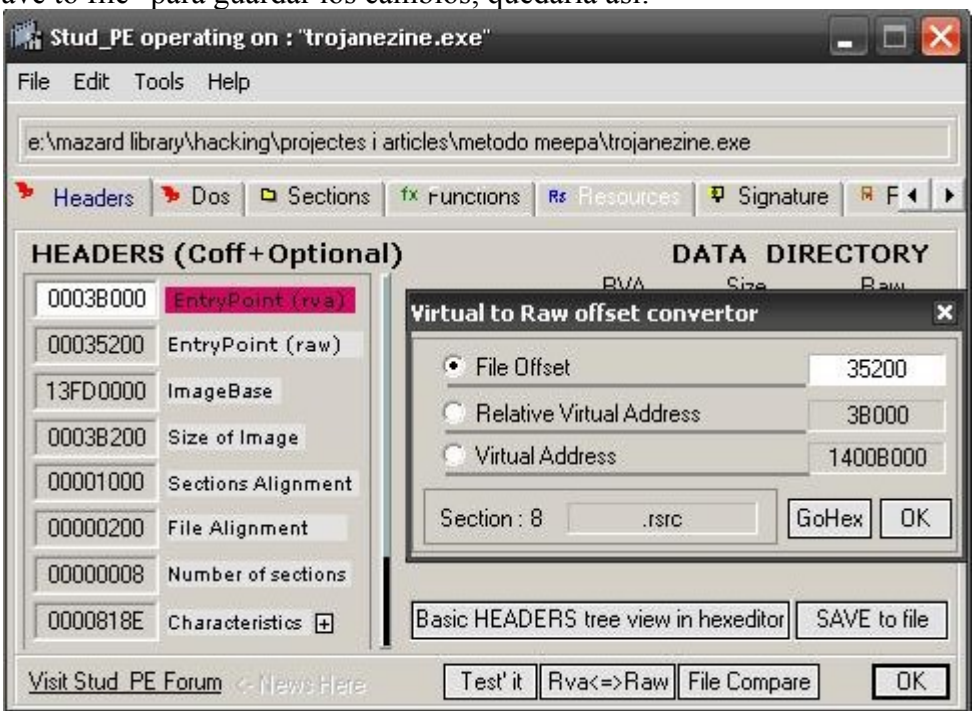
Modificando el PE32

Crear espacio en el exe para nuestro código no es absolutamente necesario, podriamos utilizar espacios vacíos por la alineación del ejecutable. Para los que conozcan un poco el formato PE sería a partir del pointertorawdata+virtualsize y tendríamos un espacio libre de virtualsize-sizeofrawdata, pero es posible que no exista este espacio y el proposito del artículo es que el método sea genérico. Otra posibilidad sería crear una nueva sección para nuestro código pero en algunos casos podría darnos problemas y nos obligaría a reajustar casi todo el archivo, así que vamos a lo mas fácil, ampliaremos el espacio de la ultima sección del ejecutable, emplazando nuestro código justo al final del exe.
Vamos a sections y vemos que en nuestro caso concreto la última sección es ".rsrc" le damos clic derecho "edit header" sumamos el raw offset y el raw size y nos dará 35200, esto es el punto en el que empezará nuestro código, si cargamos el programa con un editor hexadecimal veremos que es justo en el final, si no fuera así significaría que el troyano añade los datos de configuración en el mismo archivo pero fuera del exe, por lo que deberíamos insertar el código que veremos mas adelante justo en este punto dejando lo que ya había en el final.

En raw size le sumaremos 200 (512 bytes en hexadecimal) y en el caso de que el raw size sea mas grande que el virtual size le sumaremos a este ultimo 1000. Decir que el raw size y el virtual size no pueden ser cualquier cosa, tienen que ser multiplos de la alineación del archivo y la sección. Pero nosotros sea cual sea el troyano objetivo con ampliar 512bytes tal y como lo montamos tendremos mas que suficiente. También le damos permisos de ejecución a la sección para que se pueda lanzar nuestro código. Quedaría así:



Ahora hacemos click en Rva<=>raw y en file offset ponemos el inicio de nuestro código 35200 nos dará la dirección virtual relativa (a image base) y lo introducimos en el entrypoint(rva) de este modo conseguimos que lo primero que se ejecute al lanzar el exe sea el trozo que hemos ampliado de la sección (nuestro código). Dado que hemos aumentado el tamaño de una sección en 512 bytes (200) tenemos que añadirlo al tamaño total del exe por lo que size of image quedará en 3B200.
Pulsamos "save to file" para guardar los cambios, quedaría así:



Por último le vamos a dar permisos de escritura en la sección de codigo, vamos de nuevo a sections, clic derecho a la sección “code”, “edit headers” y seleccionamos “mem_write”:
Esto lo hacemos para que se nos permita cuando este el programa cargado en memoria escribir en el punto modificado de la firma su valor real.

Escribiendo nuestras 3 lineas de código

Que hemos hecho hasta ahora?
1-Hemos añadido espacio en la última sección del pe (espacio al final del exe)
2-Hemos cambiado el punto de entrada del programa para que lo primero que se ejecute sea lo que haya en el espacio que hemos añadido
3-Hemos recolectado información necesaria:
Offset en disco del byte de la firma: 6f81
Direccion virtual del byte de la firma: 13FD7B81
Antigua Direccion virtual de entrada al programa: 13fd7dd0

Ya tenemos el exe preparado y la información necesaria. Ahora vamos al tajo, abre el archivo con tu editor hexadecimal.
Nos vamos a la dirección 6f81 que es el punto de la firma a modificar, nos apuntamos el valor que hay ahí (53) y lo sobrescribimos con cualquier cosa (11 mismo). En este punto el programa ya no es detectado, pero petará por dos motivos:
1-Hemos modificado el byte de la firma aleatoriamente y por lo tanto nos hemos cargado el programa.
2-El punto de entrada al programa va a un sitio donde no hay código.

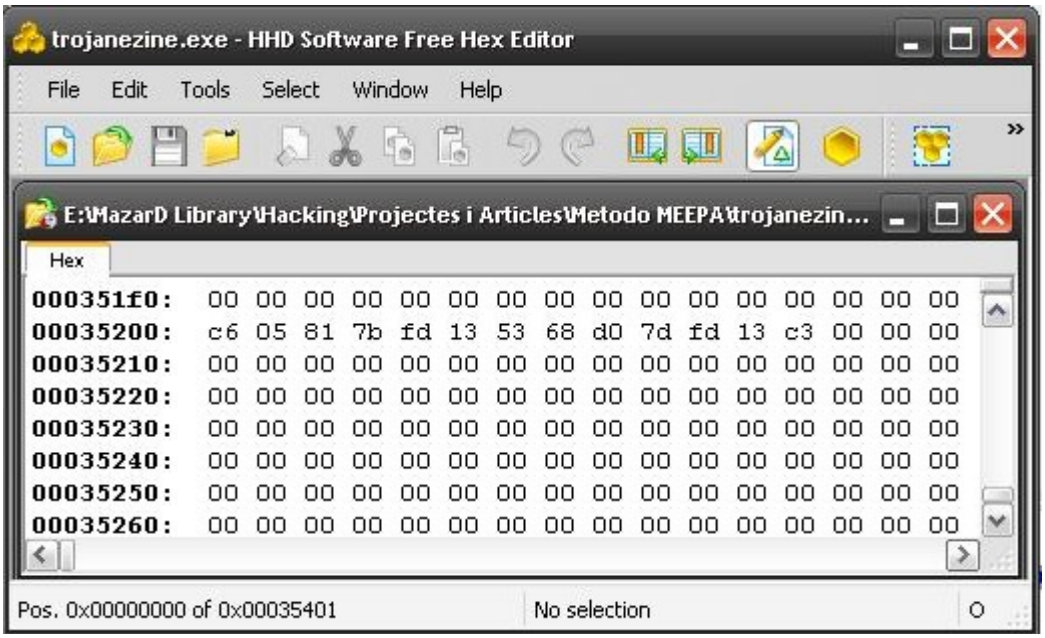
Solucionemos los problemas creando el código que reestablecerá el byte modificado:

mov byte ptr [13fd7b81],53
Con esto se copiará el byte 53 a la posición de memoria que en el archivo habiamos puesto 11.
push 13fd7dd0
ret
Con estas dos instrucciones saltaremos al punto de entrada real del programa (una envía la dirección a la pila y la siguiente coge el último valor puesto en la pila y "salta" a él. No utilizamos el jmp porque podría darnos problemas con la dirección del salto.

La representación hexadecimal de las instrucciones anteriores sería:
c605 817bfd13 53 -->c605 representa "mov byte ptr" el siguiente es la dirección y el siguiente el valor que introducimos
68 d07dfd13 -->68 representa "push" y el siguiente es el valor que introducimos
c3 -->c3 representa ret

Si te fijas las direcciones están al revés cogidas de dos en dos, esto es debido al endian, tampoco entraremos en esto, con que sepas que las direcciones se representan así es suficiente:
13 fd 7b 81 => 81 7b fd 13
13 fd 7d d0 => d0 7d fd 13
3 aa 42 12 => 12 42 aa 03

Ahora vamos a introducir este código con el editor hexadecimal, nos vamos al final del ejecutable (35200) e insertamos el código anterior. También tenemos que recordar que habíamos añadido 512bytes (200) y aunque solo rellenemos unos cuantos estos deben existir fisicamente en el archivo, así que hasta 35400 insertamos nulos.



Guardamos y listo. Troyano indetectable y 100% funcional.
Autor: MazarD
Enlaces de interés
Especificación del formato pe:
www.microsoft.com/whdc/system/platform/firmware/PECOFF.mspx