



SIN CLASIFICAR



Informe Código Dañino CCN-CERT ID-18/16

Ransom.Mischa

Agosto de 2016

SIN CLASIFICAR

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

Quedan rigurosamente prohibidas, sin la autorización escrita del Centro Criptológico Nacional, bajo las sanciones establecidas en las leyes, la reproducción parcial o total de este documento por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares del mismo mediante alquiler o préstamo públicos.

ÍNDICE

1. SOBRE CCN-CERT	4
2. RESUMEN EJECUTIVO	5
3. INFORMACIÓN DE VERSIONES DEL CÓDIGO DAÑINO	5
3.1 EXTENSIONES A CIFRAR	5
3.2 EXTENSIÓN AÑADIDA A LOS ARCHIVOS CIFRADOS	6
3.3 ARCHIVOS DE RESCATE	6
4. CARACTERÍSTICAS DEL CÓDIGO DAÑINO	8
5. DETALLES GENERALES	8
6. PROCEDIMIENTO DE INFECCIÓN.....	9
7. CARACTERÍSTICAS TÉCNICAS.....	9
8. CIFRADO Y OFUSCACION	12
8.1 PRIMERA FASE.....	12
8.2 SEGUNDA FASE.....	12
9. PERSISTENCIA EN EL SISTEMA	13
10.ARCHIVOS RELACIONADOS	13
11.DETECCIÓN	13
11.1 MANDIANT.....	13
11.2 HERRAMIENTAS DEL SISTEMA.....	13
12.DESINFECCIÓN.....	14
13.INFORMACIÓN DEL ATACANTE	15
14.REGLAS DE DETECCIÓN.....	16
14.1 INDICADOR DE COMPROMISO – IOC.....	16
14.2 YARA	17

1. SOBRE CCN-CERT

El CCN-CERT (www.ccn-cert.cni.es) es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN. 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 Centro Nacional de Inteligencia, el RD 421/2004 de regulación del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad.

De acuerdo a todas ellas, el CCN-CERT tiene responsabilidad en ciberataques sobre **sistemas clasificados** y sobre sistemas de las **Administraciones Públicas** y de **empresas y organizaciones de interés estratégico** para el país. 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.

2. RESUMEN EJECUTIVO

El presente documento recoge el análisis del código dañino "**Ransom.Mischa**", el cual ha sido diseñado para instalarse en el sistema, cifrar ciertos archivos y extorsionar a la víctima mostrando una notificación sobre el procedimiento de pago para recuperar los archivos cifrados.

El código dañino viene embebido en el ejecutable de otro código dañino "**Ransom.Petya**" en su segunda versión. Aunque se está distribuyendo inicialmente de esta forma, sería trivial poder desarrollar un ejecutable independiente.

Este código dañino se distribuyó inicialmente por correos electrónicos a departamentos de Recursos Humanos de ciertas organizaciones, ubicadas en Alemania. Posteriormente su distribución se hizo masiva.

3. INFORMACIÓN DE VERSIONES DEL CÓDIGO DAÑINO

El código dañino solo tiene una versión.

3.1 EXTENSIONES A CIFRAR

El código dañino cifra los archivos que posean alguna de las siguientes extensiones:

.3g2	.cfg	.jsp	.p12	.eip	.jge	.ptx	.m4v
.3gp	.config	.bak	.cat	.fff	.tga	.ape	.rm
.asf	.wb2	.dat	.inf	.iiq	.jfif	.aif	.srt
.asx	.msg	.pst	.mui	.k25	.emf	.wav	.aepx
.mpg	.azw	.eml	.props	.crwl	.3dm	.ram	.camproj
.mpeg	.azw1	.xps	.idl	.bay	.3ds	.ra	.dash
.avi	.azw3	.sqlite	.result	.sr2	.max	.m3u	.txt
.mov	.azw4	.sql	.localstorage	.ari	.obj	.movie	.doc
.flv	.lit	.js	.ost	.srf	.a2c	.mp1	.docx
.wma	.apnx	.jar	.default	.arw	.dds	.mp2	.docm
.wmv	.mobi	.py	.json	.cr2	.pspimage	.mp3	.odt
.ogg	.p12	.wpd	.db	.raw	.yuv	.mp4	.ods
.swf	.p7b	.crt	.sqlite	.rwl	.zip	.mp4v	.odp
.jpg	.p7c	.csv	.log	.rw2	.rar	.mpa	.odf
.jpeg	.pfx	.prf	.bat	.r3d	.gzip	.mpe	.odc
.frm	.pem	.cnf	.ico	.3fr	.vmdk	.mpv2	.odm
.wdb	.cer	.indd	.dll	.ai	.mdf	.rpf	.odb

.ldf	.key	.number	.exe	.eps	.iso	.vlc	.rtf
.myi	.der	.pages	.x3f	.pdd	.bin	.m4a	.xlsm
.vmx	.mdb	.lnk	.srw	.dng	.cue	.aac	.xlsb
.txt	.htm	.po	.pef	.dxf	.dbf	.aa	.xlk
.xml	.html	.dcu	.raf	.dwg	.erf	.aa3	.xls
.xsl	.class	.pas	.orf	.psd	.dmg	.amr	.xlsx
.wps	.java	.dfm	.nrw	.ps	.toast	.mkv	.pps
.cmf	.cs	.directory	.nef	.png	.vcd	.dvd	.ppt
.vbs	.asp	.pbk	.mrw	.jpe	.ccd	.mts	.pptm
.accdb	.aspx	.yaml	.mef	.bmp	.disc	.qt	.pptx
.ini	.cgi	.dtd	.kdc	.gif	.nrg	.vob	.pub
.cdr	.h	.rll	.dcr	.tiff	.nri	.3ga	.epub
.svg	.cpp	.lib	.crw	.gfx	.cdi	.ts	.pdf
.conf	.php	.cert					

Una característica del código dañino es que cifra archivos ejecutables, cosa que otras familias de *ransomware* no lo hacen sobre este tipo de archivos.

3.2 EXTENSIÓN AÑADIDA A LOS ARCHIVOS CIFRADOS

El código dañino añade una extensión a los archivos cifrados, ésta se compone de una parte del identificador único del sistema comprometido, en concreto, del segundo al quinto carácter, ambos inclusive.

foto_familiar.jpg.cWrU

3.3 ARCHIVOS DE RESCATE

El código dañino crea los siguientes archivos con información sobre el secuestro del sistema, el procedimiento a seguir para efectuar el pago y recuperar los archivos:

YOUR_FILES_ARE_ENCRYPTED.TXT
YOUR_FILES_ARE_ENCRYPTED.HTML

El texto de los archivos es el siguiente, por ejemplo:

You became victim of the MISCHA RANSOMWARE!

The files on your computer have been encrypted with an military grade encryption

algorithm. There is no way to

restore your data without a special key. You can purchase this key on the darknet page shown in step 2.

To purchase your key and restore your data, please follow these three easy steps:

1. Download the Tor Browser at "<https://www.torproject.org/>". If you need help, please google for "access onion page".
2. Visit one of the following pages with the Tor Browser:

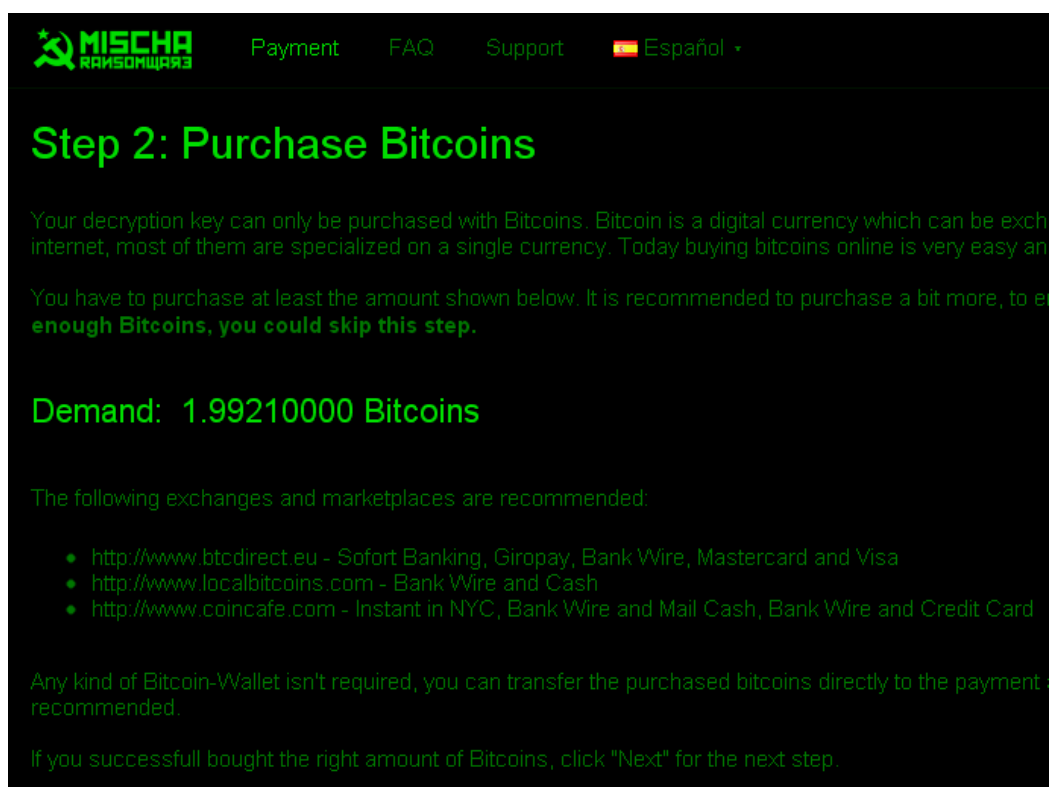
http://mischapuk6hyrn72.onion/<id_de_cinco_caracteres>

http://mischa5xyix2mrhd.onion/<id_de_cinco_caracteres>

3. Enter your personal decryption code there:

<identificador_único>

Si se accede mediante un navegador con acceso a la red Tor a cualquiera de las dos direcciones indicadas para el rescate, se puede ver el procedimiento de pago:



The screenshot shows the 'Step 2: Purchase Bitcoins' page of the Mischa ransomware. The page has a dark background with red text. At the top, there is a navigation bar with the Mischa logo (a hammer and sickle) and the word 'MISCHA' in red, followed by 'РАНСОМЩИК' in Cyrillic. To the right of the logo are links for 'Payment', 'FAQ', 'Support', and a language selector set to 'Español'. The main heading is 'Step 2: Purchase Bitcoins' in large red font. Below it, the text explains that the decryption key can only be purchased with Bitcoins and that buying them online is easy. It states that the user must purchase at least the amount shown, and recommends buying a bit more. The demand is listed as 'Demand: 1.99210000 Bitcoins'. Below this, it lists recommended exchanges and marketplaces: btcdirect.eu, localbitcoins.com, and coincafe.com. It also mentions that any kind of Bitcoin-Wallet is not required and that the purchased Bitcoins can be transferred directly to the payment address. Finally, it instructs the user to click 'Next' if they have successfully bought the right amount of Bitcoins.

Ilustración 1. Procedimiento de pago

4. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El código dañino examinado posee las siguientes características:

- Carga el código dañino en el sistema.
- Crea archivos acerca del secuestro del sistema y el procedimiento a seguir para el pago del rescate.
- Enumera las unidades de disco del sistema y los recursos de red.
- Cifra ciertos archivos en las unidades y recursos detectados.
- Muestra el archivo de texto y el HTML acerca del secuestro.

5. DETALLES GENERALES

La muestra analizada se corresponde con la siguiente firma MD5:

7fb4364669700327aea423f10c87fc3a (Dropper)
3e859ce4de1ae1b7fa3841eb073eaf68 (Ransom.Mischa)

Los binarios tienen formato PE (*Portable Executable*), es decir, son ejecutables para sistemas operativos Windows, concretamente para 32 bits. En las muestras analizadas se ha podido observar que las fechas internas de creación de los programas datan del 9 de mayo del 2016.

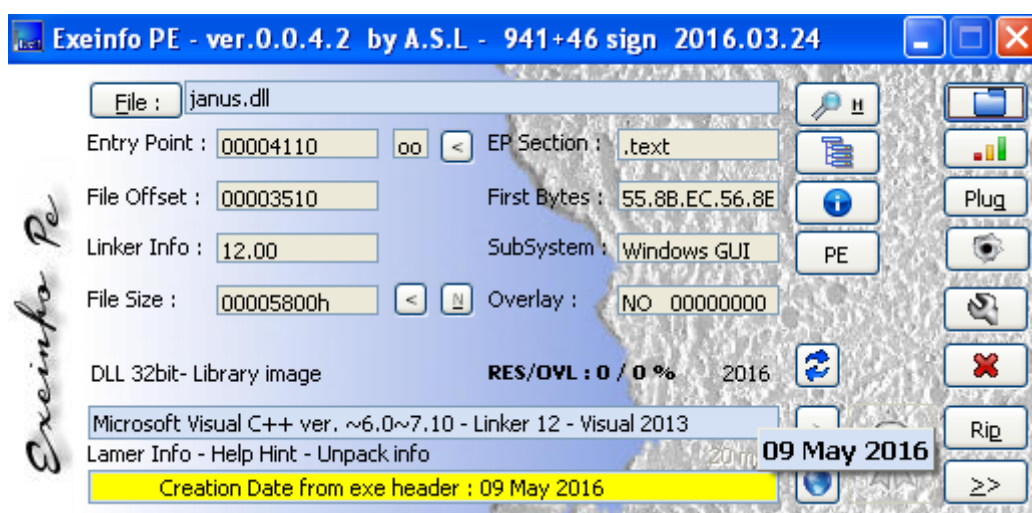


Ilustración 2. Detalles del binario

6. PROCEDIMIENTO DE INFECCIÓN

La infección en el equipo se produce al ejecutar el fichero que contiene el código dañino. Una vez ejecutado realiza las siguientes acciones en el equipo de la víctima:

- El "dropper" descifra en memoria una primera librería que lleva otro "dropper", que se encarga de comprobar si el proceso puede ejecutarse como administrador. En caso de que así sea, procede a ejecutar el código dañino "Ransom.Petya", en caso contrario, ejecuta el código dañino "Ransom.Mischa".
- Crea determinados archivos en el sistema comprometido acerca del secuestro de los archivos y el procedimiento a seguir para recuperarlos.
- Enumera las unidades del sistema y recursos de red. En todas ellas busca archivos que cumplan un patrón por extensión y los cifra.
- Muestra al usuario el archivo de texto y el HTML creados con la información del secuestro.

7. CARACTERÍSTICAS TÉCNICAS

El código dañino inicia su proceso de infección ejecutándose desde su "dropper", el cual descifra y ejecuta en memoria una librería. Está librería lleva embebidos el código dañino "Ransom.Petya" y el código dañino "Ransom.Mischa".

El paso inicial de la ejecución de dicha librería es comprobar si el usuario con el que se ejecutó puede escalar privilegios o no. Para ello, el código dañino accede al token de su proceso y obtiene la información de la clase 0x14 ("TokenElevation"). Esta clase indica si el proceso puede elevar privilegios o no.

```

push      8
call      [<&KERNEL32.GetCurrentProcess>]
push      eax
call      [<&ADVAPI32.OpenProcessToken>]
test      eax, eax
je        short 003BD84D
lea       eax, [esp+14]
mov       dword ptr [esp+14], 4
push      eax
push      4
lea       eax, [esp+18]
push      eax
push      14
push      dword ptr [esp+1C]
call      [<&ADVAPI32.GetTokenInformation>]
test      eax, eax
cmovne    esi, [esp+18]
  
```

Ilustración 3. Comprobación de privilegios

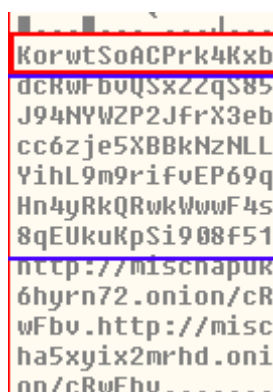
Si el proceso puede elevar privilegios, procederá a ejecutarse de nuevo mediante el comando "runas", descifrando y ejecutando el código dañino "Ransom.Petya".

En el caso de que no pueda elevar privilegios, procederá a leer su última sección llamada ".xxxx" y a extraer de ella el código dañino "Ransom.Mischa". Una vez extraído, se comprueba su integridad mediante la función "ChecksumMappedFile", siguiendo su ejecución en el caso de que el checksum sea correcto o finalizando en el caso de que no lo sea.

Posteriormente se calcula una clave aleatoria, que será usada en la segunda fase del cifrado de los archivos. Esta clave es cifrada mediante *Criptografía de Curva Elíptica*¹ de forma que, en principio, solo los autores del código dañino pueden descifrarla. Esta clave cifrada es a la que los autores del código llaman identificador único y es usado para poder generar un descifrador.

El código dañino tiene guardada la clave en texto plano en memoria durante todo el proceso de cifrado, por lo que mientras dure el proceso, se podría obtener la clave de la memoria, esto abre la puerta a poder descifrar los archivos sin pagar el rescate. Una vez que finaliza el proceso de cifrado, la clave se pierde y, actualmente, no existe otra forma de recuperar los archivos si no es pagando el rescate.

Finalmente, escribe el identificador único en la última sección del código dañino recién extraído de la sección ".xxxx".



```

KorwtSoACPrk4Kxb
dCKwFbUQSxZzqS85
J94NYWZP2JfrX3eb
cc6zje5XBBkNzNLL
YihL9m9rifvEP69q
Hn4yRkQRwkWwF4s
8qEUKuKpSi908f51
ncttp://miscnapuk
6hyrn72.onion/cR
wFbv.http://misc
ha5xyix2mrhd.oni
on/cRwFbv.....
  
```

Ilustración 4. Clave en texto plano, identificador único y enlaces de pago

A continuación, procede a inyectarse en un proceso para ocultar su actividad. Para ello, enumera todas las ventanas del sistema comprometido mediante la función "EnumWindows" y por cada una de ellas procede a obtener el identificador del proceso de la ventana al cual pertenece e intenta abrir el proceso con todos los permisos. De esta forma, va obteniendo el listado de procesos en los que puede inyectarse sin problemas.

Una vez finalizada la comprobación, por cada proceso obtenido, trata de abrirlo con todos los permisos y escribir en su memoria el código dañino para ejecutarlo, creando un hilo remoto sobre el mismo proceso. Esto se realiza hasta que uno de los procesos reciba el código dañino y se ponga en ejecución sin problemas. Una vez que lo consigue, el "dropper" finaliza su ejecución.

¹ https://es.wikipedia.org/wiki/Criptograf%C3%ADa_de_curva_el%C3%ADptica

A continuación la primera acción del código dañino inyectado, es descifrar su última sección, llamada ".xxxx", obteniendo el identificador único y las direcciones de pago.

Posteriormente, enumera todas las unidades del sistema mediante la función "GetLogicalDriveStringsA" y obtiene el tipo de unidad que es, para comprobar que sea de tipo "DRIVE_FIXED" (un disco duro), "REMOVABLE_DRIVE" (unidades extraíbles) o "DRIVE_REMOTE" (unidades de red compartidas y montadas).

Por cada una de las unidades encontradas, el código dañino procede a enumerar todos los archivos que tengan alguna de las extensiones indicadas en el apartado [Extensiones a Cifrar](#), exceptuando explícitamente las carpetas que tengan alguno de los siguientes nombres:

Windows	Internet Explorer
\$Recycle.Bin	Temp
Microsoft	Local
Mozilla Firefox	LocalLow
Opera	Chrome

Una de las diferencias de este código dañino sobre otros tipos de *ransomware*, es su capacidad de interactuar con la carpeta "Archivos de programa" o "Program Files", cifrando también los programas instalados.

El código dañino guardará en memoria todas las rutas y archivos para reducir los tiempos de acceso a disco y hacer el proceso de cifrado mucho más rápido.

Posteriormente, comienza el proceso de cifrado de los archivos encontrados anteriormente. Primero comprueba su existencia y posteriormente procede a crear un evento sin nombre y sin estar señalizado, mediante la función "CreateEventA". Después, comienza a leer el archivo usando "ReadFile" en bloques de 0x400 bytes, lo cifra y lo vuelve a escribir en el archivo. El código dañino no crea ninguna copia del archivo que está cifrando para evitar herramientas forenses de recuperación de archivos.

Una vez cifrado el archivo se le cambia el nombre, usando la función "MoveFileA", añadiéndole a su nombre y extensión originales una extensión más con cuatro caracteres, tomados del segundo al quinto caracteres del identificador único. Un ejemplo podría ser:

foto_familiar.jpg.cWrU

A continuación, en la carpeta donde ha cifrado el archivo, crea los dos ficheros con la información sobre el secuestro y el procedimiento de pago. Estos archivos son creados en cada carpeta donde haya podido cifrar un archivo como mínimo. Una vez cifrados todos los archivos del sistema, el código dañino finaliza.

8. CIFRADO Y OFUSCACION

El código dañino trabaja en dos fases a la hora de cifrar los archivos utilizando dos algoritmos distintos en bloques de 128 bits (16 bytes).

8.1 PRIMERA FASE

La primera fase de cifrado es una operación XOR en bloques de 16 bytes de forma que, el primer bloque es cifrado con un valor generado aleatoriamente mediante la función "CryptGenRandom" de la librería de Microsoft "crypt32.dll". A partir de ahí, sobre cada bloque utiliza el resultado del cifrado del bloque anterior como clave XOR.

La clave XOR inicial, se almacenará posteriormente al final del archivo cifrado junto con otra información.

8.2 SEGUNDA FASE

La segunda fase del cifrado utiliza la clave maestra generada inicialmente y de la que se deriva el identificador único. El algoritmo utilizado es un AES ligeramente modificado por los autores del código dañino, por lo que no se utiliza la librería "crypt32.dll" para cifrar.

La clave maestra permanece en texto plano en memoria, mientras el código dañino esté en ejecución. Una vez que finalice el proceso la clave se pierde y sólo los autores del código dañino pueden descifrar los archivos a través del identificador único para obtener la clave maestra.

El código dañino lee los archivos del equipo infectado en bloques de 1024 bytes y los cifra en bloques de 16 bytes, que escribe en el mismo archivo del que se lee, de forma que nunca se crea un archivo temporal donde se escriben los bytes cifrados.

Tras el cifrado, se añade al final del archivo una estructura que le servirá para saber si ese fichero está cifrado o no. De la misma forma servirá para que en el proceso de descifrado se pueda identificar y descifrar cada uno de los archivos.

La estructura está formada por tres campos:

- El tamaño original en bytes del archivo, este campo ocupa 32 bits (4 bytes), por lo tanto el código dañino no puede cifrar archivos de más de 4 gigabytes.
- La primera clave de cifrado XOR de la fase I.
- El identificador único del sistema comprometido.

9. PERSISTENCIA EN EL SISTEMA

El código dañino no utiliza ninguna técnica para obtener persistencia sobre el sistema víctima.

10. ARCHIVOS RELACIONADOS

Los archivos relacionados con el código dañino son los siguientes:

<variable>\ (según su ruta)			
Nombre	Fecha Creación	Tamaño bytes	Hash SHA1
YOUR_FILES_ARE_ENCRYPTED.TXT	<variable>	<variable>	<variable>
YOUR_FILES_ARE_ENCRYPTED.HTML	<variable>	<variable>	<variable>

11. DETECCIÓN

Para detectar si un equipo se encuentra infectado o ha estado infectado, para cualquiera de sus usuarios, se ejecutará alguna de las herramienta de Mandiant como el "Mandiant IOC Finder" o el colector generado por RedLine@ con los indicadores de compromiso generados para su detección, al igual que las Herramientas del Sistema.

11.1 MANDIANT

Se ha generado un nuevo archivo indicador de compromiso. El nombre del indicador generado es con GUID "6d2eae75-8a94-4f64-8ac5-e1c5297b9444".

Se utilizará el indicador 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.

11.2 HERRAMIENTAS DEL SISTEMA

Para detectar el código dañino se debe de buscar archivos TXT y HTML con el nombre "YOUR_FILES_ARE_ENCRYPTED" en todo el sistema. En el caso de que se encuentre al menos uno, habría que comprobar que el texto del archivo se corresponde al indicado en este informe en el punto [Archivos De Rescate](#).

Además, se buscará en la memoria del sistema, usando una herramienta como PowerTool², algún binario que posea alguno de los siguientes HASH:

```
7fb4364669700327aea423f10c87fc3a
3e859ce4de1ae1b7fa3841eb073eaf68
```

² <http://d-h.st/users/powertool>

En el caso de que se encuentre alguno, es un claro indicio de compromiso del sistema.

La desaparición de algunos archivos del usuario o programas, junto con la aparición en su lugar de otros con el mismo nombre pero una extensión de más de cuatro caracteres, es un posible indicio de compromiso. Si los archivos, además, no puedan ser abiertos de forma habitual, las probabilidades de infección aumentan considerablemente.

12. DESINFECCIÓN

Se recomienda iniciar sesión con un usuario que posea derechos administrativos en el sistema con el fin de eliminar el código dañino.

El código dañino no añade ni modifica ningún campo en el registro, por lo que la desinfección implica borrar el ejecutable del código dañino, así como eliminar todos los archivos que crea sobre el secuestro de los archivos.

El código dañino puede estar inyectado en un proceso, por lo que se recomienda usar alguna herramienta como *RogueKiller*³ para que analice la memoria de los procesos activos en el sistema y pueda encontrar al código dañino. En el caso de que sea encontrado, se finalizará el proceso afectado inmediatamente.

También se recomienda utilizar una herramienta que permita analizar los módulos cargados en los procesos como *Adlice PEViewer*⁴ y buscar en todas sus librerías cargadas (DLL) alguna que tenga una sección llamada ".xxxx". Si dicha sección ocupa 512 bytes y contiene direcciones ".onion" con referencias a la palabra "mischa", el proceso deberá ser finalizado pues está en proceso de cifrado.

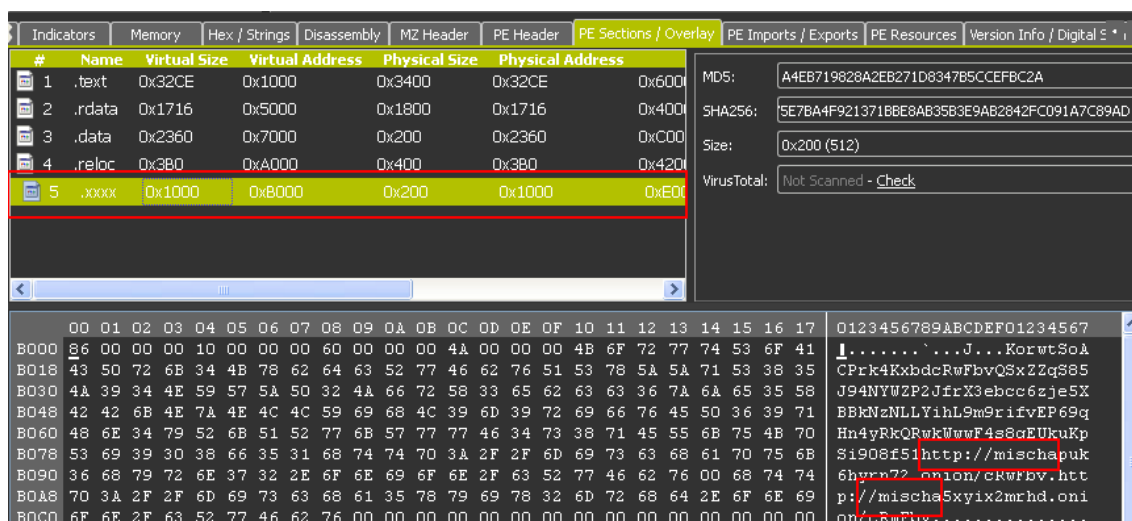


Ilustración 5. Ejemplo del código dañino inyectado en proceso víctima

³ <http://www.adlice.com/es/software/roguekiller/>

⁴ <http://www.adlice.com/software/roguekillerpe/>

El código dañino no borra las *Shadow Copies*⁵ de forma que, si estuvieran activadas en el sistema, se podrían recuperar los ficheros a través de ese servicio.

En caso de no estar activas las Shadow Copies, no existe forma conocida hasta el momento para recuperar los archivos cifrados por el código dañino, debiéndose recurrir a usar copias de seguridad previas de dichos archivos para obtener la información.

13. INFORMACIÓN DEL ATACANTE

El código dañino no establece ninguna comunicación con un C2 o algún dominio intermedio que pudiera aportar información de los atacantes. Sin embargo existe una cuenta en Twitter del portavoz del grupo, *Janus Cybercrime Solutions*, que creó el código dañino: <https://twitter.com/janussec>

Copyright © 2016 Janus Cybercrime Solutions™ / Twitter: @JANUSSEC / RaaS: janusqqdo2zx75el.onion

Ilustración 6. Nombre del grupo y cuenta de twitter

En ella se hace referencia a la siguiente URL:

<http://janusqqdo2zx75el.onion/>

Todos los nombres usados, tanto del código dañino, el grupo y la cuenta de Twitter, han sido obtenidos de la película *Goldeneye*⁶.

Si se accede a la web ".onion", mediante un navegador con acceso a la red Tor, se puede observar la publicidad con la que se vende el uso del código dañino "Ransom.Petya" y "Ransom.Mischa".

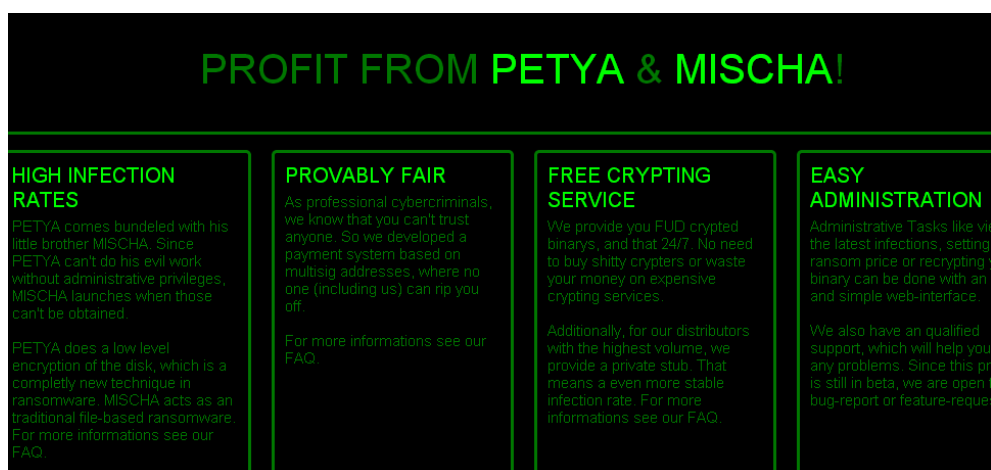


Ilustración 7. Información acerca del código dañino

⁵ [https://technet.microsoft.com/en-us/library/cc785914\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc785914(v=ws.10).aspx)

⁶ <https://es.wikipedia.org/wiki/GoldenEye>

También se puede ver los porcentajes que los autores del código dañino pagan a sus afiliados de forma que, a mayor distribución, mejor beneficio.

PAYMENT SHARE

Your share on the payments you have generated is calculated with the following table. The more volume you generate in one week, the more share on the profit you get.

Example: If you generate a volume of 125 BTC, you get a payout of 106.25 BTC. That are at the moment about 45,000 USD! To get a volume over 100 BTC is not a big deal with the right technique!

Volume/Week	Share
<5 BTC	25%
<25 BTC	50%
<125 BTC	75%
>=125 BTC	85%

Ilustración 8. Porcentajes de pago a los afiliados

14. REGLAS DE DETECCIÓN

14.1 INDICADOR 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="6d2eae75-8a94-4f64-8ac5-e1c5297b9444"
  xmlns="http://schemas.mandiant.com/2010/ioc"
  last-modified="2016-06-08T08:01:59"
  <short_description>Ransom.Mischa</short_description>
  <description>IOC para detectar el código dañino Ransom.Mischa</description>
  <authored_by>CCN-CERT</authored_by>
  <authored_date>2016-06-08T07:59:36</authored_date>
  <links />
  <definition>
    <Indicator operator="OR" id="e5d91cf6-c6d3-4234-ad48-c73b509666e5">
      <IndicatorItem id="70f128d1-9f5f-41d2-8787-bb4b66ef32e3" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">3e859ce4de1ae1b7fa3841eb073eaf68</Content>
      </IndicatorItem>
      <IndicatorItem id="98f88383-9a9b-4694-9e0c-d9bfe96ec9bd" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">7fb4364669700327aea423f10c87fc3a</Content>
      </IndicatorItem>
    </Indicator>
  </definition>
</ioc>
```

14.2 YARA

```
import "pe"

rule Ransom_Mischa_Note
{
    meta:
        description = "Regla para detectar Ransom.Mischa mediante nota de secuestro"
        author = "CCN-CERT"
        version = "1.0"
    strings:
        $a = { 4D 49 53 43 48 41 20 52 41 4E 53 4F 4D 57 41 52 45 21 }
    condition:
        $a
}

rule Ransom_Mischa_Internal_Name
{
    meta:
        description = "Regla para detectar Ransom.Mischa mediante nombre interno"
        author = "CCN-CERT"
        version = "1.0"
    strings:
        $a = { 4D 69 73 63 68 61 2E 64 6C 6C }
    condition:
        $a
}

rule Ransom_Mischa_Section
{
    meta:
        description = "Regla para detectar Ransom.Mischa por la ultima seccion"
        author = "CCN-CERT"
        version = "1.0"
    condition:
        pe.number_of_sections == 5 and pe.sections[4].name == ".xxxx" and
        pe.characteristics & pe.DLL and pe.exports("_ReflectiveLoader")
}
```