Informe Código Dañino CCN-CERT ID-15/16

Ransom.CryptoWall

Junio de 2016

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

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.CryptoWall***", el cual ha sido diseñado para instalarse en el sistema, comunicarse con un dominio de Internet, cifrar ciertos archivos y extorsionar a la víctima mostrando una notificación sobre el procedimiento de pago para rescatar los archivos cifrados. El código dañino pertenece a la familia de "*troyanos*" estando incluido en la subfamilia de "*Ransomware*".

Existen seis versiones conocidas del código dañino desde su primera aparición, en noviembre del 2013, hasta su última versión, en noviembre del 2015.

Entre dichas versiones de la familia se observan diferencias en los siguientes aspectos:

- Las extensiones a cifrar.
- El nombre y la extensión de los archivos cifrados.
- Si guarda la clave usada para cifrar en el sistema comprometido.
- El algoritmo de cifrado utilizado para proteger la clave de cifrado.
- La forma de presentación del rescate de los archivos.
- La forma de comunicarse con el servidor C2.
- El algoritmo de cifrado de archivos.

Este código dañino tiene por objetivo cualquier tipo de usuario u organización en cualquier parte del mundo.

3. INFORMACIÓN DE VERSIONES DEL CÓDIGO DAÑINO

El código dañino presenta diferencias entre sus versiones. En este apartado se muestran las diferencias y los cambios producidos entre ellas.

3.1 VERSIONES DEL CÓDIGO DAÑINO

Existen seis versiones distintas del código dañino. En este apartado se indican las diferencias entre estas versiones así como los elementos en común más significativos.

3.1.1 PRIMERA VERSIÓN

- La primera versión del código dañino data de noviembre del 2013. Inicialmente el código dañino tenía el nombre de "*CryptoLocker Clone*" porque se mostrada al usuario una copia exacta de otro ransomware llamado "*CryptoLocker*". Del mismo modo, su código era muy parecido al código de "*CryptoLocker*".
- La generación de clave de cifrado se realiza sin necesidad de conexión a Internet y por lo tanto sin interacción con un servidor C2.

- Se ofrecen múltiples opciones de pago. Esta decisión no era rentable para sus autores por lo que, en la siguiente versión, el método de pago permitido fue únicamente "*BitCoins*".
- El cifrado de archivos se realiza con una clave RSA de 2048 bytes.

3.1.2 SEGUNDA VERSIÓN

- La segunda versión recibió otro nombre por parte de sus autores: "*CryptoDefense*" e hizo su aparición en febrero del 2014.
- La información sobre el pago para el rescate de los archivos cifrados es creado en el mismo sistema comprometido en tres formatos: HTML, TXT y URL. Dichos archivos tienen el nombre "*HOW_DECRYPT.<extensión>*".
- Las claves usadas para el cifrado se almacenan en la carpeta de *%APPDATA%* del sistema comprometido lo que permitió el desarrollo de una herramienta gratuita con la que se podían descifrar los archivos sin pagar el rescate.
- Sólo admite el modo de pago mediante "*BitCoins*".
- El cifrado de archivos se realiza con una clave RSA de 2048 bytes.

3.1.3 TERCERA VERSIÓN

- Esta versión es la primera creación del código dañino con el nombre que ya mantendría hasta su última versión: "*CryptoWall*". Esta versión hizo su aparición en marzo del 2014 tras descubrirse el fallo en el sistema de cifrado de su predecesor.
- Esta versión mantiene los tres archivos con la información de rescate que ya tenía la segunda versión con las extensiones HTML, TXT y URL pero modifica el nombre siendo "*DECRYPT_INSTRUCTION.<extensión>*".
- Es la primera versión del código dañino que impide que se ejecute su carga dañina si el sistema comprometido pertenece a uno de los siguientes países:

Rusia
Ucrania
Bielorrusia
Kazajistán

- El código, en lugar de sobreescribir el archivo original con el archivo cifrado, escribe el archivo cifrado en un fichero independiente para luego borrar el fichero original y renombrar el fichero cifrado con el nombre del original. Esto significa que los archivos originales siguen estando en el disco y son recuperables a través de herramientas forenses.
- El cifrado de archivos se realiza con una clave RSA de 2048 bytes obtenida desde el servidor C2.

3.1.4 CUARTA VERSIÓN

- La cuarta versión del código dañino data de octubre de 2014 y se dio a conocer como "*CryptoWall 2.0*".
- La conexión con el servidor C2, que está en la red TOR[1], se realiza a través de un *proxy*. Ésta fue una de las razones por las que esta versión del código dañino fue actualizada dado que, analizando el tráfico de red, se podía ver fácilmente la conexión al *proxy* pudiéndose dar de baja muy fácilmente dado que es *proxy* no está en la red TOR.
- El cifrado de archivos se sigue realizando con una clave RSA de 2048 bytes obtenida desde el servidor C2.
- Al igual que ocurría con la tercera versión, el código dañino comprueba el país del sistema para no afectar en dichos países.

3.1.5 QUINTA VERSIÓN

- La quinta versión del código dañino data de enero de 2015 y se dio a conocer como "*CryptoWall 3.0*".
- El algoritmo de cifrado de archivos se cambia de RSA a AES lo que hace que el tiempo de cifrado sea mucho menor, especialmente sobre archivos de gran tamaño.
- La clave de cifrado AES se protege cifrándola con una clave RSA pública obtenida desde el servidor C2.
- Los tipos de archivo afectados aumentan con respecto a las versiones anteriores del código dañino.
- Para comunicarse utiliza túneles cifrados en la red "*I2P*" (*Internet Invisible Project*). La implementación contenía fallos de seguridad y era bastante lenta. Ésta fue la razón por la cual se modificó el código de esta misma versión para volver a utilizar un sistema de *proxy* intermedio como en la versión cuarta.
- Se añade un archivo de tipo PNG con la información del secuestro a los tres ya existentes de otras versiones: HTML, TXT y URL. El nombre del archivo cambia a "*HELP_DECRYPT.<extension>*".
- Al igual que versiones previas del código dañino, no se ejecuta sobre sistemas de una lista de países que se aumenta:

Rusia
Ucrania
Bielorrusia
Kazajistán
Armenia
Irán

[1] https://es.wikipedia.org/wiki/Tor_(red_de_anonimato)

3.1.6 SEXTA VERSIÓN

- Esta versión del código dañino data de noviembre de 2015 y se dio a conocer como "*CryptoWall 4.0*".
- Los autores del código dañino eliminaron la versión en la información sobre el secuestro y recuperación de los archivos. Además, los textos mostrados en dichos archivos son mucho mas agresivos que las versiones anteriores e incluso se burlan de la víctima afectada.
- Los archivos creados con la información acerca del secuestro y el rescate de los archivos son tres: HTML, TXT y PNG. El nombre de dichos archivos es "*HELP_YOUR_FILES.<extensión>*" aunque existe alguna subversión con el nombre "*INSTRUCTIONS_<id_único_sistema_comprometido>.<extensión>*".
- En esta verión no se utiliza una tabla de extensiones a cifrar si no que, al contrario, se usa una lista de localizaciones y extensiones que no serán cifrados, de forma que todos los que no estén en esas listas sí que se cifrarán.

3.2 EXTENSIONES A CIFRAR

El código dañino afecta a un gran número de tipos de archivo detectados por su extensión. En ningún momento comprueba binariamente si el archivo es del tipo que su extensión indica.

3.2.1 VERSIONES DE LA PRIMERA A LA CUARTA

En las primeras cuatro versiones del código dañino se cifran los archivos que posean cualquiera de las siguientes extensiones:

.c	.h	.m	.ai	.cs	.db	.db	.nd
.pl	.ps	.py	.rm	.3dm	.3ds	*3fr	.3g2
.3gp	.ach	.arw	.asf	.asx	.avi	.bak	.bay
.cdr	.cer	.cpp	.cr2	.crt	.crw	.dbf	.dcr
.dds	.der	.des	.dng	.doc	.dtd	.dwg	.dxf
.dxg	.eml	.eps	.erf	.fla	.flv	.hpp	.iif
.jpe	.jpg	.kdc	.key	.lua	.m4v	.max	.mdb
.mdf	.mef	.mov	.mp3	.mp4	.mpg	.mrw	.msg
.nef	.nk2	.nrw	.oab	.obj	.odb	.odc	.odm
.odp	.ods	.odt	.orf	.ost	.p12	.p7b	.p7c
.pab	.pas	.pct	.pdb	.pdd	.pdf	.pef	.pem
.pfx	.pps	.ppt	.prf	.psd	.pst	.ptx	.qba
.qbb	.qbm	.qbr	.qbw	.qbx	.qby	.r3d	.raf
.raw	.rtf	.rw2	.rwl	.sql	.sr2	.srf	.srt
.srw	.svg	.swf	.tex	.tga	.thm	.tlg	.txt

.vob	.wav	.wb2	.wmv	.wpd	.wps	.x3f	.xlk
.xlr	.xls	.yuv	.back	.docm	.docx	.flac	.indd
.java	.jpeg	.pptm	.pptx	.xlsb	.xlsm	.xlsx	

3.2.2 QUINTA VERSIÓN

En la quinta versión del código dañino se aumenta significativamente el número de extensiones afectadas:

.3dm	.3ds	.3fr	.3g2	.3gp	.3pr	.7z	.ab4
.accdb	.accde	.accdr	.accdt	.ach	.acr	.act	.adb
.ads	.agdl	.ai	.ait	.al	.apj	.arw	.asf
.asm	.asp	.asx	.avi	.awg	.back	.backup	.backupdb
.bak	.bank	.bay	.bdb	.bgt	.bik	.bkp	.blend
.bpw	.c	.cdf	.cdr	.cdr3	.cdr4	.cdr5	.cdr6
.cdrw	.cdx	.ce1	.cd2	.cer	.cfp	.cgm	.cib
.class	.cls	.cmt	.cpi	.cpp	.cr2	.craw	.crt
.crw	.cs	.csh	.csl	.csv	.dac	.db	.db-jounral
.db3	.dbf	.dc2	.dcr	.dcs	.ddd	.ddoc	.ddrw
.dds	.der	.des	.design	.dgc	.djvu	.dng	.doc
.docm	.docx	.dot	.dotm	.dotx	.drf	.drw	.dtd
.dwg	.dxb	.dxf	.dxg	.eml	.eps	.erbsql	.erf
.exf	.fdb	.ffd	.fff	.fh	.fhd	.fla	.flac
.flv	.fpx	.fxg	.gray	.grey	.gry	.h	.hbk
.hpp	.ibank	.ibd	.ibz	.idx	.iif	.iiq	.incpas
.indd	.java	.jpe	.jpeg	.jpg	.kc2	.kdbx	.kdx
.key	.kpdx	.lua	.m	.m4v	.max	.mdb	.mdc
.mdf	.mef	.mfw	.mmw	.moneywell		.mos	.mov
.mp3	.qbr	.st4	.st5	.zip	.yuv	.ycbcra	.xlw
.mp4	.mpg	.mrw	.msg	.myd	.nd	.ndd	.nef
.nk2	.nop	.nrw	.ns2	.ns3	.ns4	.nsd	.nsf
.nsg	.nsh	.nwb	.nx2	.nxl	.nyf	.oab	.obj
.odb	.odc	.odf	.odg	.odm	.odp	.ods	.odt
.oil	.orf	.ost	.otg	.oth	.otp	.ots	.ott
.p12	.p7b	.p7c	.pab	.pages	.pas	.pat	.pcd
.pct	.pdb	.pdd	.pdf	.pef	.pem	.pfx	.php
.pl	.plc	.pot	.potm	.potx	.ppam	.pps	.ppsm
.ppsx	.ppt	.pptm	.pptx	.prf	.ps	.psafe3	.psd
.pspimage		.pst	.ptx	.py	.qba	.qbb	.qbm

.qbw	.qbx	.qby	.r3d	.raf	.rar	.rat	.raw
.rdb	.rm	.rtf	.rw2	.rwl	.rwz	.s3db	.sas7bdat
.say	.sd0	.sda	.sdf	.sldm	.sldx	.sql	.sqlite
.sqlite3	.sqlitedb		.sr2	.srf	.srt	.srw	
.st6	.st7	.st8	.stc	.std	.sti	.stw	.stx
.svg	.swf	.sxc	.sxd	.sxg	.sxi	.sxm	.sxw
.tex	.tga	.thm	.tlg	.txt	.vob	.wallet	.wav
.wb2	.wmv	.wpd	.wps	.x11	.x3f	.xis	.xla
.xlam	.xlk	.xlm	.xlr	.xls	.xlsb	.xlsm	.xlsx
.xlt	.xltm	.xltx					

3.2.3 SEXTA VERSIÓN

En la sexta versión del código dañino se tiene una serie de tablas de tipo *lista blanca* entre las que se encuentran incluidas rutas, extensiones de archivo y nombres específicos que no serán cifrados, siendo cifrados el resto de archivos que no estén indentificados en esas tabla.

Es importante observar que el código dañino no guarda ni en texto plano ni cifrado, las cadenas a ignorar. En su lugar mantiene una serie de tablas con valores CRC32 de los *hash* de los elementos a ignorar. Es por ello que las tablas posteriores no se encuentran completas ya que desde un *hash* no es posible averiguar el texto o valor original desde el que se calculó.

Por cada archivo o carpeta que se encuentra en el proceso de cifrado, comprueba que no exista en ninguna de sus *listas blancas*. En el caso de que exista, lo ignorará y procesará el siguiente elemento, y en el caso de que no esté en esos listados procederá a cifrarlo.

La lista descubierta de las carpetas para ser ignoradas es:

windows
temp
cache
sample pictures
default pictures
sample Music
program files
program files (x86)
games
sample videos
user account pictures
packages

La lista descubierta de extensiones de archivos para ser ignoradas es:

exe	**dll**	**pif**	**scr**
sys	**msi**	**msp**	**com**
hta	**cpl**	**msc**	**bat**
cmd	**scf**		

Por último, la lista descubierta de archivos para ser ignorados es:

help_your_files.txt
help_your_files.html
help_your_files.png
iconcache.db
thumbs.db

3.3 EXTENSIÓN AÑADIDA A LOS ARCHIVOS CIFRADOS

3.3.1 VERSIONES DE LA PRIMERA A LA CUARTA

En estas versiones, el código dañino cifra directamente el archivo original sin alterar ni su nombre, ni su extensión.

3.3.2 QUINTA VERSIÓN

En esta versión del código dañino crea un nuevo archivo con un nombre aleatorio sin extensión.

3.3.3 SEXTA VERSIÓN

En esta versión del código dañino se genera un nombre aleatorio y una extensión aleatoria por cada archivo que cifra.

3.4 ARCHIVOS DE RESCATE

3.4.1 PRIMERA VERSIÓN

La primera versión del código dañino crea archivos acerca del secuestro de los archivos en el sistema comprometido utilizando una interfaz de usuario propia con la información del secuestro y los pasos a seguir.

Ilustración 1. Imagen de la información del secuestro de los archivos

3.4.2 SEGUNDA VERSIÓN

Esta versión del código dañino crea los siguientes archivos en el sistema comprometido como notas del secuestro de los archivos.

HOW_DECRYPT.HTML

HOW_DECRYPT.TXT

HOW_DECRYPT.URL

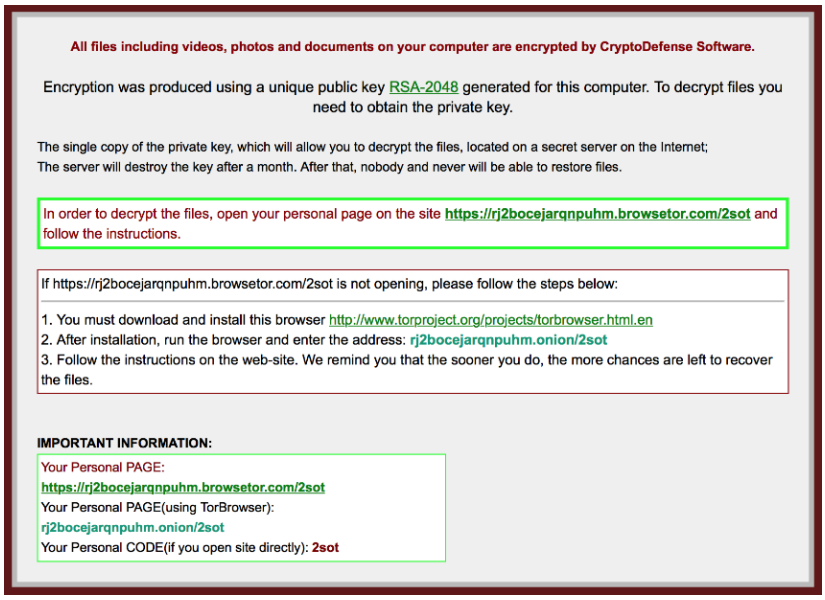


Ilustración 2. Fichero HTML con la información del secuestro de archivos

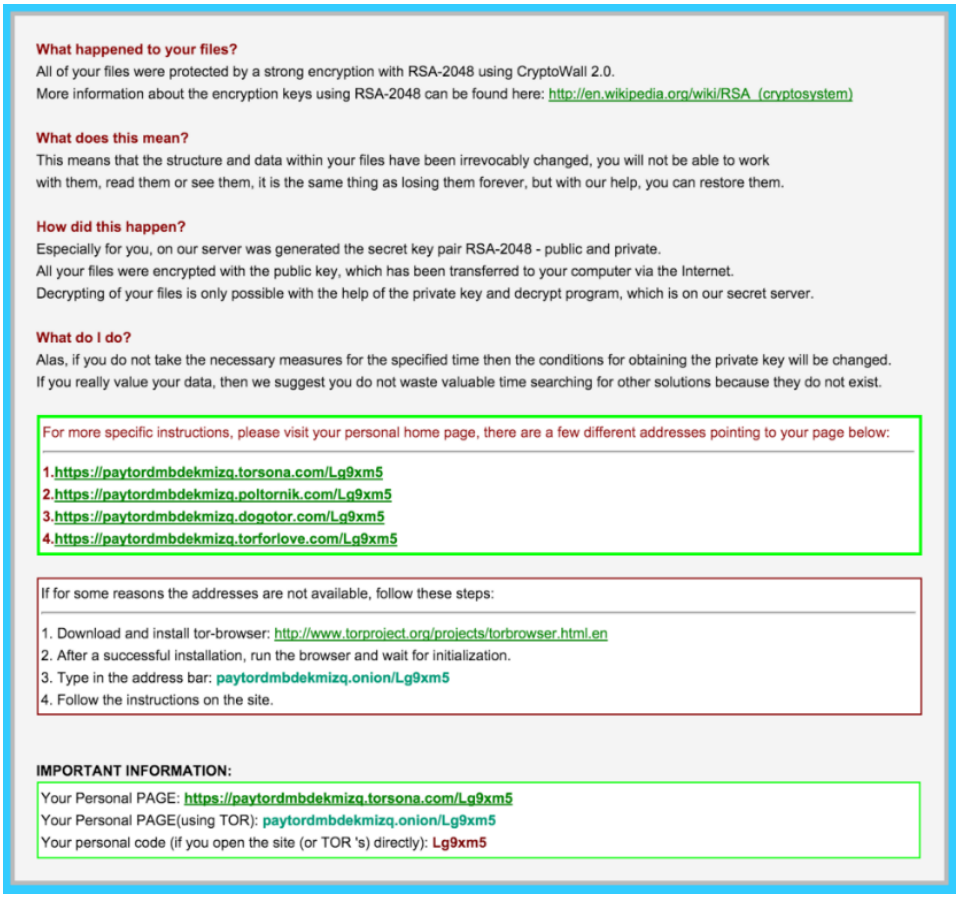
3.4.3 TERCERA Y CUARTA VERSIÓN

Esta versión del código dañino crea los siguientes archivos en el sistema comprometido como notas del secuestro de los archivos:

DECRYPT_INSTRUCTION.HTML

DECRYPT_INSTRUCTION.TXT

DECRYPT_INSTRUCTION.URL



What happened to your files?
All of your files were protected by a strong encryption with RSA-2048 using CryptoWall 2.0.
More information about the encryption keys using RSA-2048 can be found here: http://en.wikipedia.org/wiki/RSA_(cryptosystem)

What does this mean?
This means that the structure and data within your files have been irrevocably changed, you will not be able to work with them, read them or see them, it is the same thing as losing them forever, but with our help, you can restore them.

How did this happen?
Especially for you, on our server was generated the secret key pair RSA-2048 - public and private.
All your files were encrypted with the public key, which has been transferred to your computer via the Internet.
Decrypting of your files is only possible with the help of the private key and decrypt program, which is on our secret server.

What do I do?
Alas, if you do not take the necessary measures for the specified time then the conditions for obtaining the private key will be changed.
If you really value your data, then we suggest you do not waste valuable time searching for other solutions because they do not exist.

For more specific instructions, please visit your personal home page, there are a few different addresses pointing to your page below:

1.https://paytordmbdekmizq.torsona.com/Lg9xm5
2.https://paytordmbdekmizq.poltornik.com/Lg9xm5
3.https://paytordmbdekmizq.dogotor.com/Lg9xm5
4.https://paytordmbdekmizq.torforlove.com/Lg9xm5

If for some reasons the addresses are not available, follow these steps:

1. Download and install tor-browser: http://www.torproject.org/projects/torbrowser.html.en
2. After a successful installation, run the browser and wait for initialization.
3. Type in the address bar: paytordmbdekmizq.onion/Lg9xm5
4. Follow the instructions on the site.

IMPORTANT INFORMATION:

Your Personal PAGE: https://paytordmbdekmizq.torsona.com/Lg9xm5
Your Personal PAGE(using TOR): paytordmbdekmizq.onion/Lg9xm5
Your personal code (if you open the site (or TOR 's) directly): Lg9xm5

Ilustración 3. Fichero HTML con la información del secuestro de la cuarta versión

3.4.4 QUINTA VERSIÓN

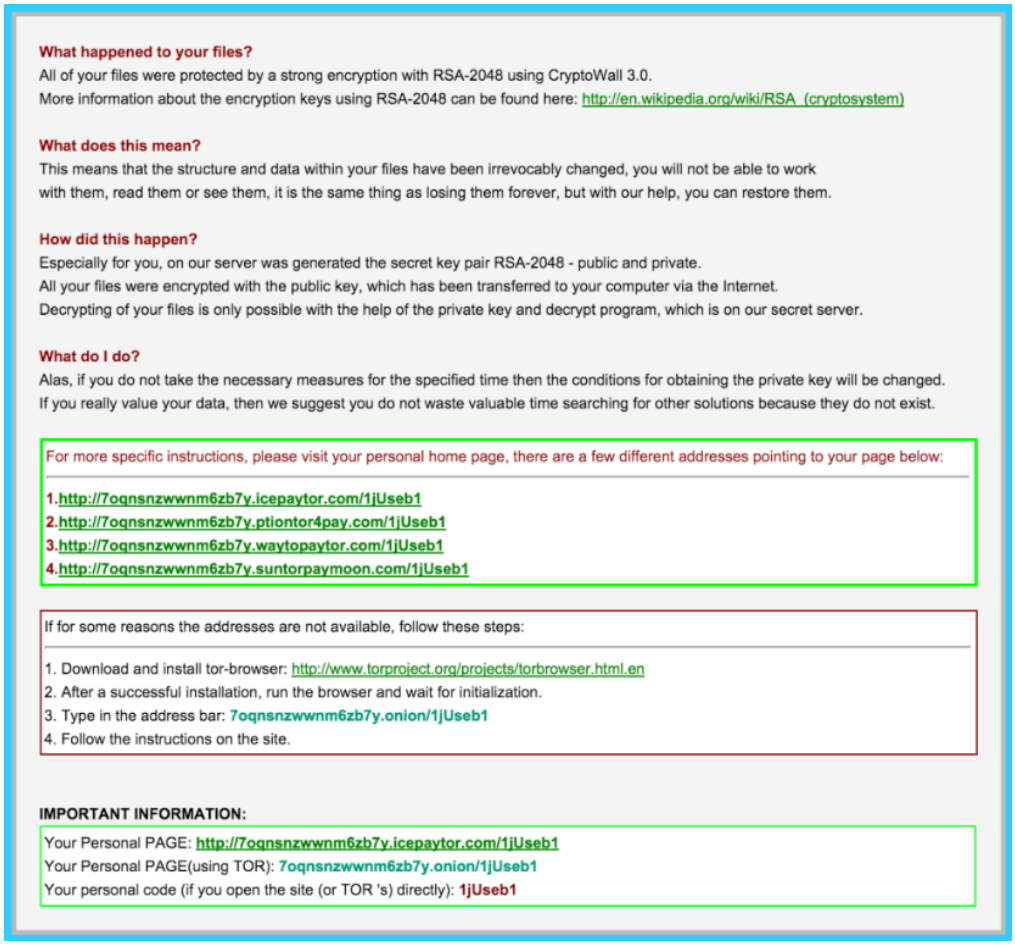
Esta versión del código dañino crea los siguientes archivos como notas del secuestro de los archivos en el sistema comprometido:

HELP_DECRYPT.HTML

HELP_DECRYPT.TXT

HELP_DECRYPT.PNG

HELP_DECRYPT.URL



What happened to your files?
All of your files were protected by a strong encryption with RSA-2048 using CryptoWall 3.0.
More information about the encryption keys using RSA-2048 can be found here: http://en.wikipedia.org/wiki/RSA_(cryptosystem)

What does this mean?
This means that the structure and data within your files have been irrevocably changed, you will not be able to work with them, read them or see them, it is the same thing as losing them forever, but with our help, you can restore them.

How did this happen?
Especially for you, on our server was generated the secret key pair RSA-2048 - public and private.
All your files were encrypted with the public key, which has been transferred to your computer via the Internet.
Decrypting of your files is only possible with the help of the private key and decrypt program, which is on our secret server.

What do I do?
Alas, if you do not take the necessary measures for the specified time then the conditions for obtaining the private key will be changed.
If you really value your data, then we suggest you do not waste valuable time searching for other solutions because they do not exist.

For more specific instructions, please visit your personal home page, there are a few different addresses pointing to your page below:

1.http://7oqnsnzwwnm6zb7y.icepaytor.com/1jUseb1
2.http://7oqnsnzwwnm6zb7y.ptiontor4pay.com/1jUseb1
3.http://7oqnsnzwwnm6zb7y.waytopaytor.com/1jUseb1
4.http://7oqnsnzwwnm6zb7y.suntorpaymoon.com/1jUseb1

If for some reasons the addresses are not available, follow these steps:

1. Download and install tor-browser: http://www.torproject.org/projects/torbrowser.html.en
2. After a successful installation, run the browser and wait for initialization.
3. Type in the address bar: 7oqnsnzwwnm6zb7y.onion/1jUseb1
4. Follow the instructions on the site.

IMPORTANT INFORMATION:

Your Personal PAGE: http://7oqnsnzwwnm6zb7y.icepaytor.com/1jUseb1
Your Personal PAGE(using TOR): 7oqnsnzwwnm6zb7y.onion/1jUseb1
Your personal code (if you open the site (or TOR 's) directly): 1jUseb1

Ilustración 4. Imagen del secuestro de los archivos en el sistema comprometido

3.4.5 SEXTA VERSIÓN

La última versión del código dañino crea los siguientes archivos como notas del secuestro de los archivos:

HELP_YOUR_FILES.HTML
HELP_YOUR_FILES.TXT
HELP_YOUR_FILES.PNG

Sin embargo, existen unas pocas muestras de esta versión que en lugar de crear los archivos con los nombres anteriormente indicados, lo hacen con los siguientes:

INSTRUCTIONS_<ID_UNICO>.HTML
INSTRUCTIONS_<ID_UNICO>.TXT
INSTRUCTIONS_<ID_UNICO>.PNG

Ilustración 5. Imagen de secuestro de la sexta versión del código dañino

4. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El código dañino examinado posee las siguientes características:

- Carga el código dañino en el sistema.
- Elimina los puntos de restauración del sistema.
- Enumera los discos duros, unidades extraíbles y recursos de red en busca de archivos, que cumplan un determinado patrón, para cifrarlos y tras ello pide un rescate para recuperar los originales.
- Conecta con un C2 preestablecido en su código.
- Enumera todos los archivos del sistema que cumplan un determinado patrón y los cifra.
- Modifica el registro del sistema para asegurar persistencia.

5. DETALLES GENERALES

Las muestras analizadas se corresponden con las siguientes firmas MD5:

e73806e3f41f61e7c7a364625cd58f65
48e00f5fce2f6e645ac3901aae3facb1

El binario tiene formato PE (*Portable Executable*), es decir, es un ejecutable para sistemas operativos Windows, concretamente para 32 bits. Las fechas internas de las muestras se corresponden con los periodos en los que se realizaron las campañas.

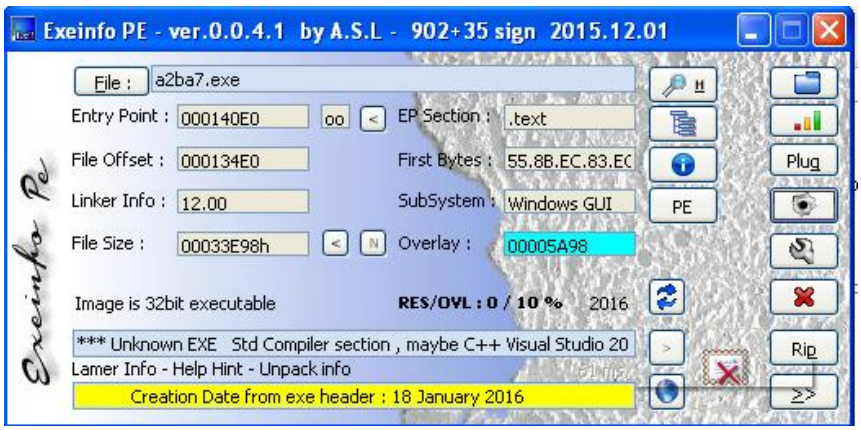


Ilustración 6. Detalle de la fecha interna del programa: 18 de enero de 2016.

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*" inyecta el código dañino en un nuevo proceso del Explorador de Windows, por defecto "*Explorer.exe*".
- Crea una copia de sí mismo en *%APPDATA%*.
- Modifica el registro para asegurar su persistencia.
- Borra los "*Shadow Volumes*" que puedan existir en el sistema.
- Elimina los puntos de restauración del sistema.
- Se inyecta en el proceso "*svchost.exe*" y en el Explorador de Windows.
- Se comunica con su servidor C2 para obtener la clave pública RSA que utilizará para cifrar las claves AES.
- Obtiene desde su servidor C2 los archivos a mostrar acerca del secuestro, los enlaces TOR para poder efectuar el pago y la imagen a mostrar.
- Cifra los archivos encontrados que sigan un determinado patrón.
- Crea archivos con información acerca del secuestro de los archivos y muestra una imagen con la misma información.

7. CARACTERÍSTICAS TÉCNICAS

El código dañino se ejecuta desde un "*dropper*" que utiliza dos métodos para dificultar el análisis. El primero es usar "*código basura*" con llamadas a funciones de diversas librerías sin sentido práctico. Esto lo realiza para confundir a los motores de análisis automáticos como, por ejemplo, los antivirus. El segundo método es usar "código spagueti"[2] para ralentizar el análisis del analista.

```
mov     [466DBC], eax
call    [<&USER32.InvalidateRect>]          USER32.InvalidateRect
mov     ecx, [esp+28]
push    dword ptr [esp+18]
lea     eax, [esp+4C]
push    dword ptr [esp+3C]
mov     [ecx+1], eax
mov     eax, [466DC8]
push    dword ptr [esp+58]
add     eax, 4
imul    eax, [466DCC]
push    dword ptr [esp+18]
sub     edi, eax
mov     [466DBC], eax
mov     [esp+2C], edi
call    [<&USER32.DefWindowProcA>]          USER32.DefWindowProcA
```

Ilustración 7. Código basura en el "*dropper*" para dificultar el análisis

El siguiente paso es crear un evento en el sistema cuyo nombre es un *hash* MD5 generado de las características del sistema comprometido. Con este evento creado, el código dañino se asegura dos cosas:

- Sincronía entre procesos.
- Evitar que otra instancia del código dañino se ejecute en el mismo sistema comprometido.

Una vez ejecutado el "*dropper*", éste inyecta código en el proceso del Explorador de Windows, por defecto, "*explorer.exe*".

Para realizar la inyección en el proceso, el código dañino usa dos formas que varían entre las muestras. La primera consiste en utilizar las funciones nativas "*ZwCreateSection*", "*ZwMapViewOfSection*" y, posteriormente, la función "*CreateRemoteThread*". La segunda opción es utilizar la función interna del sistema comprometido "*ZwQueueApcThread*" para crear un hilo mediante un APC[3] en cola en el proceso remoto, en este caso, "*explorer.exe*".

```
push    0
mov     ecx, [ebp-4]
push    ecx
mov     edx, [ebp+C]
push    edx
call    004016E0
mov     eax, [eax+A8]                       ntdll.ZwQueueApcThread
call    eax
```

Ilustración 8. Inyección del código dañino en el proceso remoto

Tras la inyección en el proceso, procede ganar persistencia en el sistema infectado. Para ello, crea una carpeta con un nombre aleatorio en la ruta de

[2] https://es.wikipedia.org/wiki/C%C3%B3digo_espagueti

[3] https://msdn.microsoft.com/es-es/library/windows/desktop/ms681951(v=vs.85).aspx

%APPDATA% del sistema comprometido y, en esta ubicación, realiza una copia del código dañino con un nombre aleatorio. Posteriormente crea una entrada en el registro de Windows que se llama exactamente igual que el nombre de la carpeta aleatoria creada en *%APPDATA%*.

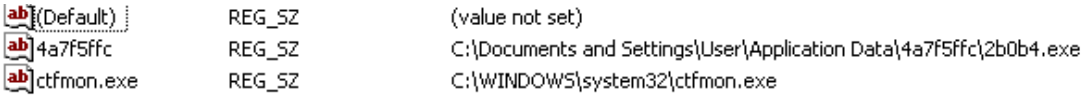


Ilustración 9. Ejemplo de persistencia del código dañino

Después procede a inyectar su código en el proceso "*svchost.exe*" usando el mismo método que usó para inyectarse en el Explorador de Windows. Esta segunda inyección la realiza para tener mayores privilegios en la máquina comprometida y así poder realizar el borrado de las *Shadow Volumes*[4] sin que aparezca ningún aviso de seguridad por el control de cuentas de usuario (UAC).

El código dañino borra todos los puntos de restauración existentes en el sistema mediante llamadas a la función "*SRRemoveRestorePoint*" que necesita, como parámetro, el índice del punto de restauración a borrar. En el caso de que exista un punto de restauración para ese índice, devolverá éxito y lo borrará, mientras que en caso de que no exista, o no pueda borrarlo, devolverá el error de "*ERROR_INVALID_DATA*".

Sabiendo esto, se utiliza el siguiente código para borrar todos los puntos de restauración del sistema:

```
int i = 0;
DWORD result = ERROR_SUCCESS;
do
{
  result = SRRemoveRestorePoint(i++);
} while (result != ERROR_INVALID_DATA);
```

Tras ello, procede a escribir en el registro una entrada "*DisableSR*" con el valor a "*1*" para deshabilitar completamente la función encargada de crear puntos de restauración del sistema comprometido.

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows Nt\SystemRestore\DisableSR]=1
```

Una vez inyectado en el segundo proceso procede a realizar el borrado de todos los "*Shadow Volumes*" del sistema comprometido. Para ello usa el comando:

```
vssadmin.exe Delete Shadows /All /Quiet
```

El código dañino se conecta al servidor C2 y obtiene un fichero de configuración comprimido con el algoritmo LZ que contiene la información necesaria para el cifrado de los archivos. Para más información acerca de la comunicación con el servidor C2 consultar el apartado de Conexiones de Red del presente documento.

[4] https://en.wikipedia.org/wiki/Shadow_volume

De ese fichero de configuración se obtiene la clave pública RSA, al igual que el texto a escribir en los archivos con la información del secuestro de los archivos y la imagen a mostrar. Toda esta información se guarda en un archivo de configuración comprimido con el algoritmo LZ mediante la función "*RtlCompressBuffer*" que se almacena en la siguiente ruta con un nombre aleatorio de 8 digitos:

<raiz>:\Users\<Nombre_del_usuario>\AppData\Roaming\<8 números aleatorios>

La estructura del archivo de configuración es la siguiente:

- Los datos de la clave pública RSA en formato binario. La conversión a formato binario se realiza mediante la función "*CryptStringToBinary*".
- El archivo HTML que será mostrado al usuario en el idioma apropiado.
- El archivo de texto que será mostrado al usuario en el idioma apropiado.
- La imagen en formato PNG con la información acerca del secuestro de los archivos que será mostrada al usuario en el idioma apropiado.

Gracias a este archivo de configuración, el código dañino puede continuar con su carga dañina en aquellos casos en los que posteriormente no haya acceso a Internet o el equipo se haya reiniciado.

El código dañino convierte la clave pública RSA en un formato que pueda ser manejado por las funciones criptográficas de Windows mediante la función "*CryptDecodeObjectEx*". Esta función devuelve una estructura de tipo "*CERT_PUBLIC_KEY_INFO*" que es importada mediante la función "*CryptImportPublicKeyInfo*" para poder manejar la clave pública.

A continuación el código dañino calcula un *hash* MD5 de la clave pública que servirá para saber si un archivo ya fue cifrado o no, tal y como se explica en el apartado Cifrado y Ofuscación del presente informe.

El código dañino enumera todos los discos duros, unidades extraibles y unidades de red conectadas y comienza a recorrerlas validando que no se hayan procesado ya. Esto lo hace de forma diferente según la versión. Por ejemplo, en ocasiones comprueba que en la raíz de cada una de las unidades no exista un archivo de rescate, por ejemplo, "*HELP_YOUR_FILES.PNG*". Por cada una de las unidades que no hayan pasado la validación de la existencia del archivo se crea un nuevo hilo al proceso de cifrado de archivos.

Cuando el proceso de cifrado ha finalizado, el código dañino crea en el menú de inicio del sistema y en el Escritorio una copia de los tres archivos obtenidos desde su archivo de configuración: el archivo HTML, el archivo de texto y la imagen. Una vez realizada esta acción, procede a borrar el archivo de configuración y a comunicarse con el servidor C2 indicandole que el proceso de cifrado finalizó con éxito.

Por último el código dañino se finaliza a sí mismo mediante la función "*ZwTerminateProcess*".

8. CIFRADO Y OFUSCACION

El código dañino tiene la funcionabilidad de cifrar los archivos del sistema que cumplan una serie de patrones. Los algoritmos usados por el código dañino son AES[5] y RSA[6].

Antes de comenzar a cifrar archivos, el código dañino procede a obtener desde su servidor C2 la clave pública RSA y guardarla en un archivo de configuración tal y como se indica en el apartado de Carácteristicas Técnicas del presente informe. En el caso de que dicha clave no pueda ser obtenida porque, por ejemplo, el servidor esté cerrado, el código dañino permanece en un bucle infinito intentando conseguirla sin afectar a los archivos del sistema.

Una vez obtenida la clave, se procede a realizar los siguientes pasos:

- Se analizan los archivos y directorios de forma recursiva para saber si deben ser cifrados o no. Ésto, tal y como se ha detallado en Información De Versiones Del Código Dañino, es diferente en cada una de las versiones.
- Los atributos del archivo son leídos y almacenados en memoria.
- Se calcula el *hash* MD5 de la clave pública obtenida del archivo de configuración.
- Se comprueba que el archivo no esté ya cifrado leyendo sus primeros 16 bytes y verificando si coinciden con el *hash* MD5 de la clave pública RSA calculada previamente. En el caso de que la comprobación sea correcta, el archivo se ignora porque ya está cifrado.
- En las últimas versiones, se genera un nombre aleatorio que se utiliza para crear un nuevo archivo en la misma localización del archivo original y almacenar, entre otras cosas, el archivo cifrado. En las primeras versiones se utiliza el mismo nombre de fichero que el original.
- Se genera una clave AES256 aleatoria y se guarda en memoria.
- El *hash* MD5 es escrito al principio del nuevo archivo.
- Se cifra la clave AES generada previamente con la clave pública RSA y el resultado es escrito en el archivo a continuación del *hash*.
- Los atributos originales del archivo son escritos a continuación de la clave AES cifrada en el archivo.
- La longitud del nombre original del archivo es escrita a continuación en el archivo.
- El nombre original del archivo es cifrado usando AES256, con la clave previamente calculada, y escrito a continuación en el archivo.

[5] https://es.wikipedia.org/wiki/Advanced_Encryption_Standard
[6] https://es.wikipedia.org/wiki/RSA

- El tamaño del archivo original es escrito a continuación del nombre original.
- Se cifra todo el contenido del archivo original con la clave AES256 generada previamente y se añade al final del archivo.
- En las últimas versiones, se procede a borrar el archivo original moviendo el nuevo a su misma ubicación, mediante la función "*MoveFileEx*", con la opción "*MOVEFILE_REPLACE_EXISTING*". Este paso lo realiza para asegurarse de que no se podrá recuperar el archivos original mediante herramientas forenses.

AES es lo suficientemente rápido como para poder cifrar un archivo grande en un tiempo razonable. Sin embargo, al ser un algoritmo simétrico, la misma clave que se usa para cifrar es la que se emplea para descifrar, por lo que requiere proteger dicha clave.

RSA permite cifrar la clave AES con un sistema de cifrado asimétrico, concreamente con la cláve pública, de forma que la clave de descifrado permanece segura al no conocerse la clave privada.

9. PERSISTENCIA EN EL SISTEMA

Para conseguir persistencia en el sistema infectado, crea una carpeta con un nombre aleatorio en la ruta de *%APPDATA%* y realiza una copia del código dañino a esa ubicación con un nombre aleatorio.

Posteriormente crea una entrada en el registro de Windows que se llama exactamente igual que el nombre de la carpeta aleatoria creada en *%APPDATA%*.

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\<nombre_aleatorio1>] = *%APPDATA%*\<nombre_aleatorio1>\<nombre_aleatorio2>

10. CONEXIONES DE RED

El código dañino se conecta con su servidor C2 mediante peticiones POST de HTTP y cifrando la información previamente con un algoritmo propietario que se basa en generar una cadena aleatoria, convertirla a sus valores en hexadecimal y usar esos valores para cifrarla.

De esta forma, se envía un par de clave-valor teniendo, como clave una letra aleatoria, y como valor la cadena de cifrado seguida de la información cifrada.

<letra_aleatoria>=<cadena_de_cifrado_en_hex><información_cifrada>

La información cifrada que envía sigue la siguiente estructura:

- *ID de acción (o Identificador)*: Un identificador numérico que le indica al C2 qué acción realizar en esa petición.

- *Hash MD5 del sistema comprometido*: Un *hash* MD5 calculado del nombre del equipo del sistema comprometido.
- *Sub ID*: Un identificador de acción secundario a realizar por parte del servidor C2 (es posible encontrar paquetes con varios Sub ID).
- *Datos ID*: Los datos necesarios o solicitados por el indicador previo

Se han podido encontrar 5 comandos distintos:

- Identificador 1 ó 3: Enviado en una conexión inicial con el servidor C2 indicando que un nuevo sistema fue comprometido.
- Identificador 7: Indica que la acción a realizar es la que viene en el campo del Sub ID en el paquete enviado. Se han encontrado las siguientes acciones posibles mediante el Sub ID:
  - *Sub ID 1*: Obtener la clave pública RSA, el PNG adecuado (en este caso el "*Datos ID*" lleva el código de idioma), los enlaces TOR en donde efectuar el pago, el código HTML y el fichero de texto en el idioma adecuado.
  - *Sub ID 2*: Indica que la infección finalizó. En este caso otro Sub ID secundario indica si fue con éxito o no (1 éxito, 2 error).

11. ARCHIVOS RELACIONADOS

Los archivos relacionados con el código dañino varian según las versiones y se resumen conjuntamente en la siguiente tabla de la que se excluye la información genérica de los ficheros de usuario cifrados.

<*%APPDATA%*\ruta_variable>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
<variable>	<variable>	<variable>	<variable>
<%DESKTOP%>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
HOW_DECRYPT.HTML	<variable>	<variable>	<variable>
HOW_DECRYPT.TXT	<variable>	<variable>	<variable>
HOW_DECRYPT.URL	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.HTML	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.TXT	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.URL	<variable>	<variable>	<variable>
HELP_DECRYPT.HTML	<variable>	<variable>	<variable>
HELP_DECRYPT.TXT	<variable>	<variable>	<variable>
HELP_DECRYPT.PNG	<variable>	<variable>	<variable>
HELP_DECRYPT.URL	<variable>	<variable>	<variable>
HELP_YOUR_FILES.HTML	<variable>	<variable>	<variable>
HELP_YOUR_FILES.TXT	<variable>	<variable>	<variable>
HELP_YOUR_FILES.PNG	<variable>	<variable>	<variable>
<Carpeta Inicio (Startup)>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
HOW_DECRYPT.HTML	<variable>	<variable>	<variable>
HOW_DECRYPT.TXT	<variable>	<variable>	<variable>
HOW_DECRYPT.URL	<variable>	<variable>	<variable>

DECRYPT_INSTRUCTION.HTML	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.TXT	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.URL	<variable>	<variable>	<variable>
HELP_DECRYPT.HTML	<variable>	<variable>	<variable>
HELP_DECRYPT.TXT	<variable>	<variable>	<variable>
HELP_DECRYPT.PNG	<variable>	<variable>	<variable>
HELP_DECRYPT.URL	<variable>	<variable>	<variable>
HELP_YOUR_FILES.HTML	<variable>	<variable>	<variable>
HELP_YOUR_FILES.TXT	<variable>	<variable>	<variable>
HELP_YOUR_FILES.PNG	<variable>	<variable>	<variable>
<todas las rutas que tienen al menos un archivo cifrado>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
HOW_DECRYPT.HTML	<variable>	<variable>	<variable>
HOW_DECRYPT.TXT	<variable>	<variable>	<variable>
HOW_DECRYPT.URL	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.HTML	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.TXT	<variable>	<variable>	<variable>
DECRYPT_INSTRUCTION.URL	<variable>	<variable>	<variable>
HELP_DECRYPT.HTML	<variable>	<variable>	<variable>
HELP_DECRYPT.TXT	<variable>	<variable>	<variable>
HELP_DECRYPT.PNG	<variable>	<variable>	<variable>
HELP_DECRYPT.URL	<variable>	<variable>	<variable>
HELP_YOUR_FILES.HTML	<variable>	<variable>	<variable>
HELP_YOUR_FILES.TXT	<variable>	<variable>	<variable>
HELP_YOUR_FILES.PNG	<variable>	<variable>	<variable>
HOW_DECRYPT.HTML	<variable>	<variable>	<variable>
HOW_DECRYPT.TXT	<variable>	<variable>	<variable>
HOW_DECRYPT.URL	<variable>	<variable>	<variable>
<raiz>:\Users\<Nombre_del_usuario>\AppData\Roaming\<8 números aleatorios>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
<8_números_aleatorios>	<variable>	<variable>	<variable>

12. DETECCIÓN

Para detectar si un equipo se encuentra, o ha estado infectado, se ejecutará 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. También se podrán utilizar herramientas propias del sistema operativo.

12.1 HERRAMIENTAS DEL SISTEMA

Un compromiso claro del sistema es la existencia en el Escritorio del usuario, en la carpeta de Menú de Inicio y en cada una de las carpetas en las que se encuentren archivos cifrados, de los archivos que informan sobre el secuestro de los archivos que varían según la versión del código dañino que haya afectado. Esta es la relación de todos ellos:

HOW_DECRYPT.HTML
HOW_DECRYPT.TXT
HOW_DECRYPT.URL
DECRYPT_INSTRUCTION.HTML
DECRYPT_INSTRUCTION.TXT
DECRYPT_INSTRUCTION.URL
HELP_DECRYPT.HTML

HELP_DECRYPT.TXT
HELP_DECRYPT.PNG
HELP_DECRYPT.URL
HELP_YOUR_FILES.HTML
HELP_YOUR_FILES.TXT
HELP_YOUR_FILES.PNG
HOW_DECRYPT.HTML
HOW_DECRYPT.TXT
HOW_DECRYPT.URL

Otro indicador de compromiso, y esto dependerá de la versión que afecte, es la desaparición de los archivos originales y la existencia de archivos nuevos con nombres y/o extensiones aleatorias. En otras versiones simplemente se verán los ficheros originales pero no se podrán abrir al haberse modificado su contenido.

También se puede comprobar ejecutando el Editor de Registro del Sistema (Inicio -> Ejecutar -> regedit.exe). Una vez arrancado el Editor del Registro, se buscará la siguiente entrada:

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\<nombre_aleatorio1>] = *%APPDATA%*\<nombre_aleatorio1>\<nombre_aleatorio2>

En el caso de que una entrada exista con ese patrón, es un posible indicio de compromiso del sistema. Para asegurar que el compromiso es real, buscará el binario al que hace referencia y se validará calculando su *hash* MD5 y comparándolo con los siguientes:

e73806e3f41f61e7c7a364625cd58f65
48e00f5fce2f6e645ac3901aae3facb1

Si no se encuentra, se puede utilizar algún servicio de antivirus que intente identificar el binario.

12.2 MANDIANT

Se ha generado un nuevo archivo indicador de compromiso. El nombre del indicador generado es con GUID "*528f072e-355b-476c-8ee6-e1c8dee69bed*". 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.

13. DESINFECCIÓN

Con el fin de eliminar el código dañino, hay que iniciar sesión con un usuario *Administrador* o que posea privilegios administrativos.

Para eliminar el código dañino del sistema comprometido se usará el Editor del Registro (Inicio -> Ejecutar -> regedit.exe) y se procederá a acceder a la siguiente entrada:

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\<nombre_aleatorio1>] = *%APPDATA%*\<nombre_aleatorio1>\<nombre_aleatorio2>

Se procederá a borrar dicha entrada con el nombre aleatorio así como la carpeta para, posteriormente, acceder a la ruta de *%APPDATA%* que se indica y borrar el binario.

Posteriormente se accederá a la siguiente rama en el registro:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows Nt\SystemRestore

En ella se buscará la entrada "*DisableSR*" y, en el caso de que exista con un valor de "*1*", se pondrá su valor a "*0*" o se borrará la entrada. De esta forma se podrá volver a utilizar los puntos de restauración en el sistema.

Finalmente, se deberían eliminar todos los archivos referentes a la información de secuestro en todas las carpetas en las que se encuentren.

No existe forma conocida en el momento actual para recuperar los archivos cifrados por el código dañino debiéndose recurrir a usar copias de seguridad previas de dichos archivos para obtener la información.

14. INFORMACIÓN DEL ATACANTE

De los análisis realizados, se obtuvieron los siguientes 3 dominios C2 a los que intentó establecer conexión.

14.1 ABELINDIA.COM

La información WHOIS de dicho dominio devuelve la siguiente información:

Whois & Quick Stats

Email	abuse@name.com is associated with ~1,432,741 domains abelindi...@protecteddomainservices.com
Registrant Org	Whois Privacy Protection Service, Inc. was found in ~2,264,119 other domains
Registrar	NAME.COM, INC.
Registrar Status	clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Dates	Created on 2006-11-30 - Expires on 2016-11-30 - Updated on 2015-12-02
Name Server(s)	NS2432.DIZINC.COM (has 77,518 domains) NS2433.DIZINC.COM (has 77,518 domains)
IP Address	66.7.210.114 - 217 other sites hosted on this server

Ilustración 10. Información WHOIS del dominio ABELINDIA.COM

14.1.1 GEOLOCALIZACIÓN

La geolocalización de la IP "66.7.210.114" muestra la siguiente información:

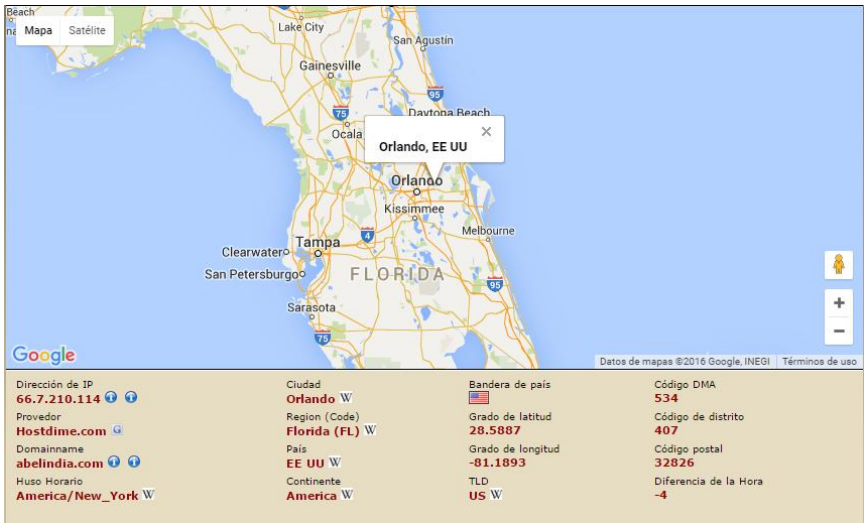


Ilustración 11. Geolocalización de la IP "66.7.210.114"

14.2 PURPOSENOWACADEMY.COM

La información WHOIS de dicho dominio devuelve la siguiente información:

Registrant Org	Domains By Proxy, LLC was found in ~11,298,354 other domains
Registrar	GODADDY.COM, LLC
Registrar Status	clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited, clientRenewProhibited https://icann.org/epp#clientRenewProhibited, clientTransferProhibited https://icann.org/epp#clientTransferProhibited, clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Dates	Created on 2015-10-28 - Expires on 2016-10-28 - Updated on 2015-10-28
Name Server(s)	NS51.DOMAINCONTROL.COM (has 40,283,810 domains) NS52.DOMAINCONTROL.COM (has 40,283,810 domains)
IP Address	184.168.47.225 - 359,874 other sites hosted on this server
IP Location	- Arizona - Scottsdale - Godaddy.com Llc
ASN	AS26496 AS-26496-GO-DADDY-COM-LLC - GoDaddy.com, LLC, US (registered Oct 01, 2002)
Domain Status	Registered And Active Website

Ilustración 12. Información WHOIS del dominio PURPOSENOWACADEMY.COM

14.2.1 GEOLOCALIZACIÓN

La geolocalización de la IP "184.168.47.225" muestra la siguiente información:

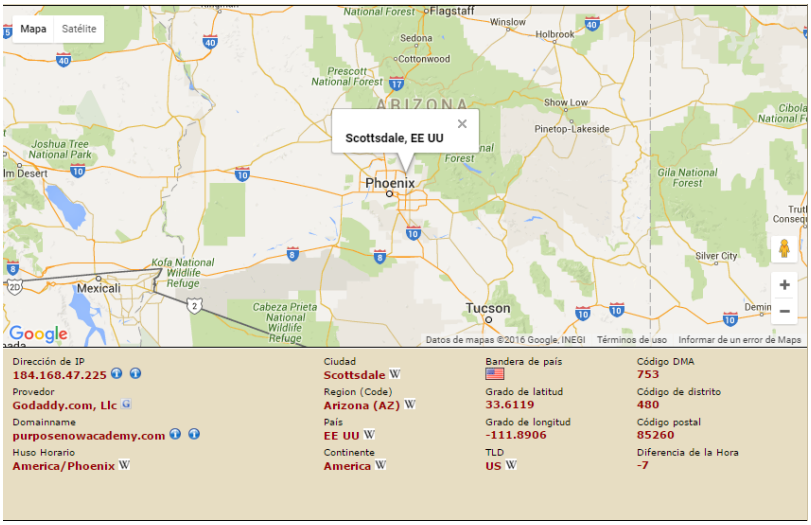


Ilustración 13. Geolocalización de la IP "184.168.47.225"

14.3 MYCAMPUSJUICE.COM

La información WHOIS de dicho dominio devuelve la siguiente información:

Email	abuse@enom.com is associated with ~11,283,769 domains peter.magala@tcs.com is associated with ~3 domains
Registrar	ENOM, INC.
Registrar Status	ok https://icann.org/epp#ok
Dates	Created on 2015-10-26 - Expires on 2016-10-26 - Updated on 2015-11-21
Name Server(s)	NS1.ARVIXESHARED.COM (has 25,192 domains) NS2.ARVIXESHARED.COM (has 25,192 domains)
IP Address	143.95.248.187 - 233 other sites hosted on this server
IP Location	- California - Los Angeles - Athenix Inc.
ASN	AS36024 COLO4-CO - Colo4, LLC, US (registered Jul 28, 2005)
Domain Status	Registered And Active Website

Ilustración 14. Información WHOIS del dominio MYCAMPUSJUICE.COM

14.3.1 GEOLOCALIZACIÓN

La geolocalización de la IP "143.95.248.187" muestra la siguiente información:

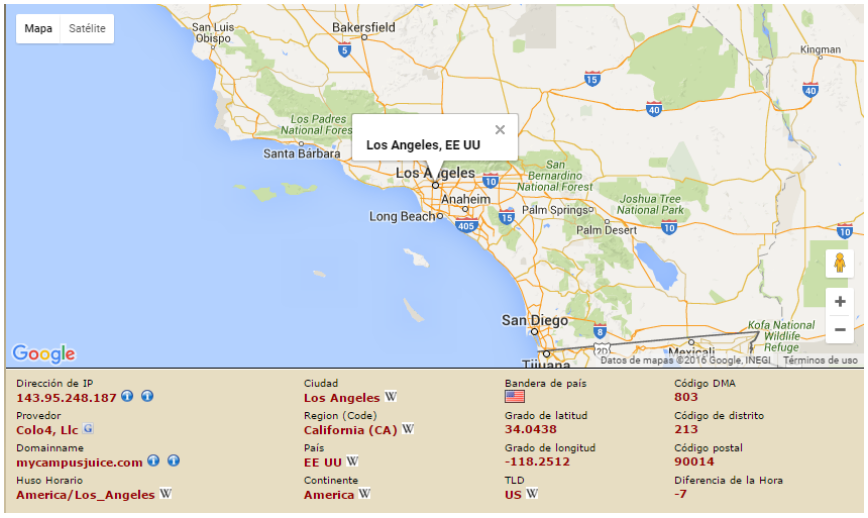


Ilustración 15. Geolocalización de la IP "143.95.248.187"

15. REFERENCIAS

https://www.cryptowalltracker.org/

ANEXOS

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="528f072e-355b-476c-8ee6-
e1c8dee69bed"                                      last-modified="2016-05-24T15:00:21"
xmlns="http://schemas.mandiant.com/2010/ioc">
  <short_description>Ransom.CryptoWall</short_description>
  <description>IOC para detectar RANSOM.CRYPTOWALL</description>
  <authored_by>CCN-CERT</authored_by>
  <authored_date>2016-04-22T05:51:02</authored_date>
  <links />
  <definition>
    <Indicator operator="OR" id="3f0b6197-3670-47c8-a49e-f0662a492925">
      <IndicatorItem id="c45da768-033b-432d-8371-5984b4b78f5e" condition="contains">
        <Context document="Network" search="Network/DNS" type="mir" />
        <Content type="string">ABELINDIA.COM</Content>
      </IndicatorItem>
      <IndicatorItem id="f9443147-5834-4f2a-a00e-dc3662d24b8a" condition="contains">
        <Context document="Network" search="Network/DNS" type="mir" />
        <Content type="string">PURPOSENOWACADEMY.COM</Content>
      </IndicatorItem>
      <IndicatorItem id="9a504e5d-597d-4cce-91eb-e0ac1b7dd4f3" condition="contains">
        <Context document="Network" search="Network/DNS" type="mir" />
        <Content type="string">MYCAMPUSJUICE.COM</Content>
      </IndicatorItem>
      <IndicatorItem id="47d05373-8b32-43fe-9c0d-d69985234295" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">e73806e3f41f61e7c7a364625cd58f65</Content>
      </IndicatorItem>
      <IndicatorItem id="2b3e1063-b123-4b2f-bfce-8817b7aabd81" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">48e00f5fce2f6e645ac3901aae3facb1</Content>
      </IndicatorItem>
      <IndicatorItem id="2a224a15-a69b-4171-9578-b12f9eefad72" condition="contains">
        <Context document="FileItem" search="FileItem/FileName" type="mir" />
        <Content type="string">help_your_files.</Content>
      </IndicatorItem>
      <IndicatorItem id="03da8635-5c96-4ed1-9810-e93093f64e64" condition="contains">
        <Context document="FileItem" search="FileItem/FileName" type="mir" />
        <Content type="string">HOW_DECRYPT.</Content>
      </IndicatorItem>
```

```
      <IndicatorItem id="d6d2a1ee-7d17-4f0e-b268-a94258f2d865" condition="contains">
        <Context document="FileItem" search="FileItem/FileName" type="mir" />
        <Content type="string">DECRYPT_INSTRUCTION.</Content>
      </IndicatorItem>
      <IndicatorItem id="a48028ed-b520-4482-9ff1-e310221f9b16" condition="contains">
        <Context document="FileItem" search="FileItem/FileName" type="mir" />
        <Content type="string">HELP_DECRYPT.</Content>
      </IndicatorItem>
    </Indicator>
  </definition>
</ioc>
```

YARA

```
rule CryptoWall {
meta:
    description = "Regla para detectar RANSOM.CRYPTOWALL"
    author = "CCN-CERT"
    version = "1.0"
strings
    $ = { 83 C4 04 0F B7 C0 33 45 F8 25 FF 00 00 00 33 34 }
    $ = { 85 30 69 41 00 89 75 F8 8B 4D F4 83 C1 02 89 4D }
    $ = { F4 EB AE }
condition:
    all of them
}
```