Informe Código Dañino CCN-CERT ID-19/16

Ransom.DMALocker

Agosto de 2016

CENTRO CRIPTOLOGICO NACIONAL
2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC,
ou=sello electrónico, serialNumber=S2800155J,
o=CENTRO CRIPTOLOGICO NACIONAL, cn=CENTRO
CRIPTOLOGICO NACIONAL, c=ES
2016.08.12 13:39:56 +02'00'

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.DMALocker***", 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.

Se distribuía inicialmente en campañas usando otros códigos dañinos que dieran acceso remoto al sistema de la víctima. Posteriormente se han usado campañas de correos electrónicos de tipo *Phishing* así como binarios que simulan ser otros ejecutables (un troyano clásico).

El código dañino tiene su origen en Polonia, existiendo versiones tanto en polaco como en inglés.

3. INFORMACIÓN DE VERSIONES DEL CÓDIGO DAÑINO

En este apartado se muestran las diferencias y los cambios producidos entre las versiones del código dañino.

3.1 VERSIONES DEL CÓDIGO DAÑINO

El código dañino se ha ido modificando desde su aparición habiéndose encontrado cinco versiones cuyas características más importantes se enumeran a continuación:

3.1.1 PRIMERA VERSIÓN

- Su primera aparición conocida se remonta a diciembre de 2015.
- El tamaño es de 89.632 bytes (88 KB).
- Existe versión en polaco y en inglés.
- Es posible descifrar los archivos con herramientas externas.
- Crea un archivo de texto con la información del proceso para poder descifrar los archivos cifrados.
- Accede a rutas predeterminadas en su código, incluyendo una con la carpeta de "*Documentos*" con el nombre en polaco: "*Dokumentary*".
- Accede a los recursos de red que han sido montados previamente.
- Contiene su clave de cifrado AES al final del archivo del código dañino, ocupa 32 bytes.
- El proceso de cifrado sólo utiliza AES en modo EBC.
- Marca los archivos cifrados con la cadena "ODSZYFRUJ".

3.1.2 SEGUNDA VERSIÓN

- Su primera aparición conocida se remonta a enero de 2016.
- El tamaño es de 98.848 bytes (96 KB).
- Existe versión en polaco y en inglés.
- Es posible descifrar los archivos con herramientas externas.
- Crea un archivo de texto con la información del proceso para poder descifrar los archivos cifrados.
- Accede a rutas predeterminadas en su código, incluyendo una con la carpeta de "*Documentos*" con el nombre en polaco: "*Dokumentary*".
- Contiene su clave de cifrado AES al final del archivo del código dañino y ocupa 32 bytes.
- Accede a los recursos de red que han sido montados previamente.
- El proceso de cifrado sólo usa AES en modo EBC.
- Marca los archivos cifrados con la cadena "ABCXYZ11".

3.1.3 TERCERA VERSIÓN

- Su primera aparición conocida se remonta a febrero de 2016.
- El tamaño es de 372.224 bytes (363 KB).
- Es posible descifrar los archivos con herramientas actuales, aunque depende de la muestra y de los archivos cifrados.
- Crea un archivo de texto con la información del proceso para poder descifrar los archivos cifrados.
- Accede a rutas predeterminadas en su código, sin embargo, está corregido el error de la carpeta en lenguaje polaco.
- Accede a los recursos de red que han sido montados previamente.
- El proceso de cifrado usa AES en modo ECB y RSA para cifrar la clave usada en cada archivo.
- La clave pública RSA está embebida en el código dañino, por lo que la misma clave es usada para todas las víctimas. Debido a esto si se pudiera acceder a un descifrador, se podría usar para cualquier víctima infectada con ese mismo código dañino.
- La marca de cifrado en los archivos es: !DMALOCK
- La clave AES ya no es única de la muestra ni se guarda al final del archivo; en su lugar se genera una clave aleatoria para cada archivo que se vaya a cifrar.
- El algoritmo de generación de claves AES es débil, por lo tanto se puede calcular por fuerza bruta.

3.1.4 CUARTA VERSIÓN

- Su primera aparición conocida se remonta al 22 de febrero de 2016.
- El tamaño es de 205.312 bytes (200 KB).
- No es posible descifrar los archivos con herramientas actuales.
- Accede a rutas predeterminadas en su código.
- Crea un archivo de texto con la información del proceso para poder descifrar los archivos cifrados.
- Accede a los recursos de red que han sido montados previamente.
- El proceso de cifrado usa AES en modo ECB y RSA para cifrar la clave usada en cada archivo.
- La clave pública RSA está embebida en el código dañino, por lo que la misma clave es usada para todas las víctimas. Debido a esto si se pudiera acceder a un descifrador, se podría usar para cualquier víctima infectada con ese mismo código dañino.
- La marca de cifrado en los archivos es: !DMALOCK3.0

3.1.5 QUINTA VERSIÓN

- Su primera aparición conocida se remonta al 19 de mayo de 2016.
- El tamaño es de 320.512 bytes (313 KB).
- No es posible descifrar los archivos con herramientas actuales.
- Accede a rutas predeterminadas en su código.
- Crea un archivo de texto con la información del proceso para poder descifrar los archivos cifrados.
- Crea un archivo de procesamiento por lotes (.BAT) en el sistema comprometido.
- Accede a los recursos de red que han sido montados previamente.
- El proceso de cifrado usa AES en modo ECB y RSA para cifrar la clave usada en cada archivo.
- La clave pública RSA es obtenida del servidor C2 (Mando y Control, al igual que la clave privada una vez se ha realizado el pago.
- Es la primera versión que funciona exclusivamente teniendo acceso a Internet.
- La marca de cifrado en los archivos es: !DMALOCK4.0

3.2 EXTENSIONES A CIFRAR

3.2.1 PRIMERA VERSIÓN A CUARTA VERSIÓN

El código dañino cifra cualquier archivo que no tenga alguna de las siguientes extensiones:

.exe	**.msp**	**.scr**
.msi	**.com**	**.sys**
.dll	**.lnk**	**.cpl**
.pif	**.hta**	**.msc**

3.2.2 QUINTA VERSIÓN

El código dañino cifra cualquier archivo que no tenga alguna de las siguientes extensiones:

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

3.3 EXTENSIÓN AÑADIDA A LOS ARCHIVOS CIFRADOS

El código dañino no añade ninguna extensión a los archivos cifrados.

3.4 ARCHIVOS DE RESCATE

El código dañino crea un archivo de texto en una ruta predeterminada en su código que cambia dependiendo de la versión del código dañino y el idioma para el que va dirigido.

También crea una ventana con información acerca del secuestro de los archivos y el procedimiento a seguir para poder recuperarlos. Esta ventana muestra el texto en inglés o en polaco dependiendo de la versión del código dañino.

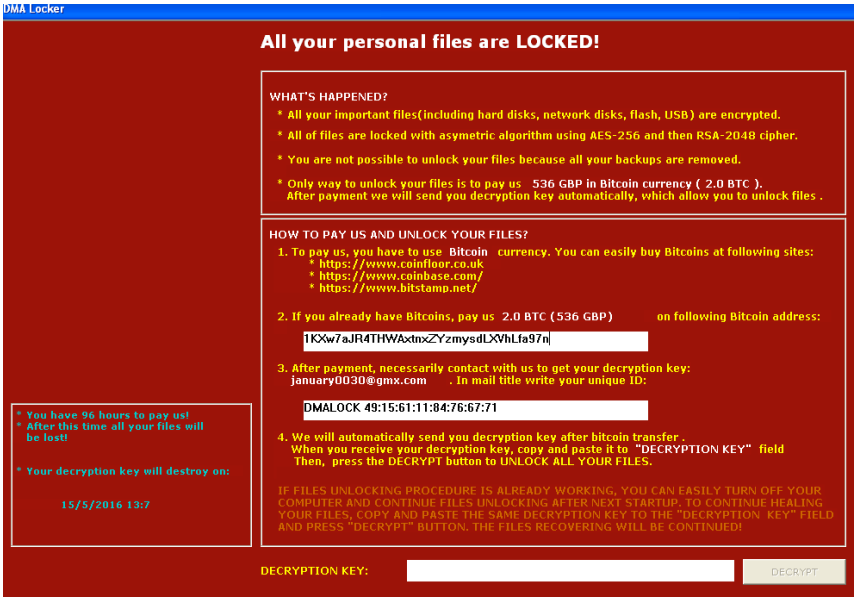


Ilustración 1. Ventana mostrada en la primera y segunda versión

La tercera y cuarta versión tienen una interfaz modificada:

Ilustración 2. Ventana mostrada en la tercera y cuarta versión

La interfaz de la quinta versión añade más campos, por ejemplo, la posibilidad de descifrar un archivo y demostrar que se tiene la clave privada para descifrar todo el sistema comprometido.

Ilustración 3. Ventana mostrada en la quinta versión

El texto mostrado en el archivo creado en el sistema, acerca del secuestro y el procedimiento a seguir en las muestras analizadas en versiones anteriores a la quinta es el siguiente:

Attention! ! !

All of your copies of your system have been permanently deleted and the data on all partitions and workstations have been encrypted!

Stay calm.

You can recover all your data by making a payment of 1072 GBP in Bitcoin currency in order to receive a decryption key.

In order to purchase Bitcions you can use www.coinbase.com

After buying BTC send the equivalent of 1072 GBP to our BTC adress:

166vHLnGB1pCQGxdBkRiMkHW5WGQDbsw6s

After payment contact us to receive your decryption key. In mail title write your unique ID: DMALOCK 43:41:90:35:25:13:61:92

Our e-mail: team4004@gmx.com

ATTENTION!

To ensure you that you can recover your data we are able to decrypt two files of your choice that are not larger than 1MB!

ATTENTION!

Even if your antivirus has removed our program, your data may be still recovered!

El archivo de texto creado en la quinta versión difiere de los anteriores en el contenido, siendo mucho más reducido e indicando que se debe visitar una página web para más información:

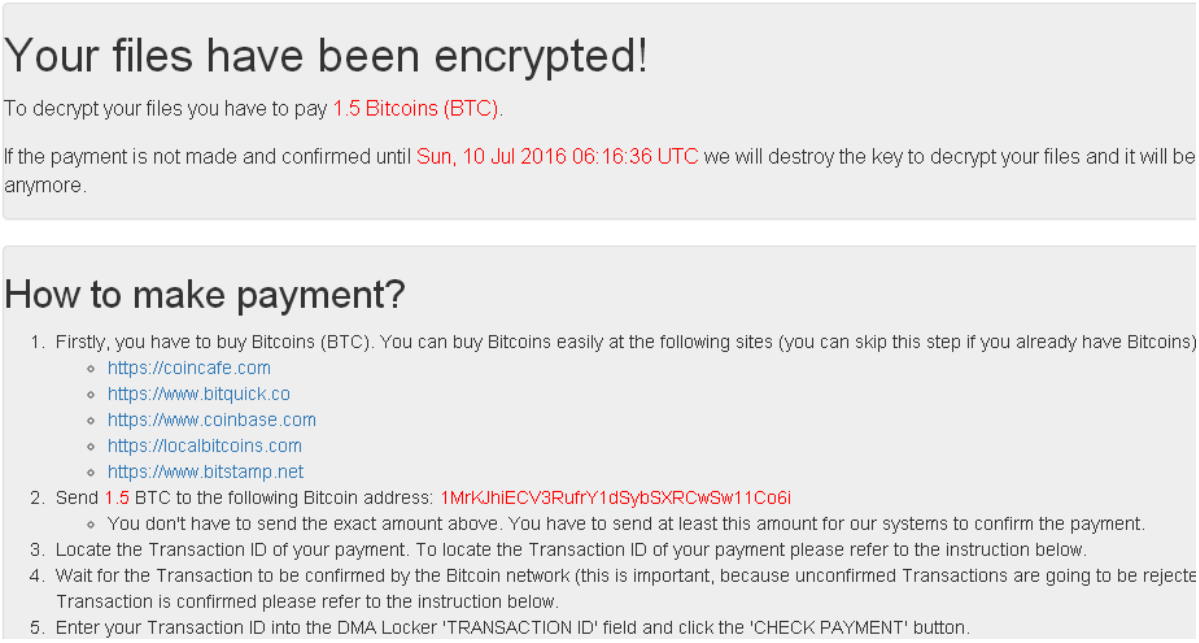
! ! ! ATTENTION ! ! !

ALL YOUR FILES HAVE BEEN ENCRYPTED!

- IF YOU WANT TO RECOVER YOUR FILES

FOLLOW THE INSTRUCTIONS AT THIS WEBSITE:

Un ejemplo de la página web indicada sería el siguiente:



Your files have been encrypted!

To decrypt your files you have to pay 1.5 Bitcoins (BTC).

If the payment is not made and confirmed until Sun, 10 Jul 2016 06:16:36 UTC we will destroy the key to decrypt your files and it will be anymore.

How to make payment?

1. Firstly, you have to buy Bitcoins (BTC). You can buy Bitcoins easily at the following sites (you can skip this step if you already have Bitcoins)
   - https://coincafe.com
   - https://www.bitquick.co
   - https://www.coinbase.com
   - https://localbitcoins.com
   - https://www.bitstamp.net
2. Send 1.5 BTC to the following Bitcoin address: 1MrKJhiECV3RufrY1dSybSXRCwSw11Co6i
   - You don't have to send the exact amount above. You have to send at least this amount for our systems to confirm the payment.
3. Locate the Transaction ID of your payment. To locate the Transaction ID of your payment please refer to the instruction below.
4. Wait for the Transaction to be confirmed by the Bitcoin network (this is important, because unconfirmed Transactions are going to be rejecte Transaction is confirmed please refer to the instruction below.
5. Enter your Transaction ID into the DMA Locker 'TRANSACTION ID' field and click the 'CHECK PAYMENT' button.

Ilustración 4. Web con información sobre el secuestro de los archivos

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 y el procedimiento a seguir para la recuperación de los archivos.
- Enumera las unidades de disco del sistema y los recursos de red.
- Cifra archivos en esas unidades y/o recursos.
- Modifica el registro del sistema para asegurar su persistencia.
- Muestra información en una ventana acerca del secuestro de los archivos y el procedimiento a seguir para recuperarlos.
- En el caso de la quinta versión, establece comunicación con su servidor C2 y, en caso de que no pueda establecerla, no cifrará ningún archivo y queda a la espera de obtener conectividad.

5. DETALLES GENERALES

Las muestras analizadas se corresponden con las siguientes firmas MD5:

d35344b1f48764ba083e51438121e6a9
4190df2af81ece296c465e245fc0caea
6fbd3cdcafd6695c384a1119873786aa
28b44669d6e7bc7ede7f5586a938b1cb
760078d6bc389ca677b14a81b7a58fc3
f676aba2d996eed2c194e9f5944446fa

Los binarios tienen formato PE (Portable Executable), es decir, son ejecutables para sistemas operativos Windows, concretamente para 32 bits.

En las muestras analizadas las fechas internas se corresponden a las campañas del código dañino, por ejemplo, 4 de febrero de 2016.

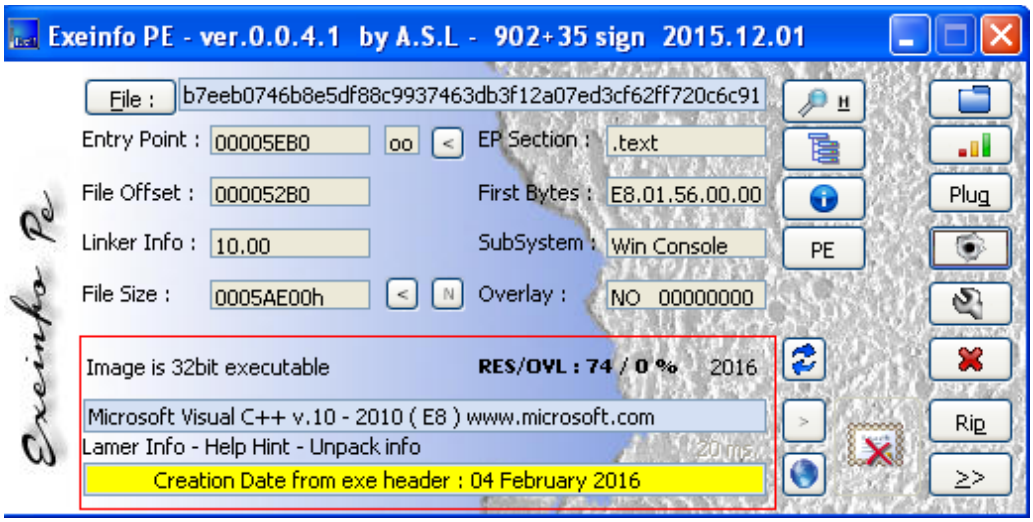


Ilustración 5. Detalles del binario

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:

- En sus primeras versiones el vector de entrada era aprovechar la infección de otros códigos dañinos de control remoto para su posterior ejecución. En posteriores versiones, se han enviado correos electrónicos de tipo *Phishing* y se han distribuido por redes P2P simulando ser otros programas.

- El código dañino, desde la primera a la cuarta versión, ni está comprimido, ni esta embebido en ningún "*dropper*". En la quinta versión sí está empaquetado y cifrado.
- Se copia en el sistema comprometido en una ruta predeterminada.
- Crea determinados archivos en el sistema comprometido sobre el secuestro de los archivos y el procedimiento a seguir para recuperarlos.
- Crea un hilo persistente que busca aplicaciones para realizar copias de seguridad y las cierra en el caso de que las encuentre excepto en la quinta versión.
- Modifica el registro para asegurarse persistencia en el sistema.
- Enumera las unidades del sistema y recursos de red (excepto en la primera versión). En todas ellas busca archivos que no cumplan unos patrones por extensión y los cifra.
- En la quinta versión crea un archivo de procesamiento por lotes y modifica el registro para asegurar su ejecución.
- Elimina los *Shadow Volume*[1] del sistema comprometido.
- Muestra una interfaz al usuario del sistema indicando lo ocurrido y los pasos a realizar para recuperar los archivos.

7. CARACTERÍSTICAS TÉCNICAS

7.1 PRIMERA VERSIÓN A CUARTA VERSIÓN

El código dañino está compilado como una aplicación de consola de comandos y por ello su primera acción tras ejecutarse es ocultar la ventana de comandos mediante el uso de la función "*ShowWindow*".

Sólo en la tercera y cuarta versión del código dañino procede a realizar una comprobación de integridad de la dirección del pago por *Bitcoin*[2] para evitar que alguien copie el código y cambie la dirección de pago. Para ello comprueba los primeros tres caracteres de forma que, en el caso de que coincidan, continúa el flujo normal y, en caso contrario, termina su ejecución.

Su siguiente acción es comprobar que la ruta "*C:\ProgramData*"[3] existe en el sistema comprometido. Para ello usa la función "*GetFileAttributesA*". En el caso de que dicha ruta exista, guarda en memoria un conjunto de rutas predeterminadas a archivos en esa carpeta y en caso contrario prepara un conjunto de rutas a archivos en la carpeta "*C:\Documents and Settings\All Users\Dokumenty*". En la tercera y cuarta versión la ruta es "*C:\Documents and Settings\All Users*". En la primera versión las rutas son "*C:\ProgramData*" y "*C:\WINDOWS*".

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

[2] https://bitcoin.org/es/

[3] En Windows 7 es una carpeta oculta que apunta a "*C:\Users\All Users*"

A continuación, crea en la ruta calculada el archivo "*cryptinfo.txt*" usando la función "*CreateFileA*". En dicho archivo escribe el texto acerca del secuestro de los ficheros y el procedimiento a seguir para poder recuperarlos. Este mensaje está escrito en inglés o en polaco dependiendo de la muestra.

Posteriormente, se crea un hilo que enumera todos los procesos activos en el sistema y comprueba que no sea alguno de los siguientes:

rstrui.exe
ShadowExplorer.exe
sesvc.exe
cbengine.exe

En el caso que encuentre alguno de los procesos indicados anteriormente, procede a finalizar dicho proceso. El hilo se ejecuta de forma continuada con un descanso entre ejecuciones de 350 milisegundos.

Las aplicaciones indicadas pertenecen al Restaurador del Sistema ("*rstrui.exe*"), el interfaz y el servicio para manipular *Shadow Volumes* en el sistema ("*ShadowExplorer.exe*" y "*sesvc.exe*") y la aplicación principal de "*Microsoft Azure Backup*" ("*cbengine.exe*").

A continuación, se crea una entrada en el registro de Windows para asegurar su persistencia. En el caso de la primera y segunda versión son las siguientes:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys = <*ruta_calculada*>\ntserver.exe
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys = <*ruta_calculada*>\ntserver.exe

En el caso de la tercera y cuarta versión la entrada tiene otro valor:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys = <*ruta_calculada*>\svhosd.exe
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys" = <*ruta_calculada*>\svchosd.exe

Para todas las versiones, se crea la siguiente entrada para mostrar el texto acerca del secuestro de los archivos en el inicio del sistema:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cryptedinfo = notepad <*ruta_calculada*>\cryptinfo.txt

A continuación intenta abrir con permisos de lectura un archivo llamado "*decrypting.txt*" en la misma ruta calculada anteriormente. En el caso de que no lo consiga, intentará abrir un archivo llamado "*start.txt*" en la misma ruta con permisos de lectura. En cualquiera de los dos casos, si el archivo es encontrado, y sin importar su contenido, el código dañino procederá a mostrar la interfaz del secuestro de los archivos para después finalizar su ejecución sin cifrar ningún archivo.

El archivo "*start.txt*" se crea cuando se comienza el proceso de cifrado, pero no se ha terminado, y el archivo "*decrypting.txt*" es creado al finalizar el proceso de cifrado. El código dañino asume que si existe el primero archivo, no tiene que volver a cifrar lo ya cifrado, y si existe el segundo, que el proceso de cifrado ya comenzó y no necesita volver a ser iniciado.

Posteriormente el código dañino obtiene el nombre de su imagen en disco utilizando el PEB[4] ("*Process Environment Block*") y accediendo a la estructura "*RTL_USER_PROCESS_PARAMETERS*[5]":

```
mov     [esp+C], ebx
mov     eax, fs:[30]
mov     eax, [eax+10]       RTL_USER_PROCESS_PARAMETERS
mov     eax, [eax+3C]       RTL_USER_PROCESS_PARAMETERS->ImagePath
mov     [esp+C], eax
mov     esi, [esp+C]
mov     ecx, [418DBC]       dma_lock.004151B0
mov     eax, esi
jmp     short 00404420
```

Ilustración 6. Obteniendo nombre de imagen y ruta mediante el PEB

Con la ruta obtenida de su propia imagen en disco procede a mover el binario del código dañino, usando la función "*MoveFileW*", a la ruta calculada anteriormente con el nombre "*ntserver.exe*".

En este punto el código dañino, en las versiones primera y segunda, obtiene la clave de cifrado que usará posteriormente. Dicha clave está al final de su archivo en disco y ocupa 32 bytes. Cuando es leída, se asegura que esos bytes no son nulos y, posteriormente, crea un archivo llamado "*time_1.txt*" en la ruta calculada donde escribe la fecha y hora del inicio del cifrado de archivos.

Después muestra la interfaz al usuario con la información del secuestro del equipo y los pasos a proceder. Es muy relevante el hecho de que, pese a mostrar ese mensaje, el código dañino no ha comenzado a cifrar archivos, cosa que hará en el siguiente reinicio del sistema.

A continuación el código dañino ejecuta un nuevo proceso de su propia imagen usando "*CreateProcess*" y finaliza su ejecución salvo que se esté ejecutando en la ruta precalculada continuará su ejecución. Por eso, en una infección normal, no empezará a cifrar hasta el siguiente reinicio.

En el caso de que continúe la ejecución, crea el archivo "*start.txt*" y obtiene la dirección de memoria a las funciones "*GetLogicalDrives*" y "*GetDriveTypeA*" a través

[4] https://www.aldeid.com/wiki/PEB-Process-Environment-Block

[5] http://foro.elhacker.net/asm/rtluserprocessparameters-t418775.0.html

del PEB: accede a todos los módulos en el contexto del código dañino y busca en su tabla de exportaciones las funciones deseadas.

Una vez obtenidas las direcciones, llama a la función "*GetLogicalDrives*" obteniendo una estructura de las unidades del sistema comprometido. Por cada una de ellas, comenzando por la "C", procede a llamar a la función "*GetDriveTypeA*" para obtener el tipo al que pertenece la unidad. Si la unidad es un disco duro, extraíble o recurso de red se la considera adecuada para enumerar archivos de ella y a cifrar ciertos archivos.

En la tercera y cuarta versión, aparte de la comprobación previa, se utiliza la función "*QueryDosDeviceA*" para descartar unidades que sean *disquetes* ("Floppy") y CD-ROM (unidades digitales entre las que se engloban CD-ROM, DVD, Blue-ray, etc.).

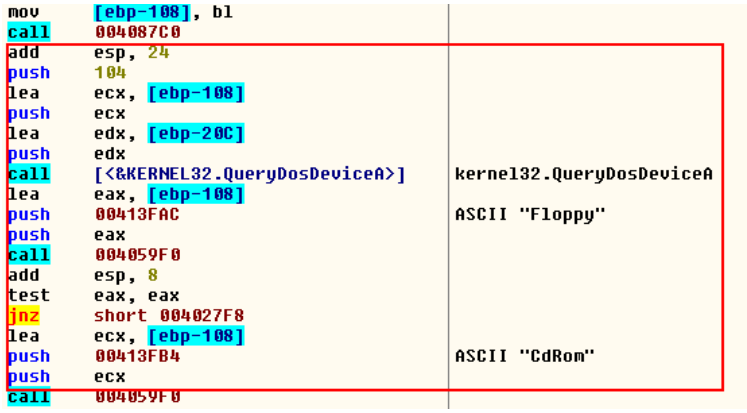


Ilustración 4. Detección de las unidades tipo *Floppy* y *CD-ROM*

En las unidades no descartadas se procede a comenzar a enumerar todos los archivos y carpetas mediante las funciones "*FindFirstFileA*", "*FindNextFileA*" y "*FindClose*".

Por cada carpeta encontrada, se llama de forma recursiva a la misma función de enumeración, y por cada archivo encontrado, se procede a comprobar que no se encuentre en una de las siguientes rutas:

"Windows"	**"Temp"**
"WINDOWS"	**"Sample Pictures"**
"Program Files"	**"Sample Music"**
"Program Files (x86)"	**"cache"**
"Games"	**"Cache"**

En el caso de que se encuentre en alguna de las rutas indicadas en la tabla anterior el archivo es ignorado. Del mismo modo, es comprobada la extensión del archivo de forma que si coincide con alguna de las expuestas en el apartado Extensiones A Cifrar, el código dañino lo ignorará y no será cifrado.

Si el archivo no es ignorado, procede a comprobar si el archivo está cifrado comparando sus primeros 8 bytes con una cadena predeterminada:

Primera versión: ODSZYFRUJ
Segunda versión: ABCXYZ11
Tercera versión: !DMALOCK
Cuarta versión: ¡DMALOCK3.0

En el caso de que no se encuentre la cadena, se procede a comprobar el tamaño del archivo de forma que, si no excede de los 953 MB, se comienza a cifrarlo.

Una vez finalizado el proceso de cifrado de todos los archivos, el código dañino finaliza su ejecución.

7.2 QUINTA VERSIÓN

El código dañino está compilado como una aplicación de consola de comandos y por ello su primera acción tras ejecutarse es ocultar la ventana de comandos mediante el uso de la función "*ShowWindow*".

Al igual que ocurre con las versiones anteriores, el código dañino comprueba la existencia de la carpeta "*C:\ProgramFiles*" para que, en el caso de que exista, usarla como ruta para crear archivos mientras que, en el caso de que no exista, usar la ruta "*C:\Documents and Settings\All Users*".

Posteriormente procede a comprobar que no tenga ninguna redirección de archivos en un sistema operativo de 64 bits. Para ello usa "*GetProcAddress*" para obtener la dirección de memoria de las funciones "*Wow64DisableWow64FsRedirection*" y "*Wow64EnableWow64FsRedirection*". En el caso de que las pueda encontrar, procede a utilizarlas y borra los *Shadow Volumes* del sistema con el comando:

```
C:\WINDOWS\system32\vssadmin.exe delete shadows /For=%c: /all /quiet
```

Esta operación es realizada en un bucle enumerando todas las letras de unidades desde la A hasta la Z.

Posteriormente crea en la ruta calculada previamente el archivo "*cryptinfo.txt*" usando la función "*CreateFileA*". En dicho archivo escribe el texto acerca del secuestro de los ficheros y el procedimiento a seguir para poder recuperarlos.

A continuación escribe en el registro unas entradas para ganar persistencia.

```
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Firewall = <ruta_calculada>\svchosd.exe
```

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

Windows Update = <*ruta_calculada*>\select.bat

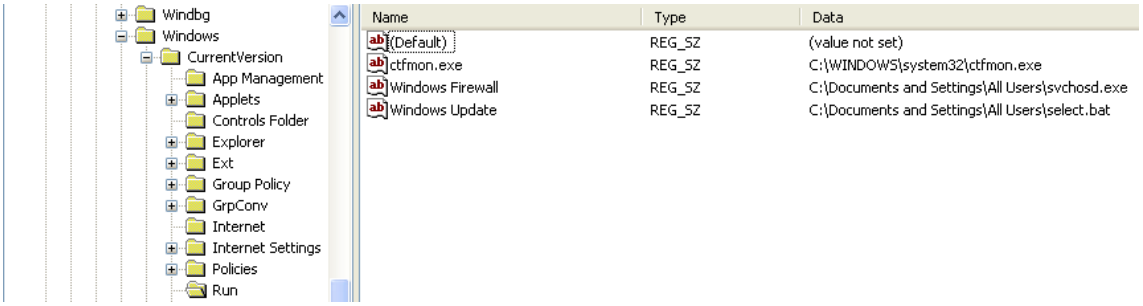


Ilustración 7. Creando entradas de persistencia en el registro

Posteriormente comprueba que no exista en el registro de Windows la entrada con el nombre "*dma_id*" en la siguiente rama.

[HKEY_CURRENT_USER\Software\dma_id]

En el caso de que no exista, procederá a intentar realizar la conexión a su servidor C2 que tiene embebido en su código: IP *5.8.63.54*.

Una vez conectado procederá a solicitar un identificador único para esa víctima al servidor que devolverá un valor hexadecimal que será escrito en el registro bajo la entrada anterior. Si no puede conectar con el servidor C2, el código dañino espera 5 minutos para volver a intentarlo y repite el proceso hasta que consigue establecer conexión.

Una vez que tiene almacenado el identificador, el código dañino procede a escribir en disco el archivo de texto sobre el secuestro al igual que un archivo de procesamiento por lotes (.bat) que se ejecutará en cada inicio del sistema a través de una de las entradas del registro antes mencionada.

El archivo de procesamiento por lotes comprueba si existe o no el binario del código dañino y, en el caso de que no exista, abre el bloc de notas con el archivo de texto acerca del secuestro de los archivos y el procedimiento a seguir para su recuperación. En caso de que exista, no hace nada. Esto lo hace por si el ejecutable del código dañino ha sido borrado, ya sea de forma manual o a través de alguna herramienta como un antivirus, y que así se pueda visualizar siempre el mensaje de secuestro.

if not exist <*ruta_calculada*>\svchosd.exe notepad <*ruta_calculada*>\cryptinfo.txt

Tras crear los archivos busca en el registro el siguiente campo:

[HKEY_CURRENT_USER\Software\dma_public_key]

En el caso de que no exista la entrada, el código dañino establecerá una conexión con el servidor C2 y solicitará una clave RSA pública enviando como parámetro el identificador único que recibió anteriormente. Una vez obtenida la clave, la guarda en ese mismo campo del registro.

Tras escribir en el registro la clave RSA, obtiene todas las unidades de disco del sistema y comprueba que sean de tipo disco duro o recurso de red. Por cada una de ellas también comprueba que no sean una unidad de disco ("*floppy*") o un CD-ROM para ignorarlos en el proceso de cifrado.

```
push    ecx             ; lpRootPathName
call    ds:GetDriveTypeA
lea     edx, [ebp+RootPathName]
push    edx
mov     esi, eax
push    esi
push    offset aDS      ; "%d %s\n"
call    _printf
add     esp, 0Ch
cmp     esi, 3          ; FIXED_DISK
jz      short _check_if_is_floppy_or_cdrom
cmp     esi, 4          ; NETWORK RESOURCE
jz      short _check_if_is_floppy_or_cdrom
test    esi, esi
jnz     short _check_letter_unit
```

Ilustración 8. Comprobando el tipo de las unidades encontradas

A continuación se inicia el proceso de cifrado enumerando todos los archivos y carpetas de cada unidad afectada de forma recursiva. Una vez que tiene todos los archivos a cifrar, comienza el cifrado de cada archivo siempre que no tenga alguna de las extensiones excluidas que se encuentran definidas en Extensiones a Cifrar y no se encuentre en ninguna de las siguientes carpetas:

"Windows"	**"Temp"**
"WINDOWS"	**"Sample Pictures"**
"Program Files"	**"Sample Music"**
"Program Files (x86)"	**"cache"**
"Games"	**"Cache"**

En el caso de que el archivo se encuentre en alguna de las carpetas indicadas o tenga alguna de las extensiones excluidas, se ignora y se pasa al siguiente.

Si el archivo es apto para ser cifrado se procede a comprobar sus primeros 8 bytes del archivo y si no coinciden con la siguiente cadena se comenzará el proceso de cifrado.

¡DMALOCK4.0

Una vez el proceso de cifrado finaliza, el código dañino establece una conexión con el servidor C2 y le solicita todos los datos de importancia para el rescate:

- Cuántos *Bitcoins* (BTC) se exigen para el descifrado de los archivos.

- La dirección de correo electrónico a mostrar al usuario de contacto.
- La dirección *Bitcoin* en la que efectuar el pago.
- Las fechas de aumento de precio y de borrado de la clave privada en el servidor C2.
- La cantidad con la que aumenta el rescate en *Bitcoins* en el caso de que se supere la fecha de aumento de precio.

Para el caso en el que no se pueda establecer conexión con servidor, el código dañino lleva embebidos unos valores por defecto para poder solicitar el pago del rescate: 4 *Bitcoins*; la dirección de *Bitcoin 16hHkyuzCDRFzoejVuqajqrnbmKHSmEfQM*; y la dirección de correo dma4004@zerobit.email.

Una vez finalizado el proceso de cifrado procede a mostrar una ventana con la información acerca del secuestro, la dirección *Bitcoin* en la que realizar el pago, la cantidad a pagar y el resto de detalles.

Esta misma ventana en la que muestra la información, actúa como descifrador ya que posee un campo donde el usuario puede introducir la identificación de la transacción realizada en el pago en *Bitcoins* si realiza el pago. Si se introduce la mencionada identificación, el código dañino procede a comprobar que dicha transacción es correcta, que ha finalizado correctamente y solicita al servidor C2 la clave privada RSA que, una vez que los autores del fraude comprueban que se ha realizado de forma correcta, es proporcionada para descifrar todos los archivos cifrados en el sistema.

8. CIFRADO Y OFUSCACIÓN

8.1 PRIMERA A CUARTA VERSIÓN

El código dañino utiliza el algoritmo AES en todas sus versiones y, además, el algoritmo RSA en la tercera y cuarta versión aunque en las notas del secuestro indique que siempre se usa RSA.

En la primera y segunda versión el código dañino utiliza una clave AES única embebida al final del archivo binario de 32 bytes y el modo de cifrado es ECB en bloques de 16 bytes. En estas dos versiones, una vez realizado el proceso de cifrado, el código dañino procede a borrar la clave del final del archivo.

En el caso de la tercera y la cuarta versión se crea una clave AES por cada uno de los archivos que se cifran y es almacenada cifrada con RSA en el fichero después de la palabra inicial que indica que es un fichero cifrado por este código dañino: en el caso de la tercera versión la palabra es "*!DMALOCK*" y en la cuarta versión "*!DMALOCK3.0*".

```
push    400h            ; size_t
mov     edi, eax
push    0               ; int
push    edi             ; void *
call    _memset
mov     ecx, [ebp+arg_4]
add     esp, 0Ch
lea     edx, [ebp+var_190]
push    edx             ; int
push    edi             ; void *
lea     eax, [ebp+var_138]
push    eax             ; int
push    ecx             ; int
mov     [ebp+var_190], 0
call    DMALocker_Encrypt_Key_With_RSA
push    ebx             ; FILE *
push    8               ; size_t
push    1               ; size_t
push    offset aDmalock ; "!DMALOCK"
call    _fwrite
mov     edx, [ebp+var_190]
```

Ilustración 9. Cifrado de la clave AES con RSA

El código dañino, en su ventana informativa acerca del secuestro de los archivos, posee un campo para introducir una clave de descifrado. En la primera y segunda versión ese campo admite texto y en el caso de la tercera y la cuarta versión el código dañino permite además seleccionar un archivo "*.key*" que contiene la clave privada RSA de descifrado.

8.2 QUINTA VERSIÓN

La quinta versión del código dañino es muy parecida a la tercera y cuarta versión pero el proceso de cifrado sigue otros pasos:

- Al igual que las versiones anteriores usa AES para cifrar los archivos y RSA para cifrar la clave usada en el algoritmo AES.
- Se utiliza una clave distinta para cada archivo, siendo ésta generada de forma aleatoria con la función "*CryptoGenRandom*".
- La clave pública RSA es solicitada al servidor C2, es decir, que en el caso de que no se pueda establecer conexión con el C2 el código dañino no cifrará nada.
- La clave RSA es guardada en el registro ofuscado y serializado. Para descifrar la clave RSA se utilizan dos funciones de Windows: "*CryptStringToBinaryA*" y "*CryptDecodeObjectEx*".
- Una vez descifrada y "deserializada" se obtiene un *blob* RSA que es utilizado mediante la función "*CryptImportKey*". Esta función extrae del *blob* la clave.
- Con dicha clave cifra la clave AES usada en cada archivo.
- Por último, se añade al principio de cada archivo la cadena de texto "*!DMALOCK4.0*" para marcarlo como cifrado seguida de la clave AES cifrada.

- El propio código dañino tiene la función de descifrado, pero sólo la usará cuando verifique un identificador de transacción del pago y se obtenga desde el servidor C2 la clave privada RSA.

9. PERSISTENCIA EN EL SISTEMA

9.1 PRIMERA A CUARTA VERSIÓN

El código dañino crea las siguientes entradas en el registro para asegurar su persistencia y para mostrar al usuario el archivo de texto con información acerca del secuestro de los archivos y el procedimiento a seguir:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **cssys = <*ruta_calculada*>\ntserver.exe**
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **cssys = <*ruta_calculada*>\ntserver.exe**
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **cryptedinfo = notepad <*ruta_calculada*>\cryptinfo.txt**

En alguna muestra de la tercera versión la entrada de reinicio es distinta en su valor de registro:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **cssys = <*ruta_calculada*>\svchosd.exe**
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **cssys = <*ruta_calculada*>\svchosd.exe**

La etiqueta "*ruta_calculada*" hace referencia a la ruta predefinida en el código dañino tal y como aparece explicado en el apartado Características Técnicas del presente informe.

9.2 QUINTA VERSIÓN

El código dañino crea las siguientes entradas en el registro para asegurar persistencia y para mostrar el archivo de texto con información acerca del secuestro.

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **Windows Firewall = <*ruta_calculada*>\svchosd.exe**
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

Windows Update = <*ruta_calculada*>\select.bat

La etiqueta "*ruta_calculada*" hace referencia a la ruta predefinida en el código dañino tal y como aparece explicado en el apartado Características Técnicas del presente informe.

10. CONEXIONES DE RED

En la quinta versión del código dañino se establece conexión con un servidor C2 la cual es obligatoria para que el código dañino pueda cifrar los archivos y secuestrarlos.

El código dañino establece hasta siete conexiones diferentes según la funcionalidad que requiera en cada momento del servidor. Los resultados de las operaciones se devuelven bajo el campo "*status*" pudiendo tener los siguientes valores:

0: Operación realizada correctamente
2: "Transaction ID confirmed! Confirming your payment, please be patient, it can take 15-20 minutes..."
4: "Your private key is currently deleted. You are late with payment".
7: "Your transaction need to be confirmed by server. It can take few hours. Check again for 1 hour".
8: "Invalid transaction ID"
9: "You have to wait 15 minutes to check again".

10.1 OBTENCIÓN DEL IDENTIFICADOR ÚNICO

El código dañino envía el siguiente comando para obtener el identificador único al servidor C2 del que recibe un *json* con el resultado de la operación y el identificador único. El identificador único devuelto por el servidor C2 sirve para saber qué víctima está siendo afectada y poder seguir su estado.

GET /crypto/gate?action=0 HTTP/1.1 Host: 5.8.63.54
{"status":0,"id":"5AE5AE09B79C49D5BD5C2881F8B05C63"}

10.2 OBTENCIÓN DE LA CLAVE RSA PÚBLICA

El código dañino envía el siguiente comando para la obtención de la clave pública RSA al servidor C2 y, tras recibirlo, devuelve un *json* con el resultado de la operación y la clave RSA pública.

GET /crypto/gate?action=1&botId=5AE5AE09B79C49D5BD5C2881F8B05C63 HTTP/1.1 Host: 5.8.63.54

```
{"status":0,"rsa_public_key":"-----BEGIN PUBLIC KEY-----
MIIBCgKCAQEAuvmHdmMomkNc0ihV+bVZvfHTx8HEG5esmoHZmZYuX0sB115+QIbBm\/
onzag2qAir9ZfEznlu8RIfrGpsMaCqyBe0COgJFqFeKndM3pP7YICm0DPVyyvRzFVwZNNAij
ACJ+C4damZkyyX0eDLJux8rsGSZE0nNifsZHTc66qgDd2vCyyC03b1LVv9GKJOv\/xP+wM
w7U6Jxdn8HXAXdZDITiPKpcQfLL2QOlBjqWmhK8\/V80fIDn0vUHniVIQG1nv3x70ujm4+iBix
9bUQmGJYEBYNeOgiMaVD9+fil165E3IEzm\/H1RRLUnO+j9xa50T7x862xJH0EOVUFYyNrm3
EZQIDAQAB-----END PUBLIC KEY-----"}
```

10.3 ORDEN DE ALMACENAR CLAVE PÚBLICA

El código dañino envía este comando al servidor C2 para indicar que almacene en su base de datos la clave RSA pública y que la asocie al identificador único del sistema afectado. El servidor únicamente devuelve el resultado de la operación.

```
GET /crypto/gate?action=2&botId=5AE5AE09B79C49D5BD5C2881F8B05C63 HTTP/1.1
Host: 5.8.63.54
```

```
{"status":0 }
```

10.4 OBTENCIÓN DE LOS DATOS ACERCA DEL SECUESTRO

El código dañino envía el siguiente comando para obtener los datos acerca del secuestro al servidor C2, el cual, tras recibirlo le devuelve un *json* con el resultado de la operación y los campos necesarios para informar al usuario del precio del rescate, tiempo máximo, etcétera.

```
GET /crypto/gate?action=3&botId=5AE5AE09B79C49D5BD5C2881F8B05C63 HTTP/1.1
Host: 5.8.63.54
```

```
{"status":0,"minimum_btc_confirmations":3,"bitcoin_address":"1MrKJhiECV3RufrY1dSyb
SXRCwSw11Co6i","ransom_amount":"1","private_key_destroy_timestamp":"2016-06-17
14.04.09","ransom_amount_increase_timestamp":"2016-06-13
14.04.42","ransom_amount_increase_amount":"1.5"}
```

10.5 COMPROBACIÓN DEL IDENTIFICADOR DE LA TRANSACCIÓN DEL PAGO

El código dañino envía el identificador único y el identificador de la transacción del pago con BitCoins al servidor C2 para que compruebe si es válido, informando de ello en la respuesta.

```
GET
/crypto/gate?action=4&botId=5AE5AE09B79C49D5BD5C2881F8B05C63&transactionId
=<id_de_la_transaccion> HTTP/1.1 Host: 5.8.63.54
```

```
{"status":2 }
```

10.6 OBTENCIÓN DE LA CLAVE RSA PRIVADA

Una vez realizada y confirmada la transacción del pago, el código dañino envía el siguiente comando solicitando al servidor C2 la clave RSA privada para poder descifrar la clave AES en cada archivo cifrado.

GET /crypto/gate?action=5&botId=5AE5AE09B79C49D5BD5C2881F8B05C63 HTTP/1.1 **Host: 5.8.63.54**
{"status":0,"rsa_private_key":"[clave_rsa_privada]"}

10.7 COMPROBAR EL ESTADO

El código dañino envía el comando siguiente para comprobar en qué estado del proceso se encuentra.

GET /crypto/gate?action=6&botId=5AE5AE09B79C49D5BD5C2881F8B05C63 HTTP/1.1 **Host: 5.8.63.54**
{"status":0,"bot_status":1}

El servidor puede devolver los siguientes estados del *bot*:

0: Estado inicial
1: Clave pública RSA guardada
3: Obteniendo la clave privada desde el servidor

11. DETECCIÓN

Para detectar si un equipo se encuentra o ha estado infectado, para cualquiera de sus usuarios, 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 al igual que las Herramientas del Sistema.

11.1 HERRAMIENTAS DEL SISTEMA

Para poder detectar el código dañino en el sistema usando herramientas del propio sistema operativo se usará el Editor del Registro (Inicio -> Ejecutar -> regedit.exe).

11.1.1 PRIMERA A CUARTA VERSIÓN

En el Editor del Registro se procederá a buscar las siguientes entradas:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys = *<ruta_calculada>*\<ntserver.exe> o < svchosd.exe>

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys = <*ruta_calculada*>\ntserver.exe
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cryptedinfo = notepad <*ruta_calculada*>\cryptinfo.txt

Sin embargo, la cuarta versión tiene las siguientes entradas:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Firewall = <*ruta_calculada*>\svchosd.exe
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Firewall = <*ruta_calculada*>\svchosd.exe
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Update = notepad <*ruta_calculada*>\cryptinfo.txt

En el caso de que se encuentre alguna de las entradas indicadas es un claro indicio de compromiso del sistema.

Se buscará un archivo ejecutable que puede llamarse "*ntserver.exe*" o "*svchosd.exe*" (en la tercera versión) en los siguientes directorios:

C:\ProgramData
C:\Documents and Settings\All Users\Dokumenty
C:\Documents and Settings\All Users

En el caso de que se encuentre un archivo con alguno de los nombres indicados es un claro indicador del compromiso del sistema.

Por supuesto, que no se puedan abrir los archivos que habitualmente se usan en el sistema es un claro indicador de que se han cifrado archivos en el sistema.

11.1.2 QUINTA VERSIÓN

En el Editor del Registro se procederá a buscar las siguientes entradas:

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Firewall = <*ruta_calculada*>\svchosd.exe
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Update = <*ruta_calculada*>\select.bat
[HKEY_CURRENT_USER\SOFTWARE]

dma_id = <identificador_único>
[HKEY_CURRENT_USER\SOFTWARE] **dma_public_key = <clave_RSA_pública_cifrada>**

En el caso de que se encuentre alguna de las entradas indicadas es un claro compromiso del sistema.

Se deberán buscar también en las siguientes carpetas:

C:\ProgramData
C:\Documents and Settings\All Users

En las carpetas que existan en el sistema comprometido se procederá a buscar los siguientes archivos:

C:\ProgramData\svchosd.exe
C:\ProgramData\select.bat
C:\ProgramData\cryptinfo.txt
C:\Documents and Settings\All Users\svchosd.exe
C:\Documents and Settings\All Users\select.bat
C:\Documents and Settings\All Users\cryptinfo.txt

En el caso de que se encuentre algún archivo de los anteriores es un claro indicador del compromiso del sistema.

Por supuesto, que no se puedan abrir los archivos que habitualmente se usan en el sistema es un claro indicador de que se han cifrado archivos en el sistema.

11.2 MANDIANT

Se ha generado un nuevo archivo indicador de compromiso. El nombre del indicador generado es con GUID "f38ae3fd-3b7c-493e-9d7d-bfa0596d8128".

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.

12. DESINFECCIÓN

Para desinfectar el sistema del código dañino se requiere una cuenta de usuario con privilegios administrativos.

12.1 PRIMERA A CUARTA VERSIÓN

Se ejecutará el Editor de Registro del sistema (Inicio -> Ejecutar -> regedit.exe), y se buscarán las siguientes entradas:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys = <*ruta_calculada*>\ntserver.exe
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cssys =<*ruta_calculada*>\ntserver.exe
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
cryptedinfo = notepad <*ruta_calculada*>\cryptinfo.txt

En el caso de la tercera versión la entrada de persistencia es distinta:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Firewall = <*ruta_calculada*>\svchosd.exe
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Firewall = <*ruta_calculada*>\svchosd.exe
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
Windows Update = notepad <*ruta_calculada*>\cryptinfo.txt

Cualquier entrada de las indicadas anteriormente deberá ser borrada.

Posteriormente se accederá a las rutas que indican en las entradas del registro y se borrarán los archivos a los que hacen referencia las entradas aunque no descifrará los archivos secuestrados.

12.1.1 DESCIFRADO DE LOS ARCHIVOS

Para la tercera y cuarta versión no existe programa para descifrarlo pero la clave pública RSA embebida es la misma para toda la campaña de forma que, si se puede acceder a un descifrador (clave privada) realizado por los autores del código dañino obtenido a través del pago, se podría usar para descifrar todas las víctimas de la misma campaña. No se ha podido encontrar ningún descifrador de alguien que haya pagado y que pueda ser usada en estas versiones para descifrar los archivos.

Para el resto de versiones, se pueden utilizar las herramientas "*Decrypter For Dmalocker*" de Emsisoft o la herramienta "*DMA Unlocker*" de Hasherezade aunque su autora no garantiza el 100% de la recuperación.

12.1.1.1 EMSISOFT DECRYPTER FOR DMALOCKER

La herramienta "*Decrypter For Dmalocker*"[6] de Emsisoft sólo puede recuperar archivos cifrados por la primera versión del código dañino.

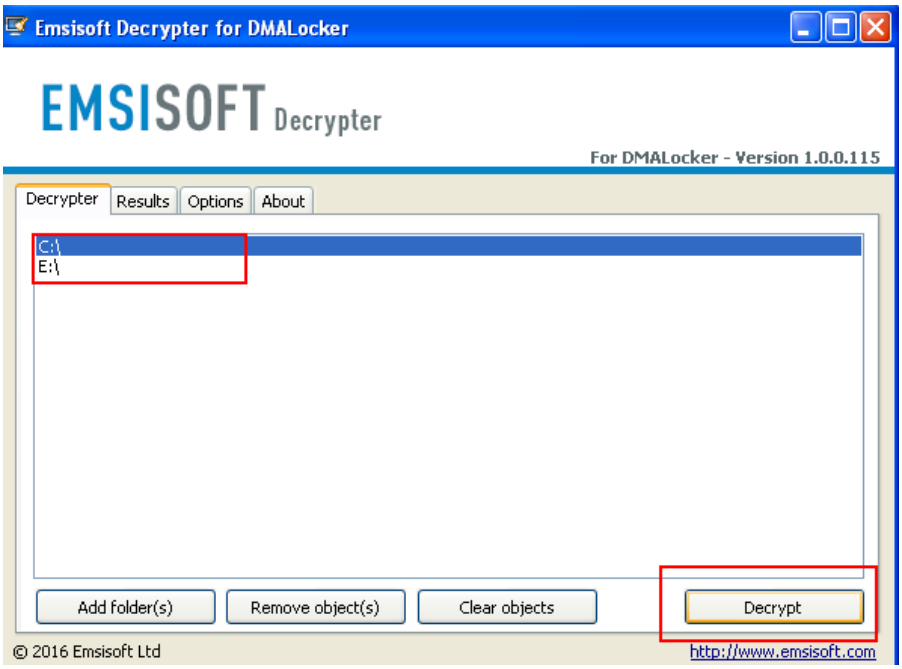


Ilustración 10.Emisoft Descifrador para DMA Locker

Su funcionamiento es simple. Posee una GUI y los pasos a seguir son los siguientes: se deben seleccionar las unidades que se quieren analizar en busca de archivos cifrados, o añadir carpetas específicas mediante el botón "*Add Folder*", y se pulsará el botón "*Decrypt*". En el caso de que se quiera deseleccionar determinados elementos se pulsará el botón "*Remove Object*" y si se quiere limpiar todo lo seleccionado se pulsará el botón "*Clear objects*".

La herramienta buscará archivos cifrados y en el caso de que encuentre y compruebe que son del código dañino *DMA Locker,* en su primera versión, intentará descifrarlo.

Se recomienda tener una copia de seguridad de los archivos cifrados previamente.

12.1.1.2 DMA UNLOCKER DE HASHEREZADE

DMA UnLocker[7] puede ser usada para descifrar los archivos cifrados con las versiones del código dañino antes de la DMALocker 3.0. Hay que tener en cuenta que la herramienta no funciona en Windows XP y que se ejecuta por línea de comandos.

[6] https://github.com/hasherezade/dma_unlocker

[7] https://drive.google.com/file/d/0Bzb5kQFOXkiSb3FWUDAzYUFaWUE

Los pasos a seguir para usar la herramienta son los siguientes:

- Descomprimir el *zip* descargado en el directorio del que se quieran descifrar los archivos.
- Ejecutar el programa indicando como parámetro la carpeta o unidad en la que se quiera descifrar archivos y aceptar su licencia de uso.
- Dependiendo de la muestra del código dañino, el tamaño de los archivos cifrados y la potencia del sistema el proceso puede consumir una cantidad muy variable de tiempo. Del mismo modo hay que indicar que no hay garantías de que se pueda encontrar la clave.
- El programa hace una copia de seguridad de los archivos que descifra y le añade la extensión "_decrypted" no borrando el archivo original cifrado.
- Una vez que finaliza el proceso, comprobar si los archivos descifrados son correctos y, en caso de que así sea, se puede borrar las copias de seguridad que el programa realizó.

12.2 QUINTA VERSIÓN

Se ejecutará el Editor de Registro del sistema (Inicio -> Ejecutar -> regedit.exe), y se buscarán las siguientes entradas:

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **Windows Firewall = *<ruta_calculada>*\svchosd.exe**
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] **Windows Update = *<ruta_calculada>*\select.bat**

Se deberán borrar las entradas anteriormente indicadas y, a continuación, se buscarán y borrarán las siguientes entradas:

[HKEY_CURRENT_USER\SOFTWARE] **dma_id = <identificador_único>**
[HKEY_CURRENT_USER\SOFTWARE] **dma_public_key = <clave_RSA_pública_cifrada>**

Por último se procederá a borrar todos los archivos con los siguientes nombres:

C:\ProgramData\svchosd.exe

https://drive.google.com/file/d/0Bzb5kQFOXkiSNmZtNXo5c0ZIaG8/

C:\ProgramData\select.bat
C:\ProgramData\cryptinfo.txt
C:\Documents and Settings\All Users\svchosd.exe
C:\Documents and Settings\All Users\select.bat
C:\Documents and Settings\All Users\cryptinfo.txt

En caso de no estar activas las Shadow Copies, 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.

13. ARCHIVOS RELACIONADOS

<C:\ProgramData> / <C:\Documents and Settings\All Users\<Dokumenty>\>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
<ntserver.exe> / <svchosd.exe>	<varia>	<varia>	<varia>
<select.bat>	<varia>	<varia>	<varia>
<start.txt>	<varia>	<varia>	<varia>
<decrypting.txt>	<varia>	<varia>	<varia>
<cryptinfo.txt>	<varia>	<varia>	<varia>

14. INFORMACIÓN DEL ATACANTE

La muestra del código dañino analizada tiene una dirección IP embebida en su código.

5.8.63.54

14.1 5.8.63.54

Haciendo un WHOIS de la dirección IP se obtiene la siguiente información:

IP Location	Russian Federation Moscow Onlineua
ASN	AS29182 ISPSYSTEM-AS ISPsystem Autonomous System, LU (registered Jun 23, 2003)
Whois Server	whois.ripe.net
IP Address	5.8.63.54
Reverse IP	1 website uses this address.

```
% Abuse contact for '5.8.63.0 - 5.8.63.255' is ' support.service@online.ua '

inetnum:        5.8.63.0 - 5.8.63.255
netname:        onlineua
descr:          Onlineua Network
country:        UA
admin-c:        DH6046-RIPE
org:            ORG-OA747-RIPE
remarks:        abuse contact  support.service@online.ua
mnt-lower:      DemianHrescak
tech-c:         DH6046-RIPE
status:         SUB-ALLOCATED PA
mnt-by:         MNT-PINSUPPORT
mnt-domains:    DemianHrescak
mnt-routes:     MNT-SYSTEM-SERVICE
mnt-routes:     ISPSYSTEM-MNT
created:        2015-09-25T14:53:48Z
last-modified:  2015-11-18T09:55:22Z
source:         RIPE

organisation:   ORG-OA747-RIPE
org-name:       onlineua
org-type:       OTHER
address:        Tavcarjeva 51 4208 ?en?ur
e-mail:          support.service@online.ua
```

Ilustración 11. Información WHOIS de la dirección IP 5.8.63.54

14.1.1 GEOLOCALIZACIÓN

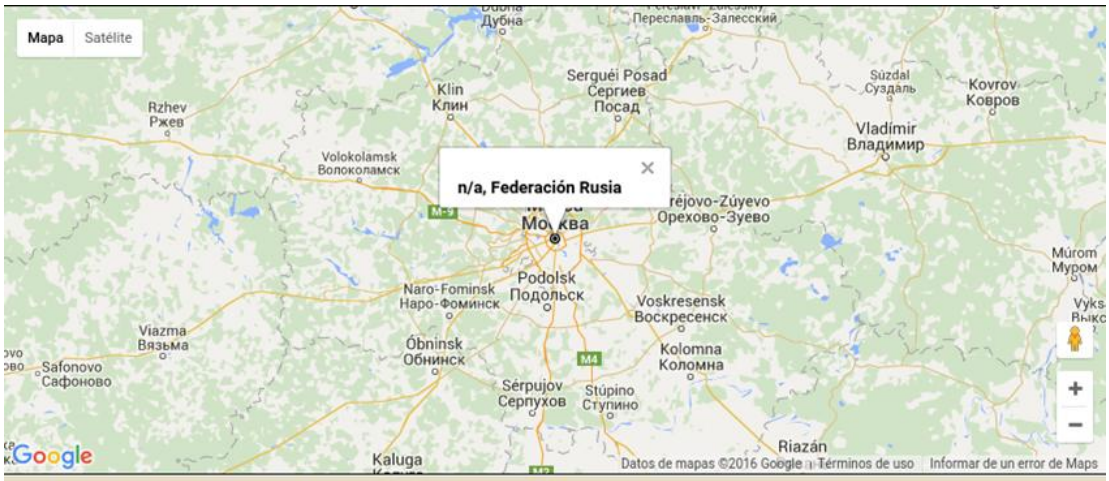


Ilustración 12. Geolocalización de la dirección IP 5.8.63.54

15. REGLAS DE DETECCIÓN

15.1 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="f38ae3fd-3b7c-493e-9d7d-
bfa0596d8128"                                    last-modified="2016-06-16T08:27:20"
xmlns="http://schemas.mandiant.com/2010/ioc">
  <short_description>Ransom.DMALocker</short_description>
  <description>Regla de deteccion del ransomware DMALOCKER.</description>
  <authored_by>CCN-CERT</authored_by>
  <authored_date>2016-05-22T17:51:09</authored_date>
  <links />
  <definition>
    <Indicator operator="OR" id="38ffa58d-5962-42a9-8e18-b9470507dd50">
      <IndicatorItem id="617e3edd-1e2e-4962-94f5-0b71e53d3ade" condition="contains">
        <Context document="RegistryItem" search="RegistryItem/Path" type="mir" />
        <Content
type="string">SOFTWARE\Microsoft\Windows\CurrentVersion\Run</Content>
      </IndicatorItem>
      <Indicator operator="AND" id="b84dd217-cda6-441c-8580-b516a69ece6b">
        <IndicatorItem id="eff1d2d0-b704-484c-aa1e-5c74ff2f0f84" condition="is">
          <Context document="RegistryItem" search="RegistryItem/ValueName" type="mir"
/>
          <Content type="string">cssys</Content>
        </IndicatorItem>
        <Indicator operator="OR" id="3da77847-208b-4681-b4ac-288bc41bc7bd">
          <IndicatorItem id="04a8ef20-d800-47bd-a162-2c50f62dcd64" condition="is">
            <Context      document="RegistryItem"      search="RegistryItem/ValueName"
type="mir" />
            <Content type="string">Windows Firewall</Content>
          </IndicatorItem>
          <Indicator operator="AND" id="bd1f768f-6114-467f-90dd-85f28912d93b">
            <IndicatorItem id="289ce902-3b0d-4633-93a7-e7a0e37341fc" condition="is">
              <Context document="RegistryItem" search="RegistryItem/Value" type="mir"
/>
              <Content type="string">svchosd.exe</Content>
            </IndicatorItem>
          </Indicator>
          <Indicator operator="OR" id="11a3747c-249e-4a30-a685-6f6f2d3de144">
            <IndicatorItem id="2f8a8eca-dba5-4bd8-b9a1-f60766bfd77e" condition="is">
              <Context     document="RegistryItem"     search="RegistryItem/ValueName"
type="mir" />
              <Content type="string">Windows Update</Content>
```

```
            </IndicatorItem>
            <Indicator operator="AND" id="8ef372b6-732a-4b25-b5e2-7166ab491b84">
              <IndicatorItem                id="af571edf-299a-445b-9a63-410ca0e2646b"
condition="is">
                <Context      document="RegistryItem"      search="RegistryItem/Value"
type="mir" />
                <Content type="string">select.bat</Content>
              </IndicatorItem>
            </Indicator>
          </Indicator>
        </Indicator>
        <Indicator operator="AND" id="236584dd-0fff-4bfe-8233-7df481c7d0bb">
          <IndicatorItem id="9e4526b1-f8b0-445c-94c4-7dc55f1dc2f6" condition="is">
            <Context  document="RegistryItem"  search="RegistryItem/Value"  type="mir"
/>
            <Content type="string">ntserver.exe</Content>
          </IndicatorItem>
        </Indicator>
      </Indicator>
    </Indicator>
  </definition>
</ioc>
```

15.2 YARA

```
import "hash"
import "pe"

rule Ransom_DMALocker
{
    meta:
        description = "Regla para detectar Ransom.DMALocker"
        author = "CCN-CERT"
        version = "1.0"
    strings:
        $a = { 40 69 6E 74 65 72 69 61 2E 70 6C 00 00 44 4D 41 4C 4F 43 4B 20 }
        $b = { 53 41 4D 20 4B 4C 55 43 5A 2C 20 41 20 4F 44 53 5A 59 46 52 4F 57 59 57
41 4E 49 45 20 44 41 4E 59 43 48 20 5A 4F 53 54 41 4E 49 45 20 4B 4F 4E 54 59 4E 55 4F
57 41 4E 45 2E 20 00 00 00 00 43 3A 5C 50 72 6F 67 72 61 6D 44 61 74 61 5C 6E 74 73 65
72 76 65 72 2E 65 78 65 00 }
        $c = { 52 45 43 4F 56 45 52 49 4E 47 20 57 49 4C 4C 20 42 45 20 43 4F 4E 54 49
4E 55 45 44 21 00 00 00 43 3A 5C 50 72 6F 67 72 61 6D 44 61 74 61 5C 6E 74 73 65 72 76
65 72 2E 65 78 65 }
        $d = { 44 4D 41 4C 4F 43 4B }
        $e = { 44 4D 41 20 4C 6F 63 6B 65 72 }
        $f = { 73 76 63 68 6F 73 64 2E 65 78 65 }
```

```
        $g = { 78 61 72 6B 74 66 61 75 2E 65 78 65 }
        $h = { 5B 2B 5D 20 53 74 61 72 74 69 6E 67 20 64 65 63 72 79 70 74 69 6E 67 20
66 69 6C 65 3A }
        $i = { 4F 44 53 5A 59 46 52 55 4A }
        $j = { 41 42 43 58 59 5A 31 31 }
        $k = { 21 44 4D 41 4C 4F 43 4B }
    condition:
        $a or $b or $c or $d or $e or $f or $g or $h or $i or $j or $k
}


rule Ransom_DMALocker_Resource
{
    meta:
        description = "Regla para detectar Ransom.DMALocker mediante su recurso"
        author = "CCN-CERT"
        version = "1.0"
    condition:
        pe.number_of_resources >= 1 and pe.resources[0].type == pe.RESOURCE_TYPE_BITMAP
and             hash.md5(pe.resources[0].offset,pe.resources[0].length)             ==
"1ded70024cfb3689e1c3165d6d968a83"
}
```