Informe Código Dañino
CCN-CERT ID-26/18

"CoinMiner.Gen"

Octubre 2018

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
2018.10.19 12:48:36 +02'00'

ÍNDICE

1. SOBRE CCN-CERT

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

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

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

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

2. RESUMEN EJECUTIVO

El presente documento recoge el análisis de la familia de troyanos identificada como "***CoinMiner.Gen***", se trata de una de las muestras vinculadas a la actividad de mineria de las monedas criptográficas, la cual se ha encontrado de manera muy activa durante este año 2018. Tiene capacidades para la descarga de otros códigos dañinos. Las variantes distribuidas se encuentran inicialmente sin la utilización de un empaquetador, aunque se identificaron capacidades criptográficas para la ofuscación de su configuración. Cuenta con una carga dinámica de las funciones necesarias en su inicio.

3. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El código dañino realiza las siguientes acciones:

- Crea una estructura de carpetas en la raíz de C.
- Crea un archivo temporal.
- Se instala en disco.
- Utiliza métodos de detección de sistemas de análisis.
- Escribe un archivo temporal.
- Lanza el ejecutable instalado y se cierra el hilo principal.
- Trata de localizar archivos utilizados por el mismo instalados en disco.
- Comprueba la existencia de un ejecutable en memoria.
- Genera una lista de User-Agents.
- La ejecución llega hasta una rutina de descifrado.
- Se realizan conexiones a internet.
- Puede comprobar el antivirus instalado y enviarlo al C&C.
- Crea un Mutex.
- Intenta lanzar un ejecutable instalado.
- Trata de eliminar un ejecutable de backup.
- Se identifican los parámetros de envío con el C&C.

4. DETALLES GENERALES

La muestra analizada se identifican con las siguiente firma SHA-1:

a910d0716525acd901355a996ee9dd40e7265fed
cdc2e1f29769add5dee5bee739176e39ccce3230

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

5. PROCEDIMIENTO DE INFECCIÓN

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

- Crea una estructura de carpetas en la raíz de C.
- Escribe un archivo temporal.
- Comprueba la existencia de un ejecutable en memoria.
- Se instala en disco.
- Utiliza métodos de detección de sistemas de análisis.
- Lanza el ejecutable instalado y se cierra el hilo principal.
- Trata de localizar archivos utilizados por el mismo instalados en disco.
- Genera una lista de User-Agents.
- La ejecución llega hasta una rutina de descifrado.
- Se realizan conexiones a internet.
- Puede comprobar el antivirus instalado y enviarlo al C&C.
- Crea un Mutex.
- Intenta lanzar un ejecutable instalado.
- Trata de eliminar un ejecutable de backup.
- Se identifican los parámetros de envío con el C&C.

6. CARACTERÍSTICAS TÉCNICAS

Una de las primeras acciones realizadas por el ejecutable durante su ejecución es la creación del directorio de instalación. Para ello utiliza las siguientes funciones:

/CALL to CreateDirectoryA from.00976A27 **|Path = "C:\$WINDOWS.~BT"** **\pSecurity = NULL**
/CALL to CreateDirectoryW from KERNELBA.75E97B78 **|Path = "C:\$WINDOWS.~BT"** **\pSecurity = NULL**

A continuación realiza la creación de una carpeta utilizada para actualizar la muestra en caso de disponer de nuevas versiones, en este caso también utiliza ambas funciones para la creación.

/CALL to CreateDirectoryA from 18c3be50.00976A6B **|Path = "C:\$WINDOWS.~BT\updates"** **\pSecurity = NULL**
/CALL to CreateDirectoryW from KERNELBA.75E97B78 **|Path = "C:\$WINDOWS.~BT\updates"** **\pSecurity = NULL**

Trata de encontrar algún archivo ejecutable dentro de la carpeta de actualizaciones "C:\$WINDOWS.~BT\updates", para llevarla a cabo.

```
/FindFirstFileA
|FileName = "C:\$WINDOWS.~BT\updates\*.exe"
\pFindFileData = 0026D438
```

Aplica el atributo de oculto a la carpeta para pasar desapercibido a los ojos del usuario infectado.

```
/FileAttributes = HIDDEN
|FileName = "C:\$WINDOWS.~BT"
\SetFileAttributesA
```

Se identifica el intento de apertura de un archivo temporal en el interior de la carpeta "$WINDOWS.~BT".

```
/CALL to CreateFileW from bicho.0120C209
|FileName = "C:\$WINDOWS.~BT\tmp-0.bin"
|Access = GENERIC_READ
|ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
|pSecurity = 0023DE9C
|Mode = OPEN_EXISTING
|Attributes = NORMAL
\hTemplateFile = NULL
```

Al no existir, realiza nuevamente la llamada a la función *CreateFileW,* aunque en este caso agregando al parámetro mode, el flag "CREATE_ALWAYS" y consecuentemente, forzando su escritura.

```
/CALL to CreateFileW from bicho.0120C209
|FileName = "C:\$WINDOWS.~BT\tmp-0.bin"
|Access = GENERIC_WRITE
|ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
|pSecurity = 0023DE1C
|Mode = CREATE_ALWAYS
|Attributes = NORMAL
\hTemplateFile = NULL
```

Seguidamente la función *WriteFile* rellena el archivo con una ristra de al menos 1000 bytes con el carácter "@", comprobando así los permisos de escritura en disco.

Comprueba la existencia del código dañino en la ruta de ejecución con el objetivo de llevar a cabo su lectura más adelante.

```
/CALL to CreateFileW from bicho.0120C209
|FileName = "C:\Users\matu\Desktop\bicho.exe"
|Access = GENERIC_READ
|ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
|pSecurity = 0023D664
|Mode = OPEN_EXISTING
|Attributes = NORMAL
\hTemplateFile = NULL
```

Tras la lectura con *ReadFile*, se escribe una copia del proceso en la carpeta creada anteriormente, con la que se lleva a cabo la instalación del mismo.

```
/CALL to CreateFileW from bicho.0120C209
|FileName = "C:\$WINDOWS.~BT\WMIC.exe"
|Access = GENERIC_WRITE
|ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
|pSecurity = 0023D664
|Mode = CREATE_ALWAYS
|Attributes = NORMAL
\hTemplateFile = NULL
```

Se ha detectado una modificación en el momento de escritura en disco de este ejecutable, incluyendo de esta manera un patrón de polimorfismo en la copia del código dañino. Se encuentra dentro de la sección ".data", justo debajo de su configuración. Permitiendo así que el ejecutable instalado contenga un hash diferente al inicial, no obstante las funcionalidades siguen intactas.

La siguiente función realiza la extracción del nombre de los procesos activos en el sistema.

```
/CALL to CreateToolhelp32Snapshot from bicho.011F7489
|Flags = TH32CS_SNAPPROCESS
\ProcessID = 0
```

Llegado a este punto el código dañino entra en una rutina de revisión de los procesos activos, en busca del nombre "FirewallGUI.exe". En caso de coincidir con alguno de la lista, el hilo de ejecución se desvía hasta una función que finaliza su ejecución para siempre con un *ExitProcess*. Las funciones encargadas de recorrer los nombres de los procesos son *Process32First*, utilizada para posicionarse sobre el primer proceso de la lista y *Process32Next* utilizado para avanzar en ella.

```
0035747A  . PUSH EBX
0035747B  . PUSH ESI
0035747C  . PUSH EDI
0035747D  . PUSH 0                                  ProcessID = 0
0035747F  . PUSH 2                                  Flags = TH32CS_SNAPPROCESS
00357481  . MOV DWORD PTR SS:[ESP+18],128
00357489  . CALL DWORD PTR DS:[<&KERNEL32.CreateToo CreateToolhelp32Snapshot
0035748F  . PUSH 10
00357491  . MOV EBX,EAX
00357493  . CALL WMIC.0035845B
00357498  . PUSH WMIC.0037809C                      Arg3 = 0037809C ASCII "FirewallGUI.exe"
0035749D  . MOV EDI,EAX
0035749F  . PUSH 10                                 Arg2 = 00000010
003574A1  . PUSH EDI                                Arg1
003574A2  . CALL WMIC.003590BE                      WMIC.003590BE
003574A7  . ADD ESP,10
003574AA  . LEA EAX,DWORD PTR SS:[ESP+10]
003574AE  . PUSH EAX                                pProcessentry
003574AF  . PUSH EBX                                hSnapshot
003574B0  . CALL DWORD PTR DS:[<&KERNEL32.Process32 Process32First
003574B6  . CMP EAX,1
003574B9  .v JNZ WMIC.003575A1
003574BF  . LEA EAX,DWORD PTR SS:[ESP+10]
003574C3  . PUSH EAX                                pProcessentry
003574C4  . PUSH EBX                                hSnapshot
003574C5  . CALL DWORD PTR DS:[<&KERNEL32.Process32 Process32Next
003574CB  . CMP EAX,1
003574CE  .v JNZ WMIC.003575A1
```

Ilustración 1.- Búsqueda del proceso FirewallGUI.exe

La copia es lanzada en ejecución mediante la función *CreateProcessA*.

```
/CALL to CreateProcessA from bicho.011F5C13
|ModuleFileName = NULL
|CommandLine = "C:\$WINDOWS.~BT\WMIC.exe"
|pProcessSecurity = NULL
|pThreadSecurity = NULL
|InheritHandles = FALSE
|CreationFlags = 0
|pEnvironment = NULL
|CurrentDir = NULL
|pStartupInfo = 0023DA28
\pProcessInfo = 0023DA18
```

El hilo de ejecución principal finaliza con la siguiente sintaxis:

```
/ExitCode = 0
\ExitProcess
```

El hilo lanzado realiza las mismas acciones hasta este punto que el proceso inicial.

Se han encontrado métodos de detección de análisis basados en tiempos mediante la API de *GetTickCount*, dejando ver el mensaje traducido al español (Mi madre está enfadada) durante la ejecución.

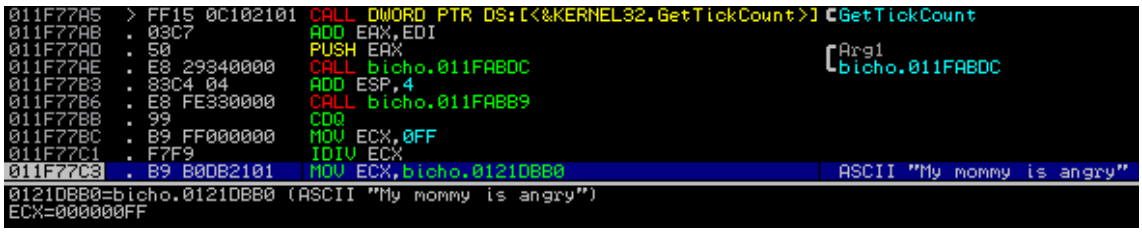

Ilustración 2.- Mensaje de detección basado en tiempos

Este proceso llega hasta la única rutina de descifrado observada durante la ejecución, la cual deja ver el *C&C* utilizado por la muestra.

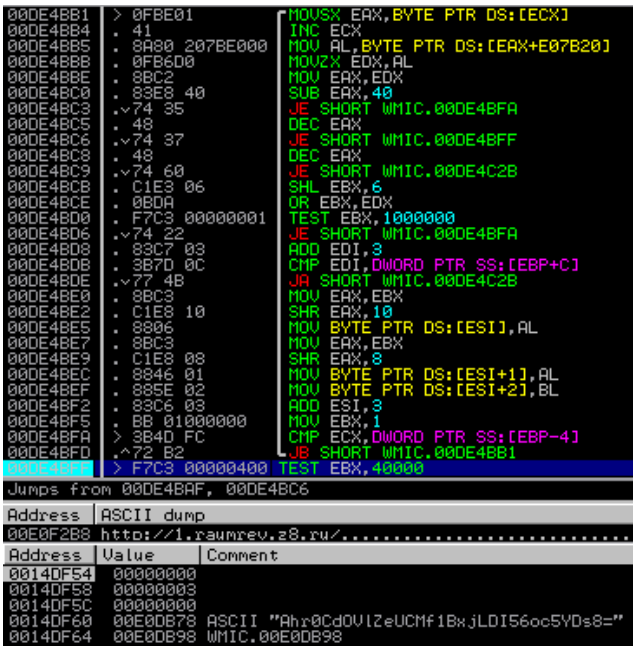

Ilustración 3.- Rutina de descifrado y dominio

Siguiendo el hilo de ejecución se realiza la unión del dominio junto a una cadena, la cual forma un archivo de descarga.

```
http://1.raumrev.z8.ru/tools/RegWriter.exe.raum_encrypted
```

Este archivo no ha sido posible de descargar debido a que el servidor actualmente se encuentra fuera de conexión, no obstante es posible identificar las conexiones DNS intentando resolver la dirección IP.

Se identifica la carga de los siguientes User-Agents:

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
Mozilla/5.0 (compatible; Konqueror/4.3; Linux) KHTML/4.3.5 (like Gecko)
Mozilla/4.8 [en] (Windows NT 5.0; U)
Mozilla/4.0 (compatible; MSIE 6.0; America Online Browser 1.1; rev1.5; Windows NT 5.1;)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.5) Gecko/20060127 Netscape/8.1
```

Es posible ver en este momento la primera petición HTTP, siendo esta a la API de la red social *Vkontakte*, una de las comunidades de internet más utilizadas en Rusia:

```
GET /method/wall.get.xml HTTP/1.0
Host: api.vk.com
Cache-Control: max-age=0, no-store
User-agent: Mozilla/4.8 [en] (Windows NT 5.0; U)
```

Esta petición se ha realizado con asiduidad desde un bucle, en el que además seleciona un User-Agent diferente de los mostrados en la lista anterior para cada petición.

A pesar de no haberse identificado otras peticiones, se han encontrado en memoria las siguientes cadenas que podrían formar parte de las enviadas al dominio "api.vk.com".

```
/method/groups.getById.xml?fields=description&group_ids=
/method/wall.get.xml?count=1&owner_id=
/signin.php?id=
```

Este segundo dominio es el utilizado para la descarga del monedero. Se han identificado las siguientes sintaxis utilizadas para la apertura del *Wallet* dentro del código dañino, en la que se puede ver la contraseña utilizada.

```
"<link>"
"reserved.raum_update</link><encryption_key>666_SamaelLovesMe_666</encryption_key>"
```

Esto puede ser de utilidad para la descarga de las actualizaciones del *Wallet* a través de un enlace. Además se han identificado multitud de interacciones entre el dominio de esta red social y el troyano, formando parte incluso del control del mismo.

Se ha identificado la posibilidad de obtener el nombre del antivirus instalado en el sistema, para ello son utilizadas las siguientes cadenas:

```
"root\SecurityCenter2"
"SELECT * FROM
AntivirusProduct"
```

La información obtenida es cotejada y preparada para el envío al panel de control. Se identificaron las siguientes cadenas que hacen alusión a este módulo con el objetivo de generar una nueva petición:

```
&aver=
Avast
Kaspersky
ESET
0
&aver_name=
```

Además se han identificado los siguientes nombres en este código dañino, los cuales podrían tratarse de archivos ha escribrir en la carpeta de instalación. Estos son utilizados para recolectar la información acerca del grupo identificativo del troyano, información sobre del proceso de minado o de la url utilizada para la copia de seguridad del monedero.

```
sign
group_ids
update_info
mining_info
backup_url
vk_timeout
checkin_simeout
```

Tras la consulta de la variable de entorno *SystemDrive*, este minero extrae la letra de la unidad donde se encuentra instalado el sistema y renombra la letra de la unidad utilizada para la siguiente ruta:

```
X:\551b45c5a2e3acc57690b0469d26f1a3\intel.exe
C:\551b45c5a2e3acc57690b0469d26f1a3\intel.exe
```

Tras no encontrar el archivo en disco, se ha identificado más adelante la siguiente acción que trataría de eliminarlo del sistema.

```
/DeleteFileA
\FileName = "C:\551b45c5a2e3acc57690b0469d26f1a3\intel.exe"
```

El código dañino crea un *Mutex* con el nombre traducido al español "Raum conmigo", refiriéndose en demonología a *Raum*, (el Gran Duque del Infierno comandante de treinta legiones de demonios). Se han encontrado diversas citas referentes a la demonología durante el análisis de su código:

```
/ CALL to CreateMutexA from WMIC.00086871
|pSecurity = NULL
|InitialOwner = TRUE
\MutexName = "Raum-with-Me"
```

Trata de identificar la posible existencia del binario de *backup*, el cual no ha sido creado durante el proceso de análisis.

```
/CreateFileW
|FileName = "C:\$WINDOWS.~BT\backup.exe"
|Access = GENERIC_READ
|ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
|pSecurity = 0026D614
|Mode = OPEN_EXISTING
|Attributes = NORMAL
\hTemplateFile = NULL
```

A pesar de no existir el archivo trata de lanzarlo en ejecución, lo que demuestra un descuido durante su desarrollo ya que la siguiente función se ejecuta provocando un error.

```
/CALL to CreateProcessA from WMIC.00085DF8
|ModuleFileName = NULL
|CommandLine = "C:\$WINDOWS.~BT\backup.exe -autorun C:\$WINDOWS.~BT\WMIC.exe"
|pProcessSecurity = NULL
|pThreadSecurity = NULL
|InheritHandles = FALSE
|CreationFlags = 0
|pEnvironment = NULL
|CurrentDir = NULL
|pStartupInfo = 0026D7A8
\pProcessInfo = 0026D798
```

Más adelante intenta eliminar el binario "backup.exe" inexistente.

```
/DeleteFileA
\FileName = "C:\$WINDOWS.~BT\backup.exe"
```

En otra de las muestras analizadas, con hash "cdc2e1f29769add5dee5bee739176e39ccce3230" además se ha encontrado la identificación del sistema operativo a través de las siguientes cadenas.

```
64bit
32bit
other
```

Esta muestra también identifica si los siguientes procesos se encuentran abiertos en el sistema.

```
Taskmgr.exe
procexp.exe
```

El *Mutex* utilizado para esta muestra contiene el nombre "SamaelLovesMe".

Se identifica en este ejecutable la posibilidad de extraer el nombre de la tarjeta gráfica y la versión de los drivers, con la que agilizar los procesos de minería debido al potencial de las GPU. Para ello crea un objeto administrativo de gestión a través de "ROOT\CIMV2", consultando a la clase de nombre "Win32_VideoController" desde la que es posible extraer esta información. Además comprueba los nombres de los drivers con los siguientes nombres de tarjetas gráficas.

```
nvidia
radeon
```

Dicha información sería almacenada en el archivo "video_info".

Los archivos identificados que podría crear cuentan con los siguientes nombres descriptivos.

\unpacked
\options
\mining.archive
last_miner_link
pre_exe
main_exe
parameters
miner_exe_name
SamaelLovesMe
all_data
group_ids
update_info
mining_info
backup_url

Además el dominio utilizado por esta y una gran cantidad de muestras recolectadas en el laboratorio, cuentan con el siguiente dominio como configuración del panel de control.

1.lalkaboy.z8.ru

El mensaje elegido por parte del desarrollador para aquel que analice el troyano, como el mostrado en el anterior caso, deja la siguiente frase:

Don't beat me, I am a child ;(

7. OFUSCACIÓN

El código dañino se ha visto muy activo desde inicios del año 2018 y se detectaron diferentes versiones hasta la fecha. El ejecutable analizado se encuentra desarrollado en *Microsoft Visual C++* v 9.0.

Durante el análisis realizado a diversas muestras de este troyano, tan solo se ha identificado una sola rutina de descifrado, la cual es utilizada durante el proceso de conversión del dominio perteneciente al panel de control. A continuación se muestra un fragmento de la rutina utilizada.

```
011F4B80 /$ 55              PUSH EBP
011F4B81 |. 8BEC            MOV EBP,ESP
011F4B83 |. 83EC 08         SUB ESP,8
011F4B86 |. 53              PUSH EBX
011F4B87 |. 56              PUSH ESI
011F4B88 |. 8B75 08         MOV ESI,DWORD PTR SS:[EBP+8]
011F4B8B |. 57              PUSH EDI
011F4B8C |. FF75 0C         PUSH DWORD PTR SS:[EBP+C]
011F4B8F |. 33FF            XOR EDI,EDI
011F4B91 |. 8D0411          LEA EAX,DWORD PTR DS:[ECX+EDX]
011F4B94 |. 57              PUSH EDI
011F4B95 |. 56              PUSH ESI
011F4B96 |. 894D F8         MOV DWORD PTR SS:[EBP-8],ECX
011F4B99 |. 8945 FC         MOV DWORD PTR SS:[EBP-4],EAX
011F4B9C |. BB 01000000     MOV EBX,1
011F4BA1 |. E8 4A920000     CALL bicho.011FDDF0
011F4BA6 |. 8B4D F8         MOV ECX,DWORD PTR SS:[EBP-8]
011F4BA9 |. 83C4 0C         ADD ESP,0C
011F4BAC |. 3B4D FC         CMP ECX,DWORD PTR SS:[EBP-4]
011F4BAF |.v73 4E           JNB SHORT bicho.011F4BFF
011F4BB1 |> 0FBE01          MOVSX EAX,BYTE PTR DS:[ECX]
011F4BB4 |. 41              INC ECX
011F4BB5 |. 8A80 207B2101   MOV AL,BYTE PTR DS:[EAX+1217B20]
011F4BBB |. 0FB6D0          MOVZX EDX,AL
011F4BBE |. 8BC2            MOV EAX,EDX
011F4BC0 |. 83E8 40         SUB EAX,40
011F4BC3 |.v74 35           JE SHORT bicho.011F4BFA
011F4BC5 |. 48              DEC EAX
011F4BC6 |.v74 37           JE SHORT bicho.011F4BFF
011F4BC8 |. 48              DEC EAX
011F4BC9 |.v74 60           JE SHORT bicho.011F4C2B
011F4BCB |. C1E3 06         SHL EBX,6
011F4BCE |. 0BDA            OR EBX,EDX
011F4BD0 |. F7C3 00000001   TEST EBX,1000000
011F4BD6 |.v74 22           JE SHORT bicho.011F4BFA
011F4BD8 |. 83C7 03         ADD EDI,3
011F4BDB |. 3B7D 0C         CMP EDI,DWORD PTR SS:[EBP+C]
011F4BDE |.v77 4B           JA SHORT bicho.011F4C2B
011F4BE0 |. 8BC3            MOV EAX,EBX
011F4BE2 |. C1E8 10         SHR EAX,10
011F4BE5 |. 8806            MOV BYTE PTR DS:[ESI],AL
011F4BE7 |. 8BC3            MOV EAX,EBX
011F4BE9 |. C1E8 08         SHR EAX,8
011F4BEC |. 8846 01         MOV BYTE PTR DS:[ESI+1],AL
011F4BEF |. 885E 02         MOV BYTE PTR DS:[ESI+2],BL
011F4BF2 |. 83C6 03         ADD ESI,3
011F4BF5 |. BB 01000000     MOV EBX,1
011F4BFA |> 3B4D FC         CMP ECX,DWORD PTR SS:[EBP-4]
011F4BFD |.^72 B2           JB SHORT bicho.011F4BB1
011F4BFF |> F7C3 00000400   TEST EBX,40000
```

Ilustración 4.- Fragmento de rutina de descifrado

8. PERSISTENCIA EN EL SISTEMA

A pesar de que ambas muestras han escrito en disco una copia de si mismas, no han llegado a generar persistencia.

9. CONEXIONES DE RED

La siguiente conexión ha sido la única identificada con el panel de control, que al encontrarse caído no realiza las conexiones HTTP pertinentes:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.15	10.0.2.3	DNS	75	Standard query 0x0607 A 1.raumrev.z8.ru
2	0.096280	10.0.2.3	10.0.2.15	DNS	135	Standard query response 0x0607 No such name A 1.raumrev.z8.ru SOA ns1.z8.ru

Ilustración 5.- Consulta DNS

De igual manera ocurre con el dominio "1.lalkaboy.z8.ru".

El bucle observado y encargado de realizar las peticiones al dominio "vk.com" deja la siguiente imagen.

Source	Destination	Protocol	Length	Info
10.0.2.15	87.240.129.75	HTTP	246	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found
10.0.2.15	87.240.129.75	HTTP	195	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found
10.0.2.15	87.240.129.75	HTTP	230	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found
10.0.2.15	87.240.129.75	HTTP	246	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found
10.0.2.15	87.240.129.75	HTTP	230	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found
10.0.2.15	87.240.129.75	HTTP	222	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found
10.0.2.15	87.240.129.75	HTTP	222	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found
10.0.2.15	87.240.129.75	HTTP	244	GET /method/wall.get.xml HTTP/1.0
87.240.129.75	10.0.2.15	HTTP	378	HTTP/1.0 302 Found

Ilustración 6.- Bucle de peticiones DNS

9.1 USER AGENT

El código dañino utiliza los siguientes campos de User-Agent específicos:

Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
Mozilla/5.0 (compatible; Konqueror/4.3; Linux) KHTML/4.3.5 (like Gecko)
Mozilla/4.8 [en] (Windows NT 5.0; U)
Mozilla/4.0 (compatible; MSIE 6.0; America Online Browser 1.1; rev1.5; Windows NT 5.1;)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.5) Gecko/20060127 Netscape/8.1

10. ARCHIVOS RELACIONADOS

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

Nombre	Fecha Creación	Tamaño bytes	Hash SHA-1
%USERPROFILE%\			
WMIC.exe	30/8/2014	214 KB	541E8031EF6BB61F6AD62B851D6AEB5930543AFB
tmp-0.bin.0	X	9 KB	14EB2BF3E4496B220ADF62026EA914D2BEBD055F
tmp-0.bin.1	X	17 KB	CD4CA6728448470EB54B8A1DF504F4C7680B31E1
1	30/8/2014	214 KB	A910D0716525ACD901355A996EE9DD40E7265FED
2	30/8/2014	209 KB	CDC2E1F29769ADD5DEE5BEE739176E39CCCE3230

11. DETECCIÓN

Para detectar si un equipo se encuentra, o ha estado infectado, para cualquiera de sus usuarios, se recomiendan utilizar alguna de las herramientas de Mandiant como el "Mandiant IOC Finder" o el colector generado por RedLine@ con los indicadores de compromiso generados para su detección.

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

11.1 MANDIANT

Se ha generado un nuevo indicador de compromiso adjunto al informe que se utilizará con alguna de las herramientas de las que dispone *Mandiant,* como "*Mandiant_ioc_finder*" Para la confección de un recolector de evidencias puede utilizarse "*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 una desinfección automática del equipo, se aconseja la utilización de herramientas antivirus actualizadas.

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

13. INFORMACIÓN DEL ATACANTE

Actualmente tan solo una de las direcciones IP encontradas dentro del archivo de configuración se encuentra activa, con lo que la información mostrada en este informe es sobre esta.

13.1 Z8.RU

13.1.1 WHOIS

Los dominios "1.raumrev.z8.ru" y "1.lalkaboy.z8.ru" actualmente no resuelven a ninguna dirección IP, no obstante estos se encontraban registrados por la misma empresa de hosting *PeterHost*. Se encuentra localizada en Moscú, Rusia. La siguiente información pertenece a la empresa registradora:

```
domain:      Z8.RU
nserver:     ns1.z8.ru. 80.93.56.2
nserver:     ns2.z8.ru. 80.93.50.53
state:       REGISTERED, DELEGATED, VERIFIED
org:         CONCORDE
registrar:   R01-RU
```

admin-contact: https://partner.r01.ru/contact_admin.khtml
created: 2003-02-16T21:00:00Z
paid-till: 2019-02-16T21:00:00Z
free-date: 2019-03-20
source: TCI

14. REFERENCIAS

Se ha encontrado la siguiente referencia tras buscar por el nombre del código dañino en internet:

- http://www.soosanint.com/wordpress/wp-content/uploads/2018/01/eWalker-MonthlyReport_1_en.pdf https://www.netskope.com/blog/analysis-godzilla-botnet-loaders-evasive-techniques/

15. REGLAS DE DETECCIÓN

15.1 REGLA SNORT

```
ccn-cert.rules:alert tcp any any -> $HOME_NET any (content: "GET /method/wall.get.xml"; http_method; content: " vk.com"; classtype:trojan-activity;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS TCP Miner"; content:"1.raumrev.z8.ru"; classtype:attempted-recon;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS TCP Miner"; content:"1.lalkaboy.z8.ru"; classtype:attempted-recon;)
```

15.2 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="ff7917cd-7d2f-489e-aa03-c05500a248a7" last-modified="2018-07-18T11:36:25" xmlns="http://schemas.mandiant.com/2010/ioc">

  <short_description>CoinMiner.gen</short_description>

  <authored_by>CCN-CERT</authored_by>

  <authored_date>2017-09-11T11:10:27</authored_date>

  <links />

  <definition>

    <Indicator operator="OR" id="4e7451b1-7e75-4633-ae3c-9d14ed8bfb71">

      <IndicatorItem id="5a052c94-c46e-46cd-b6f9-949d0579da0e" condition="is">

        <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />

        <Content type="string">541E8031EF6BB61F6AD62B851D6AEB5930543AFB</Content>

      </IndicatorItem>

      <IndicatorItem id="90a7e588-dc2c-4b5e-978c-eb04a08b3931" condition="is">

        <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />

        <Content type="string">14EB2BF3E4496B220ADF62026EA914D2BEBD055F</Content>

      </IndicatorItem>

      <IndicatorItem id="f41e2dd1-5c4b-4e23-a842-9263ae08012c" condition="is">

        <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
```

```
        <Content type="string">CD4CA6728448470EB54B8A1DF504F4C7680B31E1</Content>
      </IndicatorItem>
      <IndicatorItem id="a01cecf7-2474-4b81-9429-b4707eba21de" condition="is">
        <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
        <Content type="string">A910D0716525ACD901355A996EE9DD40E7265FED</Content>
      </IndicatorItem>
      <IndicatorItem id="037689e0-e637-4036-8c95-e1a5fb484f14" condition="is">
        <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
        <Content type="string">CDC2E1F29769ADD5DEE5BEE739176E39CCCE3230</Content>
      </IndicatorItem>
      <Indicator operator="AND" id="ad91d0e9-1d55-46a5-a71d-e838a93bb1b4">
        <IndicatorItem                    id="9ff95df6-1600-461f-9c3d-aa9ed76d99a1"
condition="contains">
          <Context document="FileItem" search="FileItem/FileExtension" type="mir" />
          <Content type="string">exe</Content>
        </IndicatorItem>
        <Indicator operator="OR" id="0b6ffb7e-dfb9-4425-8011-befcd54eceb7">
          <IndicatorItem                    id="4ac7a5a8-419c-433f-838f-2133a9a3b133"
condition="contains">
            <Context document="FileItem" search="FileItem/FileName" type="mir" />
            <Content type="string">WMIC.exe</Content>
          </IndicatorItem>
          <IndicatorItem                    id="4cf1b8a9-43d8-4252-8db5-e0230c3c60af"
condition="contains">
            <Context                                              document="ProcessItem"
search="ProcessItem/HandleList/Handle/Type" type="mir" />
            <Content type="string">Mutant</Content>
          </IndicatorItem>
          <IndicatorItem                    id="76291e15-6d25-499f-8b1a-7938459c95b6"
condition="contains">
            <Context                                              document="ProcessItem"
search="ProcessItem/HandleList/Handle/Name" type="mir" />
            <Content type="string">Raum-with-Me</Content>
          </IndicatorItem>
        </Indicator>
        <Indicator operator="OR" id="b70f5dd6-b06e-4778-8cb0-d2bcf9f25e22">
          <IndicatorItem                    id="096b3d30-6209-4969-acb2-3b64ecb3770f"
condition="contains">
            <Context document="FileItem" search="FileItem/FileName" type="mir" />
            <Content type="string">ActivateDesktop.exe</Content>
          </IndicatorItem>
          <IndicatorItem                    id="b54fd55c-da0a-4d7d-a5da-fd5d9ee31c78"
condition="contains">
            <Context                                              document="ProcessItem"
search="ProcessItem/HandleList/Handle/Type" type="mir" />
            <Content type="string">Mutant</Content>
          </IndicatorItem>
          <IndicatorItem                    id="4f18c6f0-3c37-44cb-af66-1f8f17d8463c"
condition="contains">
            <Context                                              document="ProcessItem"
```

```
search="ProcessItem/HandleList/Handle/Name" type="mir" />
            <Content type="string">SamaelLovesMe</Content>
          </IndicatorItem>
        </Indicator>
      </Indicator>
    </Indicator>
  </definition>
</ioc>
```

15.3 YARA

Utilizando sobre la memoria de un equipo las siguientes firmas YARA, es posible comprobar si el sistema se encuentra infectado.

```
rule CoinMiner_gen: CoinMiner_gen Family {
    meta:
        description = " CoinMiner_gen"
        author = "CCN-CERT"
        version = "1.0"
    strings:
          $ = "CreateFile2" wide ascii
          $ = "tmp-0.bin" wide ascii
          $ = " Mozilla" wide ascii
          $ = "GET" wide ascii
          $ = "Host" wide ascii
          $ = "User-agent" wide ascii
          $ = "/method/wall.get.xml" wide ascii
          $ = "api.vk.com" wide ascii
          $ = "X:\551b45c5a2e3acc57690b0469d26f1a3\intel.exe" wide ascii
          $ = "666_RaumWithMe_666" wide ascii
          $ = "avast" wide ascii
          $ = "Kaspersky" wide ascii
          $ = "ESET" wide ascii
          $ = "sign" wide ascii
          $ = "group_ids" wide ascii
          $ = "update_info" wide ascii
          $ = "mining_info" wide ascii
          $ = "backup_url" wide ascii
          $ = "vk_timeout" wide ascii
          $ = "checkin_simeout" wide ascii
          $ = "SystemDrive" wide ascii
    condition:
        all of them
    }
```