Informe Código Dañino
CCN-CERT ID-26/19

Ryuk



Noviembre 2019

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

CCN centro criptológico nacional

2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC, ou=sello electrónico, serialNumber=S2800155J, o=CENTRO CRIPTOLOGICO NACIONAL, c=ES
2019.11.18 09:49:42 +01'00'

© Centro Criptológico Nacional, 2019

Fecha de Edición: noviembre de 2019

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

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

ÍNDICE

1. SOBRE CCN-CERT, CERT GUBERNAMENTAL NACIONAL 4
2. RESUMEN EJECUTIVO 5
3. DETALLES GENERALES 6
4. RANSOMWARE *AS A SERVICE* 7
5. PROCESO DE INFECCIÓN 8
5.1 WAKE-ON-LAN 10
5.2 PING ESCÁNER 11
5.3 RECURSOS DE RED 12
5.4 ITERACIÓN SOBRE FICHEROS Y EXCEPCIONES 12
6. ESQUEMA DE CIFRADO 14
7. NOTA DE RESCATE 18
8. PERSISTENCIA 19
9. DESINFECCIÓN 19
10. REGLAS DE DETECCIÓN 20
10.1 REGLA YARA RYUK 20
11. INDICADORES DE COMPROMISO 20

1. SOBRE CCN-CERT, CERT GUBERNAMENTAL NACIONAL

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 muestra de código dañino identificada por la firma MD5 12147c94f3211733f893d31d587d4ad6, perteneciente a la familia de ransomware Ryuk, que deriva de la familia de ransomware Hermes.

El objetivo del binario es **cifrar los ficheros** de los sistemas afectados, para posteriormente, solicitar el **pago de un rescate** a cambio de la herramienta de descifrado.

Esta familia de ransomware se está viendo involucrada en despliegues masivos en redes internas de empresas, suponiendo una amenaza severa a la continuidad de negocio de la misma. Desde el pasado 2018, la nueva tendencia que se está observando en malware bancario como TrickBot, Dridex o QBot, es aprovechar la naturaleza modular de estos proyectos para mediante el uso de módulos adicionales, comprometer redes internas enteras a través de movimientos laterales. Una vez finalizados los procesos de exfiltración de información sensible o a través de la identificación de objetivos valiosos (redes con un número significativo de equipos que afectar), se procedería, de forma manual, a distribuir a través de una cuenta comprometida con los privilegios necesarios, una muestra de ransomware a cada equipo de la organización, parando por completo la actividad de la misma.

Aunque no se trata de una relación de exclusividad, un despliegue masivo de Ryuk indicaría un compromiso previo de la red por parte de TrickBot, de la misma forma que un despliegue masivo de la familia de ransomware BitPaymer, indicaría un compromiso previo de la red por parte de Dridex. Debido al grave impacto que supondría el ciclo de explotación completo que pueden desencadenar estas familias de malware, una detección en una fase temprana es vital de cara a proteger la continuidad de negocio, en cuanto a la amenaza que supone el malware en este contexto.

En los puntos posteriores se entra en detalles técnicos sobre los antecedentes de Ryuk, su proceso de infección y su esquema de cifrado.

3. DETALLES GENERALES

El fichero analizado se corresponde con una muestra de ransomware de la familia Ryuk, protegida mediante un *custom packer*, que tratará de ocultar a soluciones de seguridad el payload responsable del cifrado de los ficheros. El ejecutable facilitado para análisis, así como el payload extraído tras el *unpacking*, se identifican con las siguientes firmas SHA256.

Fichero	SHA256
ryuk_packed.exe	e75622957decf1594c2cbe726ff0aaba4a509dab7b77721d3db16977f224ae4a
ryuk.exe	d7b324dfead641207c85cb18cdfc00bbfd37932f27f41c3af441d6912f235849

Buscando referencias al hash del fichero inicial, se ha encontrado que la primera fecha de la que se tiene constancia es del 4 de Noviembre de 2019.

Sin embargo, la fecha de compilación del payload data del 25 de octubre de 2019.

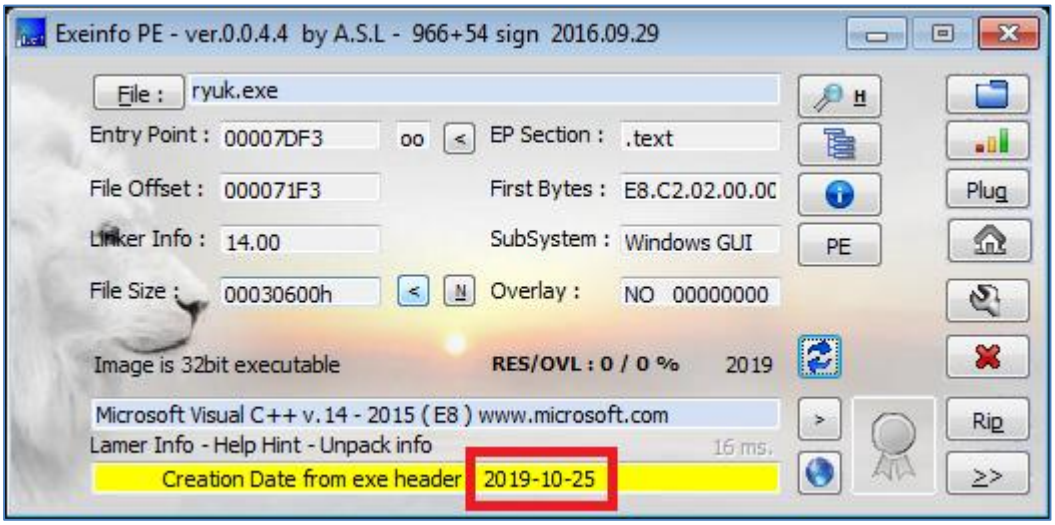


Figura 1. Fecha de compilación del payload

Debido al diseño del esquema de cifrado y gestión de claves, tema sobre el que se entrará en detalle a lo largo del informe, estas muestras son compiladas para objetivos específicos, no siendo reutilizadas para múltiples ataques.

El motivo de que no se reutilicen se basa en que, según el esquema de cifrado empleado, si varias entidades se viesen afectadas por la misma muestra de ransomware, con que una única entidad realizase el pago para recibir la herramienta de descifrado, todas las demás se podrían beneficiar de tal utilidad.

4. RANSOMWARE *AS A SERVICE*

De forma previa a detallar el proceso de infección, resulta interesante mencionar los orígenes de la familia de ransomware Hermes, proyecto del que deriva Ryuk y del que aún presenta múltiples trazas, así como cambios que se van a destacar.

Hermes surgió en 2017 y era vendido en foros por el actor **CryptoTech**.

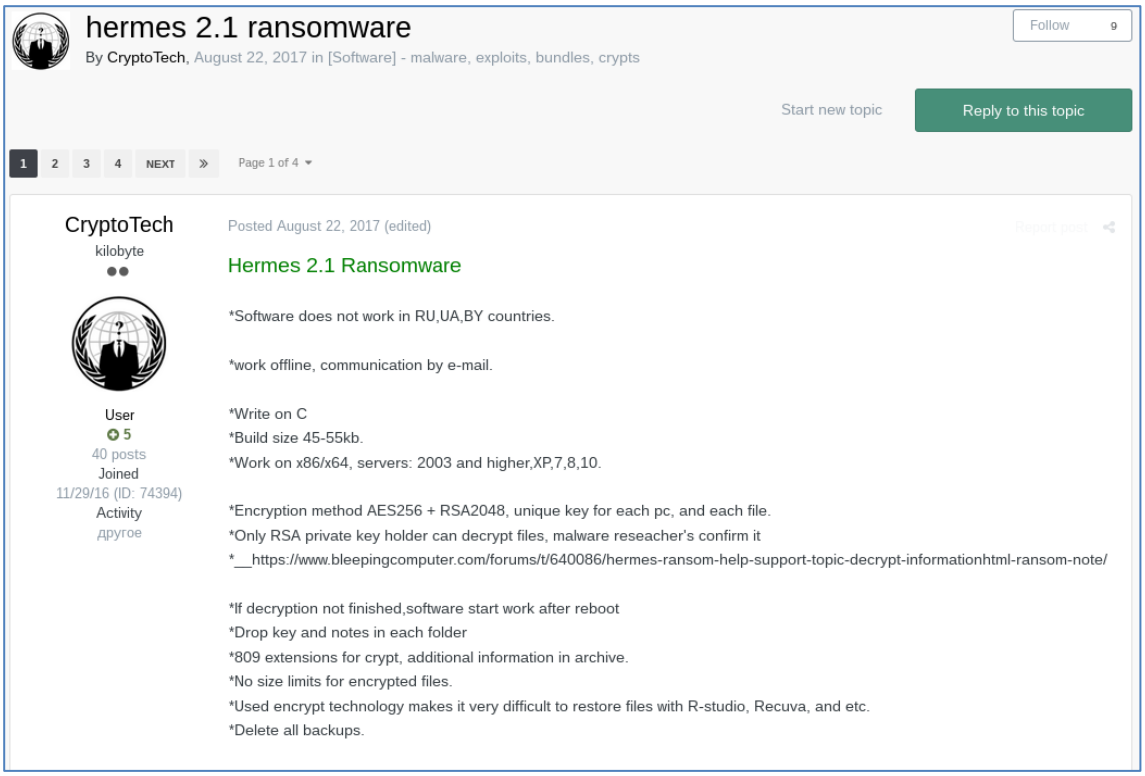


Figura 2. Venta de Hermes 2.1 en foros

En este momento, el código dañino implementaba controles para no afectar a usuarios localizados en Rusia, Ucrania o Bielorrusia. Como parte del modelo de **Ransomware *as a Service*** en el que está basado el negocio de **CryptoTech**, el kit a la venta por $300 incluía el ejecutable compilado con dos direcciones de e-mail en la nota de rescate, la herramienta de descifrado y el par único de claves RSA que garantiza que sólo el grupo criminal que compre el kit, podrá descifrar los ficheros de los usuarios afectados por el ransomware.

Mientras que no cabe duda de que Ryuk esté basado en Hermes, sí que sería cuestionable que **CryptoTech** siga contribuyendo al desarrollo del proyecto en la variante de Ryuk. El código ha cambiado de forma significativa y determinadas características sugerirían que posiblemente un nuevo grupo de actores tuviera acceso al código fuente y sean ellos quienes incorporarían estas nuevas funcionalidades.

5. PROCESO DE INFECCIÓN

La muestra de Ryuk analizada controla los posibles flujos de ejecución que tiene implementados mediante el uso de argumentos, admitiendo hasta un total de tres valores.

Si no se indica **ningún** parámetro por línea de comandos, el código dañino se replicará en el mismo directorio bajo un nombre aleatorio, se ejecutará esta réplica facilitando los argumentos "**8 LAN**" y se continuará con el proceso de infección.

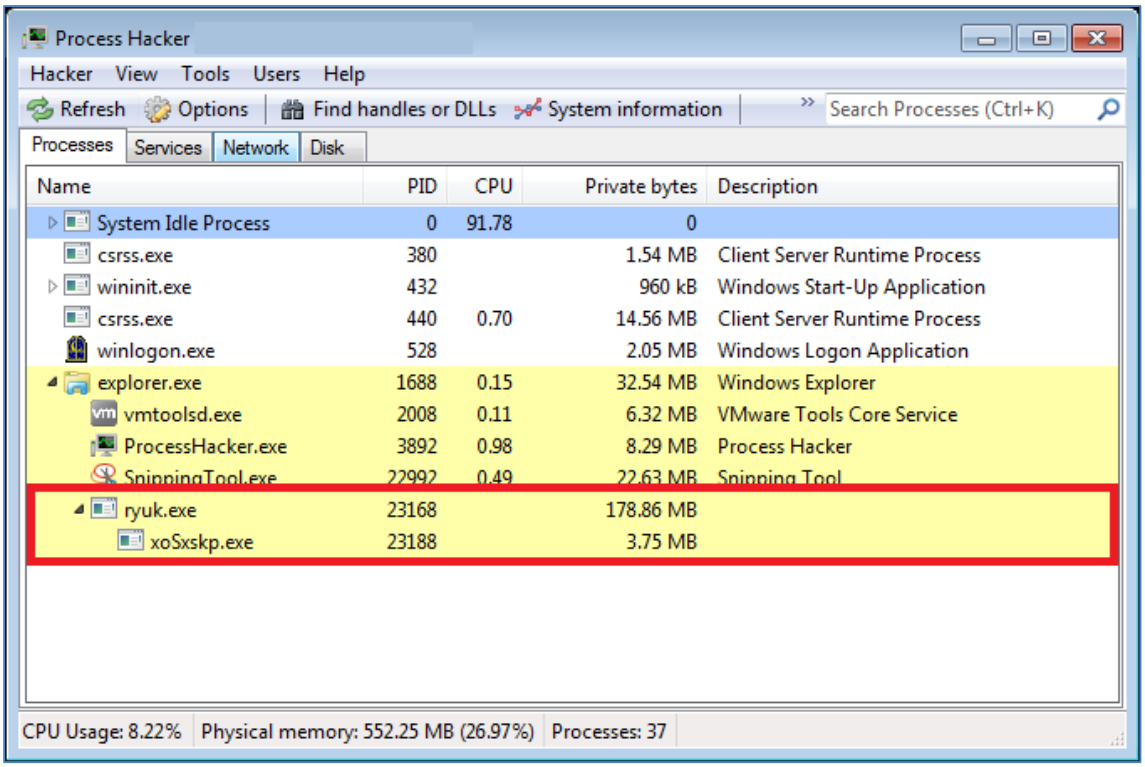


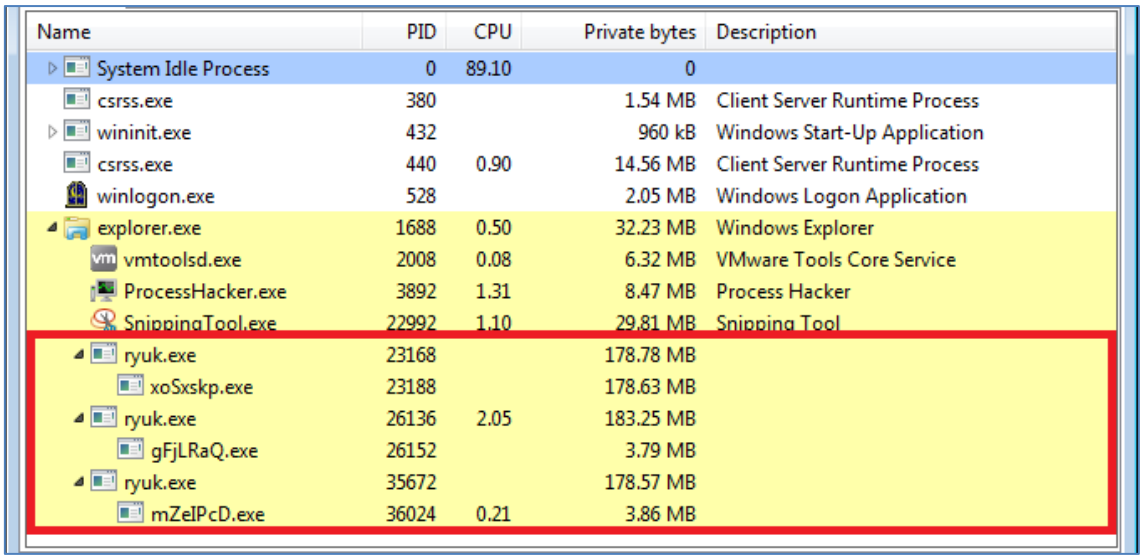
Figura 3. Ejecución de la muestra sin argumentos

Si sólo se especifica **un** argumento, el código dañino asume que es un fichero y lo tratará de eliminar mediante la función DeleteFileW, para posteriormente replicarse en el mismo directorio bajo un nombre aleatorio, ejecutar esta réplica facilitando los argumentos "**8 LAN**" y se continuará con el proceso de infección

Si se especifican **dos** argumentos y el primero de ellos es "**encrypt**" o "**-e**", se tomará el segundo argumento como la ruta del directorio cuyo contenido debe ser cifrado. En el caso de que los dos argumentos sean "**8 LAN**" (realmente el código dañino sólo comprueba que el segundo argumento empiece por "**L**"), se activarán dos métodos para extender la afectación por la red interna, Wake-on-LAN y un ping escáner, mecanismos sobre los que se entrará en detalle a lo largo del informe. Nótese que si no se especifican argumentos o si sólo se especifica uno, Ryuk lanzará una copia de sí mismo con los parámetros necesarios para activar esta funcionalidad.

Si se especifican **tres** argumentos, el código dañino asume que el primero es un fichero y lo tratará de eliminar mediante la función DeleteFileW y, si el segundo parámetro coincide con la cadena de texto "**encrypt**", se tomará el tercer argumento como la ruta del directorio cuyo contenido debe ser cifrado.

El código no implementa mecanismos de control de ejecución, como mutexes, por tanto, múltiples procesos de Ryuk se pueden encontrar corriendo en la misma máquina si la muestra es ejecutada varias veces.



Name	PID	CPU	Private bytes	Description
System Idle Process	0	89.10	0	
csrss.exe	380		1.54 MB	Client Server Runtime Process
wininit.exe	432		960 kB	Windows Start-Up Application
csrss.exe	440	0.90	14.56 MB	Client Server Runtime Process
winlogon.exe	528		2.05 MB	Windows Logon Application
explorer.exe	1688	0.50	32.23 MB	Windows Explorer
vmtoolsd.exe	2008	0.08	6.32 MB	VMware Tools Core Service
ProcessHacker.exe	3892	1.31	8.47 MB	Process Hacker
SnippingTool.exe	22992	1.10	29.81 MB	Snipping Tool
ryuk.exe	23168		178.78 MB	
xoSxskp.exe	23188		178.63 MB	
ryuk.exe	26136	2.05	183.25 MB	
gFjLRaQ.exe	26152		3.79 MB	
ryuk.exe	35672		178.57 MB	
mZeIPcD.exe	36024	0.21	3.86 MB	

Figura 4. Múltiples instancias del código dañino

Hermes comprobaba si el usuario pertenecía a Rusia, Ucrania o Bielorrusia a través del lenguaje del sistema, en cuyo caso afirmativo, la ejecución concluía sin mayor afectación. Ryuk no solamente no incluye esta comprobación, si no que las trazas que han dejado los desarrolladores indican que los códigos de mencionados países de la Comunidad de Estados Independientes han sido cambiados por el código correspondiente a Corea.

```
.rdata:30018397                 db    0
.rdata:30018398 language        db 'SYSTEM\CurrentControlSet\Control\Nls\Language\',0
.rdata:300183C7                 db    0
.rdata:300183C8 install_language db 'InstallLanguage',0
.rdata:300183D8 hex_0412        db '0412',0
.rdata:300183DD                 db    0
.rdata:300183DE                 db    0
.rdata:300183DF                 db    0
.rdata:300183E0 dec_1042        db '1042',0
```

Figura 5. Códigos de lenguaje asociados a Corea

Las trazas mostradas en la imagen anterior no se usan para ninguna comprobación y forman parte de los extensos valores y contenido que se encuentran sin referenciar a lo largo del código dañino.

Ryuk importa las funciones de Windows, que le permitirán interactuar con el sistema, de forma dinámica, y tras este paso, cederá la ejecución al procedimiento principal del código, con un argumento que modificará su flujo de ejecución.

```
.text:30007509          call    build_iat
.text:3000750E          push    8               ; main_argument
.text:30007510          call    main
.text:30007515          add     esp, 4
.text:30007518          push    6ACFC0h         ; dwMilliseconds
.text:3000751D          call    ds:Sleep
```

Figura 6. Llamada al procedimiento principal tras resolver las funciones a importar

Para tratar de causar el mayor impacto posible en las redes afectadas, dentro del procedimiento principal, Ryuk implementa tres mecanismos para cifrar contenidos en dispositivos y recursos remotos. Estas tres técnicas se van a describir a continuación.

5.1 WAKE-ON-LAN

El primero de los métodos, al que sólo se accede si la muestra es ejecutada con los parámetros "8 LAN", se denomina Wake-on-LAN [1] y mediante llamadas a la función GetIpNetTable se trataría de acceder a las entradas de la caché de la tabla del protocolo de resolución de direcciones (ARP).

Por cada entrada en mencionada tabla se enviará un paquete **Wake-on-LAN** de longitud 102 bytes (0x66 en hexadecimal) compuesto por 6 bytes con valor 255 (FF FF FF FF FF FF en hexadecimal) seguido de la dirección *media access control* (MAC) del equipo objetivo, repetida 16 veces.

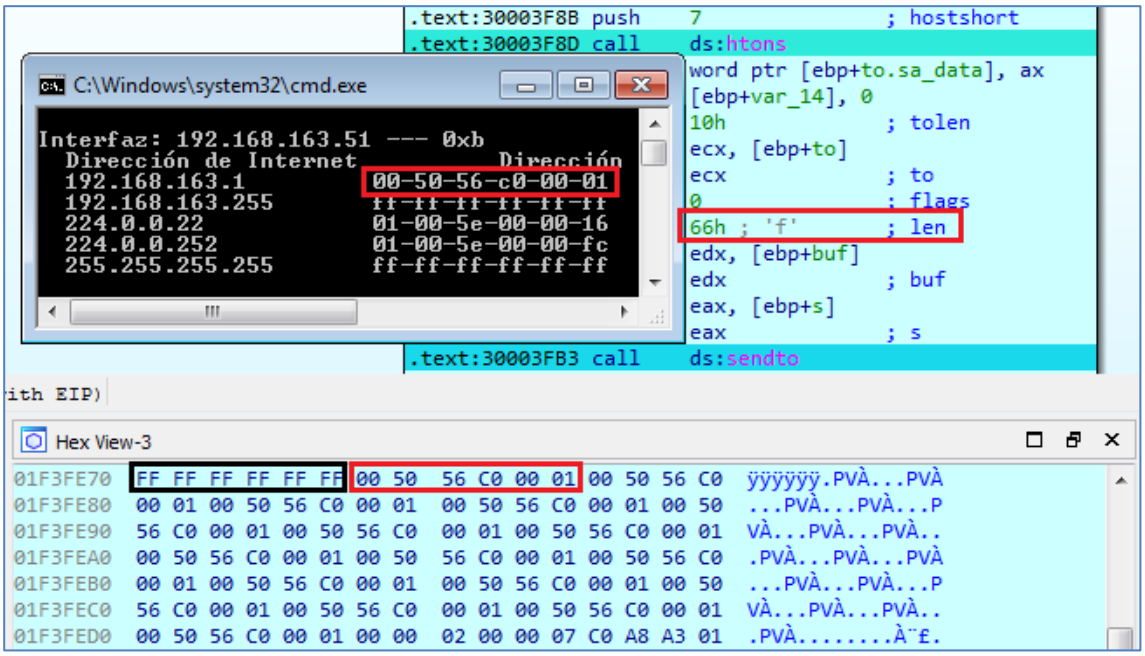


Figura 7. Envío del paquete Wake-on-LAN

[1] https://www.crowdstrike.com/blog/wizard-spider-adds-new-feature-to-ryuk-ransomware/

Si el sistema que recibe el paquete soporta **Wake-on-LAN** y se encuentra en *standby*, sería activado y se trataría de montar como una unidad de red para proceder con las rutinas de cifrado.

5.2 PING ESCÁNER

Otra técnica que Ryuk implementa para tratar de descubrir equipos en la red es un ping escáner, que se servirá de la función GetAdaptersAddresses para obtener las direcciones de las subredes a escanear. Solo se accederá a esta rutina si la muestra es ejecutada con los parámetros "**8 LAN**".

Cuando a través de mencionada llamada al sistema se obtenga una dirección IP, se comprobará si empieza por alguno de los siguientes valores.

- 10.
- 172.16.
- 192.168.

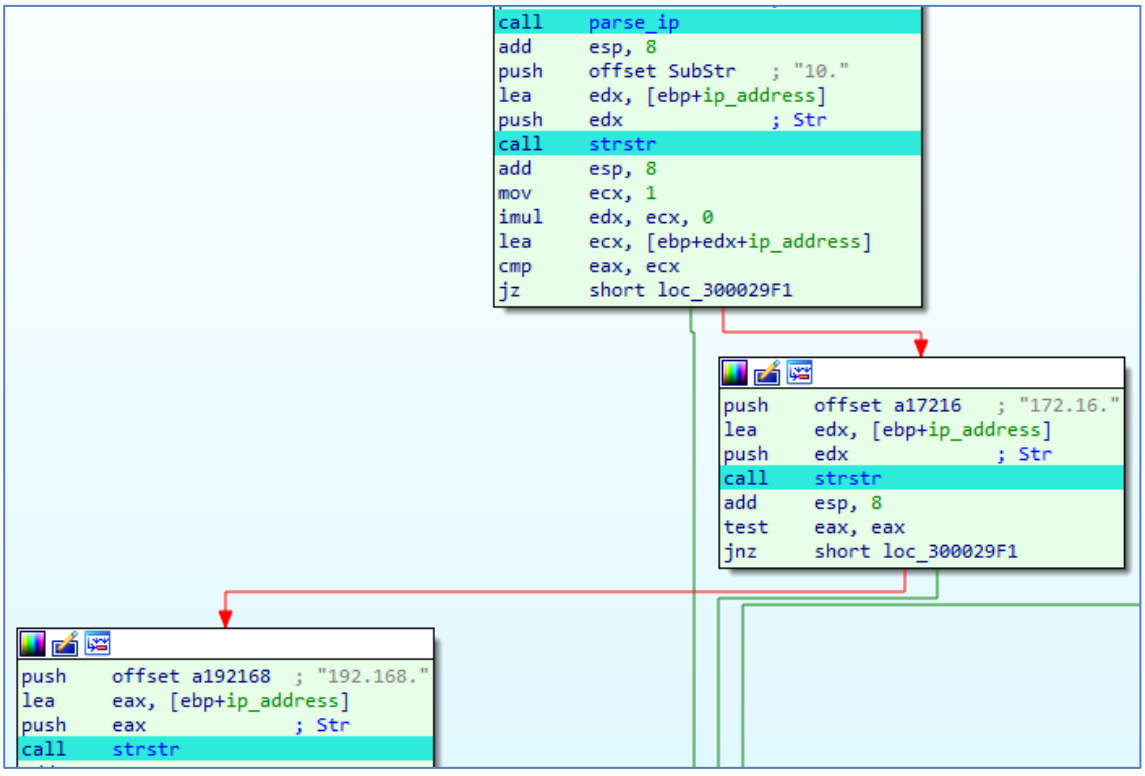


Figura 8. Comprobación de las posibles redes internas

En caso afirmativo, se escanearía todo el rango de la clase C para la dirección obtenida y si se obtuviese respuesta por parte de algún equipo, se trataría de montar como una unidad de red para proceder con las rutinas de cifrado.

5.3 RECURSOS DE RED

Por último, el tercer mecanismo intentará enumerar los recursos de red compartidos, para que en caso de tener acceso a los mismos, se proceda con las rutinas de cifrado sobre los ficheros almacenados en ellos.

Para enumerar los recursos de red, Ryuk importará las siguientes tres funciones de la librería mpr.dll.

- WNetOpenEnumW
- WNetEnumResourceW
- WNetCloseEnum

A través del fragmento de código que se lista a continuación, los recursos de red serían enumerados de forma recursiva para posteriormente, tratar de cifrar su contenido.

```
v6 = mpr_WNetOpenEnumW(2, 0, 0, lpNetResource, &lphEnum);
if ( v6 )
  return 0;
NETRESOURCE = kernel32_GlobalAlloc(64, lpBufferSize);
if ( !NETRESOURCE )
  return 0;
do
{
  overwrite_data(NETRESOURCE, 0, lpBufferSize);
  v6 = mpr_WNetEnumResourceW(lphEnum, &lpcCount, NETRESOURCE, &lpBufferSize);
  if ( v6 )
    return 1;
  if ( **(_WORD **)(NETRESOURCE + 20) == 92
    && *(_WORD *)(*(_DWORD *)(NETRESOURCE + 20) + 2) == 92
    && _LDint((wchar_t *)(*(_DWORD *)(NETRESOURCE + 20) + 6), 92) )
  {
    wcscat(Destination, *(const wchar_t **)(NETRESOURCE + 20));
    wcscat(Destination, L";");
  }
  if ( *(_DWORD *)(NETRESOURCE + 12) & 2 && !enum_network_resources(NETRESOURCE, Destination) )
    return 0;
}
while ( v6 != 259 );
mpr_WNetCloseEnum(lphEnum);
```

Figura 9. Enumeración recursiva de los recursos de red compartidos

Una vez listados los mecanismos para cifrar contenidos en dispositivos y recursos remotos, el proceso para elegir los ficheros susceptibles de cifrar, tanto en local como para esos contenidos externos, se cubrirá en el siguiente punto del documento.

5.4 ITERACIÓN SOBRE FICHEROS Y EXCEPCIONES

Mediante las funciones FindFirstFileW y FindNextFileW, el código dañino iterará sobre los ficheros del equipo, creando un hilo por cada fichero a cifrar. Para proceder con el cifrado del contenido del fichero, este debe ser superior a 24 bytes y no caer entre las excepciones que se van a listar a continuación.

Los siguientes directorios quedan exentos de cifrar su contenido y la nota de rescate no se escribiría en su interior.

Directorios en *whitelist*
AhnLab
Chrome
Mozilla
$Recycle.Bin
WINDOWS

La comprobación que se realiza no trata de buscar una coincidencia exacta en el nombre de los directorios, por tanto la carpeta "**Chrome"** quedaría libre de ser cifrada, como también lo haría la carpeta "aaa**Chrome**aaa". Este error en la comprobación se aplica a todas las reglas de *whitelisting* descritas en este apartado.

En un intento de excepcionar ficheros por extensión, si un fichero muestra alguna de las siguientes, tampoco sería cifrado.

Extensiones en *whitelist*
dll
hrmlog
exe
.ini
.lnk

De nuevo, debido al error señalado previamente, el fichero "test.**dll**" no sería cifrado, como tampoco lo sería el fichero "aaa**dll**aaa.txt". Cabe mencionar en este punto que la extensión **hrmlog** referenciaría a un artefacto del predecesor Hermes, artefacto que se usaría como fichero de log, actualmente en desuso.

En cuanto a los ficheros excluidos de ser cifrados, Ryuk excepciona los siguientes cuatro.

Ficheros en *whitelist*	
RyukReadMe.html	PUBLIC
UNIQUE_ID_DO_NOT_REMOVE	PRIVATE

Mientras que **RyukReadMe.html** sería la nota de rescate del ransomware, los otros tres se corresponden con artefactos heredados de Hermes, diseñados para asistir con la gestión de las claves en el proceso de cifrado, actualmente en desuso.

Finalmente, otro listado de directorios que se excluirían de cifrar su contenido, aunque la nota de rescate sí que se incluiría en su interior, se muestra a continuación.

Directorios en *whitelist*	
\Windows\	lib
sysvol	initrd
netlogon	sbin
bin	sys
boot	vmlinuz
dev	run
etc	var

Cabe destacar, que si una muestra se ejecuta con privilegios elevados, los mecanismos de *whitelisting* descritos podrían no ser suficientes para garantizar el correcto funcionamiento de los equipos tras ser reiniciados. Tras realizar las comprobaciones descritas, si un fichero es susceptible de ser cifrado, se crearía un hilo de ejecución adicional y el esquema empleado se detallará en el siguiente punto del informe.

6. ESQUEMA DE CIFRADO

Si bien se identificaban prácticas de desarrollo que dejaban trazas en el código sin referenciar y se indicaban fallos en la implementación de las comprobaciones, el esquema de criptografía y gestión de claves es sólido.

La muestra analizada incorpora embebida una clave pública RSA, cuya pareja privada se encuentra en posesión del grupo cibercriminal detrás de la campaña de Ryuk. De esta manera se asegura que sólo este grupo pueda garantizar el descifrado de los ficheros.

Una consideración es, que al únicamente utilizar un par de claves RSA, si múltiples organizaciones se viesen afectadas por esta misma muestra, con que sólo una de ellas adquiriese el software de descifrado, todas las demás se podrían beneficiar de esta utilidad. Sin embargo, el grupo encargado de distribuir los binarios es sensible a esta información y los compila específicamente, renovando el par de claves, antes de desplegarlos de forma masiva en redes comprometidas.

La clave pública RSA que incorpora el código dañino es importada, mediante la Crypto API de Windows, en los primeros pasos del procedimiento principal.

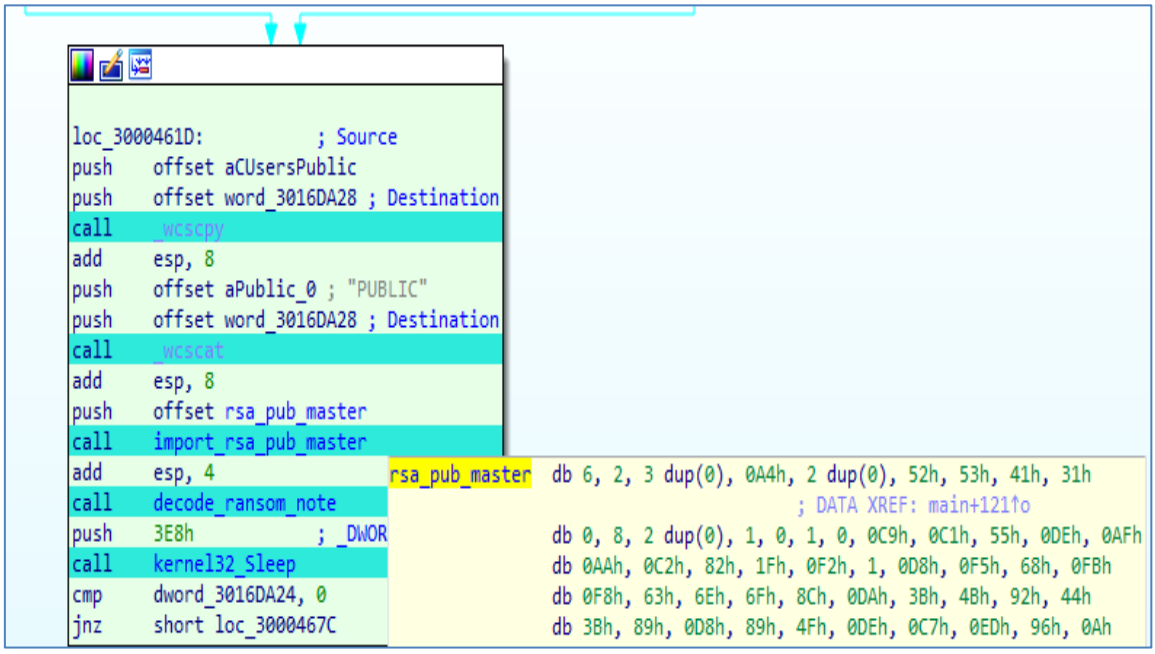


Figura 10. Importación de la clave pública RSA

El formato en que se encuentra la clave es propietario de Microsoft y su estructura se descompone en la siguiente imagen.

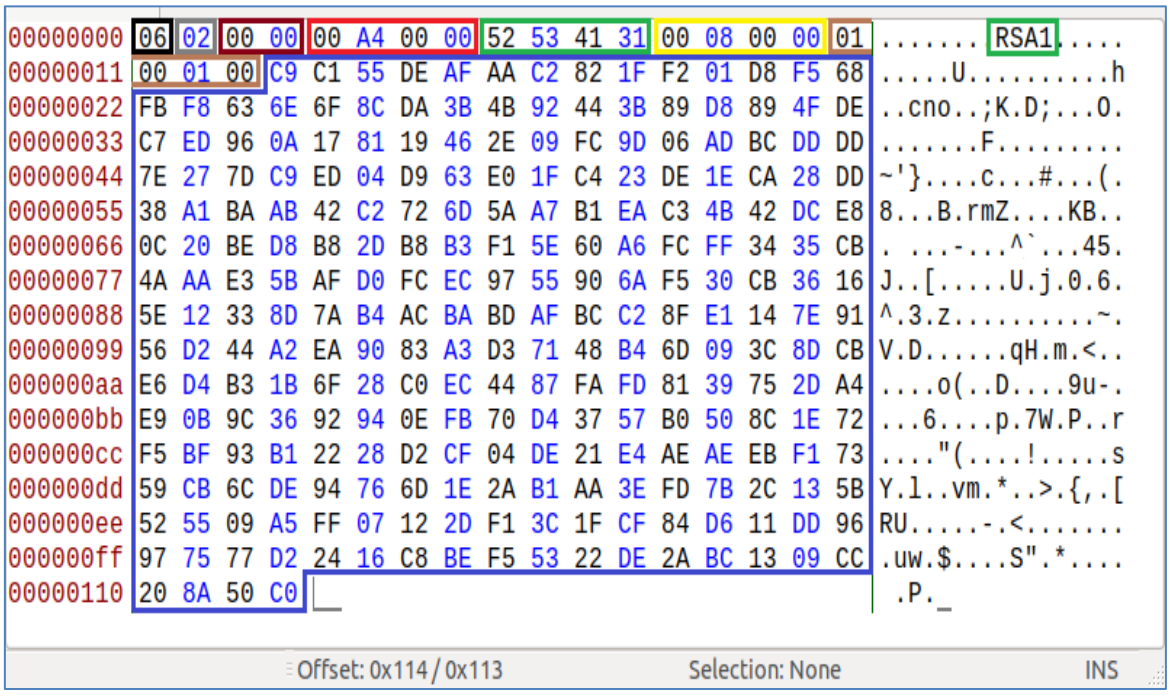


Figura 11. Estructura de la clave pública RSA embebida

Para cada campo, remarcado en un color diferente, se detalla su contenido en la tabla mostrada a continuación

Campo	Valor	Descripción
bType	0x06	PUBLICKEYBLOB
bVersion	0x02	CUR_BLOB_VERSION
reserved	0x0000	-
aiKeyAlg	0x0000A400	CALG_RSA_KEYX
magic	0x52534131	RSA1 para claves públicas y RSA2 para privadas
bitlen	0x00000800	Número de bits del módulo (2048 bits)
pubex	0x00010001	Exponente público
modulus	0xC9C1...50C0	Módulo de la clave pública RSA

Se adelantaba que cada fichero se cifra con una clave AES de 256 bits generada de forma aleatoria. Para la generación, se utiliza la función CryptGenKey.

```
.text:30001499 ; ---------------------------------------------------------------------------
.text:30001499
.text:30001499 generate_aes256_key:                    ; CODE XREF: encrypt_file+29F↑j
.text:30001499                                         ; encrypt_file+47D↑j
.text:30001499                 lea     edx, [ebp+hKey_aes256]
.text:3000149C                 push    edx             ; _DWORD
.text:3000149D                 push    1               ; _DWORD
.text:3000149F                 push    6610h           ; CALG_AES_256
.text:300014A4                 mov     eax, [ebp+hProv]
.text:300014A7                 push    eax             ; _DWORD
.text:300014A8                 call    advapi32_CryptGenKey
.text:300014AE                 test    eax, eax
.text:300014B0                 jnz     short loc_300014D0
```

Figura 12. Generación de claves AES aleatorias

Una vez cifrado el fichero a través de la función CryptEncrypt con la clave generada, esta clave AES se cifra con la pública RSA embebida, a través de un correcto uso de la función CryptExportKey, empleando el manejador a la clave pública RSA importada como parámetro en hExpKey.

```
.text:300019D7                 lea     edx, [ebp+pdwDataLen_ExpKey]
.text:300019DA                 push    edx             ; _DWORD
.text:300019DB                 lea     eax, [ebp+pbData_ExpKey] ; encrypted AES key
.text:300019E1                 push    eax             ; _DWORD
.text:300019E2                 push    0               ; _DWORD
.text:300019E4                 push    1               ; _DWORD
.text:300019E6                 mov     ecx, [ebp+hExpKey]
.text:300019E9                 push    ecx             ; _DWORD
.text:300019EA                 mov     edx, [ebp+hKey_aes256]
.text:300019ED                 push    edx             ; _DWORD
.text:300019EE                 call    advapi32_CryptExportKey
```

Figura 13. Exportación segura de la clave AES

Finalmente, cada fichero cifrado se marca con la etiqueta HERMES y se le añade la clave AES, tras ser cifrada con la pública RSA, a continuación de la marca HERMES. La

siguiente captura muestra un ejemplo de visualización de un fichero cifrado en un editor hexadecimal.

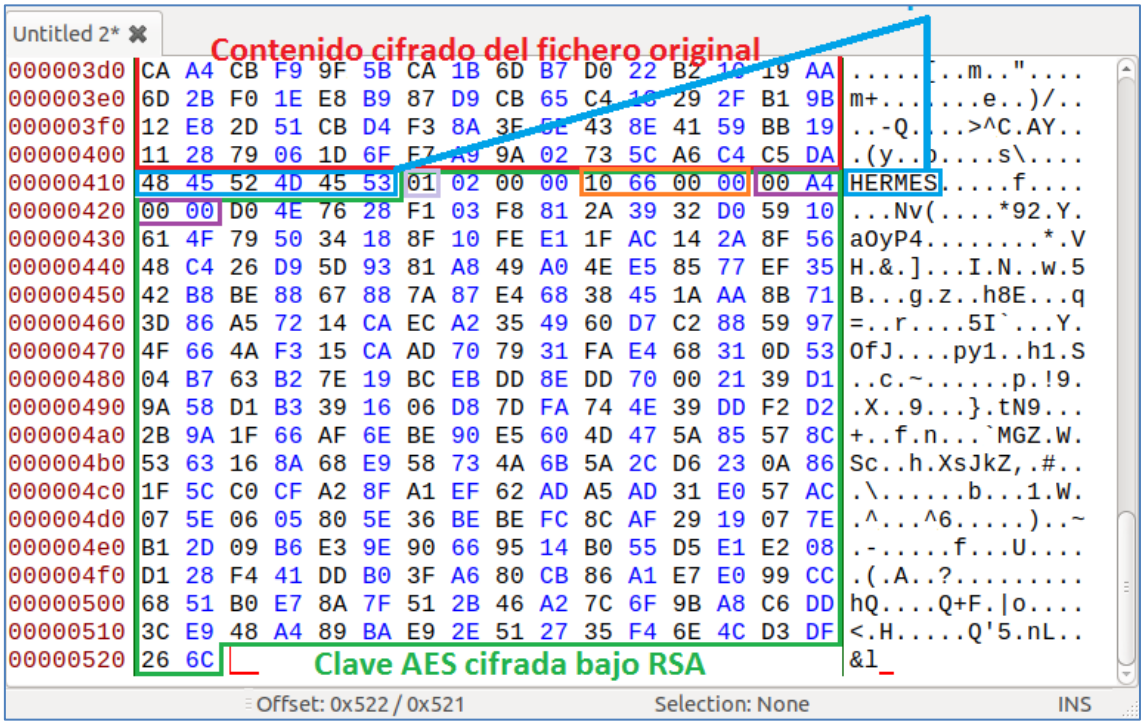


Figura 14. Estructura de la clave AES exportada

En la representación hexadecimal de la clave AES cifrada, se puede observar como la estructura desarrollada por Microsoft indica en el primer byte que se trata de una clave de sesión (0x01, SIMPLEBLOB) bajo el algoritmo AES con un tamaño de 256 bits de clave (0x00006610, CALG_AES_256) y que se encuentra cifrada bajo RSA (0x0000A400, CALG_RSA_KEYX).

La marca de cifrado se utiliza para no cifrar un mismo fichero dos veces, pues esta marca siempre se encontrará en la misma posición dentro de un fichero (a 274 bytes del final debido al tamaño fijo que posee la clave AES cifrada con la pública RSA). Antes de proceder a cifrar un fichero, la marca de cifrado será la última comprobación que se realizará para considerar si cifrar su contenido o no, evitando así cifrar un fichero múltiples veces.

```
.text:3000135C          movsx   ecx, byte ptr [ebp+eax+var_10C]
.text:30001364          cmp     ecx, 48h ; 'H'
.text:30001367          jnz     loc_3000146B
.text:3000136D          mov     edx, [ebp+var_1C]
.text:30001370          movsx   eax, byte ptr [ebp+edx+var_10C+1]
.text:30001378          cmp     eax, 45h ; 'E'
.text:3000137B          jnz     loc_3000146B
.text:30001381          mov     ecx, [ebp+var_1C]
.text:30001384          movsx   edx, byte ptr [ebp+ecx+var_10C+2]
.text:3000138C          cmp     edx, 52h ; 'R'
.text:3000138F          jnz     loc_3000146B
.text:30001395          mov     eax, [ebp+var_1C]
.text:30001398          movsx   ecx, byte ptr [ebp+eax+var_10C+3]
.text:300013A0          cmp     ecx, 4Dh ; 'M'
.text:300013A3          jnz     loc_3000146B
.text:300013A9          mov     edx, [ebp+var_1C]
.text:300013AC          movsx   eax, [ebp+edx+var_108]
.text:300013B4          cmp     eax, 45h ; 'E'
.text:300013B7          jnz     loc_3000146B
.text:300013BD          mov     ecx, [ebp+var_1C]
.text:300013C0          movsx   edx, [ebp+ecx+var_107]
.text:300013C8          cmp     edx, 53h ; 'S'
.text:300013CB          jnz     loc_3000146B
.text:300013D1          mov     eax, [ebp+hFile]
.text:300013D4          push    eax             ; _DWORD
.text:300013D5          call    kernel32_CloseHandle
```

Figura 15. Comprobación de la marca HERMES de cifrado

Finalmente, el nuevo contenido cifrado sobrescribe el del fichero original y se le añade la extensión **.RYK**.

7. NOTA DE RESCATE

La nota con las instrucciones de rescate consiste en una simple plantilla HTML mal formada con una dirección de e-mail y una cita que los desarrolladores de Ryuk suelen incluir, cuyo significado queda a libre interpretación.

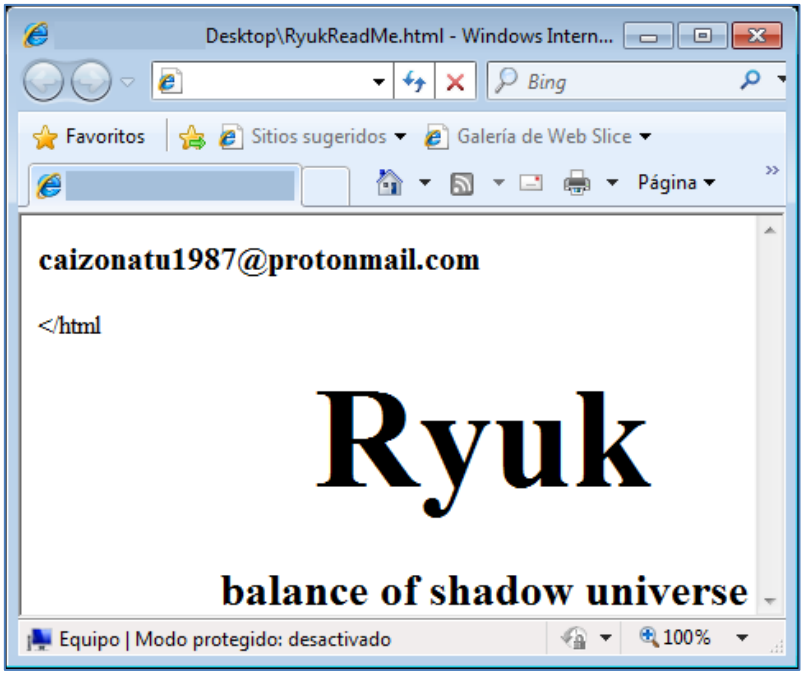


Figura 16. Nota de recate

A diferencia de otras familias de ransomware que cuentan con dominios o servicios en TOR para gestionar el pago de los usuarios afectados, en esta variante se incluye la dirección de correo electrónico **caizonatu1987@protonmail.com** para negociar el precio de la herramienta de descifrado y el método de pago.

8. PERSISTENCIA

La variante de Ryuk objeto de análisis no adquiere persistencia en el equipo. Aunque otras familias de ransomware, Ryuk/Hermes en versiones anteriores, sí tratan de asegurar su ejecución entre reinicios, esta funcionalidad no es implementada por esta versión.

Sin embargo, la nota con las instrucciones de rescate sí que sería lanzada al arrancar el sistema. No tanto como un mecanismo de persistencia, si no como una copia sin control del fichero con las instrucciones en todas las carpetas en las que el código dañino tiene permisos de escritura. Mencionada copia sin control situaría la nota de rescate dentro de la carpeta **Startup**, como se muestra en la siguiente captura.

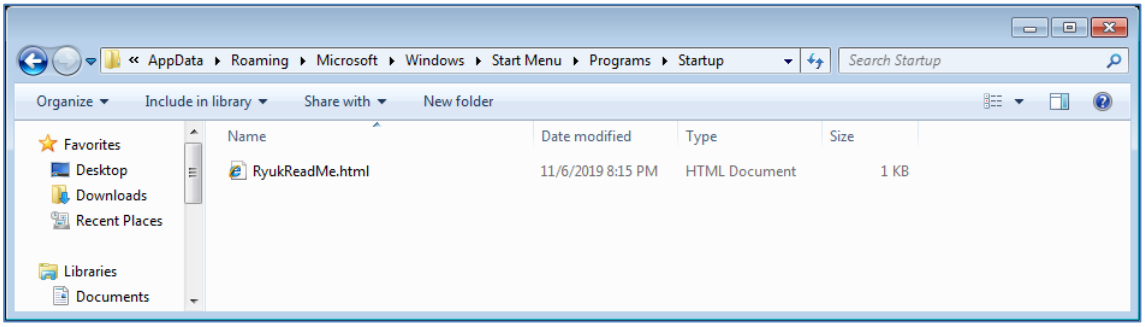


Figura 17. Nota de rescate en Startup

9. DESINFECCIÓN

Puesto que la variante analizada no adquiere persistencia, los equipos afectados con Ryuk no requieren de una rutina de desinfección más allá de reiniciar el sistema, con objeto de terminar los procesos relativos al ransomware. En el caso del descifrado de los ficheros, el modelo de criptografía utilizado asegura que el único método factible para el descifrado, sea mediante el uso de la clave privada RSA en manos del grupo cibercriminal.

10. REGLAS DE DETECCIÓN

10.1 REGLA YARA RYUK

```
rule ryuk {
   meta:
      date = "2019-11-18"
   strings:
      $ransom_note = "RyukReadMe.html" wide
      $key_file_0 = "UNIQUE_ID_DO_NOT_REMOVE" wide
      $key_file_1 = "PUBLIC" wide
      $key_file_2 = "PRIVATE" wide
      $extension_0 = "dll" wide
      $extension_1 = "hrmlog" wide
      $extension_2 = "exe" wide
      $extension_3 = ".ini" wide
      $extension_4 = ".lnk" wide
      $extension_ryk = ".RYK" wide
      $ip_range_0 = "10." ascii
      $ip_range_1 = "172.16." ascii
      $ip_range_2 = "192.168." ascii
      $rsa_pub = {0602000000A40000}
   condition: uint16(0) == 0x5A4D and (all of them)
}
```

11. INDICADORES DE COMPROMISO

Ryuk ransomware
e75622957decf1594c2cbe726ff0aaba4a509dab7b77721d3db16977f224ae4a
Nota de rescate
RyukReadMe.html
Extensión añadida a ficheros cifrados
.RYK