Informe Código Dañino
CCN-CERT ID-03/21

Ryuk Ransomware

Marzo 2021

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

CCN centro criptológico nacional

Fecha de Edición: abril de 2021

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

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 40878dbaeedf8b5cedf878623c9f519d, perteneciente a la familia de *ransomware* Ryuk, derivado de la familia de *ransomware* Hermes [REF.-1].

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 [REF.-2], suponiendo una amenaza severa a la continuidad de negocio de la misma. Desde el pasado 2018, la nueva tendencia que se está observando en código dañino bancario como TrickBot, Dridex o QBot, es aprovechar la naturaleza modular de estos proyectos para comprometer, mediante el uso de módulos adicionales, 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 podría implicar un compromiso previo por parte de Emotet, TrickBot o BazarLoader. Debido al grave impacto que supondría el ciclo de explotación completo que pueden desencadenar estas familias de *código dañino*, una detección en una fase temprana es vital de cara a proteger la continuidad de negocio, en cuanto a la amenaza que supone el *código dañino* en este contexto.

La característica más destacable y significativa de la muestra analizada es su capacidad de propagación a modo de gusano. Además de aprovecharse del protocolo *Wake-on-Lan* para arrancar equipos apagados hace uso de SMB y RPC para propagar y ejecutar el *ransomware* en otras máquinas a las que tiene acceso.

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 las soluciones de seguridad el *payload* responsable del cifrado de los ficheros. La firma del código dañino es la siguiente:

MD5	40878dbaeedf8b5cedf878623c9f519d
SHA1	6cb642afe300c337338517faa49b60c94316ed1b
SHA256	fb748c70ad17c2b27096958d34b703f3707e6bf2668962b7a36e6bf19daef249

Cabe destacar que la muestra analizada no se ha identificado, a fecha de realización del presente informe, en plataformas de análisis de código dañino *online* tales como Any.Run, Virustotal, Joe Sandbox, Hybrid-Analysis o similares.

El binario tiene un tamaño de 304.728 bytes. Está compilado para arquitecturas de 32 bits y, mientras la fecha de compilación de su *File Header* refleja el 14 de diciembre de 2019, el *TimeDateStamp* de su *Export Directory* indica el 9 de marzo de 2021. Cabe mencionar que únicamente algunos *linkers,* tales como Visual Studio, reflejan este *TimeDateStamp* en su directorio de exportación cuando éste es creado.

Figura 1. Diferencias en el campo *TimeDateStamp*

Nota: la fecha de compilación del binario embebido (el propio Ryuk tras el *unpacking*) es del 3 de marzo de 2021. Teniendo en cuenta la información anterior y, siempre y cuando no se hayan alterado intencionadamente los *TimeDateStamp,* el *ransomware* podría haberse compilado el 3 de marzo y empaquetado el día 9. No obstante, esto es una hipótesis sustentada por datos que son fácilmente manipulables.

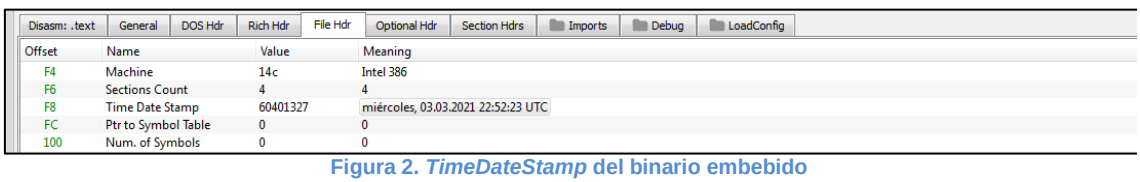
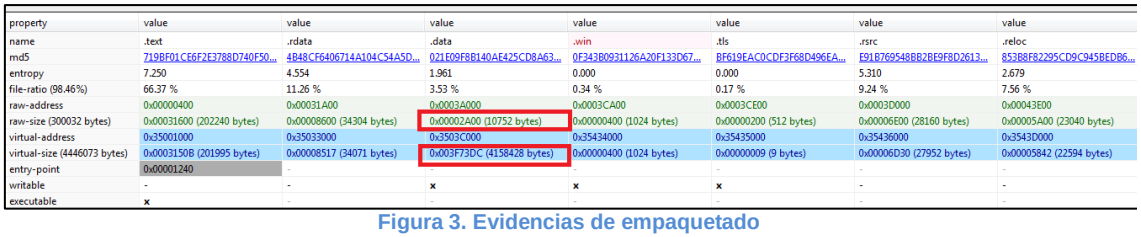


Figura 2. *TimeDateStamp* del binario embebido

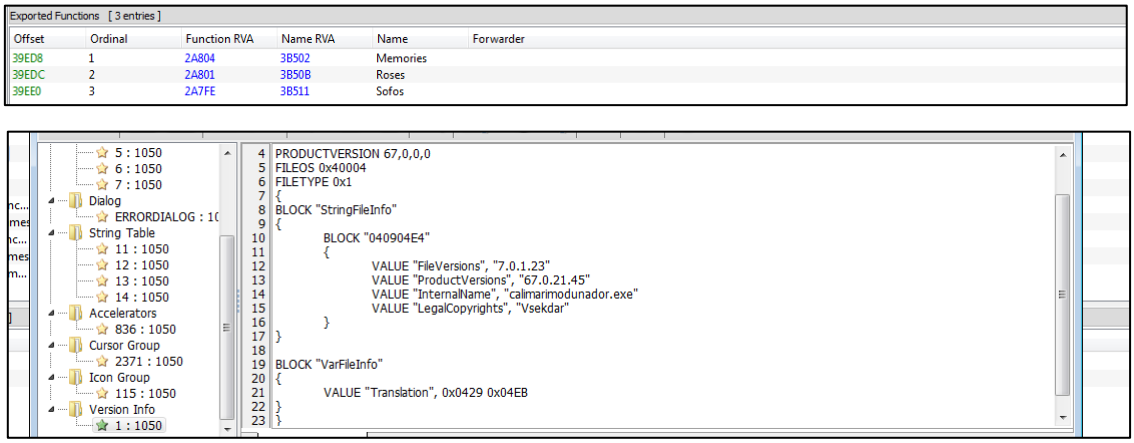
La discrepancia existente en algunas de sus secciones (por ejemplo, la sección *.data*) entre su *raw* y *virtual size,* junto con la elevada entropía de la sección *.text,* evidencian que el binario se encuentra ofuscado/empaquetado.



property	value	value	value	value	value	value	value
name	.text	.rdata	.data	.win	.tls	.rsrc	.reloc
md5	719BF01CE6F2E3788D740F50...	4B48CF6406714A104C54A5D...	021E09F8B140AE425CD8A63...	0F343B0931126A20F133D67...	BF619EAC0CDF3F68D496EA...	E91B769548BB2BE9F8D2613...	853B8F82295CD9C945BEDB6...
entropy	7.250	4.554	1.961	0.000	0.000	5.310	2.679
file-ratio (98.46%)	66.37 %	11.26 %	3.53 %	0.34 %	0.17 %	9.24 %	7.56 %
raw-address	0x00000400	0x00031A00	0x0003A000	0x0003CA00	0x0003CE00	0x0003D000	0x00043E00
raw-size (300032 bytes)	0x00031600 (202240 bytes)	0x00008600 (34304 bytes)	0x00002A00 (10752 bytes)	0x00000400 (1024 bytes)	0x00000200 (512 bytes)	0x00006E00 (28160 bytes)	0x00005A00 (23040 bytes)
virtual-address	0x35001000	0x35033000	0x3503C000	0x35434000	0x35435000	0x35436000	0x3543D000
virtual-size (4446073 bytes)	0x0003150B (201995 bytes)	0x00008517 (34071 bytes)	0x003F73DC (4158428 bytes)	0x00000400 (1024 bytes)	0x00000009 (9 bytes)	0x00006D30 (27952 bytes)	0x00005842 (22594 bytes)
entry-point	0x00001240	-	-	-	-	-	-
writable	-	-	x	x	x	-	-
executable	x	-	-	-	-	-	-

Figura 3. Evidencias de empaquetado

El binario exporta tres símbolos bajo los nombres "*Memories*", "*Roses*" y "*Sofos*". Asimismo, entre las propiedades del fichero se encuentra como *InternalName* el valor "*calimarimodunador.exe*" y en el campo *LegalCopyrights* el valor "*Vsekdar*".



Exported Functions [3 entries]

Offset	Ordinal	Function RVA	Name RVA	Name	Forwarder
39ED8	1	2A804	3B502	Memories	
39EDC	2	2A801	3B50B	Roses	
39EE0	3	2A7FE	3B511	Sofos	

Figura 4. Símbolos exportados / Propiedades fichero

Utilizando como criterio dichos valores junto a otras características estáticas del binario, se han identificado múltiples muestras similares en plataformas de análisis como VirusTotal y JoeSanbox. Estas muestras, en su mayoría, han sido remitidas para su análisis en días e incluso horas cercanas a la identificación de la muestra actual (11 de marzo de 2021). Algunas de estas muestras se corresponden con *stealers.* El hecho de haber sido generadas por el mismo *builder/packer* explicaría las similitudes entre las mismas. A continuación, se listan algunas muestras con características similares:

https://www.virustotal.com/gui/file/5386671f80dea3cde63c7418cb91f29540d145dd6129414140a7d9c5fca1a149/details

https://www.virustotal.com/gui/file/79d0da906de6dc170337e0063c28235fb2e0e86a0c2c73f2701d2b3f56b38c7d/details

https://www.virustotal.com/gui/file/78f5967d8bd5d31343632e8558b9315d9d44d009c923b9c41d417943906f2849/details

https://www.virustotal.com/gui/file/043c9b0bca213f980d6aa027e1c4f594ab4fea278e2c9582b134bafc3d859d7d/details

Figura 5. Ejemplo de muestras con características estáticas similares

Asimismo, el binario dañino está firmado por "*Certum Trusted Network CA*".

Figura 6. Firma del binario dañino

Cabe mencionar que un gran número de muestras [REF.-3] vinculadas con BazaLoader, identificadas entre noviembre de 2020 y febrero de 2021, han sido firmadas con dicho emisor.

BazaLoader, desde mediados de septiembre de 2020, ha formado parte de la cadena de infección de Ryuk (generalmente por medio de campañas de phishing) de forma análoga a la empleada anteriormente por Emotet y TrickBot.

Figura 7. BazaLoader (*Certum Trusted Network CA*)

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 describirán a continuación.

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

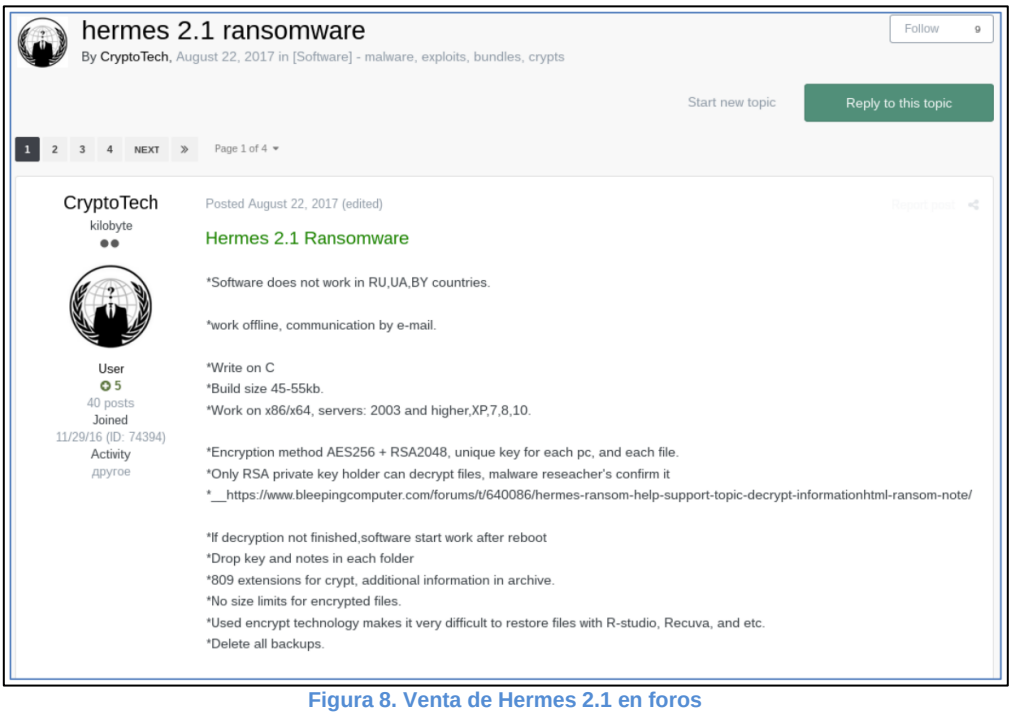


Figura 8. Venta de Hermes 2.1 en foros

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 quepa 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. CARACTERÍSTICAS TÉCNICAS

La muestra analizada se encuentra empaquetada. Diversos *shellcodes* serán los responsables de desofuscar y extraer el *payload* con el código dañino de Ryuk.

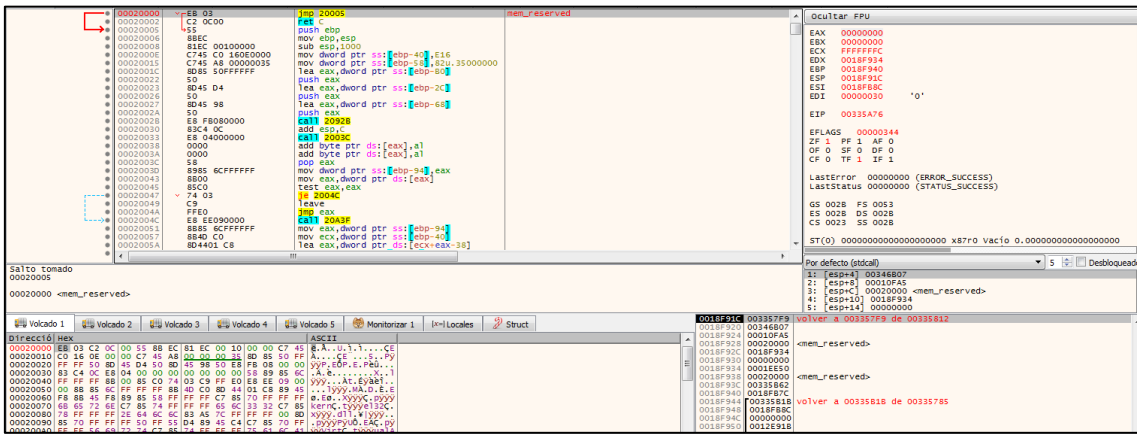


Figura 9. Shellcode 1

La siguiente imagen muestra el *payload* embebido tras finalizar las rutinas de desempacado (un binario de unos 120 KB).

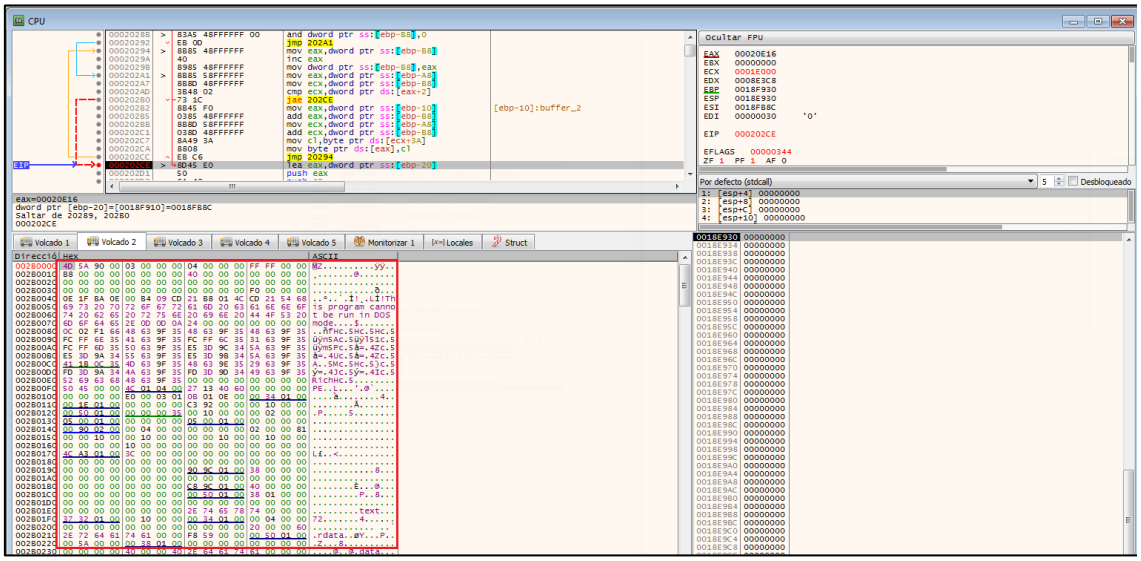


Figura 10. Payload (desempaquetado)

Posteriormente el *payload* se mapeará sobre el propio espacio de direcciones del binario principal reemplazando algunas de sus secciones. Asimismo, se resolverá la IAT y se hará un *jump* al OEP del mismo.

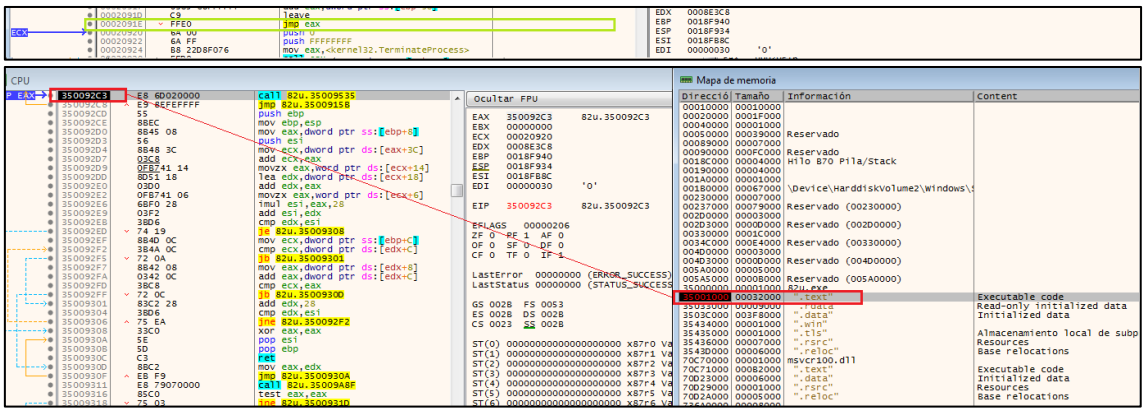


Figura 11. *Original Entry Point*

Ryuk implementa de forma reiterativa técnicas *anti-disassembly* para dificultar la ingeniería inversa desde herramientas de análisis estático.

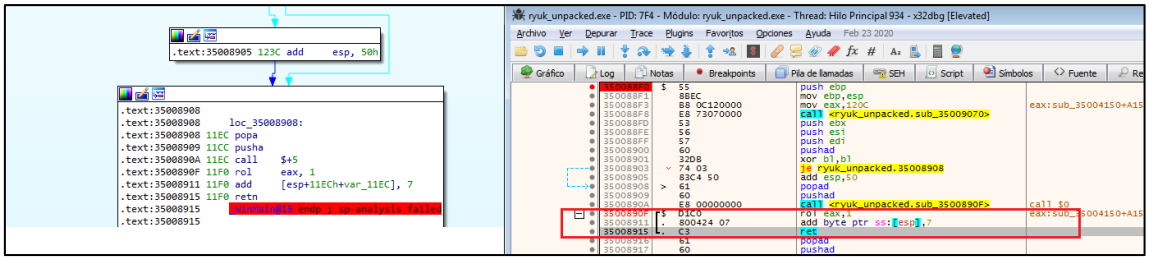


Figura 12. *Anti-Disassembly*

Asimismo, el código dañino implementará técnicas *anti-debug*, mediante el uso, por ejemplo, de APIs como *ZwQueryInformationProcess* junto con diversos *flags* (*ProcessDebugPort, ProcessDebugFlags, ProcessDebugObjectHandle*) que permitirán determinar si hay un *debugger* presente.

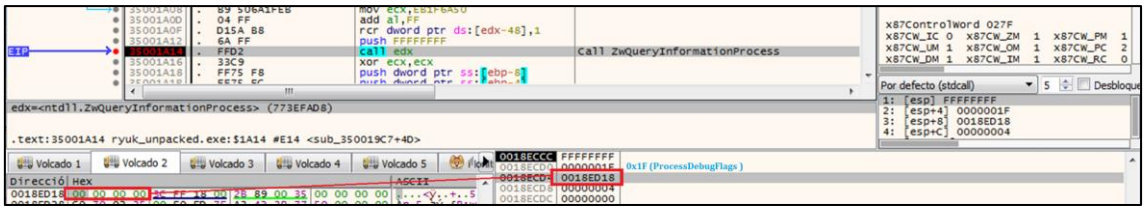


Figura 13. *Anti-Debug*

Como medida *anti-debugging* adicional comprobará el *flag BeingDebugged* alcanzable desde el PEB (*Process Environment Block*).

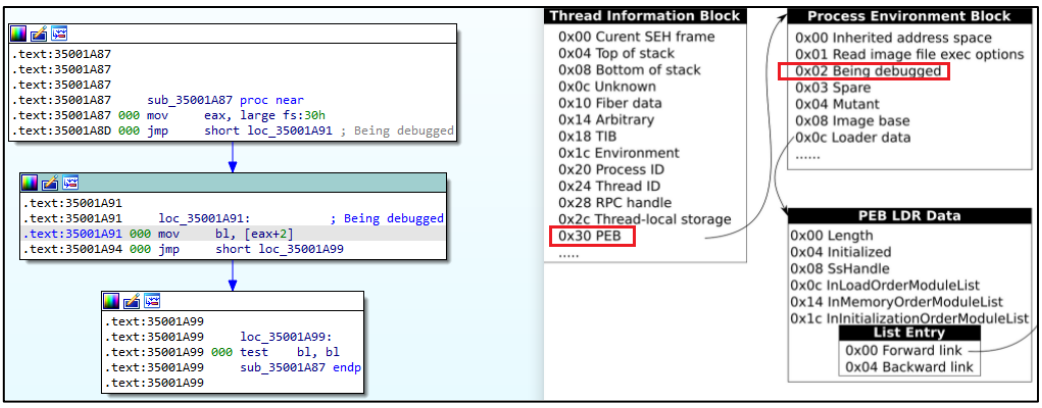
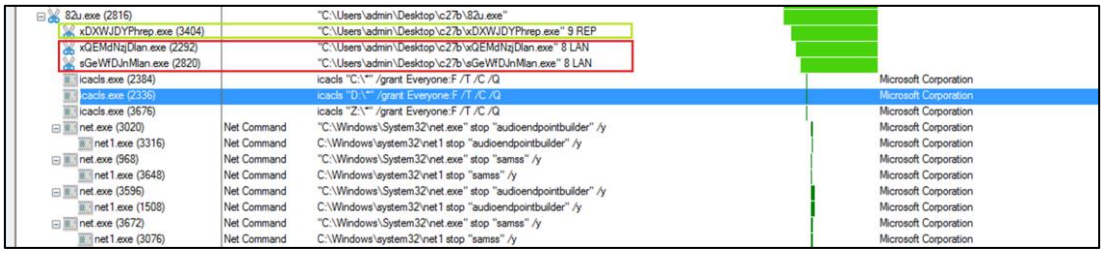


Figura 14. *Anti-Debug*

En el código se observan controles heredados de Hermes para verificar el host y asegurarse que no se está ejecutando en equipos de usuarios localizados en Rusia, Ucrania o Bielorrusia. Para verificar el idioma del host, consulta la clave de registro "*SYSTEM\CurrentControlSet\Control\Nls\Language*" y el valor *InstallLanguage*. Si la máquina tiene el valor 0419 (ruso), 0422 (ucraniano) o 0423 (bielorruso), invoca a *ExitProcess* para detener la ejecución.

Figura 15. Comprobación idioma

El árbol de procesos que generará Ryuk puede observarse en la siguiente imagen. Por un lado, se ejecutarán diversas copias de sí mismo con los argumentos "9 REP" y "8 LAN". Estos procesos serán los responsables de arrancar máquinas apagadas mediante el protocolo *Wake-on-Lan* y de replicarse a través de recursos compartidos y tareas programadas (más información en el punto 7).



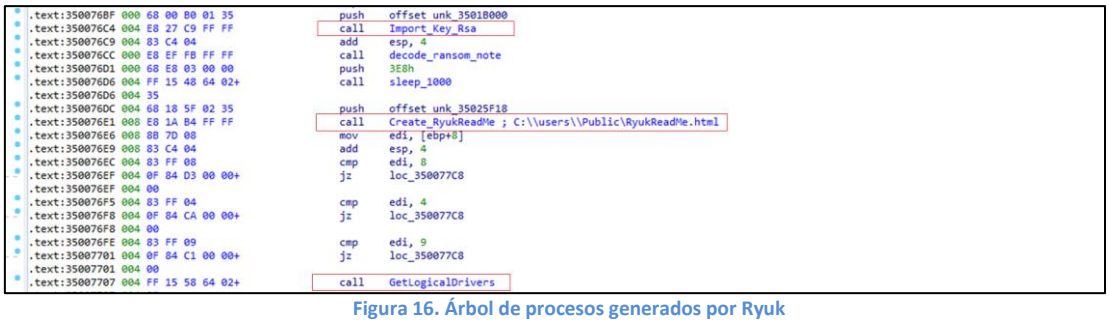


Figura 16. Árbol de procesos generados por Ryuk

De forma paralela, el binario principal importará la clave RSA que participará en el proceso de cifrado de los ficheros (más información en el punto 0), identificará las unidades locales montadas (vía *GetLogicalDrives*) y, por cada una de ellas, cambiará sus permisos mediante la herramienta de Windows *icacls* para otorgar *full access* a los mismos. Ej: *icacls "C:*" /grant Everyone:F /T /C /Q*

En la imagen anterior pueden apreciarse las instancias de *icacls* para las unidades C:, D: y Z: (procesos con PID: 2384, 2336, 3676).

Obsérvese además, que se ejecutarán múltiples comandos "*net stop*" con el objetivo de finalizar los servicios "*Windows Audio Endpoint Builder*" (*audioendpointbuilder*) y el servicio "*Security Accounts Manager*" (*samss*).

Previo al comienzo del cifrado de ficheros, Ryuk decodificará el contenido del fichero *RyukReadMe.html,* que será guardado en los directorios afectados y en donde se referencia a la URL (*.onion*) a partir de la cual, es posible ponerse en contacto con los atacantes junto con la *password* necesaria.

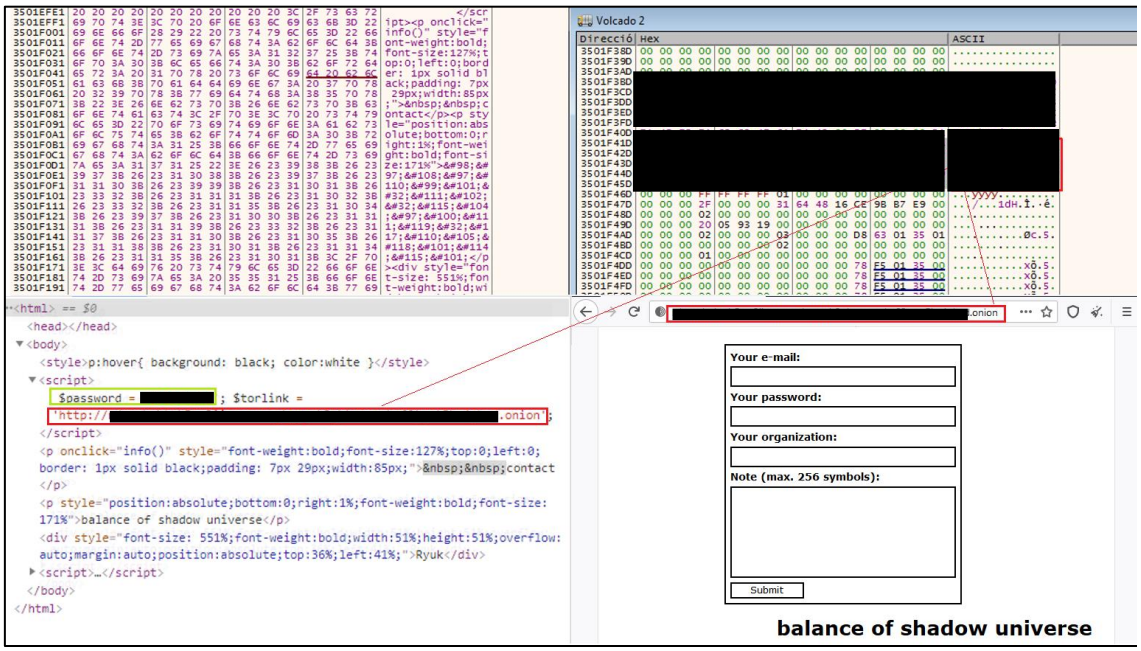


Figura 17. Contenido fichero *RyukReadMe.html*

Aparte de la creación de ficheros *RyukReadMe.html,* el código dañino podrá instalar una tarea programada para imprimir (en la impresora por defecto) el mensaje de Ryuk junto con la contraseña. La tarea mostrada a continuación imprimiría durante una semana, a una determinada hora, 50 páginas con el contenido de *RyukReadMe.*

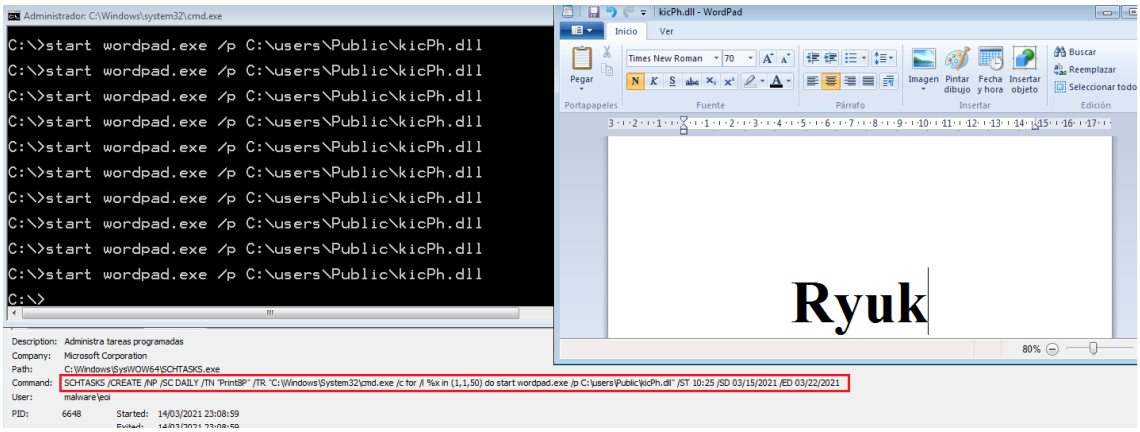


Figura 18. Tarea programada: impresión de la notificación Ryuk

6. 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.

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, éste debe ser superior a 24 bytes.

Nota: El código dispone de una lista de exclusión con determinados *strings* (por ejemplo: “Mozilla”, “Chrome”, etc.) de forma que si el nombre del fichero contiene el mismo no procederá con su cifrado. Es decir, que si un fichero tiene como nombre xxxChromeyyy.txt, éste no se vería afectado por la rutina de cifrado. Asimismo, los ficheros en el directorio “c:\Windows” quedarán excluidos

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

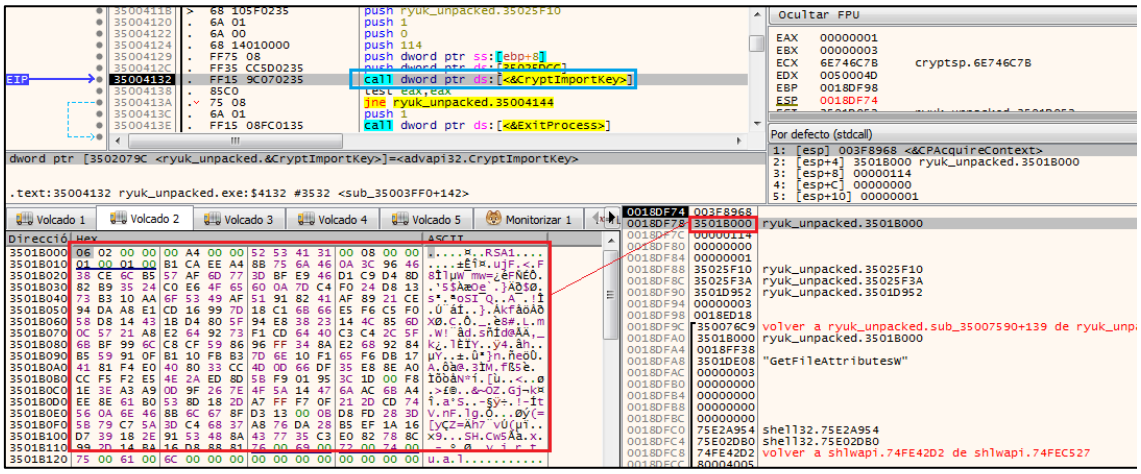


Figura 19. Importación clave pública

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

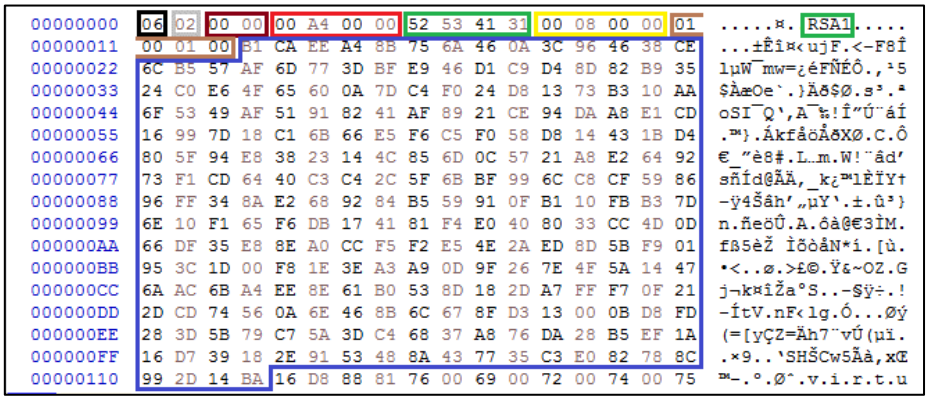


Figura 20. Estructura de la clave pública RSA embebida

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	0x B1CA...14BA	Módulo de la clave pública RSA

Cada fichero será cifrado con una clave AES de 256 bits generada de forma aleatoria por medio de la API *CryptGenKey.*

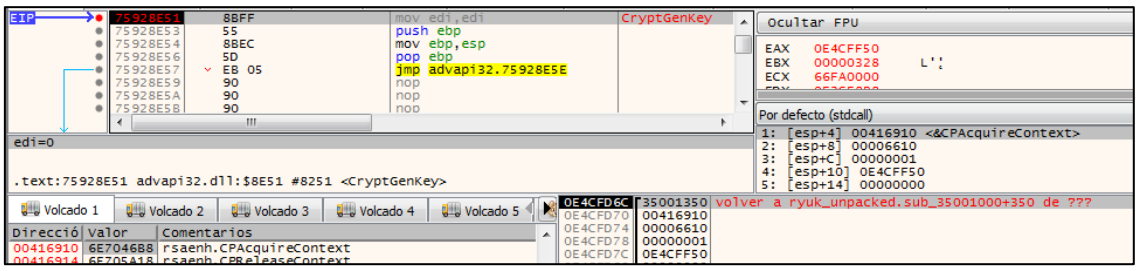


Figura 21. *CryptGenKey* (generación clave AES)

Posteriormente se cifrará el fichero por medio de la función *CryptEncrypt* haciendo uso de la clave previamente generada.

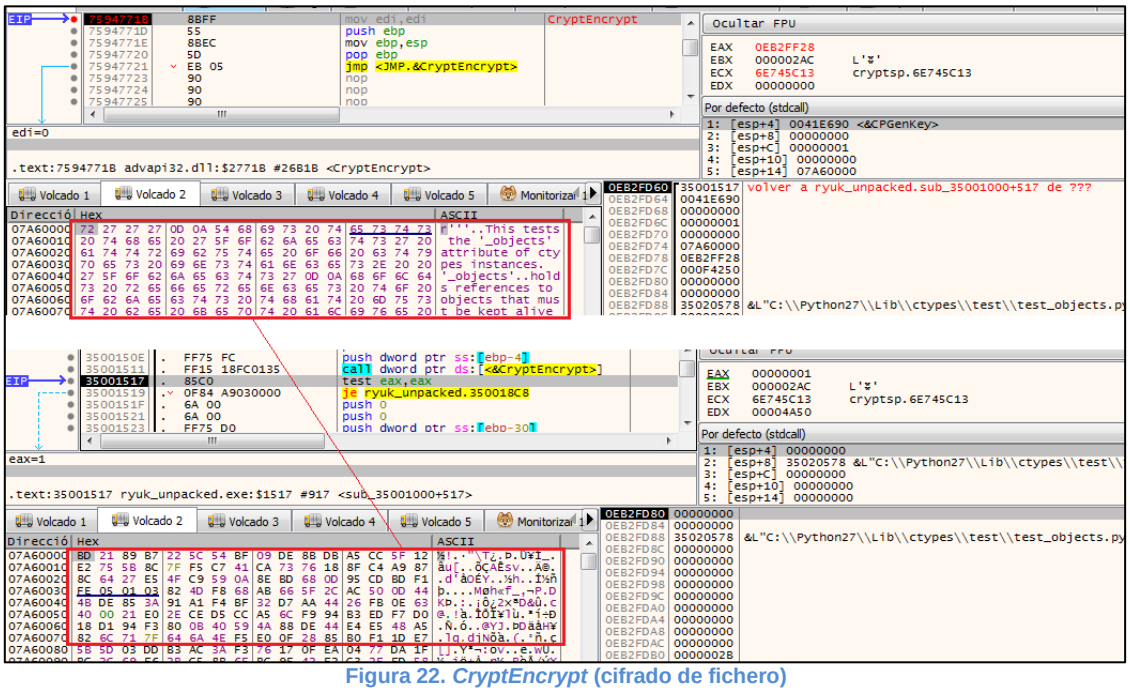


Figura 22. *CryptEncrypt* (cifrado de fichero)

Finalmente, la clave AES es cifrada 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.*

Figura 23. Rutinas de cifrado

Finalmente, cada fichero cifrado se marca con la etiqueta RYUKTM y se le añade, a continuación, la clave AES tras ser cifrada con la pública RSA. Esta marca siempre se encontrará en el mismo *offset* dentro del fichero cifrado (a 274 bytes del final debido al tamaño fijo que posee la clave AES cifrada con la pública RSA).

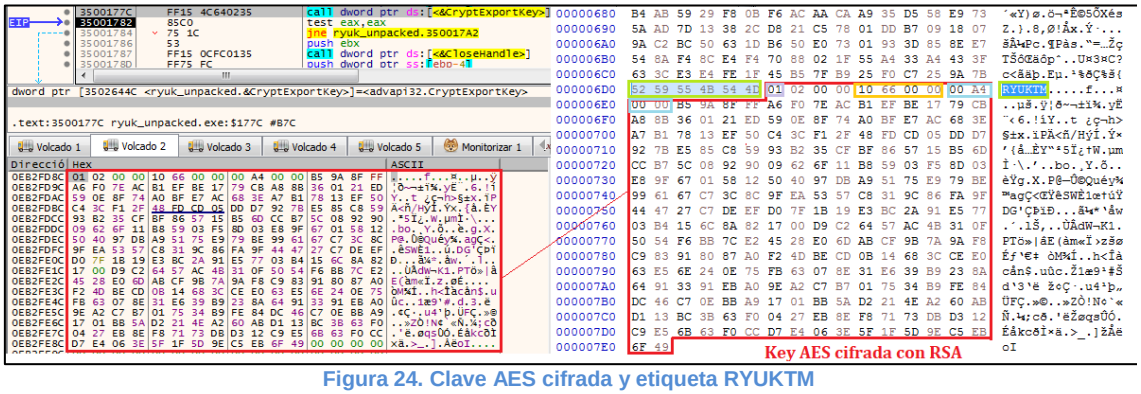


Figura 24. Clave AES cifrada y etiqueta RYUKTM

En la representación hexadecimal de la clave AES cifrada (justo después de la etiqueta **RYUKTM**), 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).

7. PROCEDIMIENTO DE INFECCIÓN A OTRAS MÁQUINAS

La variante analizada tiene capacidad de propagarse a modo de gusano si dispone de los permisos y privilegios necesarios para acceder a los *shares* de otras máquinas. Además, tal y como se ha visto en versiones anteriores [REF.-4], hace uso del protocolo *Wake-on-Lan* [REF.-5] para encender equipos que se encuentren apagados. El binario principal creará varias copias de sí mismo y generará diversos procesos hijos con los argumentos "8 LAN" y "9 REP" para implementar dichas funcionalidades. Para la creación de las nuevas copias utilizará la siguiente convención:

xxxxxxxlan.exe 8 LAN
xxxxxxxrep.exe 9 REP

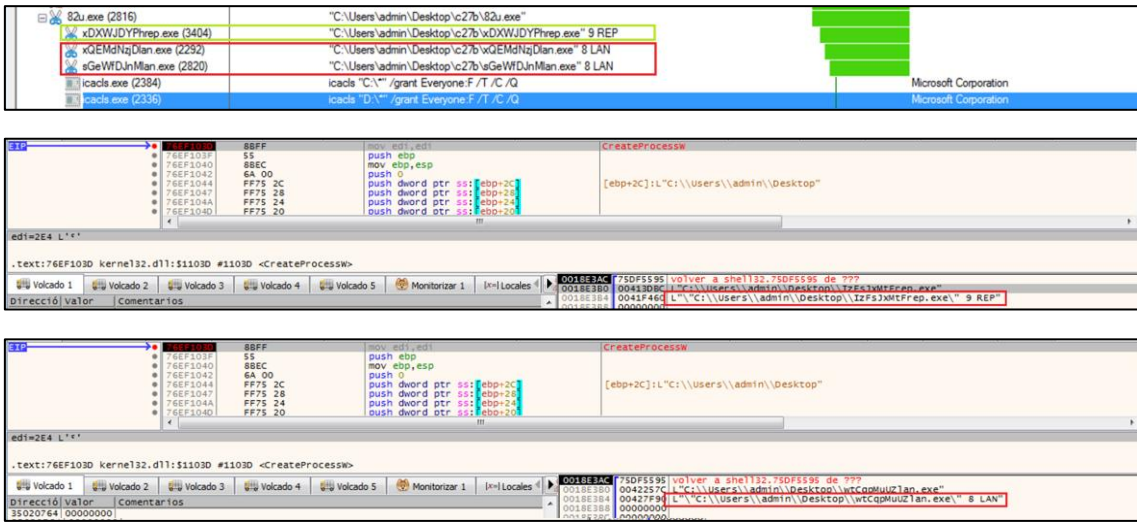


Figura 25. Procesos hijos de Ryuk ("8 LAN", "9 REP")

Por un lado, la instancia de Ryuk con "8 LAN" será la responsable de leer las entradas de la caché de la tabla del protocolo de resolución de direcciones (ARP) y, por cada una, enviar paquetes *Wake-on-LAN* de longitud 102 bytes (0x66 en hexadecimal). Para extraer la tabla ARP, se apoyará en la API *GetIpNetTable* (*iphlpapi.dll*) tal y como se muestra en la siguiente imagen.

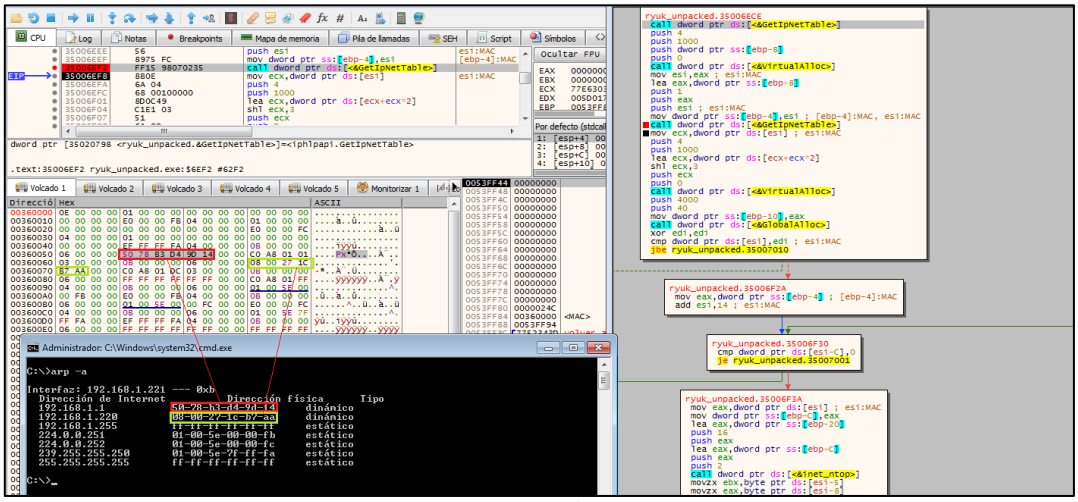


Figura 26. Recuperación de tabla ARP

Tras recuperar la tabla ARP comenzará a enviar los paquetes haciendo uso de *Winsock.* En la siguiente imagen se observa el envío de uno de estos paquetes mediante la API *sendto.*

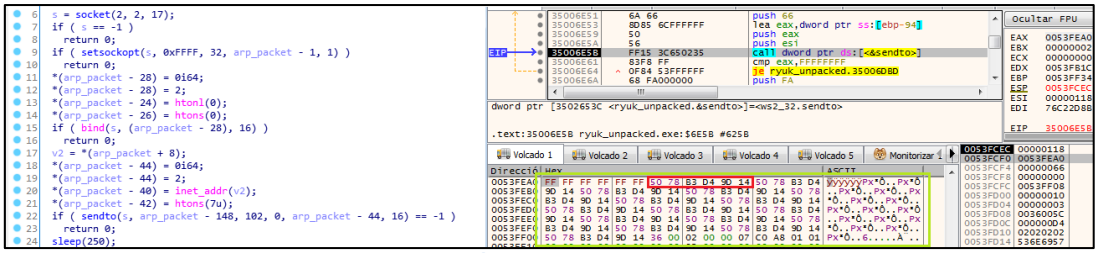


Figura 27. Envío de paquetes Wake-on-Lan

Los paquetes están compuestos por 6 bytes con valor 255 (FF:FF:FF:FF:FF:FF en hexadecimal) seguido de la dirección MAC del equipo objetivo.

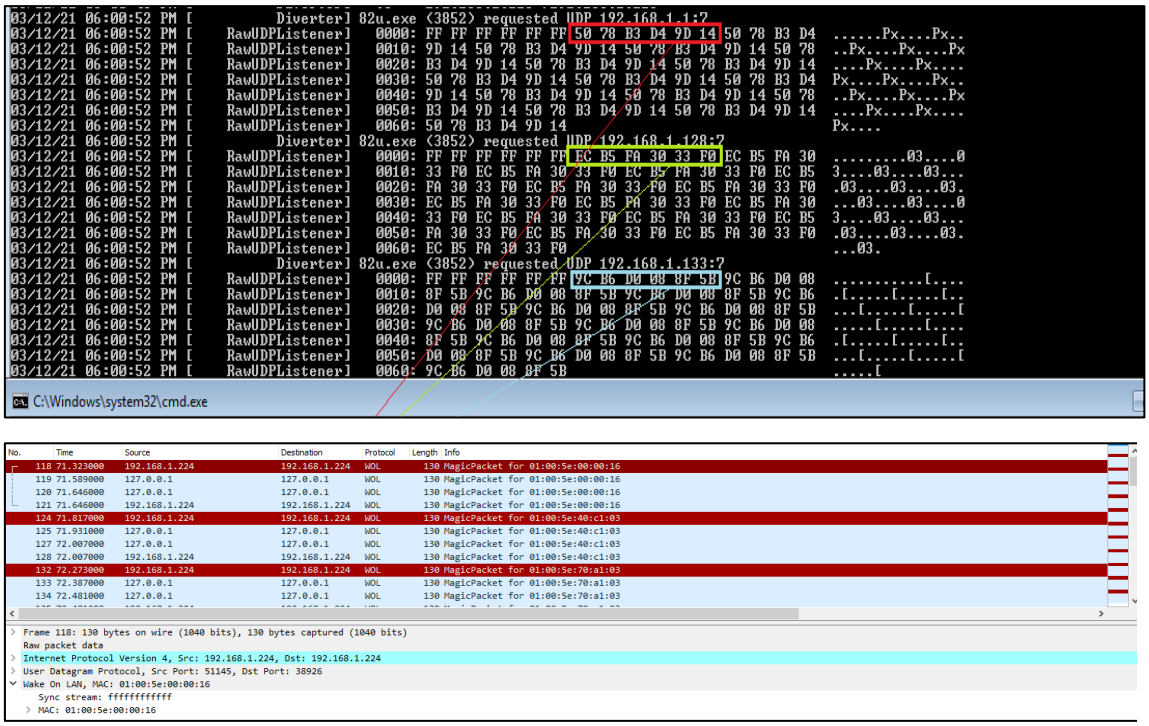


Figura 28. Tráfico *Wake on Lan*

El código dañino también intentará moverse lateralmente a otros hosts de la red comprobando, en primer lugar, la dirección IP asignada al sistema. Si esta dirección IPv4 pertenece a direccionamiento privado (comprobando los rangos de red 10.x.x.x, 172.16.x.x y 192.168.x.x) procederá a enviar paquetes *IcmpEchoRequest* usando la API *IcmpSendEcho* (en lugar de usar el comando *ping* nativo) para descubrir máquinas vecinas.

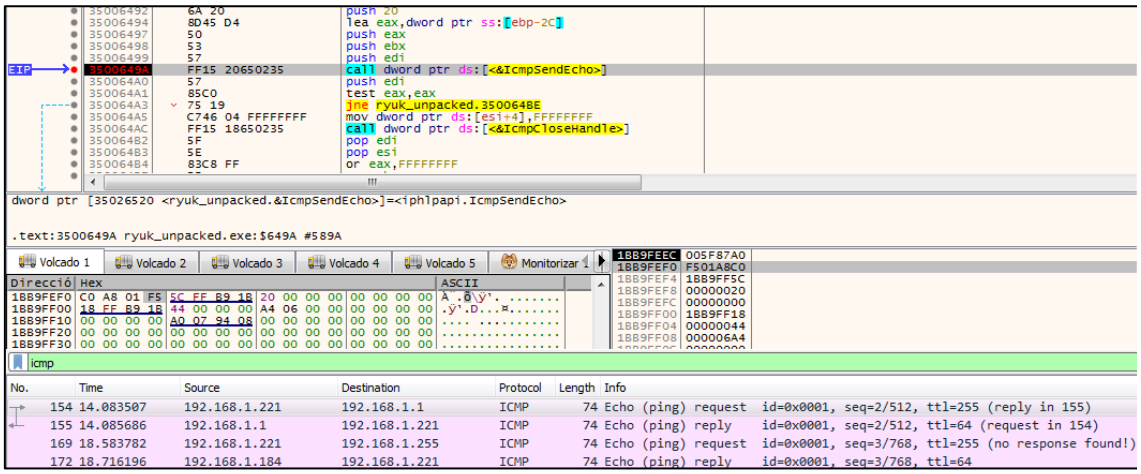


Figura 29. Envió de paquetes ICMP

Por cada máquina identificada comenzará a realizar peticiones SMB intentando acceder de forma iterativa a las unidades $ del equipo (desde la unidad A$ a Z$)

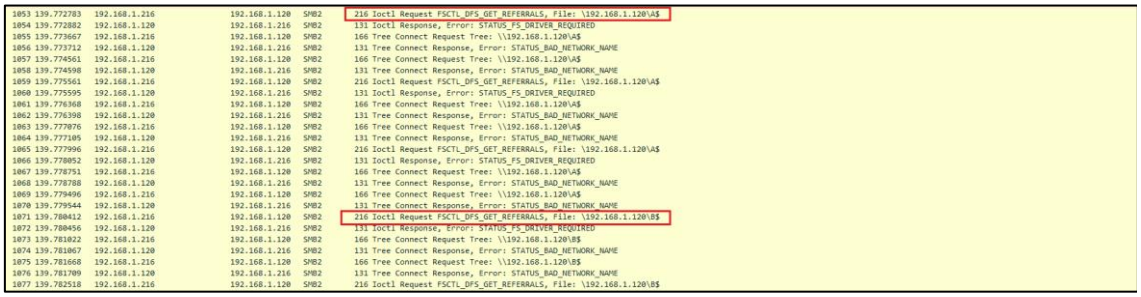


Figura 30. Identificación de shares $

En el caso de tener acceso a uno de estos recursos comenzará a cifrar los ficheros siguiendo el mismo *modus operandi* que el empleado para cifrar los ficheros locales.

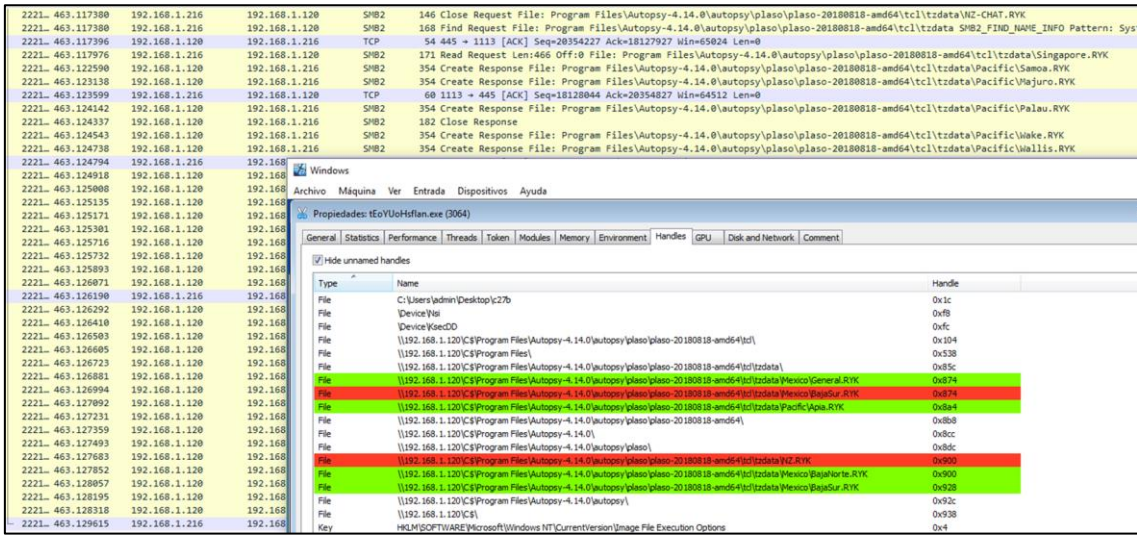


Figura 31. Cifrado unidad C$ remota

De forma paralela, si se comprueba que hay acceso a una máquina, la instancia de Ryuk que se está ejecutando con el argumento “9 REP”, intentará replicarse en la misma. Para ello, en primer lugar, creará una copia de Ryuk, que remitirá vía SMB, en el directorio “*c:\Users\Public*” de dicho host.

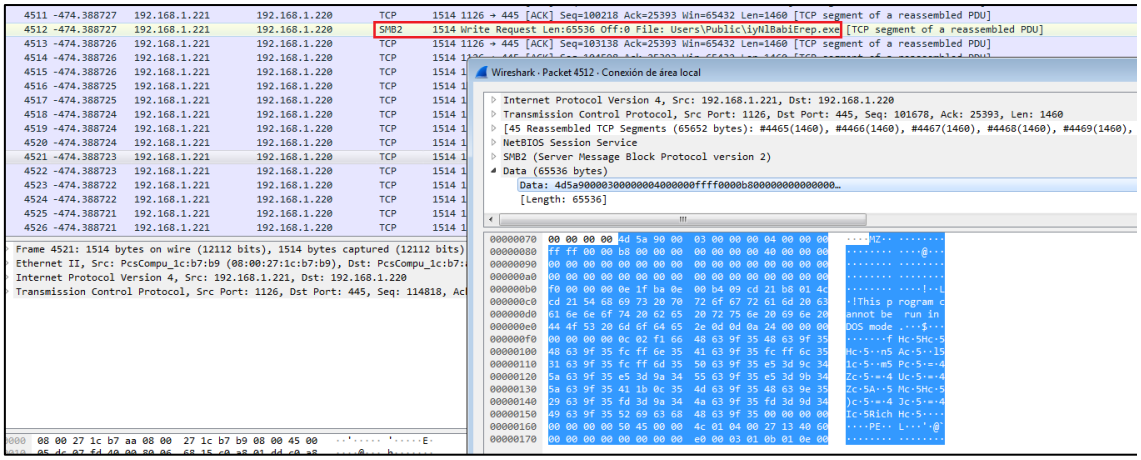


Figura 32. Envío de Ryuk vía SMB

Posteriormente, creará una tarea programada, por medio de la herramienta legítima *schtasks.exe,* para ejecutar la copia previamente creada y forzar así su ejecución inmediata.

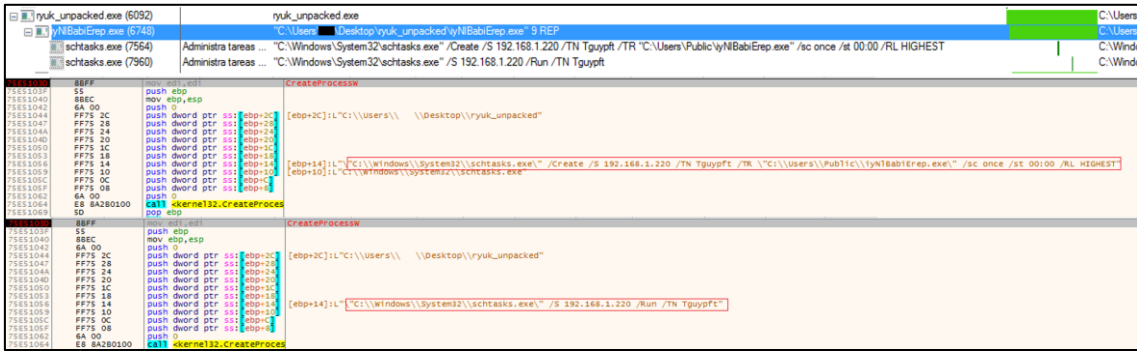


Figura 33. Subprocesos *schtasks*

Por tanto, por cada máquina remota comprometida se ejecutarán dos instancias de *schtasks.exe*:

schtasks.exe /Create /S 192.168.1.220 /TN Tguypft /TR "C:\Users\Public\iyNlBabiErep.exe" /sc once /st 00:00 /RL HIGHEST

schtasks.exe /S 192.168.1.220 /Run /TN Tguypft

La primera orden instruye al equipo comprometido (en este caso la máquina 192.168.1.220) a crear una nueva tarea (creada con nombre aleatorio) para ejecutar la copia de *Ryuk* (guardada en *C:\Users\Public\iyNlBabiErep.exe*) una vez a las 00:00. No obstante, esta hora hace referencia al primer segundo del mismo día por lo que nunca se ejecutará (obsérvense los argumentos: */sc once /st 00:00*). Será como consecuencia de la segunda instrucción, que la copia de Ryuk se ejecute de forma inmediata.

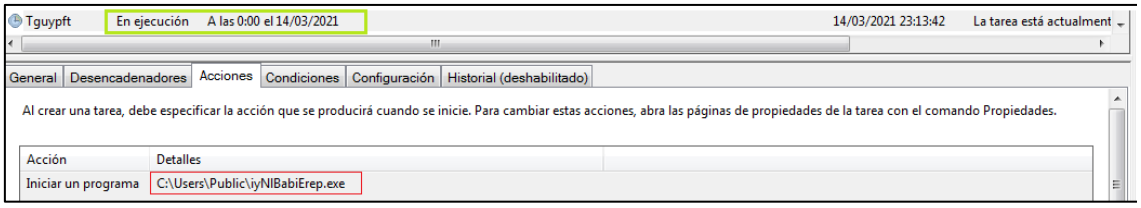


Figura 34. Tarea Programada

La ejecución de *schtasks* se realizará una vez se copie, vía SMB, el binario en el directorio "*c:\Users\Public*" del equipo remoto. Esta acción forzará la generación de tráfico RPC (135/TCP) para comunicarse con el *Task Scheduler* en la máquina remota.

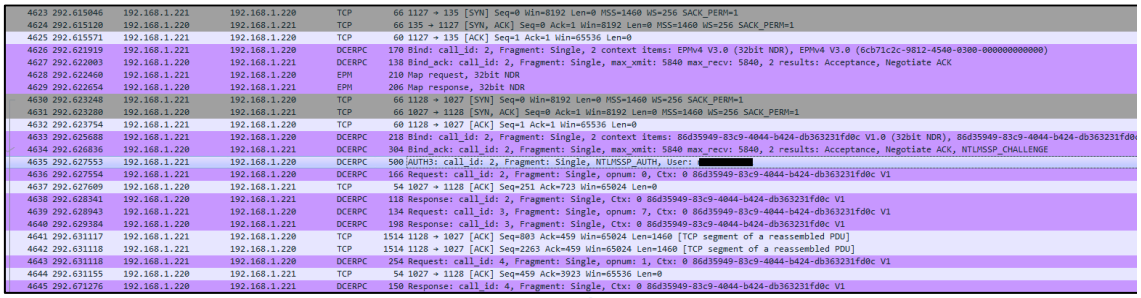


Figura 35. Tráfico RPC

8. DESINFECCIÓN

Puesto que la variante analizada no adquiere persistencia (únicamente crea en equipos remotos una tarea programada que será ejecutada una única vez), 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.

9. MITIGACIÓN

Debido a las capacidades de propagación de la muestra actual, la manera más estratégica para contener y reducir el impacto del *ransomware* es realizar un rotado de contraseñas en el dominio (incluyendo la cuenta KRBTGT). Téngase en cuenta que estas medidas posiblemente interfieran con el correcto funcionamiento del dominio o redes afectadas y puedan causar diversos imprevistos (por ejemplo, el reinicio de máquinas); no obstante, permitiría contener la amenaza de una manera resolutiva.

10. REGLAS DE DETECCIÓN

10.1 REGLAS DE YARA

```
import "pe"
rule ryuk_features{
meta:
		description = "Ryuk Ransomware"
		author = "CCN-CERT"
		version = "1.0"
		date = "2020-03-15"
		hash1 = "6cb642afe300c337338517faa49b60c94316ed1b"

strings:
		$x1 = "Vsekdar" nocase wide
		$x2 = "calimarimodunador.exe" nocase wide
		$x3 = "sufasey.exe" nocase
		$x4 = "Certum Trusted Network CA" nocase
		$x5 = "Application Delivery Solutions Ltd"

condition:
		uint16(0) == 0x5A4D and
		3 of ($x*) and (pe.number_of_sections > 6) and pe.number_of_signatures > 0 and
		filesize > 150KB and filesize < 350KB

}

rule ryuk_symbols
{
meta:
		description = "Ryuk Ransomware"
		author = "CCN-CERT"
		version = "1.0"
		date = "2020-03-15"
		hash1 = "6cb642afe300c337338517faa49b60c94316ed1b"

   condition:
      int16(0) == 0x5A4D and (pe.number_of_sections > 6) and filesize > 150KB and filesize < 350KB
      and pe.exports("Memories") and pe.exports("Roses") and pe.exports("Sofos")
}
```

Figura 36. Regla Yara

11. REFERENCIAS

[REF.- 1] Ryuk Ransomware (CERT FR)

https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-006.pdf

[REF.- 2] Malpedia: Ryuk

https://malpedia.caad.fkie.fraunhofer.de/details/win.ryuk

[REF.- 3] BazarLoader (Abuse)

https://bazaar.abuse.ch/browse.php?search=issuer_cn:Certum%20Trusted%20Network%20CA

[REF.- 4] Ryuk Ransomware Uses Wake-on-Lan To Encrypt Offline Devices

https://www.bleepingcomputer.com/news/security/ryuk-ransomware-uses-wake-on-lan-to-encrypt-offline-devices/

[REF.- 5] Wikipedia: Wake-on-LAN

https://en.wikipedia.org/wiki/Wake-on-LAN