

NETWALKER

AFECTACIÓN A MIGRACIONES ARGENTINAS

AUTOR: ING. PEDRO ALBIOL

REVISIÓN: LIC. BORGHELLO CRISTIAN (Director Segu-Info)

DISCLAMER:

Pedro Albiol deja expresamente asentado que el presente trabajo de investigación se realizó en base a fuentes públicas, no pretende ser un análisis forense exhaustivo y no debe tomarse como parte de ninguna investigación oficial en curso o futura.

Cristian Borghello y Segu-Info no representan ni aprueban la exactitud o fiabilidad de los datos, información u otro material proporcionado por el presente trabajo de investigación, por lo que se descarga explícitamente cualquier responsabilidad por el contenido de dichos datos, información y materiales proporcionados o divulgados a través de la presente.

Cristian Borghello y Segu-Info no serán responsables, por el plazo de duración de la divulgación, por los errores u omisiones en la Información y por el uso y los resultados del uso de esta Información.

Índice

| | |
|---|----|
| Introducción..... | 3 |
| Reclutamiento de <i>partners</i> en foro underground..... | 5 |
| Análisis de la fuga de datos (data leak) | 6 |
| Pruebas de compresión | 15 |
| Pedido de rescate | 17 |
| Breve análisis del malware | 18 |
| Pruebas del malware en nuestro laboratorio | 20 |
| Resultados preliminares de las pruebas realizadas | 23 |
| Informe de CrowdStrike | 25 |
| Mecanismo de cifrado según informe CrowdStrike..... | 26 |
| Nuestra interpretación del mecanismo de cifrado | 29 |
| Conclusiones..... | 32 |
| Futuras investigaciones..... | 33 |
| Referencias bibliográficas | 34 |

Introducción

Netwalker es un **fileless ransomware as a service (RaaS)** que ha afectado ya varias compañías y gobiernos desde su descubrimiento en 2019.

Es **Fileless** porque se ejecuta en memoria sin necesidad de un archivo físico en el disco: utiliza la técnica llamada **Reflective Dynamic-link Library Injection** para levantar en memoria una DLL, evitando así, las herramientas de detección/monitoreo.

Es un **Ransomware** porque cifra los archivos y solicita dinero a cambio de la llave necesaria para recuperar la información (ciber-secuestro). En caso de no pagar el rescate, a modo de presión y de garantizar una mayor probabilidad de cobro, la información secuestrada es publicada.

“**As a service**” significa que los desarrolladores no participan directamente. Solo brindan el soporte y plataforma para que otras personas (*partners/afiliates*) hagan utilización del mismo a cambio de un porcentaje (%) de las ganancias obtenidas. Los *partners* son los que realizan la infección como *insiders*, explotando una vulnerabilidad web o a través de técnicas de ingeniería social, spear phishing adjuntando un código malicioso (VBS o Powershell), etc.

Se estima que en tan solo 4 meses (entre marzo y julio) recaudaron aproximadamente 25 millones de dólares, transformándolo en el malware de mayor beneficio de la historia (Mcafee, 2020).

El malware cuenta con estricto “código de conducta” que prohíbe su utilización contra **Rusia** o la **Commonwealth of Independent States (CIS)**. Según la firma de seguridad CrowdStrike, se atribuye su creación a un grupo hacker de habla rusa (**Circus Spider Moniker**). Varias empresas de antivirus realizaron análisis del malware, su vinculación con otros ransomware y el seguimiento de las criptomonedas (ver figura 1).

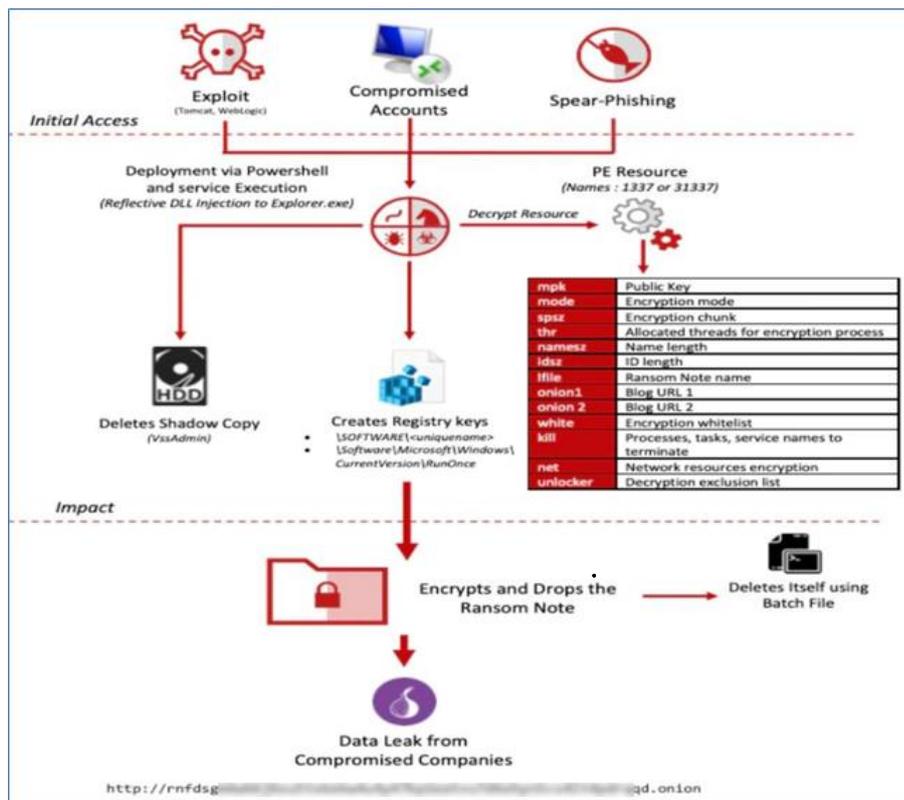


Figura 1

Esquema del malware.

Nota. Captura provista por McAfee.

Reclutamiento de *partners* en foro underground

Según el informe de McAfee, los desarrolladores del RaaS buscan reclutar gente con conocimiento en un foro de la Deep Web ofreciendo beneficios económicos por las infecciones con cobros exitosos (ver figura 2 y 3).

Luego de realizar una búsqueda en la Deep Web, creemos haber hallado el foro en cuestión. Sin embargo, cuenta con un estricto código de admisión. Habiendo agotado dicha instancia, continuamos la investigación por otros medios.

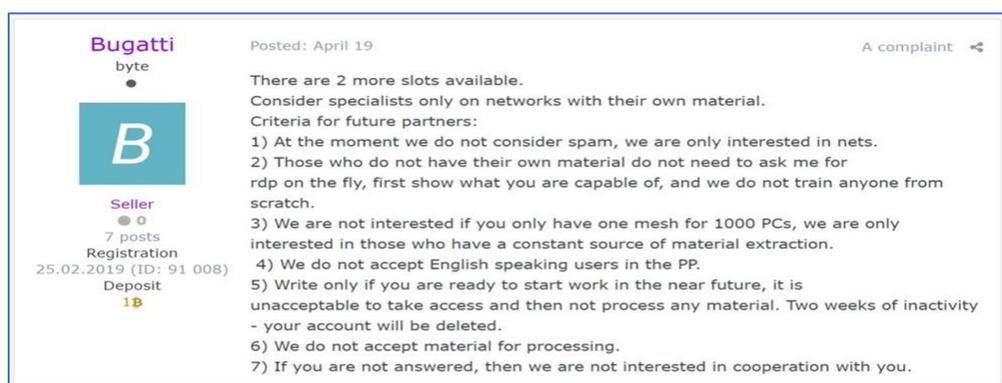


Figura 2

Captura reclutamiento de partners

Nota. Captura provista por McAfee. Reclutamiento durante 2019.

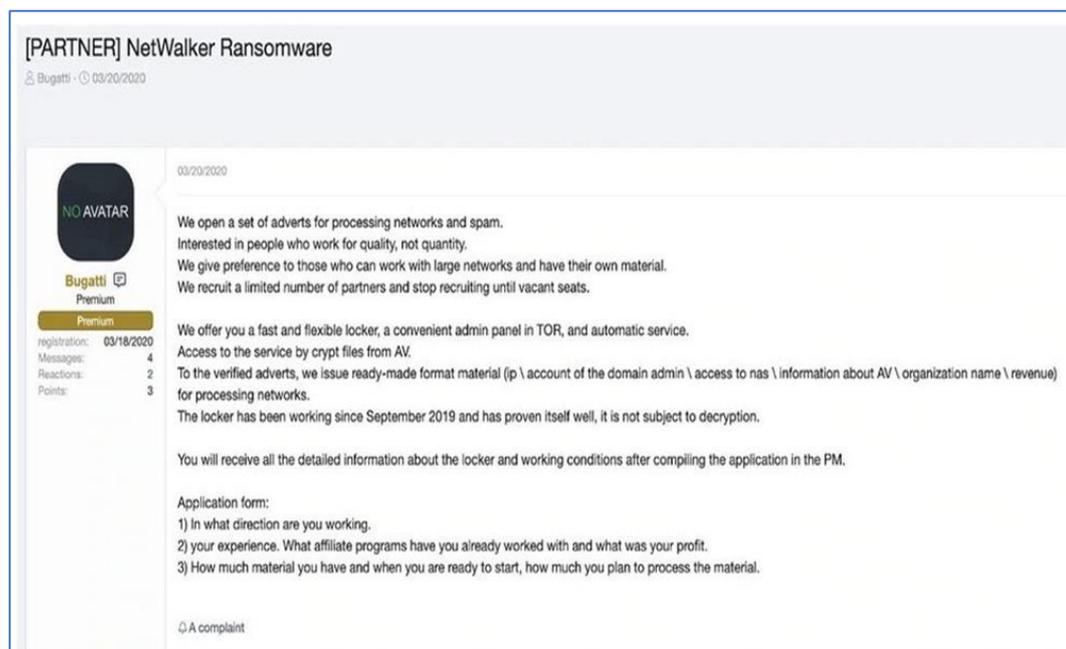


Figura 3

Captura reclutamiento de partners

Nota. Captura provista por McAfee. Reclutamiento durante 2020.

Análisis de la fuga de datos (data leak)

Puntualmente en el caso de Argentina, no podemos realizar un seguimiento o pericia ya que no contamos con acceso a la infraestructura, el log de eventos en el SIEM o la billetera digital para el pago del rescate.

A pesar de ello, el padre de la criminalística, **Edmon Locard**, decía en su **principio de intercambio: siempre algo se llevan y siempre algo dejan**. ¿Acaso es su principio aplicable a las TIC (Tecnologías de la información)?

La investigación comienza con el acceso al Onion del malware en la Deep Web donde se puede apreciar los diferentes *leak* de datos (ver figura 4).

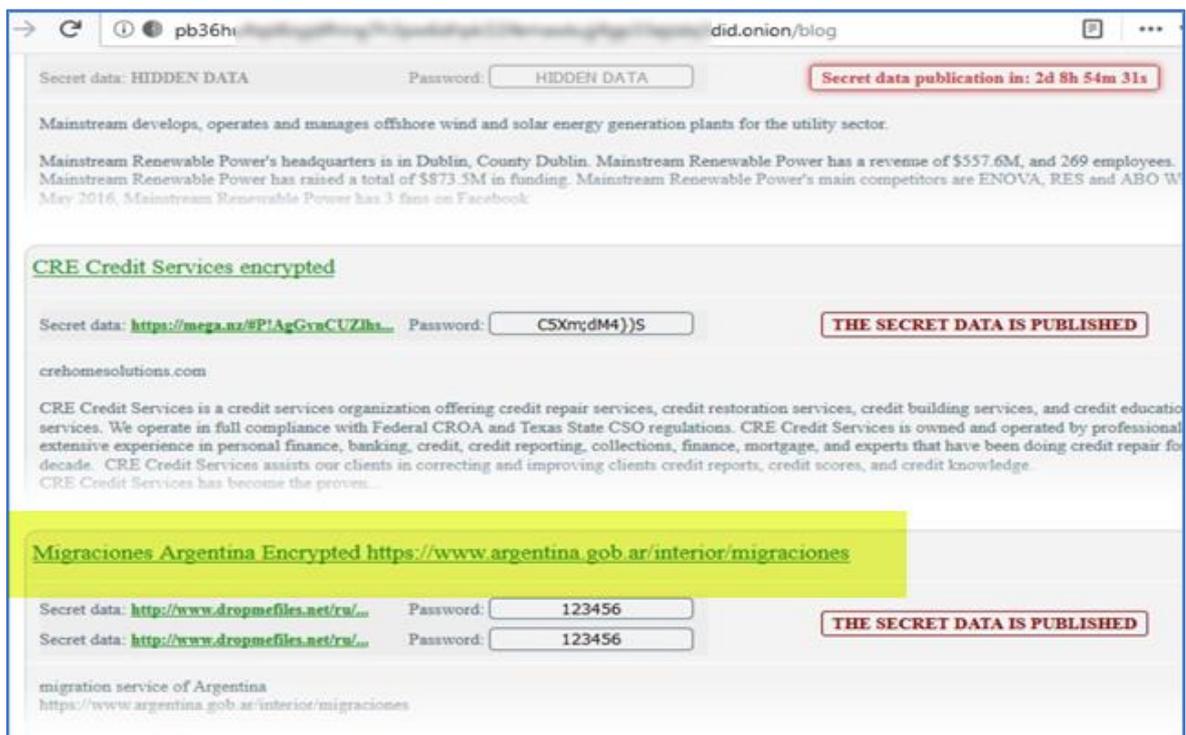


Figura 4

Captura de pantalla del sitio del malware (Onion)

Nota. En el sitio web de la Deep Web, se observan los leak de datos.

Lo primero que se aprecia es que tanto las claves, los links de descarga y formatos de compresión son diferentes en cada *leak*. Lo cual podríamos llegar a suponer que se trata de un proceso manual realizado por el *partner* y no forma parte de la automatización del RaaS (hipótesis a verificar).

Lo segundo que se observa es que a diferencia de otros *password* complejos, en el caso de Argentina, se le colocó una clave simple (“123456”), ya sea por una cuestión de rapidez o simplemente a modo de burla.

El link de descarga corresponde al sitio **DropMeFiles.net** que dispone de idioma inglés, ucraniano y ruso. Según los registros **Whois** figura registrado en Kiev Ucrania.

Ingresando directamente a la URL principal del sitio web, se obtiene la versión en inglés por defecto, inclusive entrando desde una IP rusa (VPN) y configurando el navegador en idioma ruso. Por lo cual quien subió el archivo podría haber seleccionado deliberadamente el idioma ruso a modo de atribuirle el hecho a Rusia (de la misma manera que utilizó un malware ruso sin intentar siquiera disimularlo). Ver figuras 5, 6 y 7.

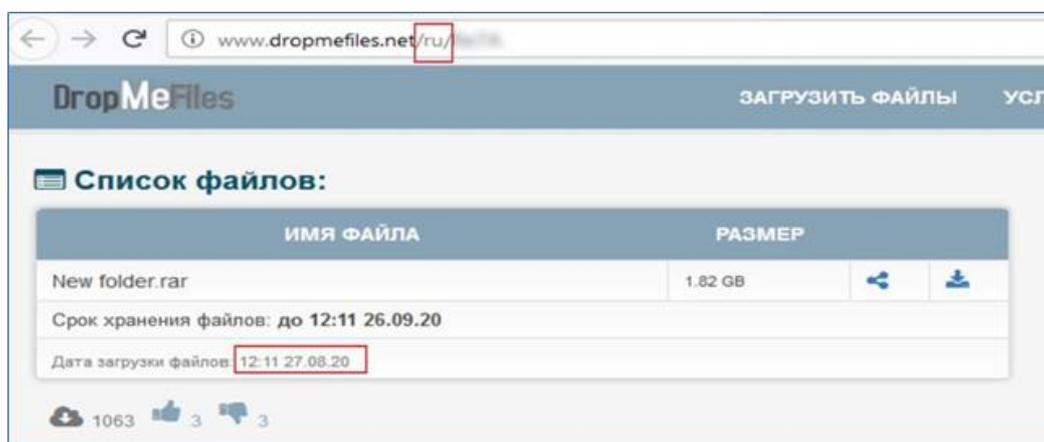


Figura 5

Captura de DropMeFiles.net

Nota. En rojo se aprecia el idioma y la fecha y hora de upload.

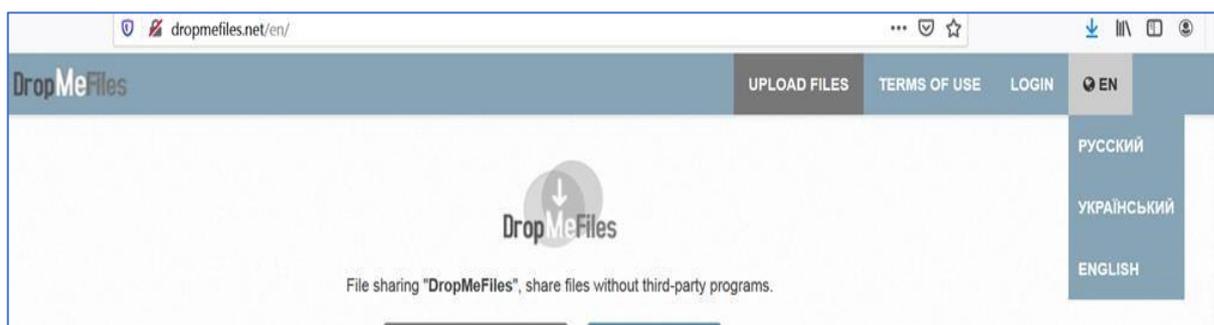


Figura 6

Captura de DropMeFiles.net

Nota. En el menú se aprecian los idiomas disponibles

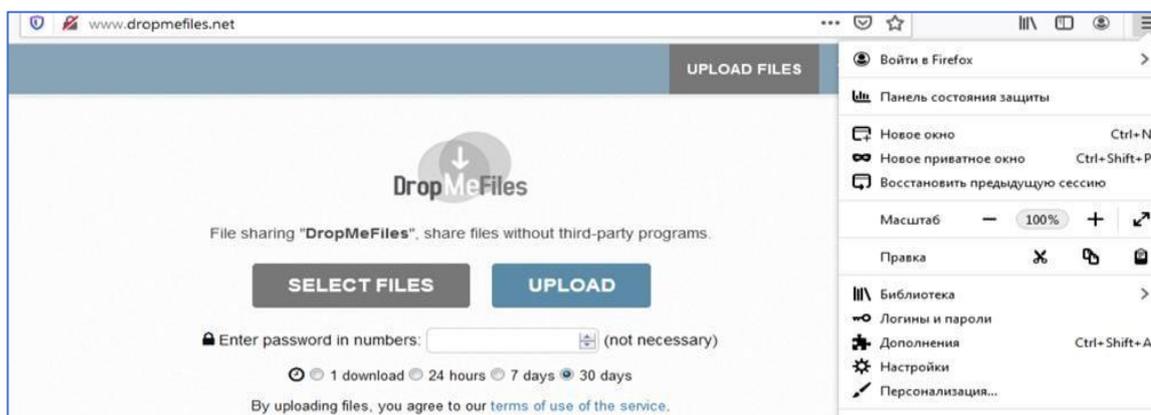


Figura 7

Captura de DropMeFiles.net

Nota. Idioma inglés por defecto

Respecto del horario de subida del archivo, figura **12:11 del 27/08/20**. Pruebas realizadas confirman que el horario corresponde a **UTC+3** (coincidente con el huso horario del host ucraniano), lo que en **UTC-3** (Horario Argentina) sería **06:11AM del 27/08/20**. Las pruebas también indican que la hora corresponde a la finalización y no al inicio del upload. Por lo tanto, **el ataque fue realizado previo a esta fecha/horario.**

El archivo se encuentra en formato RAR y es identificado por TrID en la familia 5.0 como se aprecia en la figura 8.

```
TrID/32 - File Identifier v2.24 - (C) 2003-16 By M.Pontello
Definitions found: 11624
Analyzing...

Collecting data from file: c:\users\test\Downloads\www.dropmefiles.net_New folde
r.rar
61.5% (.RAR) RAR compressed archive (v5.0) (8000/1)
38.4% (.RAR) RAR compressed archive (gen) (5000/1)
```

Figura 8

Captura de TrID

Nota. Identificado dentro de la v5.0

En el caso de 7ZIP, la figura 9 identificada al host Windows y método de compresión m0. Aclaremos que el método de compresión de RAR va desde *m0* a *m5*, donde 0 es el método sin compresión. Pruebas posteriores demostraron que la información mostrada por 7ZIP es incorrecta ya que se utilizó posiblemente el método *M2* (Fast).

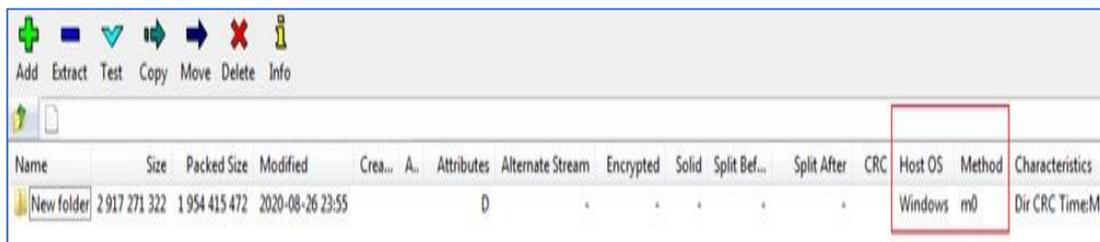


Figura 9

Captura de 7ZIP

Nota. Identifica host Windows y erróneamente método de compresión m0.

Respecto a la información indicada por WinRAR (ver figura 10), nuevamente confirma host Windows, versión utilizada de WinRAR familia 5.0 y un 66% ratio de compresión.

Una inspección del file signature (los primeros valores en hexadecimal del encabezado) nos confirma que se trata de una versión superior a 5.0. (figura 11)

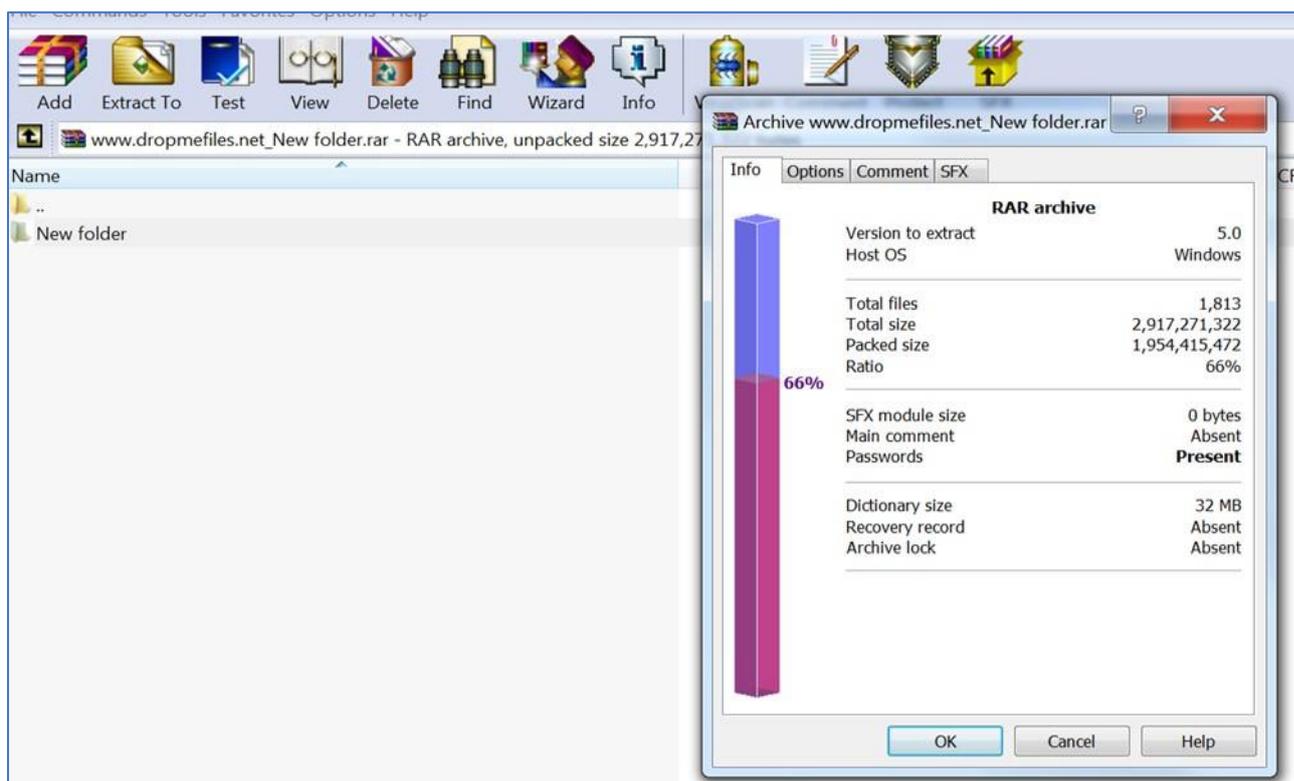


Figura 10

Captura de WinRAR.

Nota. Identifica host Windows y erróneamente método de compresión m0.

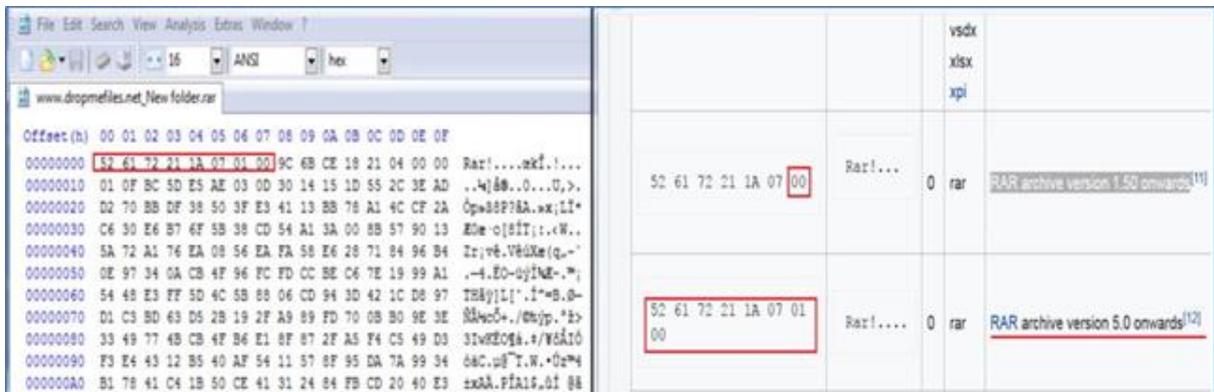


Figura 11

Captura file signatura

Nota: Confirma la identificación WinRAR familia v5.0.

¿Los archivos se comprimieron y luego se exfiltraron? ¿O, se exfiltraron y luego se comprimieron? Al abrir el archivo, se aprecia que mantiene la fecha original de los archivos, pero no la de las carpetas. Este comportamiento no es habitual del WinRAR que mantiene tanto la fecha de los archivos como de las carpetas. Esto al principio resultó confuso, pero luego se pudo entender el motivo: Las carpetas fueron actualizando su fecha durante la descarga de los archivos (fecha y horario de cada carpeta corresponde a la fecha de guardado del último archivo en dicha carpeta). Algo similar sucede con la carpeta “New folder” que contiene todo el árbol y por ende la última fecha al finalizar (figura 12).



Figura 12

Captura fecha “New folder”

Nota. Corresponde a la fecha del ultimo archivo descargado.

La distribución de tiempo en las fechas de creación de las carpetas, no se condice con el tamaño o cantidad de archivos contenidos en las mismas. Quizás se deba a fluctuaciones de la red durante la descarga. Observando la fecha de las carpetas, podemos conocer el tiempo que tomo la descarga (figura 14). A partir de ellos, podemos aproximar la **velocidad de descarga para 1.96GB en aproximadamente 13,5Mbits/s (1687KB/s)**, como indica la figura 15.

| Name | Size | Packed | Type | Modified | CRC32 |
|------|------|--------|-------------|-------------------|-------|
| ESC | | | File folder | 27-Aug-20 3:36 AM | |
| ES1 | | | File folder | 27-Aug-20 3:36 AM | |
| ES1 | | | File folder | 27-Aug-20 3:36 AM | |
| ES1 | | | File folder | 27-Aug-20 3:36 AM | |
| ES1 | | | File folder | 27-Aug-20 3:37 AM | |
| INF | | | File folder | 27-Aug-20 3:37 AM | |
| Inf | | | File folder | 27-Aug-20 3:38 AM | |
| REF | | | File folder | 27-Aug-20 3:42 AM | |
| VAI | | | File folder | 27-Aug-20 3:42 AM | |
| SUI | | | File folder | 27-Aug-20 3:43 AM | |
| RA | | | File folder | 27-Aug-20 3:43 AM | |
| CO | | | File folder | 27-Aug-20 3:45 AM | |
| 201 | | | File folder | 27-Aug-20 3:46 AM | |
| BO | | | File folder | 27-Aug-20 3:46 AM | |
| PEI | | | File folder | 27-Aug-20 3:47 AM | |
| PRV | | | File folder | 27-Aug-20 3:47 AM | |
| RIA | | | File folder | 27-Aug-20 3:47 AM | |
| REC | | | File folder | 27-Aug-20 3:49 AM | |
| REC | | | File folder | 27-Aug-20 3:49 AM | |
| TRV | | | File folder | 27-Aug-20 3:50 AM | |
| SO | | | File folder | 27-Aug-20 3:51 AM | |
| BA | | | File folder | 27-Aug-20 3:52 AM | |
| Arc | | | File folder | 27-Aug-20 3:52 AM | |
| BA | | | File folder | 27-Aug-20 3:52 AM | |
| REC | | | File folder | 27-Aug-20 3:54 AM | |
| REC | | | File folder | 27-Aug-20 3:55 AM | |
| REC | | | File folder | 27-Aug-20 3:55 AM | |

Figura 14

Captura de las carpetas.

Nota. Indica inicio 27/08/20 3:36AM - finalización 3:55AM. Aproximadamente 19 minutos de descarga.

Un dato importante es que la hora de modificación en el formato **RAR v5 mantiene el esquema UTC y no el horario local como era en la v4**. Esto permite saber la hora exacta de modificación de los archivos independientemente de su ubicación. Este hecho es fácilmente verificable. Al cambiar la zona horaria de la PC podrá observarse un cambio en las fechas mostradas por WinRAR.

Download/Upload Time Calculator

Result

Download or upload time needed is: **~19 minutes 21.481481481482 seconds**

File Size: Gigabytes (GB)

Bandwidth: Mbit/s

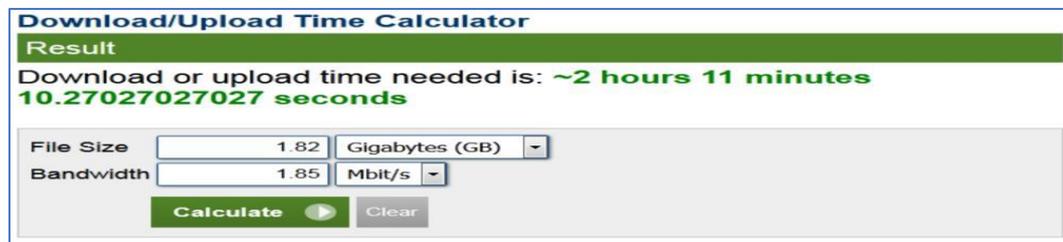
Figura 15

Velocidad de descarga

Nota. Cálculo aproximado de la velocidad de descarga en base al tamaño de los datos y el tiempo utilizado.

Esto nos permite establecer que quien descargó los archivos lo hizo el 27/08/20 entre las 3:36AM y las 3:55AM UTC-3 (horario argentino). Además, conociendo el horario de subida a DropMeFiles, sabemos que contó con 2 horas 16 minutos (diferencia horaria) para comprimir y realizar la subida al sitio mencionado.

Si bien el tiempo de compresión depende mucho del hardware utilizado, diferentes pruebas realizadas mostraron que el tiempo promedio en realidad no es significativo (Aprox. 5 minutos o incluso mucho menos). Suponiendo que comprimió y los subió los archivos inmediatamente, de forma apresurada a fin de reducir el tiempo de exposición y solicitar el rescate lo antes posible: podemos nuevamente estimar el **ancho de banda de subida aproximadamente en 1.9Mbit/s (237KB/s)** como nuestra la figura 16.



| Download/Upload Time Calculator | | |
|---|------|----------------|
| Result | | |
| Download or upload time needed is: ~2 hours 11 minutes 10.27027027027 seconds | | |
| File Size | 1.82 | Gigabytes (GB) |
| Bandwidth | 1.85 | Mbit/s |
| Calculate | | Clear |

Figura 16

Captura Estimación velocidad de subida

Nota. Cálculo de la velocidad de subida en base al tamaño de los datos comprimidos y el tiempo utilizado.

Solamente analizando los archivos, hasta aquí, pudimos establecer que el atacante contaba al menos con una conexión asimétrica de **más de 13,5Mbits/s (1687kB/s) downstream y más de 1,9Mbit/s (237KB/s) upstream**. Nuevamente, la velocidad puede ser afectada por varios factores, pero de ninguna manera puede haber utilizado una conexión por debajo de esos valores. ¿Acaso hay algún ISP con parámetros similares? Este indicio puede ser utilizado como un *fingerprint*.

A partir de los archivos “*Thumbs.db*”, podemos obtener un indicio de que el atacante efectivamente no perdió tiempo. Como muchos sabrán, estos archivos son generados y actualizados automáticamente por Windows. Se trata de un cache de las carpetas que contienen imágenes y es utilizado por el OS por razones de performance. Como estos archivos no fueron modificados, sabemos que el atacante no abrió las carpetas antes de comprimirlas. Ver figuras 17 y 18

Si bien es cierto que, con el conocimiento necesario, esta opción puede deshabilitarse desde el Group Policy, no es lo habitual. También podría plantearse que en realidad el atacante descargó los archivos desde un OS Linux, miró los archivos y luego los copio a

otros OS Windows a fin de comprimirlos. Esta última explicación resulta forzada y compleja. Posible pero no probable. Tal como dice el principio de Parsimonia: “*la explicación más simple suele ser la más probable*” (Ockham, 1280~1349).

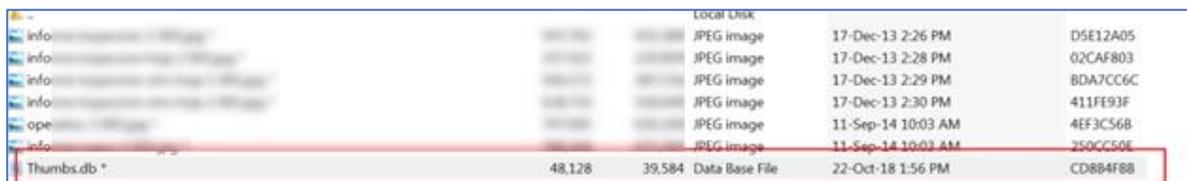


| | | | Local Disk | | |
|-------------|--------|--------|----------------|--------------------|----------|
| in | | | JPEG image | 17-Dec-13 2:26 PM | D5E12A05 |
| in | | | JPEG image | 17-Dec-13 2:28 PM | 02CAF803 |
| in | | | JPEG image | 17-Dec-13 2:29 PM | 8DA7CC6C |
| in | | | JPEG image | 17-Dec-13 2:30 PM | 411FE93F |
| op | | | JPEG image | 11-Sep-14 10:03 AM | 4EF3C56B |
| | | | JPEG image | 11-Sep-14 10:03 AM | 250CC50E |
| Thumbs.db * | 48,128 | 39,584 | Data Base File | 22-Oct-18 1:56 PM | CD8B4F88 |

Figura 17

Archivos *Thumbs.db*

Nota: Indicio de que el atacante no miró los archivos antes de comenzar



| | | | Local Disk | | |
|-------------|--------|--------|----------------|--------------------|----------|
| info | | | JPEG image | 17-Dec-13 2:26 PM | D5E12A05 |
| info | | | JPEG image | 17-Dec-13 2:28 PM | 02CAF803 |
| info | | | JPEG image | 17-Dec-13 2:29 PM | 8DA7CC6C |
| info | | | JPEG image | 17-Dec-13 2:30 PM | 411FE93F |
| ope | | | JPEG image | 11-Sep-14 10:03 AM | 4EF3C56B |
| info | | | JPEG image | 11-Sep-14 10:03 AM | 250CC50E |
| Thumbs.db * | 48,128 | 39,584 | Data Base File | 22-Oct-18 1:56 PM | CD8B4F88 |

Figura 18

Archivos *Thumbs.db*

Captura archivos Thumbs.db actualizados

Pruebas de compresión

Respecto al formato de compresión utilizado, se hicieron varias pruebas desde GUI (Graphic User Interface), CLI (Command Line Interface), con diferentes opciones y diferentes versiones de WinRAR a fin de constatar cual fue utilizado.

Aun así, no se logró obtener exactamente el mismo resultado (archivo comprimido). Este dato podría utilizarse como un tipo de *fingerprint* de quien realizó la compresión. Los resultados más parecidos se obtuvieron con la compresión **FAST**, lo cual nuevamente refuerza la idea del apuro. Ver figura 19.

| Name | Date modified | Type | Size |
|--|---------------|----------------|--------------|
| test_cli_m0(4min25seg)-ver5.91.rar | 11-Sep-20 | WinRAR archive | 2,849,643 KB |
| test_gui_fastest_encrypt-ver5.50.rar | 11-Sep-20 | WinRAR archive | 2,034,251 KB |
| test_cli_m2_encrypt(2min40seg)-ver5.50.rar | 11-Sep-20 | WinRAR archive | 1,909,444 KB |
| test_gui_fast_encrypt(2min4seg)-ver5.91.rar | 11-Sep-20 | WinRAR archive | 1,909,358 KB |
| www.dropmefiles.net_New folder.rar | 10-Sep-20 | WinRAR archive | 1,909,358 KB |
| test_gui_fast_encrypt(32b)-ver5.90.rar | 12-Sep-20 | WinRAR archive | 1,909,356 KB |
| New folder5.90beta3.rar | 12-Sep-20 | WinRAR archive | 1,909,355 KB |
| test_cli_m2_encrypt-ver5.61.rar | 12-Sep-20 | WinRAR archive | 1,909,289 KB |
| finaltest.rar | 12-Sep-20 | WinRAR archive | 1,909,250 KB |
| test_gui_fast_encrypt-ver5.50.rar | 11-Sep-20 | WinRAR archive | 1,909,250 KB |
| test_gui_fast_encrypt(7min15sec virtual 1 core)-ver 5.61.rar | 12-Sep-20 | WinRAR archive | 1,909,238 KB |
| test_cli_m3_encrypt(3min18seg)-ver5.91.rar | 11-Sep-20 | WinRAR archive | 1,901,618 KB |
| test_gui_normal_encrypt(3min23seg)ver5.50.rar | 11-Sep-20 | WinRAR archive | 1,901,542 KB |
| test_gui_normal_encrypt-ver5.91.rar | 11-Sep-20 | WinRAR archive | 1,901,542 KB |
| test_cli_m3(1min54seg)-ver5.91.rar | 11-Sep-20 | WinRAR archive | 1,901,525 KB |
| test_gui_normal(2min17seg)-ver5.50.rar | 11-Sep-20 | WinRAR archive | 1,901,450 KB |
| New folder5.40.rar | 12-Sep-20 | WinRAR archive | 1,477,961 KB |

Figura 19

Nota. La opción de cache de imágenes fue deshabilitada durante las pruebas a fin de no contaminar la evidencia.

En pruebas adicionales permitieron identificar que WinRAR solo genera archivos idénticos, en la misma versión, cuando no se utiliza la opción de cifrado (figura 20 y 21). Esto es porque WinRAR debe utilizar internamente algún tipo de función aleatoria vinculado al proceso de cifrado y por lo tanto no es posible saber exactamente que versión se utilizó. Aun así, por la similitud de los resultados obtenidos, se sospecha que se usó alguna variante de la versión 5.90. (figura 22)

| Nombre de archivo | MD5 | SHA1 |
|--|----------------------------------|--|
| New folder (2)_64_defecto.rar | 41499FCA7E2488FC7967551778E9E2AA | AC29F51233B978A183425845FB54B8D3BD42A8F6 |
| New folder (2)_64_defecto_again.rar | 41499FCA7E2488FC7967551778E9E2AA | AC29F51233B978A183425845FB54B8D3BD42A8F6 |
| New folder (2)_64_defecto_again_aga... | 41499FCA7E2488FC7967551778E9E2AA | AC29F51233B978A183425845FB54B8D3BD42A8F6 |
| New folder (2)_64_fast.rar | 41499FCA7E2488FC7967551778E9E2AA | AC29F51233B978A183425845FB54B8D3BD42A8F6 |
| New folder (2)_64_fast_again.rar | 18180C449D13A410DCF376E4A5F56FA1 | 8A0B9AE39238D00C6E6A7980EE4798FB242ED5C |
| New folder (2)_64_fast_again_again.rar | 18180C449D13A410DCF376E4A5F56FA1 | 8A0B9AE39238D00C6E6A7980EE4798FB242ED5C |
| New folder (2)_pass.rar | BD050C329C8BE5C52EBF7E23B97C9E | 7135DCEA40C33185C42C3DB1839899075568F5B9 |
| New folder (2)_pass_again.rar | 54F87F55FDBDCB768C09FEC187B9A75 | E77921FFAFC27C74C48183B4D0FA14922C2FE222 |
| New folder (2)_pass_again_again.rar | C7514FCD746C0BC4C14F695C4208669 | 3946F3A18533C753902036FD2940474825835FF |

Figura 20

Captura Hash idénticos y diferentes

Nota. Solo se obtuvieron archivos idénticos al utilizar la misma versión sin cifrado de archivos.

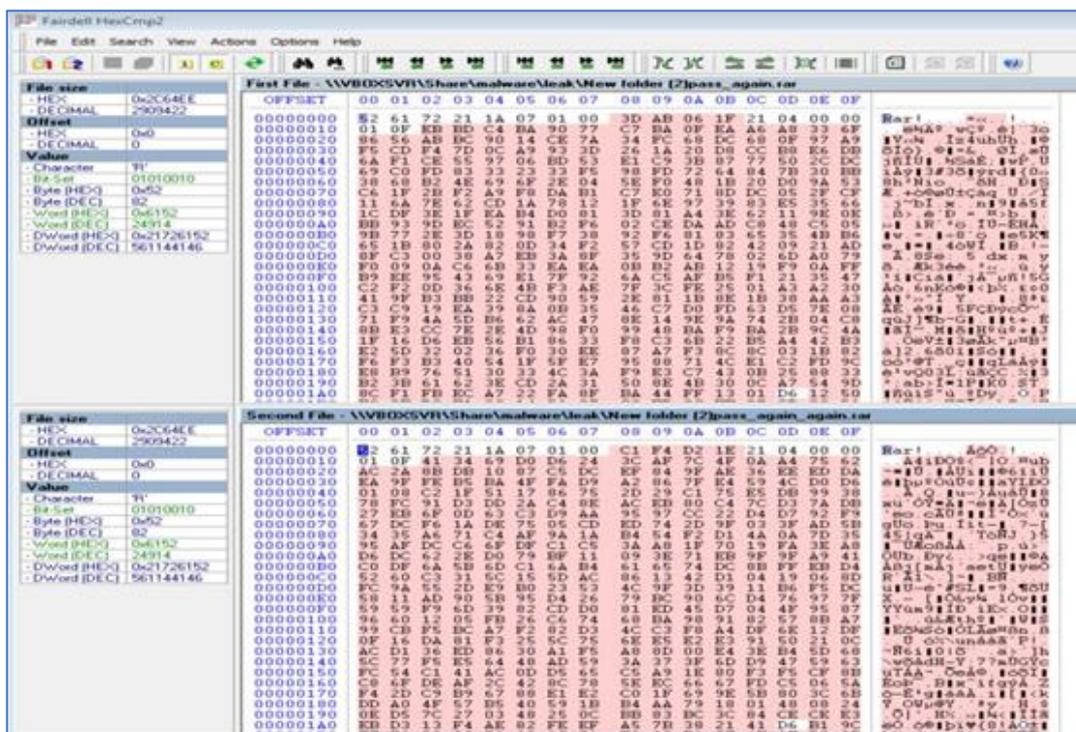


Figura 21

Captura comparación hexadecimal de una misma compresión con opción de cifrado

Nota. Con la opción de cifrado únicamente se mantiene idéntico el encabezado.

| | |
|---|--------------|
| test qui fast encrypt(2min4seq)-ver5.91.rar | 1,909,358 KB |
| www.dropmefiles.net_New folder.rar | 1,909,358 KB |
| test qui fast encrypt(32b)-ver5.90.rar | 1,909,356 KB |
| New folder.5.90beta3.rar | 1,909,355 KB |

Figura 22

Captura similitud de resultados en versión 5.90

Nota. Los resultados más similares se obtuvieron con la versión 5.90 y opción "fast".

Pedido de rescate

Según trascendió en los medios, el pedido de rescate fue de 2 millones de dólares y posteriormente de 4 millones ya que se incrementa a medida que se vencen los plazos (figura 23). Conociendo la situación de Argentina, podemos preguntarnos: ¿realmente se pensaba tener éxito en dicha solicitud? **El atacante no estaba familiarizado con nuestro país o su intención no era económica.**



Figura 23

Captura del pedido de rescate

Nota. El monto de los rescates se incrementa a medida que se vencen los plazos.

Breve análisis del malware

A partir de una muestra del malware **Netwalker** se puede analizar el formato **PE** (Portable Executable). Los resultados denotan que la muestra fue compilada en 32 bits para mayor compatibilidad y por otra parte se observa una protección VMProtect (figura 24). Esto último, significa que el código está cifrado y se ejecuta en una VM (protect) con su propio set de instrucciones, lo cual dificulta un análisis de ingeniería inversa (figura 25).

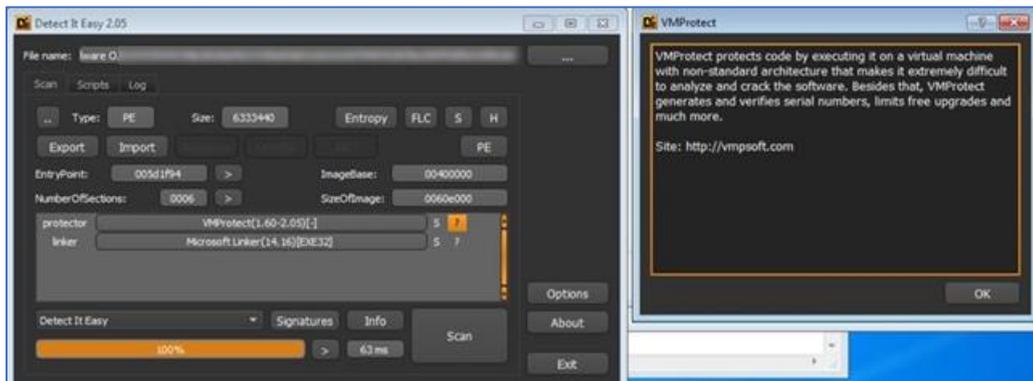


Figura 24

Nota. Compilado en 32 bits para mayor compatibilidad y una protección VMProtect.

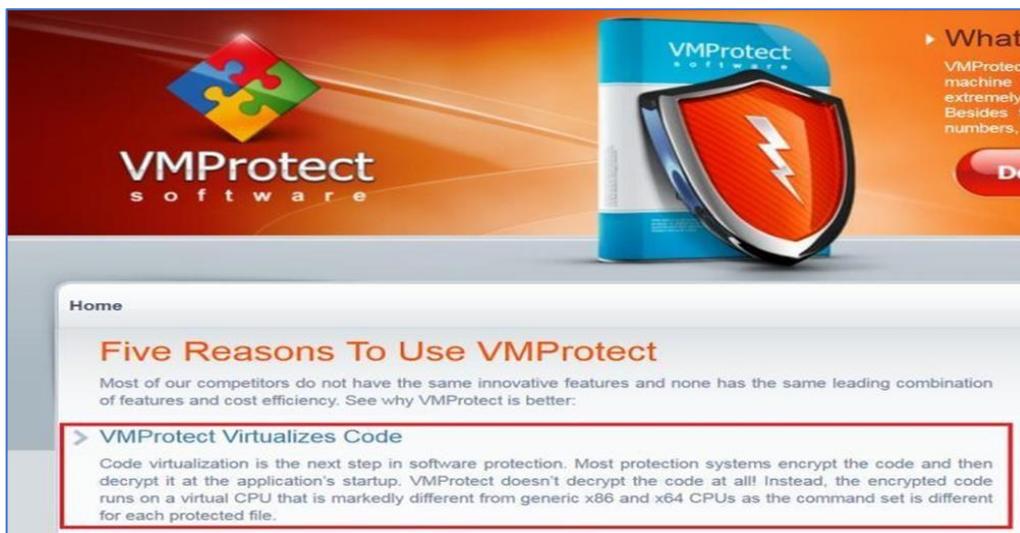


Figura 25

Captura del website de VMProtect

Nota. VMProtect es utilizado por desarrolladores para proteger su software.

Al simular la muestra en la plataforma *any.run* (figura 26) se aprecia el mecanismo de cifrado de archivos, pero no la exfiltración de datos. ¿Por qué? Lo veremos más adelante.



Figura 26

Captura del website *any.run*

Nota. Diferentes muestras de *Netwalker* en la plataforma *any.run*

Según la plataforma *any.run*, además de destruir los *shadow copy* de los archivos (para evitar una posible recuperación de los archivos), también realiza un acceso a las credenciales guardadas en el navegador Chrome (figura 27). Si bien no es recomendable guardar credenciales en el navegador, lamentablemente debe ser una práctica habitual. Por lo cual, quien haya sufrido un ataque de este tipo, como de cualquier otro estilo, siempre es recomendable el cambio de credenciales.

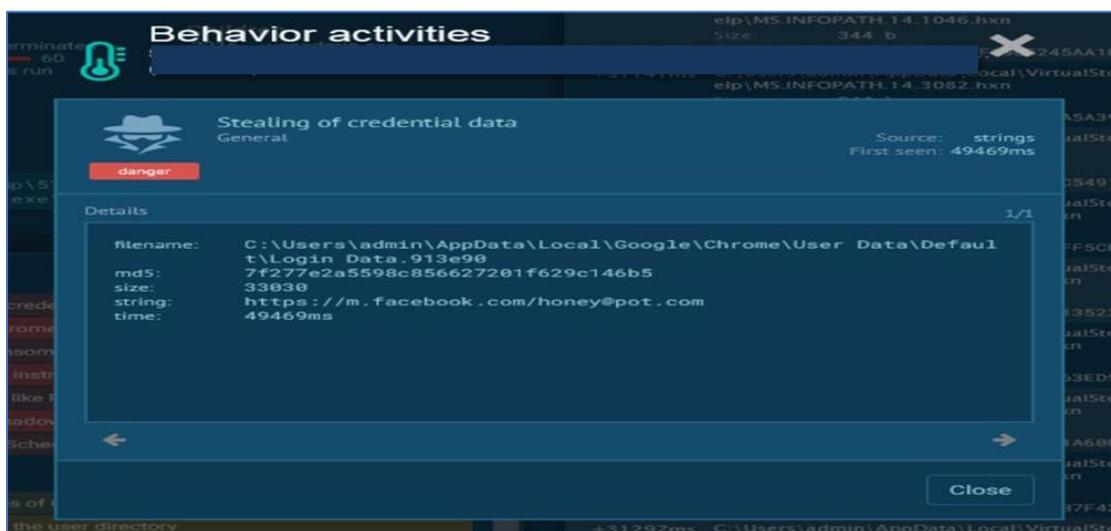


Figura 27

Captura de acceso a credenciales de navegador en plataforma *any.run*

Nota. No es recomendable almacenar credenciales en el navegador.

Pruebas del malware en nuestro laboratorio

Al ejecutar la muestra en el laboratorio, se escucha el temeroso crujido del disco arrojando archivos a la licuadora de cifrado. El pánico está presente al pensar que el malware pudiera saltar de la virtual machine al host. Afortunadamente, no es lo suficientemente avanzado en este aspecto.

Al finalizar el cifrado, el malware nos muestra en el Notepad la nota de rescate (figura 28). La misma indica que debemos ingresar a un sitio en la Deep Web y colocar un código que nos provee. ¿El precio del rescate? Un millón de dólares.

Al realizar un seguimiento de los movimientos sobre la billetera digital que nos asignaron para el pago, se aprecia que no existen movimientos (figura 29). Recordemos al lector que en la tecnología de Blockchain todas las transacciones son de público acceso y son almacenadas en la cadena de bloques desde sus inicios.

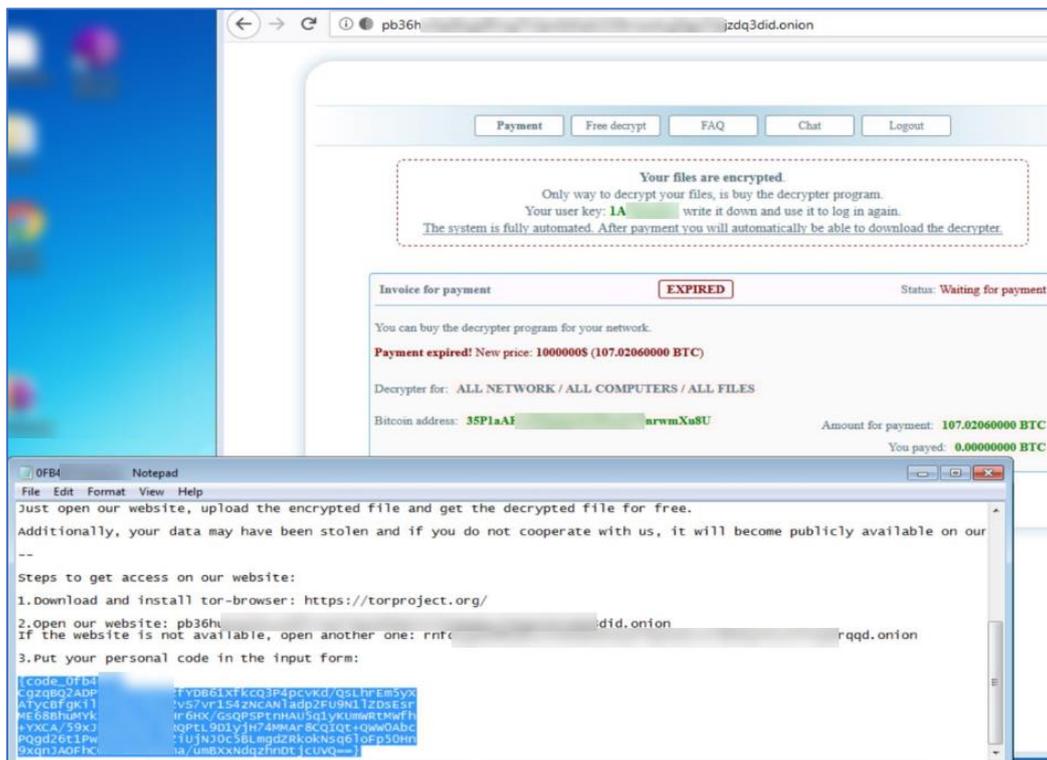


Figura 28

Captura de la nota de rescate

Nota. Nota de rescate con las instrucciones a seguir para el pago del secuestro.

| | | | | |
|--|------------------|---------|---|------------------|
| BTC Address | 3SP1aAFn | SnrwXU8 | # Website Appearances | 0 |
| Wallet Name | - | | Last Transaction IP | - |
| Current Balance | 0.00000000 = \$0 | | Total Received | 0.00000000 = \$0 |
| # Transactions | 0 | | # Output Transactions | |
| First Transaction | | | Last Transaction | |
| Last Known Input | None | | Last Known Output | None |
| Repeated Inputs From (50 most recent transactions) | None | | Repeated Outputs To (50 most recent transactions) | None |

Figura 29

Captura de la cuenta donde transferir los bitcoin

Nota. La cuenta asignada no contiene transacciones.

Respecto al *leak* de datos, a diferencia de migraciones, no existe ni captura de pantalla ni link de descarga. ¿Por qué?

Al mismo tiempo que se ejecutaba la muestra en el laboratorio, se realizaba un monitoreo la red con Wireshark. Esperando alguna señal pudiera salir por la red o alguien pudiera llegar a ingresar desde el exterior, lo único que sucedió fue haberle ganado al solitario a **Netwalker** (figura 30).

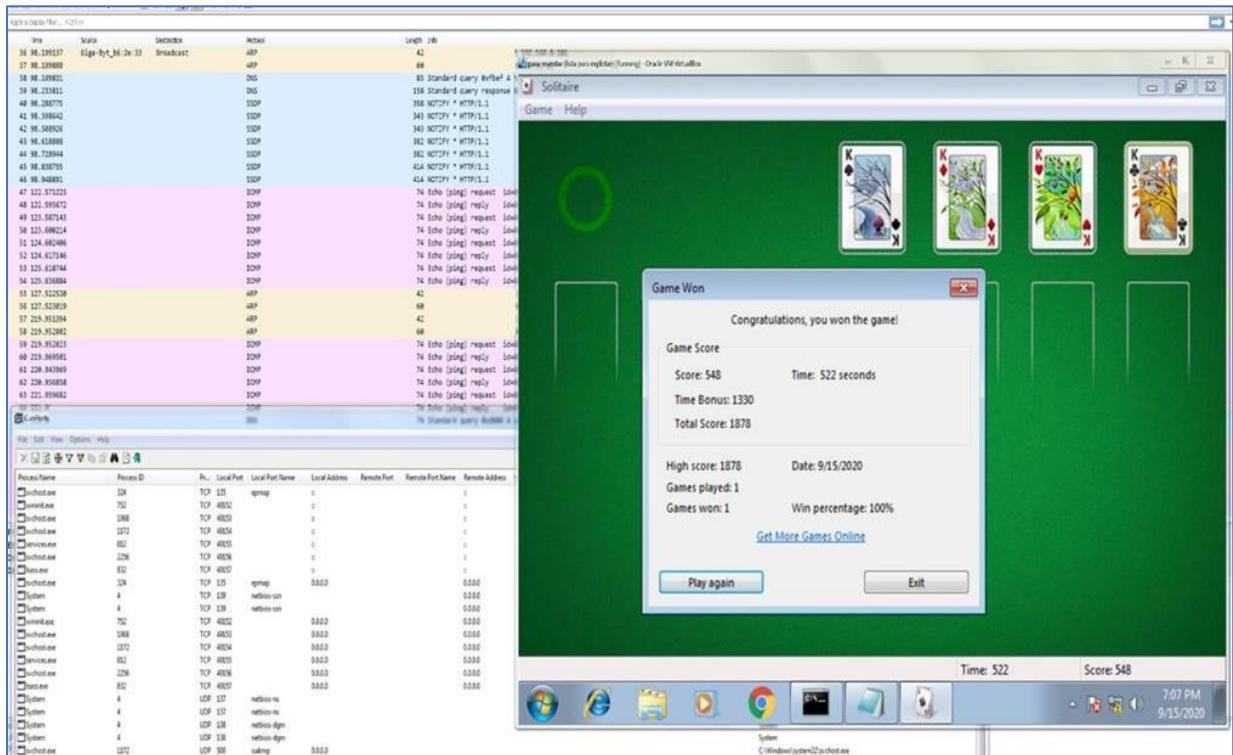


Figura 30

Captura de análisis de paquetes con Wireshark

Nota. Sin movimientos relevantes en la red.

A fin de tener una segunda nota comparativa, se recuperó la VM desde un *snapshot* y se ejecutó el malware nuevamente. Al comparar ambas notas, se aprecia que el cuerpo del “*Personal code*” variaba (posiblemente debido a parámetros aleatorios del cifrado), pero no así el primer parte del texto (figura 31). ¿Es la parte fija un ID para identificar al *partner*? No, no lo es. Esta pregunta se responde más adelante

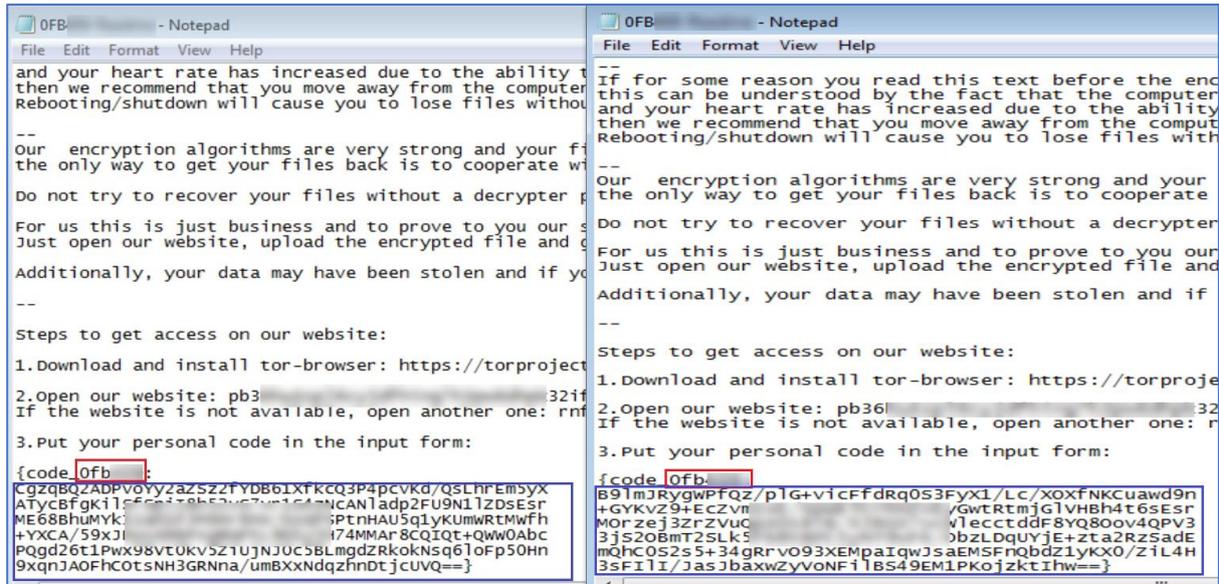


Figura 31

Comparativa entre 2 notas

Nota. Se observa que la parte del recuadro color rojo se mantiene mientras que el recuadro azul varía.

Resultados preliminares de las pruebas realizadas

A lo largo del trabajo, hemos estado hablando del atacante y de los desarrolladores, pero es importante aclarar que en realidad se trata de una formalidad lógica, ya que no sabemos si cada parte se conforma por una o varias personas.

Habiendo aclarado dicho punto y a partir de las pruebas realizadas, podemos entonces enumerar algunos resultados preliminares.

Primero, la llave de cifrado de los archivos no se envía a ningún C&C por la red. ¿Es posible que la llave por la cual debemos pagar, sea la misma que nosotros mismos entregamos en el “*personal code*” a los secuestradores?

La llave podría estar escondida en dicho texto, cifrada de manera simétrica o asimétrica. En caso de ser simétrica, ingeniería inversa podría generar un algoritmo reversible. Aunque, lo más probable es que trate de un esquema llave pública / privada. Considerando que no contamos con una computadora cuántica de 53qbits como Google o 40mil profesionales como la NSA, podemos considerarlo irreversible.

Segundo, los análisis y las pruebas realizadas permitieron identificar que el malware no exfiltra ningún dato. **Por lo tanto, el ataque que sufrió migraciones se trató de un ataque combinado. En la primera etapa se produjo el robo de datos (ya sea con un *insider* o con técnicas de hacking convencionales). Durante la segunda etapa, se lanza el ransomware en la retirada y es utilizado como mecanismo de extorsión a través de su plataforma “*as a service*”.**

De este mismo punto, se desprende que el atacante tiene contacto con los desarrolladores para subir a la plataforma las capturas de pantalla y los links de descarga. Por lo tanto, la evidencia indica que los desarrolladores y los *partners* están estrechamente vinculados. Esto no necesariamente significa que se conozcan ya que pueden interactuar anónimamente por un foro o plataforma. Tampoco debe descartarse la posibilidad que el *partner* y los desarrolladores sean el mismo individuo.

Recapitulando el proceso completo: los desarrolladores ofrecen el servicio en un foro donde se produce el primer contacto con el *partner* al quien le entregan el ransomware. El *partner* vulnera el sistema de la víctima con técnicas de hacking (phishing, vulnerabilidades web, intrusiones en la red, reconocimiento, movimientos laterales, elevación de privilegios, entre otros). Seguido a ello, realiza el robo de datos y los sube a algún sitio web.

Posteriormente, ejecuta el ransomware en sistema de la víctima e informa los datos al RaaS a fin de solicitar su porcentaje de la extorsión. Por su parte, la víctima se ve forzada a ingresar a la plataforma y negociar el rescate de sus datos. (ver figura 32).

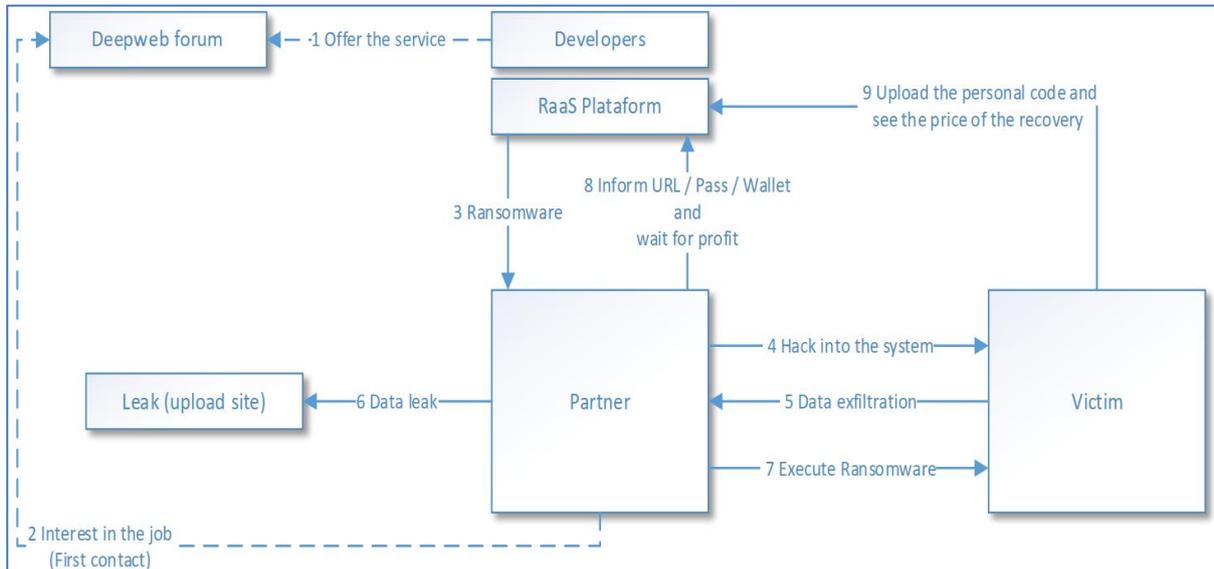


Figura 32

Diagrama paso a paso del ataque

Nota. El diagrama resume los pasos del ataque incluyendo tanto al RaaS como al Partner.

Imaginemos por un instante ser los desarrolladores y ver en una consola el listado de rescates (algunos exitosos, otros de investigadores o incluso agencias gubernamentales persiguiendo a los criminales). Por otro lado, tenemos un listado de *partners* informando los links de las filtraciones y solicitando su parte de las ganancias. ¿Cómo hacen los desarrolladores para saber que víctima corresponde a cada *partner*? Posibles opciones:

- 1) Cada ransomware está hecho a medida para cada *partner*. Con un ID interno o con una llave específica que luego es identificable en el personal code.
- 2) El *partner*, luego de lanzar el ransomware, de la misma manera que robó los documentos, descarga el archivo de rescate e informa el personal code al RaaS.
- 3) El *partner* forma parte del mismo equipo del RaaS (developers) y por lo tanto no es necesario ninguna coordinación.
- 4) Coordinan la fecha y hora del ataque. Esta opción es la menos probable por no ser eficiente y ser poco escalable.

Informe de CrowdStrike

Continuando la investigación, nos encontramos con un informe técnico de la firma CrowdStrike que brinda información valiosa sobre el malware y que compartiremos a continuación.

El informe aclara que el malware, a fin de evitar persecuciones por fuerzas de seguridad locales, no funciona en caso de detectar una configuración de teclado en los idiomas ruso / ucraniano / bielorruso / kazajo. Esto es lo que se conoce como **kill switch** y le garantiza cumplir con sus políticas descriptas al principio del presente trabajo.

El ID que habíamos identificado dentro del “*Personal code*” NO es el ID del *partner*. Se trata de un “**Infection ID**” que es calculado con un **HMAC-SHA256** basado en datos del sistema: **Computer Name** y **GUID** (Globally Unique Identifier). Este punto lo verificamos cambiando el nombre de la máquina y volviendo a ejecutar el malware, generando así, un id diferente.

Características destacables del malware:

- Cuenta con un archivo de configuración en formato JSON donde almacena parámetros pre configurables del comportamiento del ransomware. Este archivo cifrado con RC4 se accede en tiempo de ejecución utilizando una *hardcoded key*.
- Modifica el registro de Windows (según la versión del OS) con el fin de eludir el UAC (User Account Control). Esto significa que intenta una elevación de privilegios durante la ejecución.
- Contiene una lista configurable de servicios, procesos y tareas a matar.
Ejemplo: veeam*, protecciones Fortinet, psexec, etc.
- Borra los *shadows copies* de los archivos (evita recuperación de archivos).
- Utiliza una cantidad configurable de threads para un cifrado más veloz (entre 1000 y 1500 hilos).
- Busca el proceso *explorer.exe* ($CRC32 = BE037055$) para obtener los *Users Security Tokens*. Los mismos son utilizados para autenticar con los *Network Shares* en caso de no poder acceder directamente.
- La configuración incluye la opción de realizar un *discovery* de shares en la red.
- Cifra tanto archivos locales como de red (para la red utiliza *threads* adicionales).

- Además, crea un listado de todos los archivos que no pudo acceder (ej. bloqueados por otro proceso/usuario). Al finalizar el cifrado de los archivos accesibles, intenta matar los procesos que bloqueaban los archivos inaccesibles y reintenta el cifrado nuevamente.
- El cifrado se realiza con un algoritmo simétrico (ChaCha) + asimétrico de los más veloces dentro de curvas elípticas (Curve25519)
- Posee 3 modos configurables para el cifrado de archivos:
 - 0 = Cifra 3 partes del archivo.
 - 1 = Cifra una parte al principio del archivo.
 - 3 = Cifra el archivo completo.

Mecanismo de cifrado según informe CrowdStrike

El malware crea un par de llaves publicas/privadas para el host (**priv0/pub0**) utilizando el algoritmo **Curve25519**. Para dicho proceso, llama 32 veces la función *RtlRandom* que genera números pseudo-aleatorios a partir de la hora del sistema como *seed*. Estas llaves son almacenadas en el registro de Windows en la ubicación “*HKLM\Software\<INFECTION ID>*” y “*HKCU\Software\<INFECTION ID>*”. Ver figura 33

Este punto lo verificamos en el laboratorio realizando una captura del registro de Windows antes y después de la infección como se aprecia en la figura 34.

```

000000  60 7F 1E 52 8E AC E9 7D 72 0D D6 AB 1D 2D DE A8  `..R.-é}r.Ö«.-P"
000010  59 55 55 E9 52 EF 86 22 95 EA E5 7C 9D E0 45 1C  YUUErî.".êâ|.àE.
000020  23 4E 73 53 01 AF 4B A6 77 83 67 56 FA D4 F5 E4  #NsS.¯K|w.gVúÔöä
000030  1C A4 52 36 32 9A 95 60 BF 43 14 A1 09 43 D7 17  .µR62..`¿C.¡.C×.
000040  D2 54 F3 0A B8 2D AD 56 BD 63 CA 27 C8 FB BB 88  ÒTó. .-V½cÊ'Èû».
000050  84 9C 55 1F C6 AD 81 62 3C 5F 9B 29 FD 0E 6C AA  ..U.Æ..b<_.)ý.lª
000060  56 63 6B 9F DA 5A 59 68 23 BB 40 15 F6 61 41 4F  Vck.ÚZYh#»@.öaAO
000070  D0 37 54 AF E7 25 EC 5C 89 70 93 EB 76 4A D5 4A  Ð7T¯ç%i\ .p.ëvJÖJ
00008C  2F 71 A2 2E 6E 08 2D A5 D3 7D 92 8C          /qç.n.-¥Ó}..

Host-specific public Curve25519 key (pub0)
CRC32 sum of pub0
One-time public Curve25519 key (pubr)
ChaCha nonce
HMAC-SHA256 from ChaCha encryption
ChaCha Encrypted data (priv0)

```

Figura 33

Captura del informe crowdstrike

Nota. Datos almacenados en el registro de Windows.

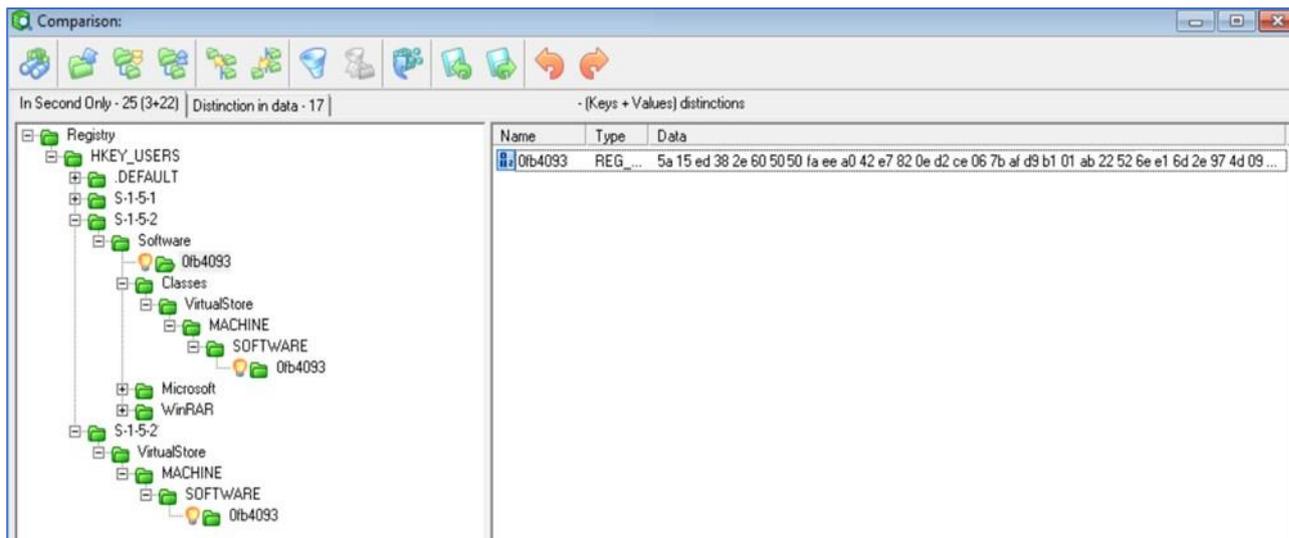


Figura 34

Captura snapshot del registro

Nota. La foto previa y posterior a la infección verifica los cambios en el registro de Windows.

Es importante aclarar que la llave **priv0** se almacena cifrada utilizando el algoritmo **ChaCha** con una llave derivada de una llave pública del RaaS presente en el archivo de configuración del malware (variable **mpk**) y una pública generada (**pubr**).

Entendemos que esto último en rojo es un error del informe y quisieron colocar llave privada (**privr**), ya que la **pubr** se almacena en el registro y es igual al esquema utilizado para los archivos que se describe a continuación.

Para cada archivo que debe cifrar el malware, se genera otro par de llaves publica/privadas (**priv1/priv0**) con **Curve25519** (como si fuera una de clave de sesión). Con la llave privada generada (**priv1**) y la pública del host (**pub0**), se deriva un **shared secret** de 32 bytes que es utilizado como **key** del cifrado del archivo bajo el algoritmo **ChaCha**. Cada archivo se cifra con la llave **ChaCha** mencionada y un **nonce**. Este **nonce** se genera a partir del hash **SHA256** del **shared secret**, que a su vez le incrementa +1 al primer byte.

Adicionalmente se utiliza un hash **HMAC-SHA256** para integridad y autenticación. Además, al archivo se le adicionan algunos parámetros para su posterior descifrado como ser la **pub1**, el **nonce**, modo de cifrado, etc. Ver figura 35.

```

0060BDA8 8E 77 1A 44 A2 75 A8 9A 56 E5 A0 79 21 7E 57 CF .w.Dçu".Vâ y!~Wİ
0060BDB8 52 BE 67 26 0C 51 4C ED 8C EA F9 BB F7 F7 3F 97 R³g&.QLí.èù»÷÷?.
0060BDC8 1E EE 10 2A 24 00 00 00 A7 D3 93 56 58 78 81 62 .i.*$....$Ó.VXx.b
0060BDD8 32 23 3D CA D3 58 1D ED 95 B3 2F 02 FB 00 65 65 2#=#ÈOX.i.³/.û.ee
0060BDE8 DD B1 83 63 C5 CF A1 93 83 0B C0 D0 EA D2 10 6C Ý±.cÁÍj...ÀÐèÒ.l
0060BDF8 A3 80 90 61 09 6F BE 23 C5 79 5E 15 3D 15 9C 34 £..a.o³#Áy^.=..4
0060BE08 0B 9E BB 13 10 66 D5 F2 83 0B C0 D0 EA D2 10 6C ..»..fÕò..ÀÐèÒ.l
0060BE18 A3 80 90 61 09 6F BE 23 C5 79 5E 15 3D 15 9C 34 £..a.o³#Áy^.=..4
0060BE28 0B 9E BB 13 10 66 D5 F2 83 0B C0 D0 EA D2 10 6C ..»..fÕò..ÀÐèÒ.l
0060BE38 A3 80 90 61 09 6F BE 23 C5 79 5E 15 3D 15 9C 34 £..a.o³#Áy^.=..4
0060BE48 0B 9E BB 13 10 66 D5 F2 C4 C7 A4 D9 4E A0 BE 76 ..»..fÕòÀÇ=ÛN ³v
0060BE58 00 00 00 00 00 3C 00 00 23 4E 73 53 01 AF 4B A6 .....<..#NsS.¯K!
0060BE68 77 83 67 56 FA D4 F5 E4 1C A4 52 36 32 9A 95 60 w.gVúÔöä.R62..`
0060BE78 BF 43 14 A1 09 43 D7 17 D2 54 F3 0A B8 2D AD 56 çC.j.C×.ÔTó.¸-V
0060BE88 BD 63 CA 27 C8 FB BB 88 84 9C 55 1F C6 AD 81 62 ²cÈ'Èù»...U.Æ..b
0060BE98 3C 5F 9B 29 FD 0E 6C AA 56 63 6B 9F DA 5A 59 68 <_)Ý.lªVck.ÚZYh
0060BEA8 23 BB 40 15 F6 61 41 4F D0 37 54 AF E7 25 EC 5C #»@.òAOD7T¯ç%i\
0060BEB8 89 70 93 EB 76 4A D5 4A 2F 71 A2 2E 6E 08 2D A5 .p.évJÖJ/qç.n.-¥
0060BEC8 D3 7D 92 8C 83 2F 4F 14 5F 02 05 93 90 C0 9A 32 ó).../O. ....À.$
0060BED8 78 43 3D 64 C9 D7 FA 52 EC 87 E4 B1 F4 1E 1B 24 xC=dÉ×úRì.à±ò...$
0060BEE8 3B 9E AA 16 XX XX XX XX ;.ª.....

```

```

Original File Information (Inner Layer, Encrypted)
Size of Encrypted File Information
HMAC-SHA256 Hash of Encrypted File Information
HMAC-SHA256 Hashes of the Three Encrypted Chunks
ChaCha Nonce
Encryption Mode
Chunk size (spsz)
Key Recovery Blob
One-time Curve25519 public key (publ)
CRC32 of the configuration public key (mpk, campaign-specific censored)

```

Figura 35

Captura del informe de CrowdStrike

Nota. Datos incorporados al final de cada archivo para su posterior recuperacion.

Luego de cifrar todos los archivos, el malware realiza un **wipe** de la llave privada (*priv0*) en memoria, para evitar un volcado de la misma. El mensaje en la nota de rescate incluye algunos parámetros que requiere el RaaS a fin de crear el proceso de descifrado en caso de pago del rescate. Para descifrar los archivos se necesita la llave **priv0** que puede descifrarse del registro de Windows con una llave derivada de la **pubr** del registro y la privada que posee únicamente el RaaS.

Nuestra interpretación del mecanismo de cifrado

Respecto al cifrado, el informe Crowstrike es confuso de leer y no especifica cómo se derivan los *shared secrets*. Por favor, no malinterpretar dicha afirmación. El trabajo que realizaron es realmente excelente. Simplemente consideramos que una explicación gráfica y una aclaración del posible mecanismo de derivación de claves ayudaría al entendimiento del lector.

En la siguiente figura 36 se aprecia el mecanismo de cifrado de archivos.

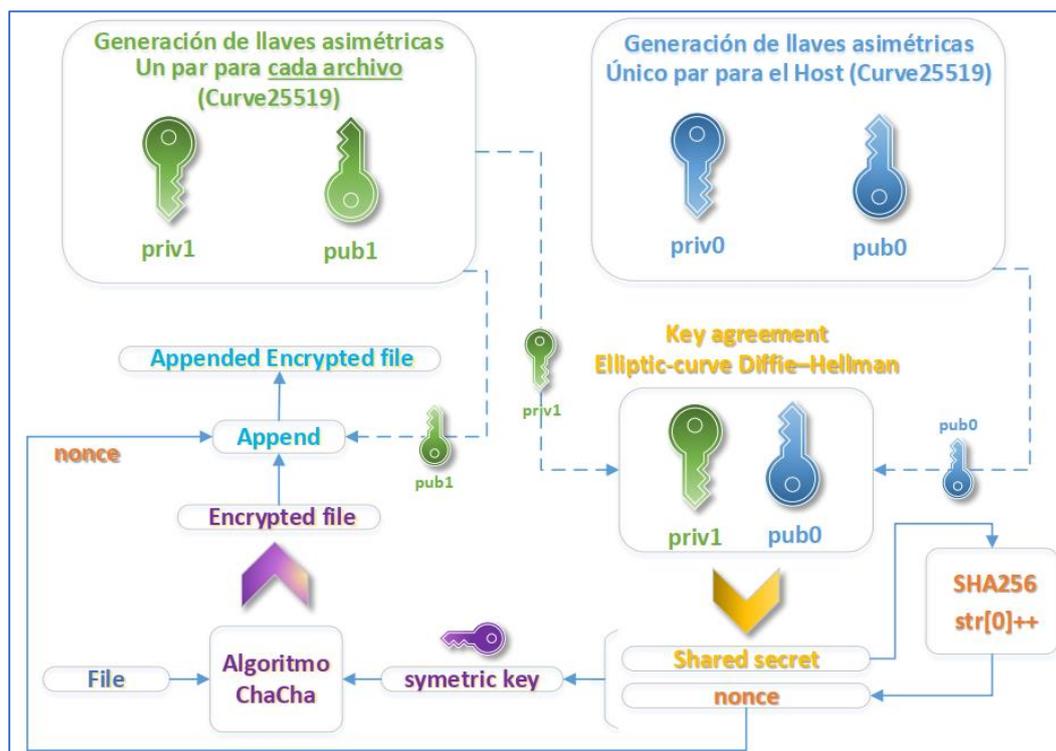


Figura 36
Cifrado de archivos

En nuestra explicación gráfica asumimos que la derivación de claves se realiza con un mecanismo de **Key Agreement** utilizando el **algoritmo de Diffie-Hellman de curvas elípticas (ECDH)**. Además, hemos quitado los hashes de verificación y algunos detalles menores a fin de facilitar la interpretación.

En la figura 37, se aprecia el mecanismo de cifrado de la llave **Priv0** que es almacena en el registro de Windows y se necesita para la recuperación de los archivos.

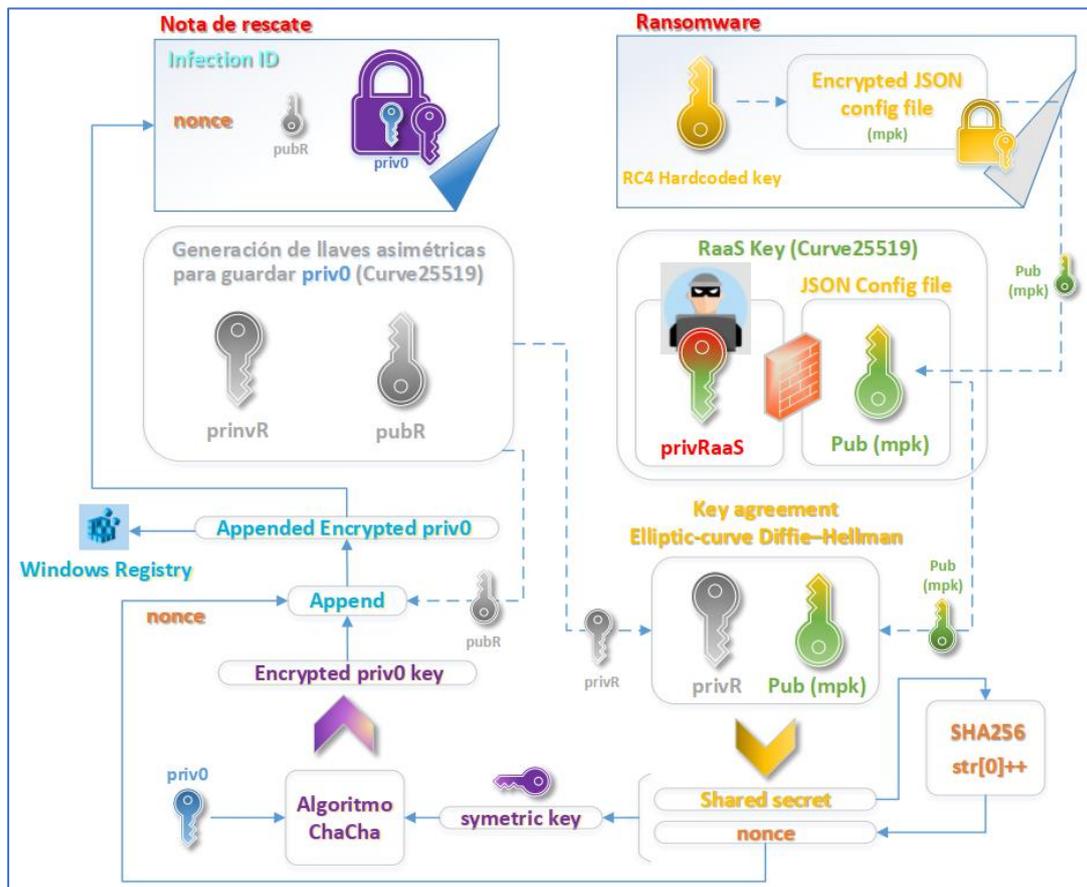


Figura 37

Cifrado de llave priv0

Antes de explicar el mecanismo de descifrado es importante recordar la propiedad del algoritmo de *Diffie-Hellman* que permite realizar un *Key Agreement* entre 2 personas y que es aprovechado por el malware. Ver figura 38.

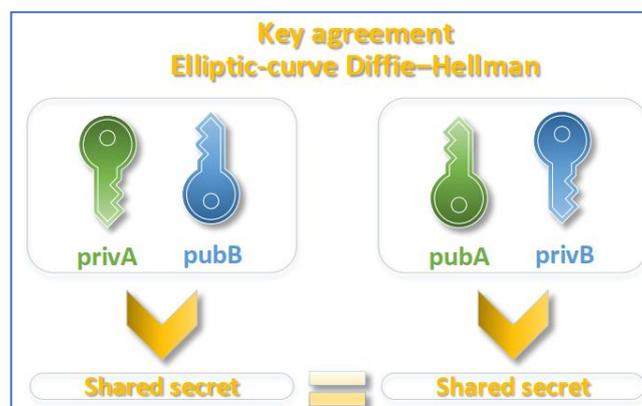


Figura 38

Propiedad Diffie-Hellman

Para el proceso de recuperación de archivos primero el desarrollador debe recuperar la llave *priv0* sin entregar su llave privada. Ver figura 39.

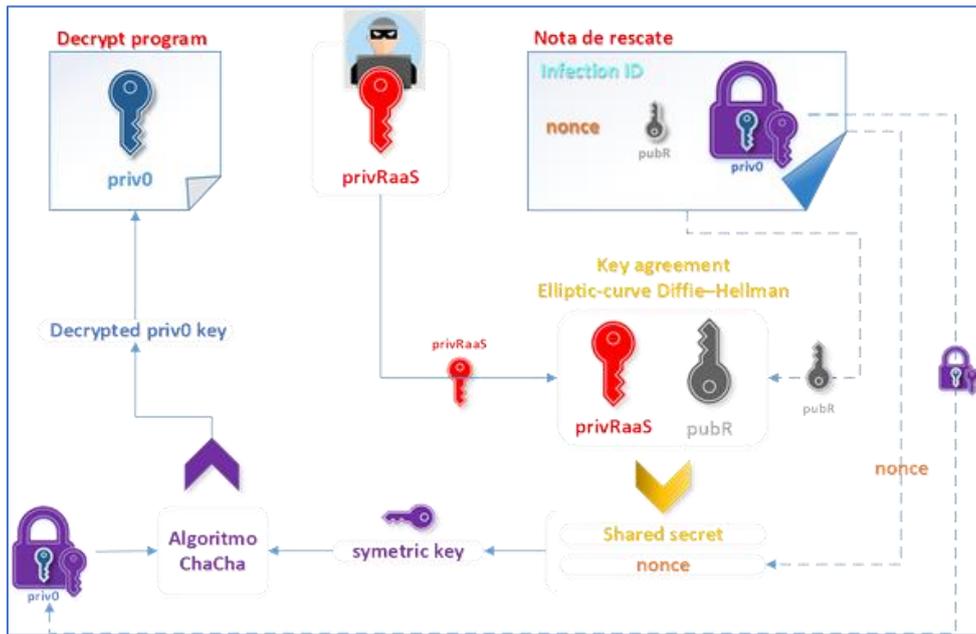


Figura 39
Descifrado *priv0*

Finalmente puede realizarse la recuperación de los archivos con el programa entregado por el RaaS. Ver figura 40.

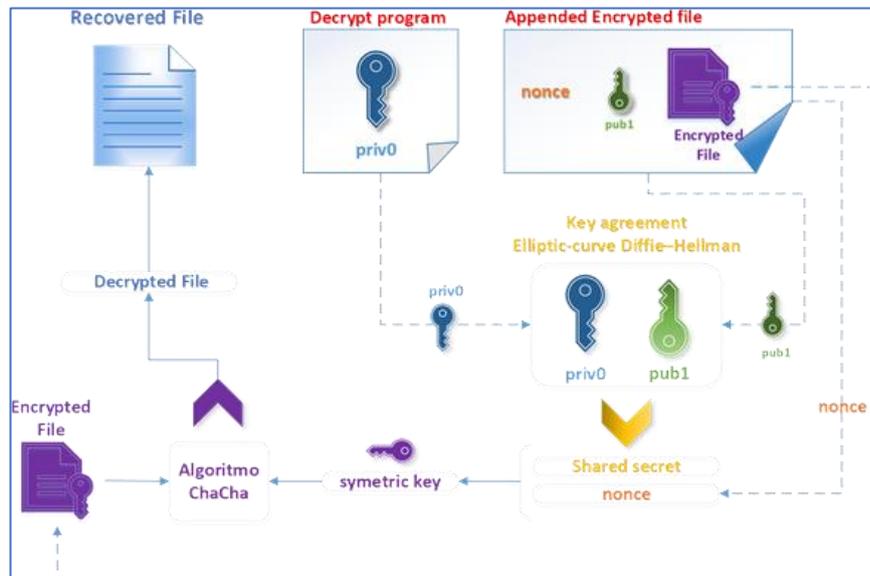


Figura 40
Recuperación del archivo

La llave pública del RaaS (presente en el archivo de configuración del malware) podría ser específica a fin de poder identificar a cada *partner*.

Conclusiones

En el trabajo expuesto, se pudo realizar una investigación a pesar de no contar con acceso a la infraestructura afectada en migraciones argentinas. Recreamos el proceso y obtuvimos *fingerprints* (huellas) que podrían ayudar a identificar al atacante. Algunos puntos relevantes que surgen del trabajo:

- El atacante posiblemente utilizaba un Windows en idioma inglés y estaba apurado.
- No estaba familiarizado con el contexto argentino o su intencionalidad no era económica.
- Dispone de un conocimiento avanzado para conocer el malware y contactar a los desarrolladores en algún foro de cibercrimen en la Deep Web. Tienen un vínculo estrecho. No descartar que puedan ser ellos mismos.
- El ataque fue combinado y se produjo en 2 etapas.
- Primero tuvo que vulnerar el sistema de la víctima con técnicas de hacking (phishing, vulnerabilidades web, intrusiones en la red, reconocimiento, movimientos laterales, elevación de privilegios, entre otros). Descargó la información, la comprimió y la subió a DropMeFiles. Finalmente, lanzó el malware en la retirada para exigir el rescate mediante ciber extorsión.
- Se utilizó un WinRAR > 5.0 (posiblemente 5.90)
- El *leak* de datos se produjo el 27/08/20 durante las 3:36AM y las 3:55AM UTC-3 (horario argentino). Una verificación en los logs de red identificaría la IP origen. Posterior a ello, se produce el cifrado de datos y pedido de rescate.
- La conexión de internet utilizada por el atacante tiene aproximadamente 13,5Mbits/s (1687KB/s) downstream más de 1,9Mbit/s (237KB/s) upstream. Podría ser mayor pero nunca mucho menor (considerar cierto margen de error por los cálculos estimados). Es importante aclarar que, al igual que la IP, esta conexión no necesariamente sea propia, bien podría ser un wifi ajeno o una maquina pivót.

Futuras investigaciones

- Verificar si el algoritmo de *Shared Key* que asumimos es el correcto.
- Verificar si la llave pública del RaaS identifica al *partner*.
- La investigación estuvo orientada más hacia el *partner* que a los

desarrolladores. Futuras investigaciones pueden enfocarse en los creadores del RaaS: *fingerprint* en el código, identificación basada en estilo de escritura (nota de rescate), seguimiento del flujo del dinero, investigación en foros, entre otros.

- ¿La Intencionalidad fue económica o política? ¿se trató de un simple acto de cibercrimen vinculado a grupos hacker rusos? ¿existe un estado extranjero con intención de interferir en nuestro país que a su vez inculpar a Rusia? Investigar diferentes tipos de hipótesis.

El análisis del malware no solo ayuda a resolver el cibercrimen, sino que también brinda importante información a los profesionales de seguridad a la hora de desarrollar mecanismos efectivos de defensa y de mitigación de riesgos. Es por ello que es importante estar atentos a los proyectos de ley que pretenden penalizar la tenencia de malware, ya que criminaliza nuestro trabajo de investigación y nuestra labor profesional diaria a la hora de proteger las infraestructuras críticas del país y a los ciudadanos.

Referencias bibliográficas

- McAfee (2020, Aug 3). Take a “NetWalk” on the Wild Side. Recuperado: <https://www.mcafee.com/blogs/other-blogs/mcafee-labs/take-a-netwalk-on-the-wild-side/>
- CrowdStrike (2020, Jun 10). Technical analysis of the **Netwalker** ransomware. Recuperado: <https://www.crowdstrike.com/resources/reports/Netwalker-ransomware-technical-analysis/>