

Informe Código Dañino

CCN-CERT ID-14/20

Maze



Mayo 2020



Edita:



© Centro Criptológico Nacional, 2019

Fecha de Edición: mayo de 2020

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	5
4. RANSOMWARE AS A SERVICE (RaaS)	6
5. EXTORSIÓN	7
6. PROCESO DE INFECCIÓN.....	9
6.1 SHELLCODE.....	9
6.2 DLL PRINCIPAL.....	9
6.2.1 OFUSCACIÓN	9
6.2.2 OPCIONES DE EJECUCIÓN.....	10
6.2.3 IMPORTACIÓN DE FUNCIONES.....	11
6.2.4 HOOKS	12
6.2.5 FINALIZACIÓN DE PROCESOS	12
6.2.6 LENGUAJES PROTEGIDOS	14
6.2.7 EXPLOITS.....	15
6.2.8 ESQUEMA DE CIFRADO	17
6.2.9 COMUNICACIÓN	20
7. RESCATE	20
8. DESINFECCIÓN	23
9. REGLAS DE DETECCIÓN.....	23
9.1 REGLA YARA	23
10. INDICADORES DE COMPROMISO	24



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 perteneciente a la familia de **ransomware Maze**, identificada por la firma SHA256 db617d3ca09f78673aef2a706a0161b9a7e160f58891f14a1e7250b39e3d9fb2.

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

Las primeras muestras de Maze datan de principios de mayo de 2019, aunque en aquel momento se identificaba la familia de ransomware bajo el nombre ChaCha, en alusión al algoritmo que emplea para cifrar los ficheros de los sistemas infectados. En los últimos meses, Maze se ha perfilado como una de las **principales amenazas** a las que se enfrentan las empresas, en lo que respecta a infecciones por malware y su **impacto en la continuidad de negocio, filtrado de información sensible y daño a la imagen corporativa**. El modo de operación del grupo cibercriminal no se limita al cobro de rescates a cambio de la recuperación de los datos cifrados. Maze ha ido un paso más allá posicionándose como uno de los grupos precursores de la extorsión en público, amenazando a las empresas afectadas con filtrar su información confidencial en caso de negarse a colaborar, es decir, proceder con el pago del rescate.

En puntos posteriores del informe se entra en detalles sobre el modelo de extorsión, así como en detalles técnicos sobre la muestra analizada. Finalmente, se proporciona una regla YARA con la que identificar muestras similares de la familia de ransomware objeto de análisis.

3. DETALLES GENERALES

El binario inicial, ejecutable para sistemas Windows de 32-bit, se identifica con la firma SHA256 que se muestra a continuación.

Fichero	SHA256
maze.exe	db617d3ca09f78673aef2a706a0161b9a7e160f58891f14a1e7250b39e3d9fb2

La fecha de compilación del código dañino protegido mediante un *custom packer* coincide con la fecha que muestra la DLL obtenida tras el proceso de *unpacking*, datando del 8 de marzo de 2020.

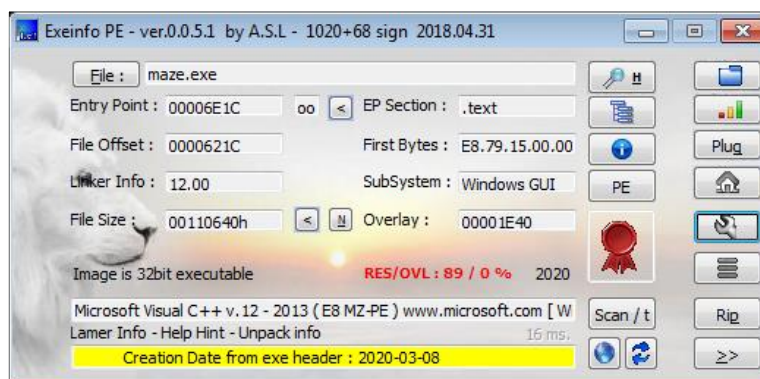


Figura 1. Propiedades del binario inicial

Sin embargo, no ha sido hasta el 22 de abril de 2020 cuando la muestra se ha subido ha sido detectada por primera vez.

La DLL resultante del *unpacking* contiene toda la funcionalidad del código dañino, en su **versión 2.3.2**, oscurecida por múltiples técnicas de ofuscación destinadas a romper herramientas de desensamblado. Probablemente, el nombre elegido por los desarrolladores (Maze, sinónimo de laberinto en su traducción del inglés) haya sido elegido tras el resultado obtenido al ofuscar el código. Además de la ofuscación, en este punto cabe destacar que el código dañino contiene dos *exploits*, destinados a elevar privilegios explotando las vulnerabilidades CVE-2016-7255 y CVE-2018-8453.

4. RANSOMWARE AS A SERVICE (RaaS)

De forma previa a entrar en detalles técnicos sobre el binario analizado, resulta de interés entender el contexto en el que se encuentra el ransomware Maze. Aunque se desconocen los detalles de cómo se estructura el modelo de negocio detrás de Maze, todo indica a que sigue un programa de afiliados que adquieren el ransomware para monetizar accesos a empresas que han conseguido por otra vía.

En un primer momento, la distribución se realizaba por medio de campañas de SPAM y ExploitKits, sin embargo, ninguno de estos vectores explicaría ninguno de los *outbreaks* causados últimamente por el grupo criminal. El giro que ha posicionado a Maze como una de las principales amenazas a fecha de hoy, fue la tendencia que adoptaron a finales de 2019 de distribuir el ransomware en la etapa de post-explotación, tras comprometer previamente las redes internas de las compañías afectadas.

En esta última tendencia, se diferenciaría por un lado a los desarrolladores del código dañino y por otro a los clientes afiliados al programa. Los afiliados contarían de forma ideal con accesos ilícitos a redes comprometidas de entidades, desde donde podrían desplegar el ransomware a placer. Si se suma a la ecuación la exfiltración de información de forma previa al despliegue del ransomware, el impacto para la entidad



afectada no solo pone en riesgo su continuidad de negocio y su marca, sino la seguridad de sus clientes y proveedores.

Como medida de protección, se recomienda la lectura del Informe de Amenazas [CCN-CERT IA-11/18](#) de medidas de seguridad contra ransomware disponible desde el portal del CCN-CERT.

5. EXTORSIÓN

Se adelantaba que una vez que los actores se encuentran con un acceso garantizado a la red de la empresa comprometida, el último giro en la escena del despliegue masivo de ransomware es la exfiltración de información de forma previa al cifrado. Esta última tendencia dota al grupo criminal de una nueva herramienta, que no es otra que la extorsión. Para hacer eco del éxito de sus ataques, el grupo detrás de Maze mantiene un blog en el que publican los últimos acontecimientos y señalan públicamente a las compañías que se niegan a cooperar, es decir, que se niegan a proceder con el pago solicitado.

En los comunicados, los actores señalan a las empresas afectadas e incrementan la gravedad del incidente haciendo pública información sensible, que bien puede poner en mayor riesgo a la propia compañía afectada, o dependiendo de los datos publicados, a los clientes o proveedores. En la imagen a continuación se muestra una entrada del 5 de mayo de 2020 en el que tras la negativa por parte de una entidad a admitir el éxito del ataque y cooperar con el grupo Maze, se publica la estructura de su red interna y se amenaza con continuar divulgando información sensible en posteriores entregas.



Figura 2. Comunicado del grupo Maze del 5 de mayo de 2020

Además del portal con las últimas noticias, en el blog se listan los diez últimos clientes (eufemismo para empresas afectadas) y se listan los documentos de las compañías que, tras la extorsión, se han negado definitivamente a cooperar.

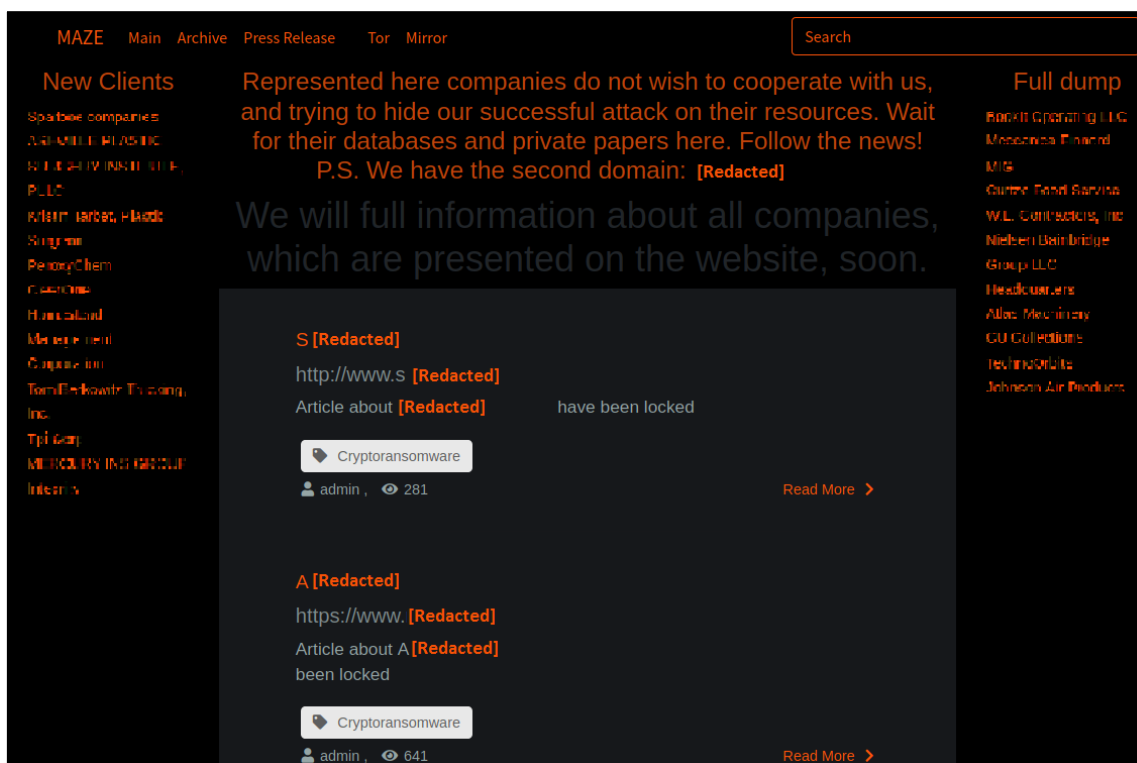


Figura 3. Portal con los últimos clientes y documentos publicados



Finalmente cabe destacar que el grupo criminal parece no tener predisposición por áreas geográficas o por áreas de actividad empresarial, pues no se observa ningún patrón en las empresas listadas en su web.

6. PROCESO DE INFECCIÓN

Cuando el binario inicial se ejecuta, iniciando así el proceso de infección, un fragmento de *shellcode* se carga en memoria destinado a cargar a su vez la DLL principal de Maze.

6.1 SHELLCODE

El fragmento de *shellcode* contiene embebida la DLL principal de Maze, que carga sin necesidad de usar el *loader* de Windows.

```
IDA View-A | Hex View-1 | Structures
Library function | Regular function | Instruction | Data | Unexplored | External symbol | Lumina function

seg000:000004C7 ;
seg000:000004CC aVirtualFree db 'VirtualFree',0
seg000:000004D8 ;
seg000:000004D8 loc_4D8: ; CODE XREF: sub_4B1+16↑p
seg000:000004D8 pop ecx
seg000:000004D9 mov [esp+4], ecx
seg000:000004DD call eax
seg000:000004DF sub esp, 8
seg000:000004E2 mov [ebp-10h], eax
seg000:000004E5 mov eax, [ebp-24h]
seg000:000004E8 mov ecx, [ebp-4]
seg000:000004EB mov [esp], ecx
seg000:000004EE call loc_502
seg000:000004EE ;
seg000:000004F3 aVirtualProtect db 'VirtualProtect',0
seg000:00000502 ;
seg000:00000502 loc_502: ; CODE XREF: seg000:000004EE↑p
seg000:00000502 pop ecx
seg000:00000503 mov [esp+4], ecx
seg000:00000507 call eax
```

Figura 4. *Shellcode* destinado a cargar en memoria la DLL principal de Maze

Esta técnica denominada *reflective DLL loading* permite ejecutar PEs (*Portable Executables*) que se encuentren en memoria, sin necesidad de que el fichero toque el disco.

6.2 DLL PRINCIPAL

Una vez extraída la DLL principal, es posible continuar con el análisis de Maze. La muestra implementa varias técnicas para complicar el proceso, que se desglosan a lo largo de la esta sección.

6.2.1 OFUSCACIÓN

El análisis de Maze está marcado por los múltiples mecanismos de ofuscación que dificultan el desensamblado previniendo la ingeniería inversa.

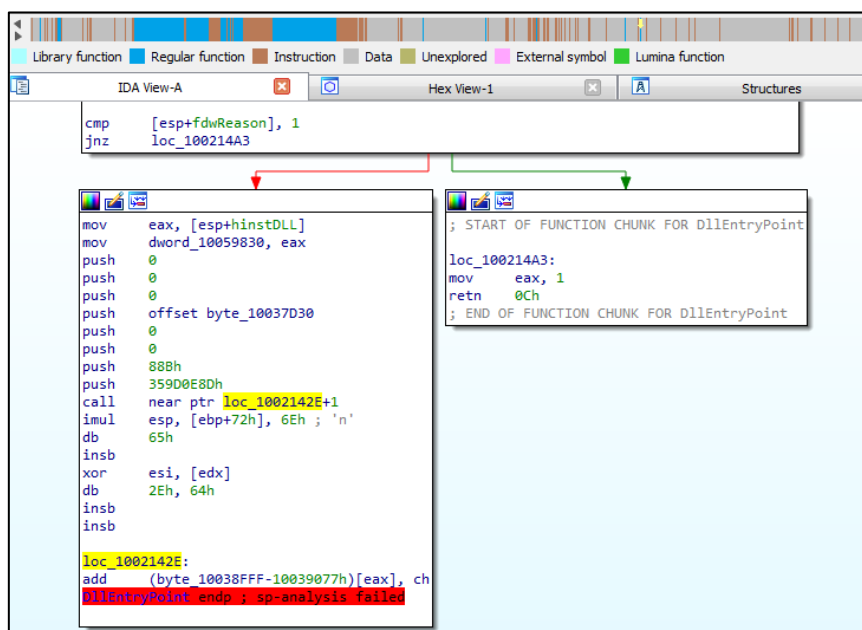


Figura 5. Punto de entrada la DLL principal

La implementación de rutinas que eliminan mencionadas capas de ofuscación queda fuera del alcance del presente documento, sin embargo, han sido cubiertas recientemente por otros autores.

- <https://www.crowdstrike.com/blog/maze-ransomware-deobfuscation/>

6.2.2 OPCIONES DE EJECUCIÓN

Antes de entrar en detalles sobre las características técnicas del código dañino, conviene adelantar que su flujo de ejecución puede ser modificado a través de una serie de parámetros a suministrar por línea de comandos.

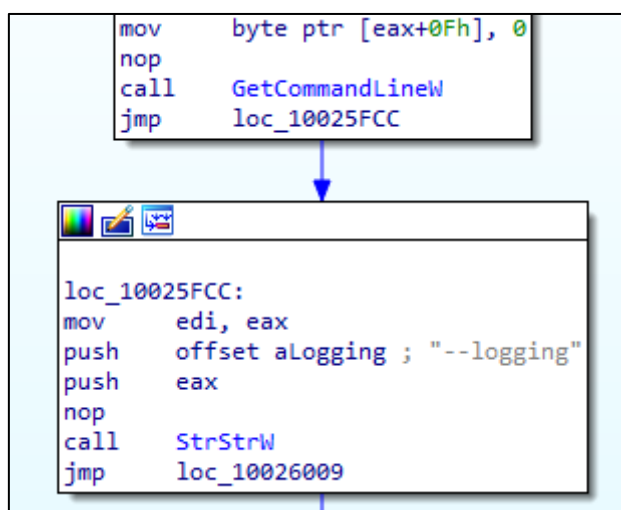


Figura 6. Comprobación de los parámetros de entrada

La tabla a continuación recoge la descripción de cada uno de los posibles parámetros contemplados.



Parámetro	Descripción
--logging	Activa los mensajes de <i>debug</i>
--nomutex	Deshabilita el control de instancias vía <i>mutex</i>
--noshares	Deshabilita el cifrado de los recursos de red compartidos
--path	El cifrado se realiza únicamente en la ruta especificada

Cabe destacar que en los últimos despliegues masivos de ransomware protagonizados por Maze, los actores ejecutan los binarios desde posiciones privilegiadas dentro de la red comprometida. Estas opciones en los argumentos del programa les permite controlar que el resultado sea el esperado en tiempo real. Si por ejemplo se desarrollase una vacuna basada en el nombre del *mutex*, relanzar el binario con la opción **--nomutex** dejaría la vacuna inservible.

6.2.3 IMPORTACIÓN DE FUNCIONES

Otra de las técnicas destinadas a ocultar la funcionalidad del binario es la resolución de las funciones de Windows a importar de forma dinámica.

```

.text:10021406      push     0
.text:10021408      push     0
.text:1002140A      push     offset main_thread
.text:1002140F      push     0
.text:10021411      push     0
.text:10021413      push     88Bh          ; XOR word key
.text:10021418      push     359D0E8Dh     ; XORed dword adler32 hash
.text:1002141D      push     offset kernel32 ; "kernel32.dll"
.text:10021422      call     import_api    ; CreateThread
.text:10021427      jmp      loc_10021488
.text:10021427 ; -----
.text:1002142C      kernel32 db 'kernel32.dll',0 ; DATA XREF: .text:1002141D↑to
.text:10021439      db      0

```

Figura 7. Ejemplo de resolución de la función CreateThread

Para ello, se pasa a la función responsable de la resolución una serie de parámetros.

- Clave XOR
- *Checksum* Adler-32 XORreado de la función a importar
- Nombre de la librería que exporta la función a importar

Para obtener la función, se recorren los *exports* de la librería calculando el *checksum* Adler-32 para cada uno de ellos (en mayúsculas), partiendo de un valor inicial de **666 (0x29A)** en hexadecimal). La función deseada será aquella cuyo *checksum* coincida con el resultado de la operación XOR entre la clave y el *checksum* XORreado. Mediante esta técnica, se importan funciones de las librerías que se listan a continuación.

- user32.dll



- kernel32.dll
- ole32.dll
- advapi32.dll
- gdi32.dll

6.2.4 HOOKS

Con el fin de impedir que un *debugger* pueda interrumpir la ejecución del código dañino, se implementa un *hook* para la función **DbgUiRemoteBreakin**.

ntdll32.dll:773DF7EA ntdll_DbgUiRemoteBreakin proc near	ntdll32.dll:773DF7EA ntdll_DbgUiRemoteBreakin:
ntdll32.dll:773DF7EA push 8	ntdll32.dll:773DF7EA retn
ntdll32.dll:773DF7EC push offset unk_7736BA30	ntdll32.dll:773DF7EA ; -----
ntdll32.dll:773DF7F1 call near ptr unk_7736DEB4	ntdll32.dll:773DF7EA ; -----
ntdll32.dll:773DF7F6 mov eax, large fs:18h	ntdll32.dll:773DF7EB db 8
ntdll32.dll:773DF7FC mov eax, [eax+30h]	ntdll32.dll:773DF7EC ; -----
ntdll32.dll:773DF7FF cmp byte ptr [eax+2], 0	ntdll32.dll:773DF7EC push offset unk_7736BA30
ntdll32.dll:773DF803 jnz short loc_773DF80E	ntdll32.dll:773DF7F1 call near ptr unk_7736DEB4
	ntdll32.dll:773DF7F6 mov eax, large fs:18h

Figura 8. Hook en la función DbgUiRemoteBreakin

El *hook* consiste en sustituir el primer *opcode* de la función por un **RETN**, impidiendo de tal manera que se ejecuten las instrucciones del cuerpo de la función.

6.2.5 FINALIZACIÓN DE PROCESOS

Maze trata de finalizar 52 procesos, entre los que se encuentran aquellos correspondientes a herramientas de análisis de malware y procesos relacionados con aplicaciones ofimáticas y manipulación de bases de datos. Mientras que las herramientas de análisis se finalizarían con el objetivo de complicar el estudio de la muestra, el objetivo de finalizar el segundo tipo de procesos nada tiene que ver con el primero. Si un archivo de interés para el cifrado se encuentra abierto por un programa, este programa lo bloquearía de posibles modificaciones por parte de un segundo proceso. Por este motivo, es usual observar en implementaciones de ransomware rutinas para la finalización de procesos de interés y liberar así los ficheros objetivo del cifrado.

La rutina que implementa Maze sigue el siguiente procedimiento.

- Se obtiene la lista de procesos corriendo en el equipo y se itera sobre ella.
- El nombre del proceso sufre una serie de transformaciones.
- Se calcula el *checksum* Adler-32 (**con un valor inicial de 666**) sobre el nombre transformado.
- Los dos bytes más significativos del *checksum* sufren otra modificación.
- El valor final obtenido se compara con un total de 52 *checksums* que identifican los procesos susceptibles de ser finalizados.

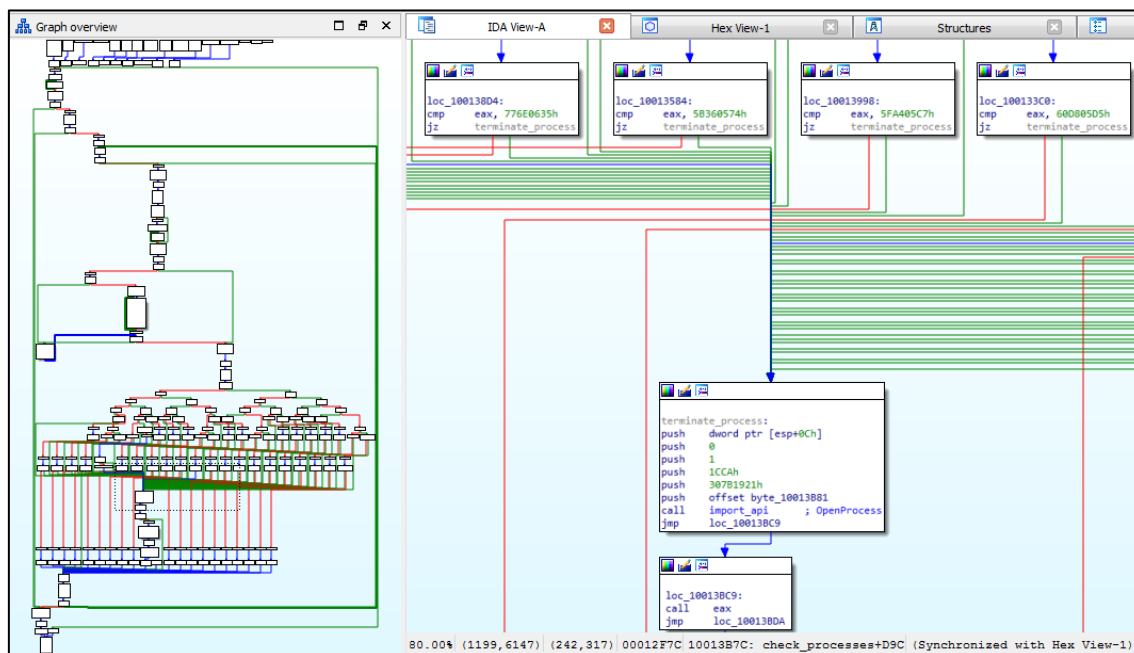


Figura 9. Comprobación del *checksum* de los procesos en ejecución

En la siguiente tabla se listan algunos de los procesos susceptibles de ser finalizados por Maze junto con el *checksum* que identifica a cada uno de ellos.

Proceso	Checksum	Proceso	Checksum
agntsvc.exe	0x5f1a05c1	outlook.exe	0x615605dc
dbeng50.exe	0x5b360574	powerpnt.exe	0x6de00630
dbsnmp.exe	0x53380575	procexp.exe	0x606805d4
dumpcap.exe	0x5fb805c5	procexp64.exe	0x78020640
encsvc.exe	0x52fc056b	procmon.exe	0x600005c9
excel.exe	0x48780528	procmon64.exe	0x776e0635
Fiddler.exe	0x5e0c05b1	python.exe	0x55ee0597
firefoxconfig.exe	0xb0200770	sqbcoreservice.exe	0xc14a07c9
ida.exe	0x33840485	sqlagent.exe (*)	0x6c2e0614
ida64.exe	0x45e204f1	sqlbrowser.exe	0x87e606bd
infopath.exe	0x6b88060e	sqlservr.exe	0x6dec062f
isqlplussvc.exe	0x97c2071c	sqlwriter.exe	0x7b8e0680
msaccess.exe	0x6a9c05ff	steam.exe	0x491a0531
msftesql.exe	0x6d100624	synctime.exe	0x6d680621



Proceso	Checksum	Proceso	Checksum
mspub.exe	0x499a053a	tbirdconfig.exe	0x928606d6
mydesktopqos.exe	0xa6300768	thebat.exe	0x537c0571
mydesktopservice.exe	0xe72e0886	thebat64.exe	0x698a05dd
mysqld-opt.exe	0x87ba06b4	thunderbird.exe	0x950c06f2
mysqld.exe	0x55b00593	visio.exe	0x49780539
ocautoupds.exe	0x873806bc	winword.exe	0x60d805d5
ocomm.exe	0x4866052a	wordpad.exe	0x600205c4
ocssd.exe	0x485c0527	x32dbg.exe	0x5062053b
onenote.exe	0x5fa405c7	x64dbg.exe	0x50dc0542
oracle.exe	0x52f00567	xfssvccon.exe	0x79a40660

(*) El *checksum* del proceso **sqlagent.exe** colisiona con el del proceso **taskkill.exe**.

6.2.6 LENGUAJES PROTEGIDOS

El código dañino no inicia el proceso de cifrado si el lenguaje del sistema se encuentra entre los idiomas protegidos. Las funciones de Windows empleadas en la obtención del idioma del sistema se listan a continuación.

- GetUserDefaultUILanguage
- GetSystemDefaultLangID
- GetUserDefaultLangID

Si el código del idioma se encuentra entre alguno de los 21 contemplados en la siguiente tabla, el proceso de cifrado de ficheros no se inicia.

Identificador	Idioma
0x419	Ruso
0x422	Ucranio
0x423	Bielorruso
0x428	Tayiko
0x42B	Armenio
0x42C	Azerbaiyano (latín)
0x437	Georgiano



Identificador	Idioma
0x43F	Kazajo
0x440	Kirguís
0x442	Turcomanos
0x443	Uzbeko (latín)
0x444	Tártaro
0x818	Rumano (Moldavia)
0x819	Ruso (Moldavia)
0x81A	Serbio (latín, Serbia y Montenegro (antiguo))
0x82C	Azerbaiyano (cirílico)
0x843	Uzbeko (cirílico)
0x1C1A	Serbio (cirílico, Bosnia y Herzegovina)
0x281A	Serbio (cirílico, Serbia)
0x6C1A	Serbio (cirílico)
0x7C1A	Serbio

6.2.7 EXPLOITS

La DLL principal de Maze incorpora dos *exploits* para elevar privilegios aprovechándose de las vulnerabilidades identificadas por **CVE-2016-7255** y **CVE-2018-8453**.

En ambos casos, el código de los *exploits* se encuentra cifrado mediante el algoritmo ChaCha8, utilizando como clave "**37432154789765254678988765432123**" y como *nonce* "**09873245**" para el descifrado.

6.2.7.1 CVE-2016-7255

Para explotar la vulnerabilidad registrada en el CVE-2016-7255, tras el descifrado se obtienen dos DLLs destinadas a arquitecturas de 64 y 32-bits, respectivamente.



```

.text:1001A48D CVE_2016_7255: ; CODE XREF: .text:1001A448↑j
.text:1001A48D      mov     edi, [esp+14h]
.text:1001A491      xor     eax, eax
.text:1001A493      mov     esi, ecx
.text:1001A495      mov     ecx, offset CVE_2016_7255_x64
.text:1001A49A      mov     ebp, offset CVE_2016_7255_x86
.text:1001A49F      cmp     dword ptr [edi+28h], 40h ; '@'
.text:1001A4A3      setz    al
.text:1001A4A6      cmovz   ebp, ecx
.text:1001A4A9      shl     eax, 0Bh
.text:1001A4AC      or      eax, 2400h
.text:1001A4B1      mov     [esi+8], eax
.text:1001A4B4      push    eax
.text:1001A4B5      nop
.text:1001A4B6      call    virtual_alloc_2
.text:1001A4B8      jmp     chacha_decrypt_exploit

```

Figura 10. DLLs incorporadas para la elevación de privilegios

Las DLLs con los *exploits* se identifican con las firmas SHA256 recogidas en la siguiente tabla.

Arquitectura	SHA256
x64	953d574ef0f49e886e270c2887586aab5c0d37616dd99f422838a1257abdbe3b
x86	d339da79ac85ba14dcde6c0e2263e17b9382126baff1a8a8248b071f2027e1f

Para mayor detalle sobre el funcionamiento de la vulnerabilidad y su explotación se recomienda visitar los recursos listados a continuación.

- <https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2016-7255>
- <https://blog.trendmicro.com/trendlabs-security-intelligence/one-bit-rule-system-analyzing-cve-2016-7255-exploit-wild/>

6.2.7.2 CVE-2018-8453

Por su parte, el *exploit* destinado a elevar privilegios a través de la vulnerabilidad registrada en el CVE-2018-8453, se presenta en forma de *shellcode* una vez descifrado.

Arquitectura	SHA256
x86-64	987bec3c79c8d19cfa4ec66e8e05ab8ccd8700f9d22083c9e61f596451e8f4d2

Para mayor detalle sobre el funcionamiento de la vulnerabilidad y su explotación se recomienda visitar los recursos listados a continuación.

- <https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2018-8453>
- <https://securelist.com/cve-2018-8453-used-in-targeted-attacks/88151/>



6.2.8 ESQUEMA DE CIFRADO

Tras haber comentado ciertos aspectos de la funcionalidad del código dañino, en este punto se centra la atención en la principal funcionalidad del ransomware: el cifrado de ficheros.

Antes de proceder al cifrado, se eliminan las **shadow copies** impidiendo recuperar la información desde back-ups generados por el sistema mediante esta herramienta de Windows. Para ello, el código dañino ejecuta un comando similar al ejemplo que se muestra a continuación.

```
C:\gcph\ncubk\gmbc\..\..\Windows\ih\dd\thjl\..\..\system32\l\r\t\..\..\wbem\nvxc\..\wmic.exe shadowcopy delete
```

La ofuscación que se aprecia consiste en introducir directorios aleatorios que luego en la práctica se retroceden con las secuencias de dos puntos consecutivos. Al concluir el proceso de cifrado, el comando para eliminar las **shadow copies** se ejecuta por segunda vez.

En cuanto al esquema de cifrado, se combina criptografía simétrica, **ChaCha8** con claves de 256-bit y **nonces** de 64-bit, con asimétrica, usando claves **RSA** de 2048-bit. La muestra incluye la clave pública del par máster RSA, cuya pareja privada se encuentra en poder del grupo ciber-criminal. En cada ejecución, el código dañino generaría un nuevo par de claves RSA asignadas al usuario infectado. Para evitar usar más de un par de claves RSA para un mismo usuario, en caso de múltiples ejecuciones del ransomware, el proceso de generación sigue los siguientes pasos, con la correspondiente medida de seguridad para evitar tal escenario.

- Se genera el par de claves RSA asignadas al usuario mediante **CryptGenKey**.
- Se importa la RSA pública del par master mediante **CryptImportKey**.
- Mediante **CryptGenRandom**, se genera una clave de 256-bit y un **nonce** de 64-bit para el algoritmo ChaCha8.
- Se cifra la clave RSA privada del par asignado al usuario mediante ChaCha8 y las claves del paso anterior.
- Se cifra la clave usada en el cifrado ChaCha8 con la RSA pública del par máster.
- Se cifra el **nonce** usado en el cifrado ChaCha8 con la RSA pública del par máster.

De esta manera se garantiza que la clave RSA privada del usuario solo pueda ser recuperada a través del uso de la clave RSA privada del par máster, en manos del atacante. Una vez generado el par de claves del usuario y asegurada la parte privada por medio del cifrado, se crea en el directorio %ProgramData% el siguiente artefacto.

```
%ProgramData%\memes.tmp
```

Este fichero contiene la información que se lista a continuación.

- Clave RSA privada del usuario cifrada mediante ChaCha8.



- Clave ChaCha8 de 256-bit cifrada con la clave RSA pública del par máster.
- *Nonce* ChaCha8 de 64-bit cifrado con la clave RSA pública del par máster.
- Clave RSA pública del par asignado al usuario.

De esta manera, si al iniciar el proceso de cifrado este fichero ya existe, se tomaría la parte pública de la pareja RSA asignada al usuario, garantizando así que una única RSA privada pueda devolver los ficheros al estado original. Cabe destacar que esta información no se escribe en el fichero **memes.tmp** de una manera convencional, sino que se emplean los **atributos extendidos** del fichero para almacenar en base64 todos los valores listados anteriormente.

```

C:\ProgramData>EaQuery.exe /Target:C:\ProgramData\memes.tmp /Mode:0 /Verbose:2
EaQuery v1.0.0.1

TargetFile: C:\ProgramData\memes.tmp
NextEntryOffset: 0
Flags: 0x00
EaNameLength: 6
EaName: GFT0AU
EaValueLength: 2616
EaValue:
0000 70 4e 39 4f 53 69 61 41 33 6f 52 42 44 42 4a 45 pN90Sia03oRDBJE
0010 44 46 6c 41 71 50 42 59 32 2b 71 38 58 2b 54 6b DF1aQPBV2+q8X+Tk
0020 70 4f 52 6c 73 52 2b 2b 42 64 41 49 6c 63 67 76 pORlsR++BdAl lcgV
0030 47 34 74 46 50 62 39 45 6a 6e 42 6d 41 61 45 63 G4tFPb9EjnBm0aEc
0040 43 30 65 51 46 47 6c 70 4e 67 53 54 41 63 30 33 C0eQFG1pNgSTAc03
0050 6c 73 4c 47 55 52 45 78 78 62 63 59 70 37 68 55 lsLGURExxbcYp7hU
0060 54 4d 69 45 4c 64 6b 62 6d 4b 38 37 6a 2b 2b 6b TMiELdkbmK82j++k
  
```

Figura 11. Atributos extendidos del fichero memes.tmp

El proceso utilizado para cifrar cada fichero sigue los siguientes pasos.

- Se genera un clave de 256-bit y *nonce* de 64-bit para usar con el algoritmo ChaCha8.
- Se cifra el contenido del fichero mediante ChaCha8 y las claves del paso anterior.
- Se concatenan clave (rojo) y *nonce* (azul) utilizados en ChaCha8.

Address	Hex	ASCII
00A60000	88 F7 C6 2B 83 2B CE EA DB C8 79 D5 DB 7D 5F 38	.+4+.+ie0ey00}_8
00A60010	19 8E 8B CD CD C9 38 CE 99 CF 3B 4F 92 22 60 B9	...iie8i.i;0."^,
00A60020	54 E4 37 B1 A6 D7 F2 A6 00 00 00 00 00 00 00 00	Ta7±;x0;.....
00A60030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A60040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A60050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A60060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A60070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A60080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A60090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A600A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A600B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A600C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A600D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A600E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A600F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00A60100	00 00 00 00 66 11 61 66 00 00 00 00 00 00 00 00	...f.af.....
00A60110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

Figura 12. Clave ChaCha8 y *nonce* antes de ser cifrados

- Se cifran con la clave RSA pública del par asignado al usuario.



Address	Hex	ASCII
00A60000	A2 13 75 F5 74 CE BF 5C 2F 10 EB AD 8D 5A 24 E0	c.uötî¿\./ë..Z\$a
00A60010	8D 1C A9 C7 B2 83 B8 53 92 36 92 42 D7 88 DC 75	..@Ç*. S.6.Bx.Uu
00A60020	08 38 94 53 F3 1D B3 E2 A1 5D 13 DC F5 E9 A5 62	.8.Só.*à;j.Üöëyb
00A60030	6F 08 72 07 D5 43 98 00 A1 E8 27 32 46 09 61 64	o.r.ÖC..ië'2F.ad
00A60040	30 58 AF 9F 31 1B 21 23 9E C6 78 4B 3B DE 32 2F	OX .1. !#.AxK; p2/
00A60050	FA F7 64 77 C0 C9 AF 92 77 71 AE D8 71 8C E0 0C	ú÷dwAE-.wq°0q.à.
00A60060	59 11 CE 81 0D E2 53 6E 71 B8 AC D3 CC 27 73 43	Y.î..àSng»-0I'sC
00A60070	D5 22 4A 06 B6 EE F2 9C EE EB FE CF 76 EC 31 EE	Ö"J.ñib.îëpîvîîî
00A60080	D2 58 12 36 E2 B9 B1 F1 82 51 DE 97 A1 7A 56 71	ÖX.6â'±ñ.Qb.izVq
00A60090	A0 DE 76 34 C1 C5 4C 7C E0 85 6A 2A 18 09 AB 48	pV4AAL à.j*..«H
00A600A0	09 B3 E8 87 A7 37 77 CC 25 1F 37 BA 32 98 5F 98	.*ë.\$7wi%.7°2.-.
00A600B0	F1 26 BA CE 30 78 A5 CE 3C E8 99 EC B0 8E B8 1F	ñ&°10x*I<è.î'...
00A600C0	B3 C8 65 DD B8 66 53 02 A7 4E 72 FE 1E E0 55 CD	*ÊeY.fS.\$Nrp.àUi
00A600D0	11 07 DA 66 C9 1D 6F 59 2D 5A ED 13 B7 8D 0C EA	..ÜfÉ.oy-Zí....ê
00A600E0	B4 87 58 CE C9 B9 84 43 8A 1C 5A E4 4E 6B B6 86	.XIÉ'.C..Zänk¶.
00A600F0	3C 68 01 75 08 98 ED 22 5A F2 B3 48 F5 5F 62 49	<k.u..i"zâ*Hö_bI
00A60100	00 00 00 00 66 11 61 66 00 00 00 00 00 00 00 00	f.af.....
00A60110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

Figura 13. Clave ChaCha8 y *nonce* cifrados junto con la marca de cifrado

El fichero resultante se compone del contenido cifrado, claves cifradas y la marca de cifrado **0000000066116166** (para evitar múltiples cifrados de un mismo fichero), además de una nueva extensión calculada de forma aleatoria.

En cuanto a los ficheros exentos de ser cifrados, se incluyen los que cuenten con alguna de las siguientes extensiones.

Extensiones exentas del cifrado			
Ink	exe	sys	dll

Además, los ficheros con los siguientes nombres también están exentos de sufrir modificaciones.

Ficheros exentos del cifrado	
DECRYPT-FILES.txt	iconcache.db
autorun.inf	bootsect.bak
boot.ini	ntuser.dat.log
desktop.ini	thumbs.db
ntuser.dat	Bootfont.bin

En última instancia, el contenido de los directorios a continuación también se encuentra libre de ser cifrado.

Directorios exentos del cifrado	
:\\Windows	\\User Data\\Default\\Cache\\
\\Games\\	\\All Users
\\Tor Browser\\	\\IETIdCache\\



Directorios exentos del cifrado	
\ProgramData\	\Local Settings\
\cache2\entries\	\AppData\Local
\Low\Content.IE5\	AhnLab

Al concluir el proceso de cifrado, se genera el *wallpaper* personalizado a la vez que una alerta sonora informa al usuario de la necesidad de comprar la herramienta de descifrado para recuperar los ficheros.

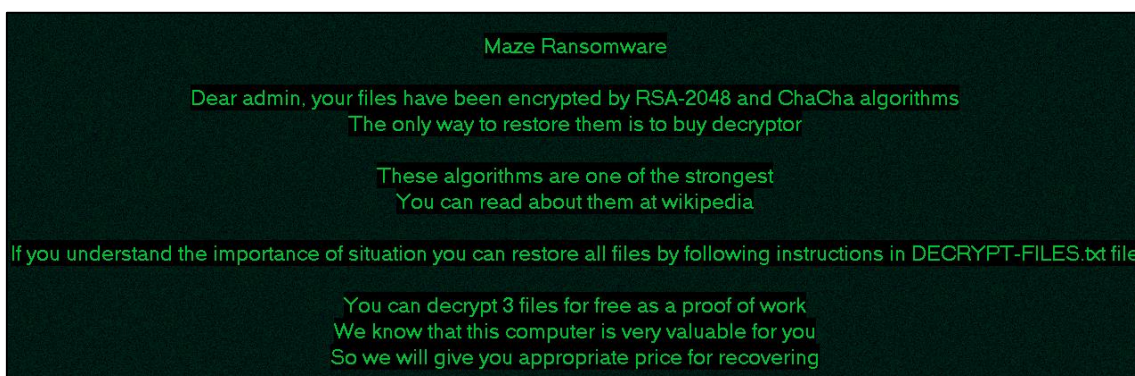


Figura 14. Fondo de escritorio establecido por el código dañino

Finalmente, el fichero **DECRYPT-FILES.txt**, que entre otras localizaciones se podrá encontrar en el escritorio, contiene las instrucciones del rescate.

6.2.9 COMUNICACIÓN

Mientras que investigadores han reportado previamente sobre la capacidad del código dañino de exfiltrar ficheros a servidores controlados por el grupo criminal, no se ha detectado tal comportamiento en la muestra analizada.

Cabe destacar, que los atacantes se encuentran dentro de la red en el momento de desplegar el ransomware, la exfiltración de información la pueden hacer por cualquier otra vía, sin necesidad de depender en el propio binario del ransomware para esta tarea.

7. RESCATE

Cuando un equipo se ve afectado por el ransomware Maze, las instrucciones para descifrar los ficheros se incluyen en la nota de rescate bajo el nombre **DECRYPT-FILES.txt**.

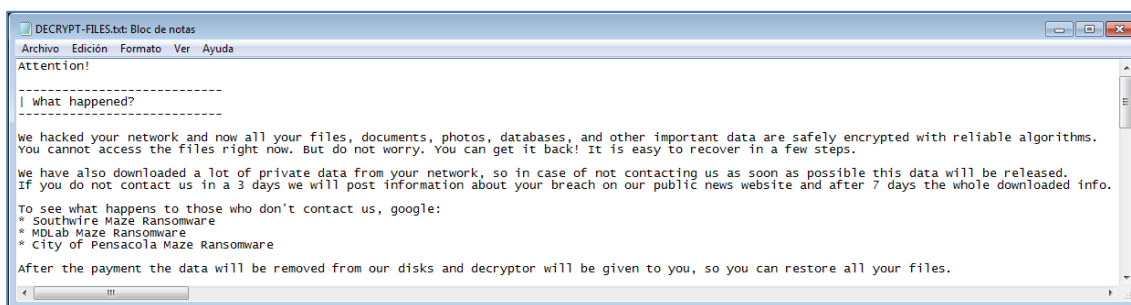


Figura 15. Explicación de los acontecimientos en la nota de rescate

En la nota se avisa de la intención de publicar la información privada exfiltrada y se facilitan los pasos para ponerse en contacto con el grupo Maze.

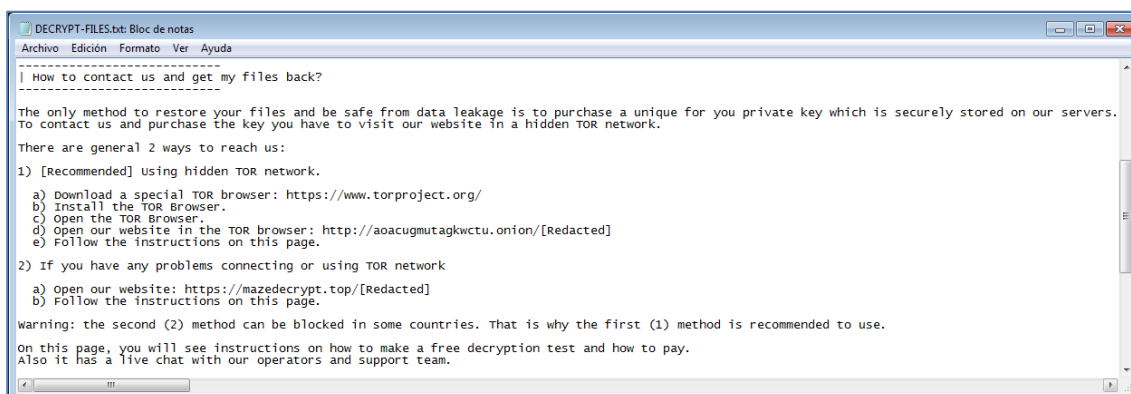


Figura 16. Pasos para contactar con el grupo ciber-criminal

En la parte final de la nota se encuentra la **MAZE KEY**.

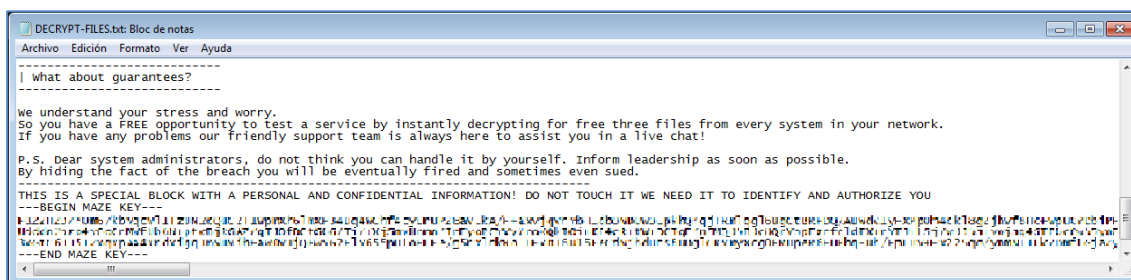


Figura 17. MAZE KEY incluida en la nota de rescate

Este artefacto, de forma similar al fichero **memes.tmp**, contiene la siguiente información.

- Clave RSA privada del usuario cifrada mediante ChaCha8.
- Clave ChaCha8 de 256-bit cifrada con la clave RSA pública del par máster.
- *Nonce* ChaCha8 de 64-bit cifrado con la clave RSA pública del par máster.
- Datos del usuario infectado como su ID, nombre de usuario y nombre de equipo, entre otros.



Como la parte atacante se encuentra en poder de la clave RSA privada del par máster, pueden descifrar la clave ChaCha8 y su *nonce* para finalmente descifrar la clave RSA privada del usuario y poder proceder al descifrado de los ficheros.

El fichero **DECRYPT-FILES.txt** es solicitado en el portal de pago para identificar al usuario y poder empezar la negociación.

Maze support system

What's just happened?

If you see this page it means you have a vulnerability in your system.
This vulnerability was used to modify your valuable data in a way, which temporary disallow further usage of it.
Please upload DECRYPT-FILES.txt using the form below and start recovering your data.
If this file is recognized by our parser, you will be successfully authorized and provided with further instructions.

Please upload **DECRYPT-FILES.txt**

No file selected.

Guarantees?

We can recover your files, as our software is carefully designed to keep the integrity and safety of your files.

Don't be afraid and start recovering!

Antivirus corporations?

If you are waiting for a free solution to come, we must disappoint you.

Our cryptography scheme is military grade. It will require decades to crack.

Start working with us and get your files back.

Price?

We understand that the customer cannot always pay the fee. We have discounts and price can be negotiated.

Figura 18. Portal de pago

Una vez autenticados, se accede a un chat en el que se negociará el precio del rescate y se facilita la dirección del *wallet* a la que hacer la transferencia, **3N8RFydNdQrQuzP9Uhbho29WPCTEJf3ekG** en este caso. Además de las recomendaciones para adquirir *bitcoins*, se ofrece el descifrado de tres ficheros de forma gratuita como garantía.

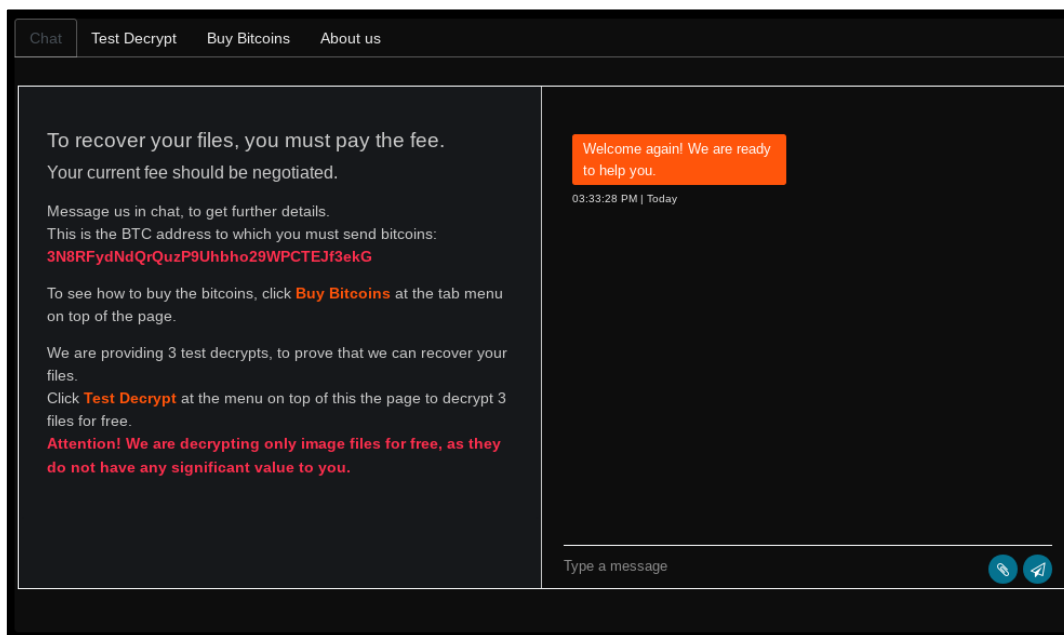


Figura 19. Chat para negociar el rescate

Llegados a este punto es preciso recordar que, por un lado, no existe ninguna garantía de que tras realizar el pago se facilite la herramienta de descifrado. Por otra parte, proceder con el pago de los rescates contribuye a la financiación del cibercrimen, potenciando que los ataques no solamente no cesen, si no que cuenten con más recursos para el desarrollo de su actividad.

8. DESINFECCIÓN

Dada la naturaleza del código dañino, no se requiere de un proceso de desinfección, puesto que no adquiere persistencia. El ejecutable responsable del cifrado no se elimina a sí mismo tras finalizar ejecución, por tanto, en caso de verse afectados por esta variante de ransomware convendría eliminar la muestra a fin de evitar futuras ejecuciones indeseadas.

Si se diese una infección por Maze, el fichero **memes.tmp** convendría no eliminarlo hasta asegurar que se ha expulsado al atacante de la red. En versiones anteriores del código dañino, este fichero se podía encontrar en la misma ruta bajo el nombre **0x29A.db**.

En el caso del cifrado de los ficheros, el modelo de criptografía utilizado garantiza que el descifrado sea únicamente posible mediante el uso de la clave privada del par master, que se encuentra en poder del grupo ciber-criminal.

9. REGLAS DE DETECCIÓN

9.1 REGLA YARA

```
rule maze
```



```
{
  meta:
    author = "CCN-CERT"
    date = "2020-05-18"
  strings:
    $w_ransom_note = "DECRYPT-FILES.txt" wide
    $w_command_line_1 = "--path" wide
    $w_command_line_2 = "--noshares" wide
    $w_command_line_3 = "--nomutex" wide
    $w_command_line_4 = "--logging" wide

    $proc_excel_exe = {28 05 78 48}
    $proc_ida_exe = {85 04 84 33}
    $proc_ida64_exe = {f1 04 e2 45}
    $proc_powerpnt_exe = {30 06 e0 6d}
    $proc_winword_exe = {d5 05 d8 60}
    $proc_x32dbg_exe = {3b 05 62 50}
    $proc_x64dbg_exe = {54 02 dc 50}

    $mod_kernel32_dll = "kernel32.dll" ascii
    $mod_advapi32_dll = "advapi32.dll" ascii
    $mod_user32_dll = "user32.dll" ascii
    $mod_gdi32_dll = "gdi32.dll" ascii
    $mod_ole32_dll = "ole32.dll" ascii

    condition: (uint16(0) == 0x5A4D and
      all of ($w*) and
      all of ($mod*) and
      3 of ($proc*))
}
```

10. INDICADORES DE COMPROMISO

Fichero	SHA256
Maze.exe	db617d3ca09f78673aef2a706a0161b9a7e160f58891f14a1e7250b39e3d9fb2



Nota de rescate

DECRYPT-FILES.txt

Artefacto con información sobre las claves

%ProgramData%\memes.tmp

Fondo de escritorio

%temp%\111.bmp

Exploit	SHA256
CVE-2016-7255 x64	953d574ef0f49e886e270c2887586aab5c0d37616dd99f422838a1257abdbe3b
CVE-2016-7255 x86	d339da79ac85ba14dcde6c0e2263e17b9382126bafff1a8a8248b071f2027e1f
CVE-2018-8453 x86-64	987bec3c79c8d19cfa4ec66e8e05ab8ccd8700f9d22083c9e61f596451e8f4d2