BlackCat Ransomware

BCSC-MALWARE-BLACKCAT

TLP:WHITE

www.basquecybersecurity.eus

TABLA DE CONTENIDO

Cláusula de exención de responsabilidad

Cláusula de prohibición de venta

SOBRE EL BCSC

El Centro Vasco de Ciberseguridad (Basque Cybersecurity Centre, BCSC) es la entidad designada por el Gobierno Vasco para elevar el nivel de madurez de la ciberseguridad en Euskadi.

Es una iniciativa transversal que se enmarca en la Agencia Vasca de Desarrollo Empresarial (SPRI), sociedad dependiente del Departamento de Desarrollo Económico, Sostenibilidad y Medio Ambiente del Gobierno Vasco. Así mismo, involucra a otros tres Departamentos del Gobierno Vasco: el de Seguridad, el de Gobernanza Pública y Autogobierno, y el de Educación, y a cuatro agentes de la Red Vasca de Ciencia, Tecnología e Innovación: Tecnalia, Vicomtech, Ikerlan y BCAM.

El BCSC es la entidad de referencia para el desarrollo de la ciberseguridad y de la confianza digital de ciudadanos, empresas e instituciones públicas en Euskadi, especialmente para los sectores estratégicos de la economía de la región.

La misión del BCSC es por tanto promover y desarrollar la ciberseguridad en la sociedad vasca, dinamizar la actividad empresarial de Euskadi y posibilitar la creación de un sector profesional que sea referente. En este contexto se impulsa la ejecución de proyectos de colaboración entre actores complementarios en los ámbitos de innovación tecnológica, investigación y transferencia tecnológica a la industria de fabricación avanzada y otros sectores.

Así mismo, ofrece diferentes servicios en su rol como Equipo de Repuesta a Incidentes (en adelante CERT, por sus siglas en inglés “Computer Emergency Response Team”) y trabaja en el ámbito de la Comunidad Autónoma del País Vasco para aumentar la capacidad de detección y alerta temprana de nuevas amenazas, la respuesta y análisis de incidentes de seguridad de la información, y el diseño de medidas preventivas para atender a las necesidades de la sociedad vasca. Con el fin de alcanzar estos objetivos forma parte de diferentes iniciativas orientadas a la gestión de incidentes de ciberseguridad:

1. RESUMEN EJECUTIVO

BlackCat, o también conocido como *ALPHV*, se trata de una nueva versión del antiguo *ransomware BlackMatter*, en el que se ha reescrito todo su código en Rust, un lenguaje de programación más seguro, más sencillo de desarrollar, concurrente, multiplataforma y multiparadigma.

Su primera aparición data de noviembre del 2021 y un alto porcentaje de las compañías que se han visto afectadas se encuentran en Estados Unidos, aunque no ha sido el único país afectado.

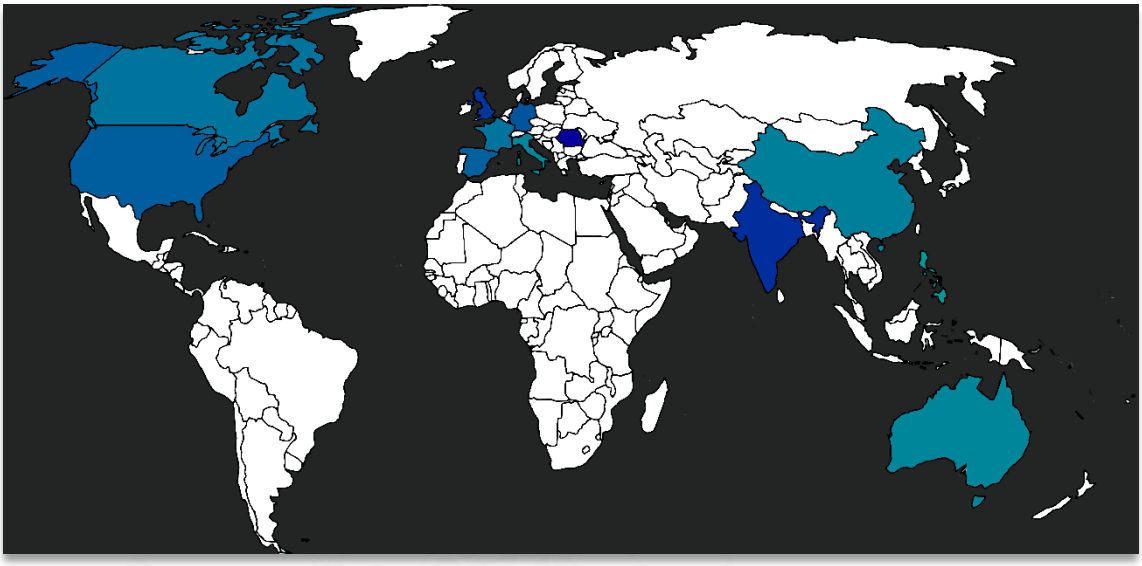

Ilustración 1: Todos los países afectados por BlackCat

Tras el análisis realizado a *BlackMatter* en enero, se ha podido observar que ambas familias utilizan como tipo de cifrado una combinación de Chacha20 y RSA. Además, varias empresas de seguridad han notificado que los dominios utilizados durante los ataques de *BlackMatter* se han visto reutilizados durante los ataques de *BlackCat* relacionando de forma directa a ambas familias.

Está información revela que el grupo encargado del desarrollo de este *ransomware* se encuentra muy activo pues en menos de un año han reescrito totalmente el código a otro lenguaje y mantienen su servicio actualizado.

Al igual que la gran mayoría del *ransomware* de la actualidad, el grupo detrás de *BlackCat* ofrecen su herramienta como un servicio (RaaS) un sistema que ha ido ganando un gran público a lo largo de los años.

Durante el análisis de esta nueva familia se ha podido acceder a la plataforma donde los atacantes publican la información extraída durante el ataque. En ella se pueden observar las diferentes empresas que se han visto afectadas debido a que no utilizan un dominio distinto por cada uno de los ataques.

Por otro lado, la web utilizada por los atacantes para explicar a las víctimas cómo puede realizar el descifrado de sus ficheros, se encuentra caída y, por lo tanto, no se ha podido comprobar el funcionamiento de la misma.

Finalmente, y debido a que la última versión de *BlackCat* no permite la ejecución de la muestra sin el conocimiento previo del valor "*--access-token*", el análisis se ha basado en la versión previa y se ha realizado una comparación final de ambas versiones donde se han verificado los cambios realizados entre ellas.

Las principales características de esta familia son:

- Eliminación de las copias de seguridad locales.
- Eliminación de los *snapshots* en sistemas ESXi.
- Eliminación de los eventos del sistema.
- Configuración en formato JSON cifrada con parte del parámetro "*--access-token*".
- Propagación por la red, haciendo uso de PsExec y credenciales de administrador.
- Uso combinado de los algoritmos Salsa20 y RSA, para la realización del cifrado de los ficheros.

2. ANÁLISIS TÉCNICO

2.1. Flujo de infección

Ilustración 2: Flujo de infección.

Teniendo en cuenta la forma en que se han realizado los ataques con este *malware*, el flujo de infección que termina derivando en que la detonación del *ransomware* puede variar de unos casos a otros.

Dado que el *ransomware* no posee altas capacidades para propagarse a través de Internet, el proceso de compromiso inicial es llevado a cabo por operadores humanos. Por otro lado, esta nueva familia de *ransomware* viene preparada para realizar un despliegue automático a través de la red interna, realizando una exploración de todos los dispositivos de red accesibles e intentando realizar la propagación haciendo uso de PsExec y el conocimiento previo de credenciales de Administrador. No obstante, deben tenerse en cuenta todas las opciones que puedan terminar derivando en una ejecución de código malicioso, como la explotación de vulnerabilidades, el envío de correos con adjuntos maliciosos o el uso de *exploit kits*.

Una vez comprometido el sistema, los atacantes recopilan credenciales e información sobre la víctima hasta que deciden ejecutar el *ransomware* que cifrará toda la información y con el cual habrá concluido el ataque.

Al igual que otros operadores de *ransomware*, el grupo detrás de *BlackCat* estaría tratando de llevar a cabo un modelo de doble extorsión. Siguiendo este modelo, además de reclamar una suma de dinero en criptomonedas a cambio de descifrar la información, amenazan con filtrar los datos que han robado,

venderlos al mejor postor si las víctimas se niegan a pagar o sencillamente haciéndolos accesibles al público y devaluando la marca.

2.2. Análisis técnico

Desde la aparición de *BlackCat* se ha producido una actualización importante: la primera versión contenía la configuración del *ransomware* embebida como una cadena de texto en formato JSON en la cual se encontraba toda la información necesaria para su ejecución. Actualmente, esa información viene cifrada dentro del código y la clave de descifrado se crea a partir de un parámetro de entrada.

Las muestras analizadas tienen los siguientes hashes SHA256:

SHA256	Descripción
731adcf2d7fb61a8335e23dbee2436249e5d5753977ec465754c6b699e9bf161	*BlackCat* - Binario PE de Windows V1
bd4e603e953d8c7803f3c7d72cd7197d996ab80ce80b9da96a4df7d10969bb55	*BlackCat* – Binario PE de Windows V2

2.1.1. BlackCat – V1

La muestra de *BlackCat* analizada no se encuentra empaquetada por lo que es posible acceder al desensamblado del código original del *malware*. Se trata de un binario PE ejecutable para sistemas Microsoft Windows y arquitectura de 32 bits:

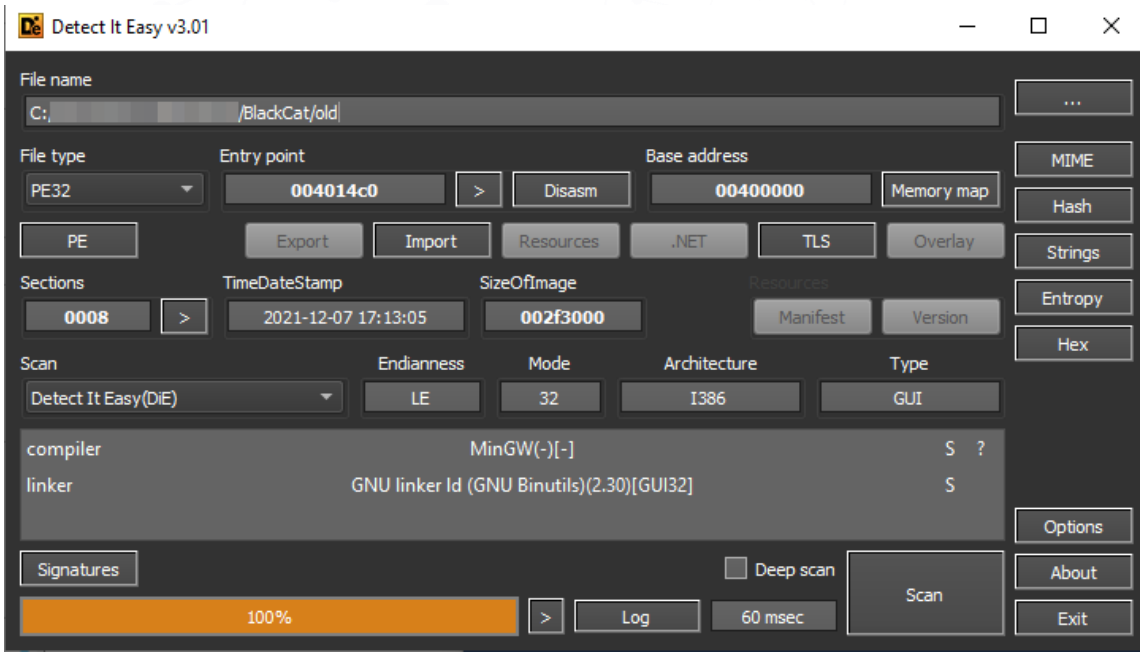

Ilustración 3: Información estática de la muestra.

Además, se encuentra desarrollado en Rust, un lenguaje de programación de tipo compilado que permite programación funcional por procedimientos, imperativa y orientada a objetos. Debido al uso de Rust, es importante resaltar que su seguridad se centra en que es complicado analizar realizando ingeniería inversa:

```
set socket configuration (3 matches)          | communication/socket
create UDP socket                             | communication/socket/udp/send
compiled with rust                            | compiler/rust
compute adler32 checksum                      | data-manipulation/checksum/adler32
encode data using Base64 (4 matches)          | data-manipulation/encoding/base64
```

Ilustración 4: Ejemplo de función que no se resuelve fácilmente lo que complica su análisis.

2.1.1.1 Ejecución

El *ransomware* está programado para ser ejecutado de forma manual y, una vez ejecutado, se comporta como una aplicación de escritorio en formato de consola de comandos que, incluso, muestra información de depuración durante su ejecución, según se ha podido observar en el código fuente:

```
USAGE:
    [OPTIONS] [SUBCOMMAND]

OPTIONS:
        --access-token <ACCESS_TOKEN>                Access Token
        --bypass <BYPASS>...
        --child                                      Run as child process
        --drag-and-drop                              Invoked with drag and drop
        --drop-drag-and-drop-target                  Drop drag and drop target batch file
    -h, --help                                       Print help information
        --log-file <LOG_FILE>                        Enable logging to specified file
        --no-net                                     Do not discover network shares on Windows
        --no-prop                                    Do not self propagate(worm) on Windows
        --no-prop-servers <NO_PROP_SERVERS>...       Do not propagate to defined servers
        --no-vm-kill                                 Do not stop VMs on ESXi
        --no-vm-kill-names <NO_VM_KILL_NAMES>...     Do not stop defined VMs on ESXi
        --no-vm-snapshot-kill                        Do not wipe VMs snapshots on ESXi
        --no-wall                                    Do not update desktop wallpaper on Windows
    -p, --paths <PATHS>...                           Only process files inside defined paths
        --propagated                                 Run as propagated process
        --ui                                         Show user interface
    -v, --verbose                                    Log to console
```

Ilustración 5: Parámetros que admite el malware.

En la imagen anterior se pueden apreciar todas las posibles configuraciones a la hora de ejecutar *BlackCat.*

Para poder realizar la ejecución de *BlackCat* es necesario introducir el valor del parámetro "*--access-token*". Sin este parámetro el binario no se puede ejecutar. Este tipo de *malware* está enfocado para ser desplegado por un operador y no para propagarse de forma descontrolada como podría ser *Ryuk*, que se desplegaba por otras familias de malware por *Trickbot* y *Emotet*.

Durante las pruebas de ejecución se ha podido comprobar el funcionamiento del sistema encargado de indicar al operador el estado de la ejecución. Para ello se puede hacer uso de los parámetros "*--log-file <fichero>*" o "*--verbose*": en el primer caso se guardará la información en un fichero de texto mientras que con la segunda opción se mostrará directamente por la consola:

```
15:39:16 MASTER [INFO] locker::core::discoverer: Fixing Permissions -> C:\Users\Default User\Menú Inicio\Programas
15:39:16 MASTER [INFO] locker::core::discoverer: Fixing Permissions -> C:\Users\Default User\Menú Inicio\Programs
15:39:16 MASTER [INFO] locker::core::discoverer: Fixing Permissions -> C:\Users\Default User\Menú Inicio\Programs\Accessibility
15:39:16 MASTER [INFO] locker::core::discoverer: Fixing Permissions -> C:\Users\Default User\Menú Inicio\Programs\Accessories
15:39:16 MASTER [INFO] locker::core::discoverer: Fixing Permissions -> C:\Users\Default User\Menú Inicio\Programs\System Tools
15:39:16 MASTER [INFO] locker::core::discoverer: Already Traversed -> \\?\C:\Users\Default\AppData\Roaming\Microsoft\Windows\Start Menu\Programs
15:39:16 MASTER [INFO] locker::core::discoverer: Already Traversed -> \\?\C:\Users\Default\Documents
15:39:16 MASTER [INFO] locker::core::discoverer: Fixing Permissions -> C:\Users\Default User\Datos de programa\Microsoft\Internet Explorer\Quick Launch
15:39:16 MASTER [INFO] locker::core::discoverer: Already Traversed -> \\?\C:\Users\Default\AppData\Local
15:39:16 MASTER [INFO] locker::core::discoverer: Already Traversed -> \\?\C:\Users\Default\AppData\Local\Microsoft\Windows\INetCache\IE
15:39:17 MASTER [INFO] locker::core::renderer: Speed: 69.10 Mb/s, Data: 5236Mb/5236Mb, Files processed: 57894/57894, Files scanned: 273162
15:39:19 MASTER [INFO] locker::core::renderer: Speed: 68.19 Mb/s, Data: 5308Mb/5308Mb, Files processed: 59742/59742, Files scanned: 273162
15:39:21 MASTER [INFO] locker::core::renderer: Speed: 66.81 Mb/s, Data: 5341Mb/5341Mb, Files processed: 62092/62092, Files scanned: 273162
15:39:23 MASTER [INFO] locker::core::renderer: Speed: 65.13 Mb/s, Data: 5342Mb/5342Mb, Files processed: 63827/63828, Files scanned: 273162
```

Ilustración 6: Ejemplo de información almacenada dentro del fichero de log.

```
PS C:\Users\[redacted]Desktop\BlackCat> .\old.exe --access-token 1231 --verbose
PS C:\Users\[redacted]Desktop\BlackCat> 15:59:58 MASTER [INFO] locker::core::stack: Starting Superviso
r
15:59:58 MASTER [INFO] locker::core::stack: Starting Discoverern
15:59:58 MASTER [INFO] locker::core::stack: Starting File Unlockers
15:59:58 MASTER [INFO] locker::core::stack: Starting File Processing Pipeline
15:59:58 MASTER [INFO] locker::core::pipeline::chunk_workers_supervisor: spawned_workers=6
15:59:58 MASTER [INFO] locker::core::pipeline::file_worker_pool: spawned_file_dispatchers=3
15:59:58 MASTER [INFO] locker::core::pipeline::file_worker_pool: spawned_chunk_work_infastructure=2
15:59:58 MASTER [INFO] locker::core::stack: Detecting Other Instances specified file
15:59:58 MASTER [INFO] locker::core::stack: Starting Cluster Servicenetwork shares on Windows
15:59:58 MASTER [INFO] locker::core::stack: Connecting to Clusterropagate(worm) on Windows
15:59:58 MASTER [INFO] locker::core::cluster: server=16718247293466428226efined servers
15:59:58 MASTER [INFO] locker::core::stack: This is a Master Processon ESXi
15:59:58 MASTER [INFO] locker::core::stack: Starting Platformop defined VMs on ESXi
15:59:58 MASTER [INFO] encrypt_app::windows: Bootstrap Routinee VMs snapshots on ESXi
15:59:58 MASTER [INFO] locker::core::os::windows::privilege_escalation: win7_plus=true Windows
15:59:58 MASTER [INFO] locker::core::os::windows::privilege_escalation: token_is_admin=false
15:59:58 MASTER [INFO] locker::core::os::windows::privilege_escalation: token_is_domain_admin=true
15:59:58 MASTER [INFO] locker::core::os::windows::privilege_escalation: masquerade_peb
15:59:58 MASTER [INFO] locker::core::os::windows::privilege_escalation: uac_bypass::shell_exec="C:\\U
sers\[redacted]\\Desktop\\BlackCat\\old.exe",Some("\"--access-token\" \"1231\" \"--verbose\""),Some("C
:\\Users\\[redacted]\\Desktop\\BlackCat")
15:59:59 MASTER [INFO] locker::core::os::windows::privilege_escalation: escalate=success
```

Ilustración 7: Ejemplo de información mostrada por consola.

Una vez finalizada la ejecución, y si no se ha modificado ninguno de los parámetros iniciales, el equipo afectado queda con la siguiente apariencia:

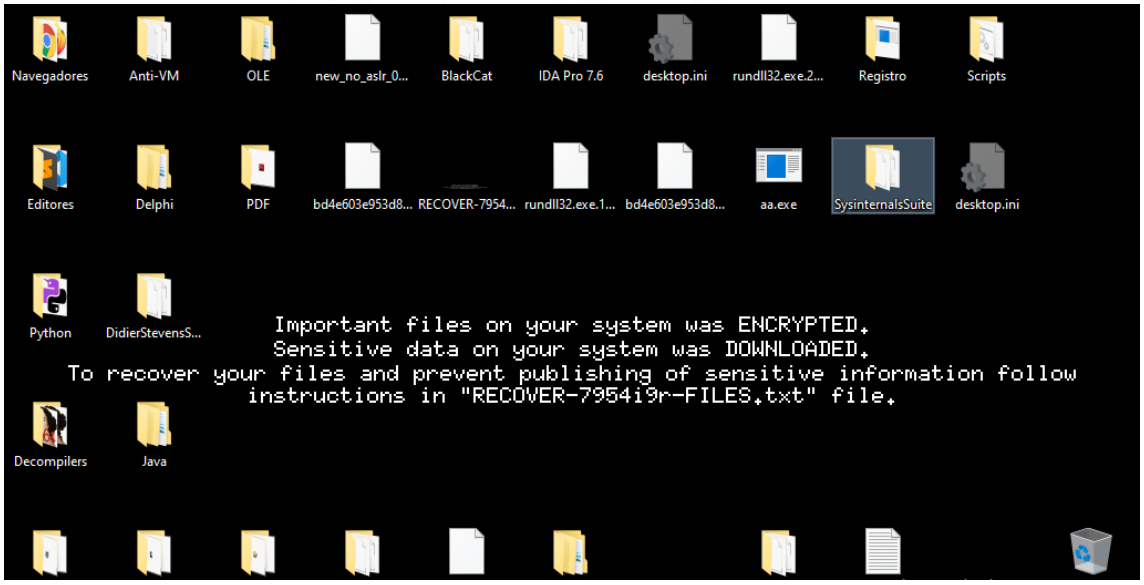

Ilustración 8: Escritorio de la máquina, tras finalizar el proceso de cifrado.

Por otro lado, la nota de rescate escrita en cada uno de los directorios por donde el proceso de cifrado ha pasado, es la siguiente:

>> Introduction Important files on your system was ENCRYPTED and now they have have "7954i9r" extension. In order to recover your files you need to follow instructions below. >> Sensitive Data

Sensitive data on your system was DOWNLOADED and it will be PUBLISHED if you refuse to cooperate.

Data includes:
- Employees personal data, CVs, DL, SSN.
- Complete network map including credentials for local and remote services.
- Financial information including clients data, bills, budgets, annual reports, bank statements.
- Complete datagrams/schemas/drawings for manufacturing in solidworks format
- And more...

Private preview is published here: **<URL>**

>> CAUTION

DO NOT MODIFY FILES YOURSELF.
DO NOT USE THIRD PARTY SOFTWARE TO RESTORE YOUR DATA.
YOU MAY DAMAGE YOUR FILES, IT WILL RESULT IN PERMANENT DATA LOSS.
YOUR DATA IS STRONGLY ENCRYPTED, YOU CAN NOT DECRYPT IT WITHOUT CIPHER KEY.

>> Recovery procedure

Follow these simple steps to get in touch and recover your data:
1) Download and install Tor Browser from: https://torproject.org/
2) Navigate to: **<URL2>**

Dentro de la primera dirección URL perteneciente al dominio "*hxxp://alphvmmm27o3abo3r2mlmjrpdmzle3rykajqc5xsj7j7ejksbpsa36ad[.]onion/*". Se puede observar la información que han exfiltrado los atacantes antes de desplegar BlackCat y que todos los ficheros del sistema quedasen inutilizados:

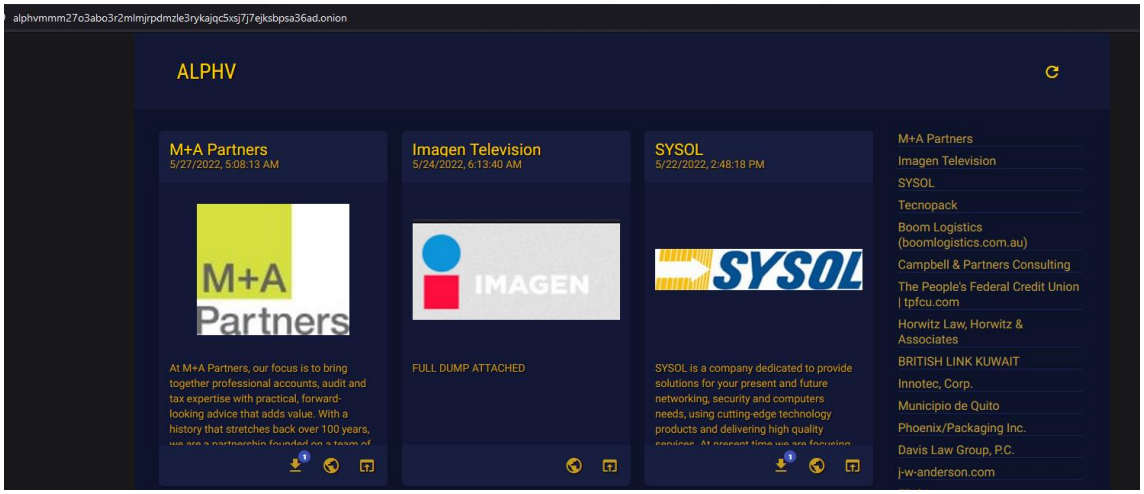

Ilustración 9: Ejemplos de algunas corporaciones afectadas por este ransomware.

Dentro de la dirección URL se encuentra una cadena de texto que sigue el formato UUID con el cual el atacante identifica el ataque. Por ejemplo, en la siguiente imagen se puede apreciar la dirección asociada al ataque recibido por la empresa "*British Link Kuwait*" y toda la información que han obtenido sobre ellos y que han hecho pública a través de su web:

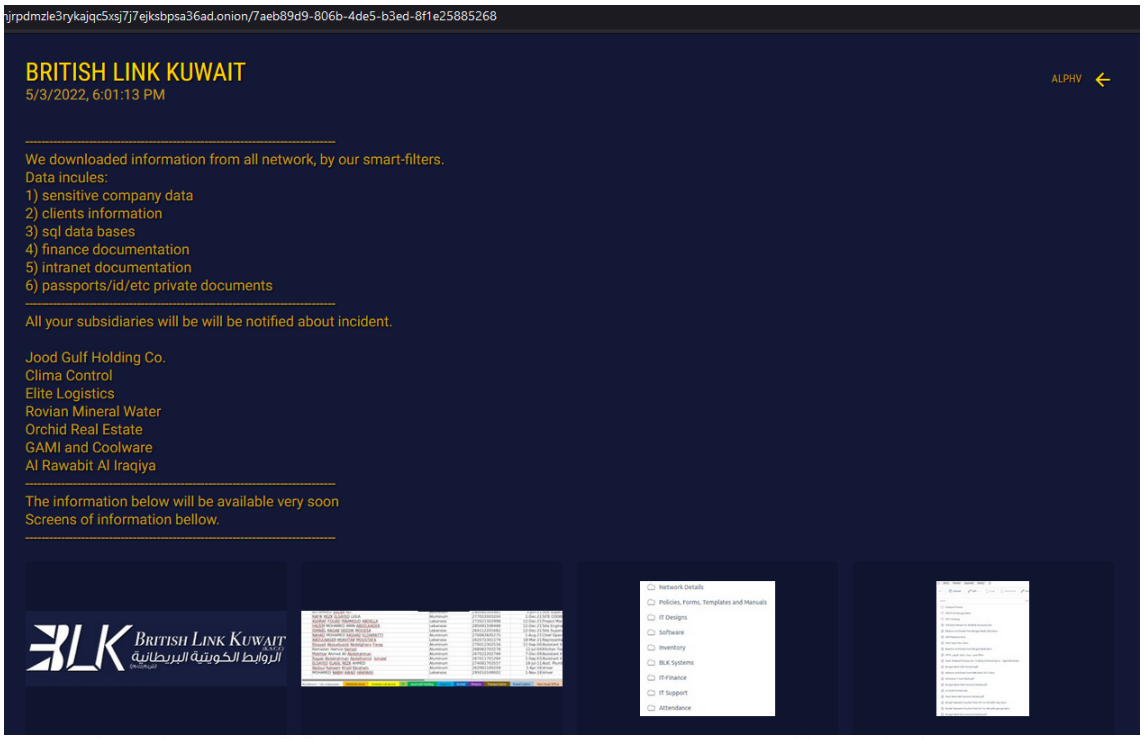

Ilustración 10: Ejemplo de empresa afectada por un ataque.

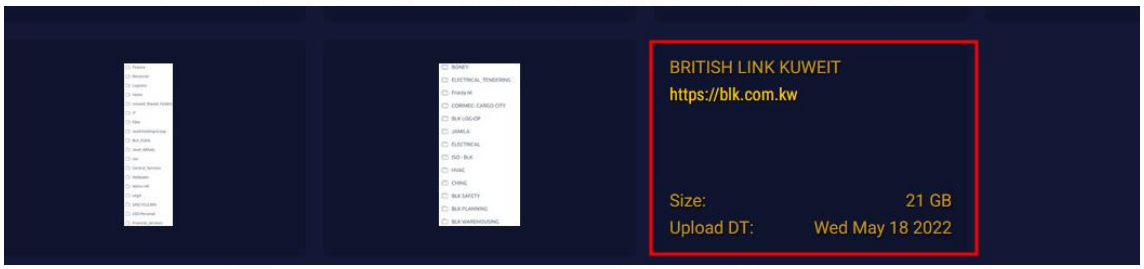

Ilustración 11: Información descargable desde la web.

Esta metodología es usada frecuentemente en este tipo de ataques de *ransomware* para extorsionar a las víctimas, provocando de esta forma que el pago no solo se produzca para recuperar sus datos, sino también para evitar que se publique información de carácter sensible en internet y aumentando así la gravedad del incidente.

Por la información que se ha podido encontrar por la red, la fase de exfiltración se realiza de forma manual y no es parte del proceso de ejecución del *ransomware.* En la publicación realizada por **Cisco Talos Intelligence Group** se detalla mucha información del proceso previo a la ejecución del *ransomware*, pudiendo observar tácticas, técnicas y procedimientos durante sus ataques:

https://blog.talosintelligence.com/2022/03/from-blackmatter-to-blackcat-analyzing.html

2.1.1.2 Análisis del código fuente.

Como se indicaba en el apartado de ejecución, *BlackCat* contiene un sistema que permite conocer el estado actual de la ejecución. Para ello hace uso de una serie de cadenas de texto que muestra por consola o que escribe en fichero de texto. Por lo tanto, se ha comenzado el análisis a partir de las cadenas de texto, para averiguar si las cadenas se encuentran cifradas o no:

Address	Length	Type	String
.rdata:0061...	00000014	C	Starting Supervisor
.rdata:0061...	00000014	C	Starting Discoverer
.rdata:0061...	00000018	C	Starting File Unlockers
.rdata:0061...	00000022	C	Starting File Processing Pipeline
.rdata:0061...	00000018	C	Starting Cluster Service
.rdata:0061...	00000012	C	Starting Renderer
.rdata:0061...	00000012	C	Starting Platform

Ilustración 12: Algunas de las cadenas de texto vistas durante la ejecución del ransomware.

El sistema utilizado por Rust para las cadenas de texto, difiere bastante del sistema utilizado por C/C++. En el caso del lenguaje Rust pueden existir cadenas que no terminen con el carácter “\x00” o también conocido como **NULL** y esto se trata sencillamente porque hace uso de estructuras intermedias las cuales guardan la referencia a la *string* y el tamaño. De esta forma se puede recuperar la cadena sin producir perdida de información o que supongo un problema que la cadena contenga caracteres **NULL**:

```
.rdata:00619D5C aInvalidConfig  db 'Invalid config.'     ; DATA XREF: .rdata:00619D6C↓o
.rdata:00619D6B                 db 10
.rdata:00619D6C                 dd offset aInvalidConfig
.rdata:00619D6C                                          ; DATA XREF: mw_parse_arguments+4EAC↑o
.rdata:00619D6C                                          ; "Invalid config."
.rdata:00619D70                 dd 16                    ; Size
.rdata:00619D74 aInvalidPublicK db 'Invalid public key.',0Ah
.rdata:00619D74                                          ; DATA XREF: .rdata:off_619D88↓o
.rdata:00619D88 off_619D88      dd offset aInvalidPublicK
.rdata:00619D88                                          ; DATA XREF: mw_parse_arguments+4DE9↑o
.rdata:00619D88                                          ; "Invalid public key.\n"
.rdata:00619D8C                 dd 20                    ; Size
```

Ilustración 13: En los recuadros rojos se puede apreciar la estructura utilizada por las cadenas de texto y como no existe el valor "\x00" para indicar el final de la cadena.

Analizando el resto de las cadenas visibles, llama la atención una cadena de gran tamaño y en formato JSON:

```
.rdata:0061... 00000010  C  Invalid config.\n
.rdata:0061... 00001FDB  C  {\"config_id\":\"\",\"public_key\":\"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApw3tWdMaWJvNf2Mejy5H0Y6kuj+lstNpwFyismGDEYhWKPps9c68xl+84o6uLKfqPzNvLnSxl
.rdata:0061... 0000044C  C  access-tokenpathsno-netno-propno-wallno-vm-killno-vm-snapshot-killno-vm-kill-namesno-prop-serverspropagatedchilddrop-drag-and-drop-targetdrag-and-droplog-fileverboseuibypas
.rdata:0061... 00000040  C  Bit region conflict: i=, old_value=, new_value=, symbol=, code=
.rdata:0067... 00000028  C  config_idWritingWaitingIOProcessingIdle
```

Ilustración 14: Configuración de BlackCat

El valor completo de la cadena es el siguiente (*por motivos de privacidad hacía la empresa afectada por el ataque, se han borrado información sensible, como por ejemplo credenciales*):

```
{
  "config_id": "",
  "public_key":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApw3tWdMaWJvNf2Mejy5H0Y6kuj+lstN
```

```
pwFyismGDEYhWKPps9c68xl+84o6uLKfqPzNvLnSxlVa6DitcJGeKJEQkzN+C1e1KsfzM63jHybR
EB2hs+dHbqBq4dbamIQcTrrr4mKzuHJ7aok4mlpRx2Un1XOJaodoV7xOHO7ui5v6uK39MJ3rvitS
EBvv5oI0WDlp3IFmtd6UM6r2nygY1ncAUuasalZgF1Vaz7VXOWyX2ReQHbYWWRCR1qyKMQcBtjT5
POXx9B8ek1pnU4p65kGe9M794Bhhh20GN24gY5a+zwXwstaNTO9luwd4xjjRQAVsDgjrjkzti27G
11ICn6wIDAQAB",
  "extension": "7954i9r",
  "note_file_name": "RECOVER-${EXTENSION}-FILES.txt",
  "note_full_text": ">> Introduction\n\nImportant files on your system was
ENCRYPTED and now they have have \"${EXTENSION}\" extension.\nIn order to
recover your files you need to follow instructions below.\n\n>> Sensitive
Data\n\nSensitive data on your system was DOWNLOADED and it will be
PUBLISHED if you refuse to cooperate.\n\nData includes:\n- Employees
personal data, CVs, DL, SSN.\n- Complete network map including credentials
for local and remote services.\n- Financial information including clients
data, bills, budgets, annual reports, bank statements.\n- Complete
datagrams/schemas/drawings for manufacturing in solidworks format\n- And
more...\n\nPrivate preview is published here:
http://alphvmmm27o3abo3r2mlmjrpdmzle3rykajqc5xsj7j7ejksbpsa36ad.onion/******
*****************************************\n\n\n>> CAUTION\n\nDO NOT
MODIFY FILES YOURSELF.\nDO NOT USE THIRD PARTY SOFTWARE TO RESTORE YOUR
DATA.\nYOU MAY DAMAGE YOUR FILES, IT WILL RESULT IN PERMANENT DATA
LOSS.\nYOUR DATA IS STRONGLY ENCRYPTED, YOU CAN NOT DECRYPT IT WITHOUT
CIPHER KEY.\n\n>> Recovery procedure\n\nFollow these simple steps to get in
touch and recover your data:\n1) Download and install Tor Browser from:
https://torproject.org/\n2) Navigate to:
http://sty5r4hhb5oihbq2mwevrofdiqbgesi66rvxr5sr573xgvtuvr4cs5yd.onion/?acces
s-key=${ACCESS_KEY}",
  "note_short_text": "Important files on your system was
ENCRYPTED.\nSensitive data on your system was DOWNLOADED.\nTo recover your
files and prevent publishing of sensitive information follow instructions in
\"${NOTE_FILE_NAME}\" file.",
  "default_file_mode": "Auto",
  "default_file_cipher": "Best",
  "credentials": [
    [
      "C*E****NE\\Ad****** ","K***@***9"
    ],
  ],
  "kill_services": [

"mepocs","memtas","veeam","svc$","backup","sql","vss","msexchange","sql*"
  ],
  "kill_processes": [

"encsvc","thebat","mydesktopqos","xfssvccon","firefox","infopath","winword",
"steam","synctime","notepad","ocomm","onenote","mspub","thunderbird","agntsv
c","sql","excel","powerpnt","outlook","wordpad","dbeng50","isqlplussvc","sqb
coreservice","oracle","ocautoupds","dbsnmp","msaccess","tbirdconfig","ocssd"
,"mydesktopservice","visio","sql*"
  ],
  "exclude_directory_names": [
    "system volume information","intel","$windows.~ws","application
data","$recycle.bin","mozilla","program files (x86)","program
files","$windows.~bt","public","msocache","windows","default","all
users","tor
browser","programdata","boot","config.msi","google","perflogs","appdata","wi
ndows.old"
  ],
  "exclude_file_names": [

"desktop.ini","autorun.inf","ntldr","bootsect.bak","thumbs.db","boot.ini","n
tuser.dat","iconcache.db","bootfont.bin","ntuser.ini","ntuser.dat.log"
  ],
  "exclude_file_extensions": [

"themepack","nls","diagpkg","msi","lnk","exe","cab","scr","bat","drv","rtp",
"msp","prf","msc","ico","key","ocx","diagcab","diagcfg","pdb","wpx","hlp","i
```

```
cns","rom","dll","msstyles","mod","ps1","ics","hta","bin","cmd","ani","386",
"lock","cur","idx","sys","com","deskthemepack","shs","ldf","theme","mpa","no
media","spl","cpl","adv","icl","msu"
  ],
  "exclude_file_path_wildcard": [],
  "enable_network_discovery": true,
  "enable_self_propagation": true,
  "enable_set_wallpaper": true,
  "enable_esxi_vm_kill": true,
  "enable_esxi_vm_snapshot_kill": true,
  "strict_include_paths": [],
  "esxi_vm_kill_exclude": []
}
```

Como se puede apreciar en el anterior cuadro de texto, esta nueva familia de malware es altamente configurable y de una forma muy sencilla, pues solamente editando este texto, puede controlar todo el comportamiento del malware, además como ya se vio en el apartado de ejecución, se pueden cambiar algunos de esos valores previamente configurados dentro del binario a través de los argumentos de ejecución, por ejemplo:

- Activar/Desactivar el cambio del fondo de escritorio.
- Activar/Desactivar el descubrimiento de red.
- Activar/Desactivar la auto-propagación.
- Activar/Desactivar el apagar las máquinas virtuales de un sistema ESXI
- Activar/Desactivar el borrado de los snapshots de las máquinas virtuales.

Otras partes interesantes que se encuentran dentro de la configuración son los valores contenidos en:

- "*public_key*": Contiene el valor de la clave pública RSA escrito en formato base64, utilizada para cifrar las claves generadas de forma aleatoria.
- "*note_file_name*": Contiene el formato utilizado para generar los nombres de los ficheros de la nota de rescate.
- "*note_full_text*": Contiene todo el formato del texto utilizado para generar las notas de rescate.
- "*credentials*": Contiene un vector con credenciales validas dentro del Controlador de Dominio o cuentas de administración local. Este listado, se utilizará durante el proceso de propagación.

Por otro lado, el algoritmo de cifrado utilizado para el contenido de los ficheros, se trata de Salsa20/ChaCha20. Esto es debido a que se utiliza la cadena "*expand 32-byte k*" como constante para la inicialización del cifrador:

```
1  __m128i *__fastcall mw_chacha20_key_setup(const __m128i *a1, __m128i *a2)
2  {
3    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]
4
5    v195 = a2;
6    v188 = 'k et';
7    v190 = 6;
8    qmemcpy(v180, "nd 32-byexpa", sizeof(v180));
9    qmemcpy(v174, "expand 3expa", sizeof(v174));
10   qmemcpy(v182, "nd 32-byte k2-bynd 32-byte k", sizeof(v182));
11   qmemcpy(v172, "expate k", sizeof(v172));
12   v194 = a1;
13   v138 = a1[2].m128i_i32[0];
14   v133 = a1[2].m128i_i32[1];
15   v168 = a1[1].m128i_i32[0];
16   LODWORD(v84) = a1->m128i_i32[0];
17   v81 = a1->m128i_i32[0];
18   v88 = a1->m128i_i32[0];
19   v102 = a1->m128i_i32[0];
20   v147 = (unsigned __int64)(a1[2].m128i_i64[0] + 1) >> 32;
21   v151 = v138 + 1;
22   v164 = a1[1].m128i_i32[1];
23   v2 = a1[1].m128i_i32[2];
24   v129 = v138 + 2;
25   v3 = a1[2].m128i_i64[0] + 3;
26   v63 = v3;
27   v159 = (unsigned __int64)(a1[2].m128i_i64[0] + 2) >> 32;
28   m128i_i64 = a1[2].m128i_i64;
29   v92 = HIDWORD(v3);
30   v155 = v2;
```

Ilustración 15: Código con las constantes de inicialización.

Ésta es utilizada después de llamar a la función "*BCryptGenRandom*" encargada de generar un valor aleatorio del tamaño indicado a través de su tercer argumento:

```
6    v4 = BCryptGenRandom(0, (PUCHAR)&random_key, 32u, 2u);
7    if ( v4 < 0xC0000000 || (v5 = v4 ^ 0x80000000) == 0 )// if Correct random key
8    {
9      v8 = *(_OWORD *)&random_key.m256_f32[1];
10     v9 = *(_QWORD *)&random_key.m256_f32[5];
11     v10 = random_key.m256_f32[7];
12     *(float *)a1 = random_key.m256_f32[0];
13     *(_OWORD *)(a1 + 4) = v8;
14     *(_QWORD *)(a1 + 20) = v9;
15     *(float *)(a1 + 28) = v10;
16     *(_DWORD *)(a1 + 36) = 0;
17     *(_DWORD *)(a1 + 32) = 0;
18     *(_DWORD *)(a1 + 44) = 0;
19     *(_DWORD *)(a1 + 40) = 0;
20     goto Todo_bien;
21   }
22   ProcessHeap = hHeap;
23   if ( !hHeap )
24   {
25     ProcessHeap = GetProcessHeap();
26     if ( !ProcessHeap )
27       goto error;
28     hHeap = ProcessHeap;
29   }
30   v7 = HeapAlloc(ProcessHeap, 0, 4u);
31   if ( !v7 )
32 error:
33     mw_fastFail();
34   *v7 = v5;
35   HeapFree(hHeap, 0, v7);
36 Todo_bien:
37   *(_DWORD *)(a1 + 64) = 0;
38   v11 = (*(unsigned int *)(a1 + 48) | 0xFFFFFFFF00000000LL) + __PAIR64__(*(_DWORD *)(a1 + 52), -256);
39   *(_DWORD *)(a1 + 56) = *(_DWORD *)(a1 + 48) - 256;
40   *(_DWORD *)(a1 + 60) = HIDWORD(v11);
41   return mw_chacha20_key_setup((const __m128i *)a1, a2);
```

Ilustración 16: Generación de un valor aleatorio.

La función encargada de matar los servicios del sistema, que coincidan con aquellos que aparecen en el listado de la configuración, se ha encontrado gracias a la cadena "*Killing Services*". Dentro de esta función se han encontrado llamadas a las funciones "*OpenSCManagerW*", "*EnumServicesStatusExW*", "*EnumDependServicesW*", "*ControlService*" y "*OpenServiceW*". Con estas llamadas *BlackCat* puede conocer el estado de un servicio y pararlo.

```
    v46 = off_5F101C;
  ((void (__cdecl *)(const char *, struct _FILETIME *))v47[5])(v46, &SystemTimeAsFileTime);
}
memset(&ServiceStatus, 0, sizeof(ServiceStatus));
if ( !ControlService(*(SC_HANDLE *)(v5 + 12), 1u, &ServiceStatus) )// Try_to_stop_Service
{
  if ( (unsigned int)dword_6DE1DC >= 3 )
  {
```

Ilustración 17: Llamadas para parar un servicio.

Con la misma metodología también se ha encontrado la función encargada de matar procesos en ejecución ("*Killing Processes*"):

```
Toolhelp32Snapshot = CreateToolhelp32Snapshot(15u, 0);
if ( Toolhelp32Snapshot )
{
  v4 = Toolhelp32Snapshot;
  pe.dwSize = 556;
  memset(&pe.cntUsage, 0, 552);
  if ( Process32FirstW(Toolhelp32Snapshot, &pe) )
  {
    hSnapshot = v4;
    p_Src = &Src;
    v6 = 0;
    for ( i = 0; ; i = v68 )
    {
      memcpy(p_Src, &pe, 0x22Cu);
      if ( v6 == i )
      {
        sub_4A4170(&v67);
        v6 = v69;
        v75 = (int)v67;
      }
      v7 = p_Src;
      memcpy((void *)(v75 + 556 * v6++), p_Src, 0x22Cu);
      v69 = v6;
      v4 = hSnapshot;
      if ( !Process32NextW(hSnapshot, &pe) )
        break;
      p_Src = v7;
    }
```

Ilustración 18: Llamadas para recorrer la lista de procesos.

Como es ya frecuente en la mayoría de los *ransomware*, se utiliza la función "*CreateToolhelp32Snapshot*" para realizar una captura de todos los procesos que se encuentran actualmente en ejecución y poder obtener de esta forma el nombre del proceso y compararlo con el listado que tiene preconfigurado.

Finalmente, si la comparación del nombre del proceso analizado con alguno de la lista resulta exitosa, se hace una llamada a la función "*TerminateProcess*" para matar al proceso.

Otras funcionalidades se han podido identificar haciendo uso de la extensión de CAPA para IDA, que permite identificar posibles funcionalidades dentro del código fuente:

- Eliminación de las "*Shadow Copies*": se ha encontrado la siguiente cadena de texto, que hace referencia a un comando que permite la eliminación de las copias de seguridad:

```
.0C8 aLockerCoreOsWi_2 db 'locker::core::os::windows::recycle_binsrc/
.0C8                                          ; DATA XREF: mw_encrypt_
.0C8                                          ; mw_encrypt_files_funct
.0C8                   db '_bin.rs'
.110 aVssadminExeDel db 'vssadmin.exe delete shadows /all /quiet'
.137 aShadowCopyRemo db 'shadow_copy::remove_all=',0
.137                                          ; DATA XREF: .rdata:ptr_
.150 ptr_shadow_remove dd offset aShadowCopyRemo
```

Ilustración 19: Eliminación de las Shadow Copies.

También se ha verificado que, durante la ejecución de *BlackCat*, se arranca un nuevo proceso a través de CMD con el comando visto en la anterior imagen:

```
• cmd.exe (PID: 5992 cmdline: C:\Windows\system32\cmd.exe" /c "vssadmin.exe delete shadows /all /quiet MD5: 4E2ACF4F8A396486AB4268C94A6A245F)
  • conhost.exe (PID: 2836 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA777DEEA782E8B4D7C7C33BBF8A4496)
  • vssadmin.exe (PID: 6920 cmdline: vssadmin.exe delete shadows /all /quiet MD5: 47D51216EF45075B5F7EAA117CC70E40)
```

Ilustración 20: Ejecución del comando para borrar las Shadow Copies.

- Listar los dispositivos de almacenamiento en red: se ha identificado la función "*NetShareEnum*":

```
v5 = sub_4DD3B0(&v52);
v46 = HIDWORD(v5);
v40 = (LPVOID)v5;
v6 = NetShareEnum((LPWSTR)v5, 1u, &bufptr, 0xFFFFFFFF, &entriesread, &totalentries, &resume_handle);
v7 = v6;
v51 = v6;
```

Ilustración 21: Código que obtiene los recursos de red.

- Listado de dispositivos locales de almacenamiento: llama por cada una de las letras del abecedario a una función encargada de comprobar si esa letra corresponde a un disco duro haciendo una llamada a "*GetDriveTypeW*". Luego haciendo uso de la función "*GetVolumePathNamesForVolumeNameW*" obtiene la ruta a partir del volumen, para crear una lista de los dispositivos que se van a recorrer durante el cifrado.
- Eliminación de los eventos del sistema: a través de otro comando de CMD, se realiza la eliminación de los eventos del sistema operativo:

```
1  int mw_remove_win_evtx()
2  {
3    int result; // eax
4    void **v1; // esi
5    _DWORD *v2; // ecx
6    _DWORD *v3; // eax
7    int v4; // [esp+0h] [ebp-18h] BYREF
8    LPVOID lpMem; // [esp+4h] [ebp-14h]
9    void **v6; // [esp+8h] [ebp-10h]
10
11   result = sub_4ECFE0(
12             (int)"cmd.exe /c  for /F \"tokens=*\" %1 in ('wevtutil.exe el') DO wevtutil.exe cl \"%1\"",
13             (int)&v4,
14             79u);
15   if ( v4 )
16   {
17     if ( (_BYTE)lpMem == 3 )
18     {
19       v1 = v6;
20       (*(void (__cdecl **)(void *))v6[1])(*v6);
21       v2 = v1[1];
22       if ( !v2[1] )
```

Ilustración 22: Eliminación de eventos del sistema.

Esta función se ejecuta una vez ha terminado el cifrado para eliminar las posibles pruebas de la ejecución que puedan aparecer en el sistema de eventos de Windows.

- Cambio del fondo de pantalla: haciendo uso de la función "*SystemParametersInfoW*" con el parámetro 0x14 se realiza el cambio de fondo de pantalla.

```
    }
    while ( v528 );
  }
  *(_QWORD *)&v792[0] = phkResult[0];
  DWORD2(v792[0]) = phkResult[1];
  v530 = sub_4DD3B0(v792);
  if ( !SystemParametersInfoW(0x14u, 0, (PVOID)v530, 3u) && dword_6EE1C0 )
  {
    HIDWORD(phkResult[0]) = GetLastError();
    LODWORD(phkResult[0]) = 0;
    v571 = "/cargo/registry/src/github.com-1ecc6299db9ec823/utfx-0.1.0/src/ucstr.r
```

SPI_SETDESKWALLPAPER
0x0014

Note When the **SPI_SETDESKWALLPAPER** flag is used, **SystemParametersInfo** returns **TRUE** unless there is an error (like when the specified file doesn't exist).

Ilustración 23: Cambio del fondo de pantalla.

2.1.2. BlackCat – V2

En la nueva versión desarrollada de *BlackCat* se ha introducido una mejora bastante relevante. Debido a que, en su anterior versión el parámetro *"--access-token"* aunque era un requerimiento ejecución, no era necesario conocer un valor secreto para introducirlo, pues se podía ejecutar con cualquier valor aleatorio. En esta nueva versión ya no es posible y esto se debe a que la cadena introducida se utiliza durante el descifrado de la configuración. Por lo tanto, sin el valor correcto de *"--access-token"* es imposible descifrar la configuración y, por lo tanto, imposible de ejecutar la muestra.

```
PS C:\Users\[redacted]\Desktop\BlackCat> .\new.exe
Invalid access token.
```

Ilustración 24: Prueba de ejecución sin "--access-token".

```
PS C:\Users\[redacted] Desktop\BlackCat> .\new.exe --access-token 123
Invalid config.
```

Ilustración 25: Prueba de ejecución con un valor aleatorio de "--access-token".

Durante el análisis del código se ha podido identificar zonas del binario, donde se supone debería existir información, pero que no se encuentran valores legibles:

```
ıta:0060DA61 blob9          db 0FDh                 ; DATA XREF: sub_469430+321C↑o
ıta:0060DA62                db 0B4h
ıta:0060DA63                db  95h
ıta:0060DA64                db  18h
ıta:0060DA65                db 0C9h
ıta:0060DA66                db  59h ; Y
ıta:0060DA67                db  15h
ıta:0060DA68                db  65h ; e
ıta:0060DA69                db  3Fh ; ?
ıta:0060DA6A                db  8Fh
ıta:0060DA6B                db  27h ; '
ıta:0060DA6C                db 0D0h
ıta:0060DA6D                db 0D1h
ıta:0060DA6E                db  32h ; 2
ıta:0060DA6F                db  9Dh
ıta:0060DA70                db 0AEh
ıta:0060DA71                db  70h ; p
```

Ilustración 26: Configuración cifrada

Siguiendo las referencias donde se utiliza esta cadena, se llega a una función donde guarda cierta similitud con la versión previa de *BlackCat*:

```
blob9_cpy = HeapAlloc(v326, 0, 0x1FC0u);
if ( !blob9_cpy )
  goto LABEL_1108;
blob9_cpy_ptr = (size_t)blob9_cpy;
memcpy(blob9_cpy, &blob9, 0x1FC0u);
v839 = v227;
Size = blob9_cpy_ptr;
if ( flag != 1 )
{
  if ( !mw_maybe_decrypt(blob9_cpy_ptr, (int)var_aes_key_cpy, 0x1FC0u ) )
  {
    v331 = hHeap;
    if ( !hHeap )
      goto LABEL_676;
    goto LABEL_678;
  }
```

Ilustración 27: Código de la versión 2 de BlackCat relacionado con la configuración.

```
    }
    while ( v13 );
  }
  if ( *(_DWORD *)&v566[4] && 12 * *(_DWORD *)&v566[4] )
    HeapFree(hHeap, 0, *(LPVOID *)v566);
  mw_parse_config((int)aConfigIdPublic, v566, 0x1FC0);
  v14 = *(_DWORD *)v566;
  v517.m128i_i32[0] = *(_DWORD *)&v566[4 * (*(_DWORD *)v566 == 1) + 8];
  if ( v517.m128i_i32[0] < 0 )
    goto LABEL_1006;
  v15 = *(const void **)&v566[4];
  if ( v517.m128i_i32[0] )
```

Ilustración 28: Código de la versión 1 de BlackCat relacionado con la configuración.

Además, una vez finalizado el descifrado, se llama a una función con el resultado, donde se pueden apreciar las diferentes cadenas correspondientes a las distintas claves del JSON:

```
      *(_DWORD *)(a1 + 4) = sub_488EC0((int)"config_id", 9);
      v217 = 0;
      *(_DWORD *)a1 = 1;
BEL_552:
      v216 = v286;
      v316 = v217;
      v292 = v10 != 0;
      goto LABEL_446;
    }
    v192 = v338;
    v298 = v319 == 0;
    if ( !v319 )
    {
      v218 = sub_488EC0(
               (int)"extensionpublic_keynote_file_namenote_full_textnote_short_textcredentialsdefault_file_
                    "file_cipherkill_serviceskill_processesexclude_directory_namesexclude_file_namesexclude
                    "ionsexclude_file_path_wildcardenable_network_discoveryenable_self_propagationenable_se
                    "nable_esxi_vm_killenable_esxi_vm_snapshot_killstrict_include_pathsesxi_vm_kill_exclude
               9);
```

Ilustración 29: Interpretación del JSON.

Otra característica que se ha encontrado en esta versión es la aparición de un *PsExec* cifrado y un pequeño *script* desarrollado en *batch*.

```
  }
  else
  {
    mw_decrypt_aes(&PS_Exec_Binary, v250, 396314);
    if ( lpBuffer && *(&lpBuffer + 1) )
      HeapFree(hHeap, 0, (LPVOID)lpBuffer);
    nNumberOfBytesToWrite = Size;
    *(_QWORD *)&lpBuffer = *(_QWORD *)v250;
    _InterlockedExchange(&dword_6DE068, 2);
```

Ilustración 30: Descifrado del PsExec.

```
else
{
  mw_decrypt_aes(&Batch_File, Src, 123);
  if ( qword_6DE07C && *(&qword_6DE07C + 1) )
    HeapFree(hHeap, 0, qword_6DE07C);
  ::dwBytes = v162[0];
  *(_QWORD *)&qword_6DE07C = *(_QWORD *)Src;
  _InterlockedExchange(&dword_6DE078, 2);
}
```

Ilustración 31: Descifrado del script en batch.

A continuación, se puede ver el resultado tras el descifrado del *script*:

```
@ECHO OFF
SETLOCAL
SET allargs=%*
"${EXECUTABLE}"  --access-token  ${ACCESS_TOKEN}  --drag-and-drop  -p
%allargs%
```

Fuera de estas modificaciones, el código sigue con las mismas funcionalidades y características. Se han comparado diferentes secciones de código, de las distintas funcionalidades encontradas en la primera versión y son totalmente iguales.

```
  v45 = (void ** &ptr_Killing_Processes;
  v46 = 1;
  v47 = 0;
  v48 = (int *)"/root/.cargo/registry/src/github.com-1ecc6299db9ec823/utfx-0.1.0/src/ucstr.rs";
  v49 = 0;
  v50 = 0;
  v52 = 20;
  v51 = "encrypt_lib::windowslibrary/encrypt-lib/src/windows.rsTrying to self propagate to ";
  v53 = 0;
  v55 = 34;
  v54 = "library/encrypt-lib/src/windows.rsTrying to self propagate to ";
  v56 = 1;
  v57 = 204;
  v2 = &off_6722F4;
  if ( dword_6DE088 == 2 )
    v2 = off_5F1020;
  if ( dword_6DE088 == 2 )
    v1 = off_5F101C;
  ((void (__cdecl *)(const char *, __int64 *))v2[5])(v1, &Src);
}
v67 = (LPVOID)4;
v68 = 0;
v75 = 4;
v69 = 0;
Toolhelp32Snapshot = CreateToolhelp32Snapshot(15u, 0);
if ( Toolhelp32Snapshot )
{
  v4 = Toolhelp32Snapshot;
  pe.dwSize = 556;
  memset(&pe.cntUsage, 0, 552);
  if ( Process32FirstW(Toolhelp32Snapshot, &pe) )
  {
```

Ilustración 32: Función encargada de parar procesos de la versión 2 de BlackCat.

2.3. Técnicas MITRE ATT&CK

Initial Access	T1078	Valid Accounts
Execution	T1059	Command and Scripting Interpreter
	T1106	Execution through API
	T1204	User Execution
Defense Evasion	T1211	Exploitation for Defense Evasion
	T1070	Indicator Removal on Host
	T1222	File and Directory Permissions Modification
Privilege Escalation	T1134	Access Token Manipulation
	T1088	Bypass User Account Control
Discovery	T1135	Network Share Discovery
	T1083	File and Directory Discovery
	T1057	Process Discovery
	T1018	Remote System Discovery
	T1082	System Information Discovery
Impact	T1486	Data Encrypted for Impact
	T1489	Service Stop
	T1490	Inhibit System Recovery

En el Apéndice A se puede consultar el mapa de tácticas y técnicas utilizadas por *BlackCat*.

3. MITIGACIÓN

3.1. Medidas a nivel de endpoint

El código de *BlackCat* no está firmado, por lo que implementar una política que no permita la ejecución de binarios que no estén firmados podría prevenir la ejecución de este *ransomware* y de otro tipo de *malware*. No obstante, gran cantidad de desarrolladores y paquetes de software no distribuyen sus productos firmados, por lo que esta estrategia podría no resultar práctica en algunos casos.

En concordancia con lo anterior, pero empleando mecanismos más generales, se recomienda que las organizaciones prohíban o, al menos, monitoricen la ejecución de binarios no conocidos previamente dentro de ella o aquellos no provenientes de fuentes confiables. Aunque imperfecto, por la forma en la que se crea y distribuye el software legítimo, esta medida puede servir como una alarma inicial para impulsar una mayor investigación y, posiblemente, limitar su propagación.

Con el objetivo de disminuir el tiempo de reacción frente a este tipo de amenazas se recomienda mantener vigilado el *endpoint* con soluciones de monitorización y de antivirus/EDR así como disponer de una política de actualizaciones que mantenga el *endpoint* con las últimas vulnerabilidades.

3.2. Medidas a nivel de red

Si se dispone de los mecanismos para inspeccionar el tráfico que ocurre dentro de la red, se debería identificar la transferencia de binarios desconocidos dentro de ella.

Por otro lado, es altamente recomendable mantener una segmentación adecuada de la red para evitar desplazamientos laterales y que finalmente se alcancen los sistemas críticos de la organización.

3.3. Medidas y consideraciones adicionales

En caso de incidente con este *malware*, se debe de reportar a las autoridades pertinentes lo más rápido posible.

4. INDICADORES DE COMPROMISO

Los indicadores de compromiso y reglas de detección también están disponibles para su consulta y descarga en el repositorio público del Basque Cybersecurity Centre:

https://github.com/basquecscentre/technical-reports

4.1. Hashes

4.1.1. SHA256:

731adcf2d7fb61a8335e23dbee2436249e5d5753977ec465754c6b699e9bf161

bd4e603e953d8c7803f3c7d72cd7197d996ab80ce80b9da96a4df7d10969bb55

4.2. YARA rules

Esta regla sirve para identificar muestras de la familia *BlackCat*, tanto **V1** como **V2**:

```
/*
BlackCat / ALPHV ransomware
*/

rule BlackCat
{
    meta:
        author = "Andrey Zhdanov"
        family = "ransomware.blackcat"
        description = "BlackCat ransomware Windows/Linux payload"
        severity = 10
        score = 100

    strings:
        $h0 = { ( B8 01 00 00 00 31 C9 | 31 C9 B8 01 00 00 00 )
                89 DE 0F A2 87 F3 89 CE [0-8]
                ( B8 07 00 00 00 31 C9 | 31 C9 B8 07 00 00 00 )
                [0-2] 81 E6 00 00 00 02 [0-2] 0F A2 [0-14] C1 E8 19 85 F6 }
        $h1 = { ( B8 01 00 00 00 31 C9 | 31 C9 B8 01 00 00 00 )
                ( 89 | 48 89 ) DE 0F A2 ( 87 | 48 87 ) F3 89 C?
                ( B8 07 00 00 00 31 C9 | 31 C9 B8 07 00 00 00 )
                [0-4] 0F A2 [0-8] C1 E? 19 ( 24 01 | 40 80 E6 01 ) }

        $x0 = { 8D ( 4D | 4C 24 ) ?? BA [4] 68 1A 0C 06 00 E8 }
        $x1 = { 8D ( 4D | 4C 24 ) ?? BA [4] 6A 7B E8 }

        $a01 = "src/bin/encrypt_app/app.rs" ascii
        $a02 = "encrypt_app::windows" ascii
        $a03 = "src/bin/encrypt_app/windows.rs" ascii
```

```
        $a04 = "encrypt_app::linux" ascii
        $a05 = "src/bin/encrypt_app/linux.rs" ascii
        $a06 = "library/encrypt-lib/src/app.rs" ascii
        $a07 = "encrypt_lib::windows" ascii
        $a08 = "library/encrypt-lib/src/windows.rs" ascii
        $a09 = "library/encrypt-lib/src/linux.rs" ascii
        $a10 = "encrypt_lib::linux" ascii
        $a11 = "psexec_args=" ascii
        $a12 = "psexec_args::args=" ascii
        $a13 = "locker::core::" ascii
        $a14 = "set_desktop_image::" ascii
        $a15 = "::pipeline::file_worker_pool" ascii
        $a16 = "::pipeline::chunk_workers_supervisor" ascii
        $a17 = "::os::windows::privilege_escalation" ascii
        $a18 = "::os::windows::samba" ascii
        $a19 = "::os::windows::system_info" ascii
        $a20 = "::os::windows::netbios" ascii
        $a21 = "hidden_partitions::mount_all::mounting=" ascii
        $a22 = "uac_bypass::shell_exec=" ascii
        $a23 = "-u-p-s-d-f-cpropagate::attempt=" ascii
        $a24 = "enum_dependent_services" ascii
        $a25 = "masquerade_peb" ascii
        $a26 = "AdvancedSmartPattern" ascii

        $b01 = "note_file_name" ascii
        $b02 = "note_full_text" ascii
        $b03 = "note_short_text" ascii
        $b04 = "default_file_cipher" ascii
        $b05 = "default_file_mode" ascii
        $b06 = "note_full_text" ascii
        $b07 = "exclude_file_path_wildcard" ascii
        $b08 = "exclude_file_extensions" ascii
        $b09 = "enable_network_discovery" ascii
        $b10 = "enable_self_propagation" ascii
        $b11 = "enable_set_wallpaper" ascii
        $b12 = "enable_esxi_vm_kill" ascii
        $b13 = "enable_esxi_vm_snapshot_kill" ascii
        $b14 = "strict_include_paths" ascii
        $b15 = "esxi_vm_kill_exclude" ascii
        $b16 = "drop-drag-and-drop-target" ascii
        $b17 = "no-vm-kill" ascii
        $b18 = "no-vm-snapshot-kill" ascii
        $b19 = "no-prop-servers" ascii

    condition:
        (((uint16(0) == 0x5A4D) and (uint32(uint32(0x3C)) == 0x00004550)) or
         (uint32(0) == 0x464C457F)) and
        (
            (1 of ($h*)) or
            (all of ($x*)) or
```

```
        (7 of ($a*)) or
        (5 of ($b*))
    )
}
```

5. REFERENCIAS ADICIONALES

- https://malpedia.caad.fkie.fraunhofer.de/details/elf.blackcat
- https://blog.talosintelligence.com/2022/03/from-blackmatter-to-blackcat-analyzing.html
- https://blog.emsisoft.com/en/39181/on-the-matter-of-blackmatter/
- https://github.com/rivitna/Malware/blob/main/BlackCat/BlackCat.yar
- https://unit42.paloaltonetworks.com/blackcat-ransomware/
- https://research.openanalysis.net/blackcat/ransomware/malware/python/dumpulator/emulation/2022/03/16/blackcat_ransomware.html
- https://www.ic3.gov/Media/News/2022/220420.pdf
- https://www.avertium.com/resources/threat-reports/blackcat-ransomware-triple-extortion-analysis-tactics

APÉNDICE A: MAPA DE TÉCNICAS MITRE ATT&CK

Execution	Privilege Escalation	Defense Evasion	Discovery	Lateral Movement	Impact
Command and Scripting Interpreter	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	File and Directory Discovery	Lateral Tool Transfer	Data Encrypted for Impact
Native API	Bypass User Account Control	Bypass User Account Control	Network Share Discovery		Inhibit System Recovery
User Execution	Access Token Manipulation	Access Token Manipulation	Process Discovery		Service Stop
Malicious Link	Token Impersonation/Theft	Token Impersonation/Theft	Remote System Discovery		
Malicious Image		Exploitation for Defense Evasion	System Information Discovery		
Malicious File		File and Directory Permissions Modification	System Owner/User Discovery		
		Indicator Removal on Host	System Service Discovery		

Reportar incidente

Si has detectado algún incidente de ciberseguridad, avísanos para que tomemos las medidas oportunas para evitar su propagación.

900 104 891

incidencias@bcsc.eus

Catálogo de ciberseguridad

¿Necesitas ayuda con tu ciberseguridad o la de tu empresa?