

Informe de Malware

Petya

Begoña Rodríguez Arteaga

Begoña Rodríguez Arteaga

Bootcamp Ciberseguridad Full Time

Mayo 2025

ÍNDICE INFORME DE MALWARE

Introducción

Objetivos del informe

Contexto del análisis

Relevancia de Petya como amenaza

Historia del malware Petya

Aparición y evolución

Versiones y variantes (Petya, Mischa, GoldenEye, NotPetya)

Impacto global y casos reales

Informe general de la muestra utilizada

Metodología del análisis

Proceso general

Consideraciones éticas y de laboratorio seguro

Herramientas utilizadas

Herramientas de análisis estático

Herramientas de análisis dinámico

Análisis de Malware

Análisis Estático

Análisis de Código

Conclusiones

Bibliografía

Introducción

El panorama de la ciberseguridad actual está marcado por una creciente sofisticación en las amenazas informáticas, siendo el ransomware una de las más extendidas y perjudiciales. Dentro de esta categoría, el malware Petya y sus variantes han destacado por su capacidad de causar interrupciones masivas, inutilizar sistemas completos y generar pérdidas económicas de gran magnitud. A diferencia de otros ransomware tradicionales, Petya introduce un modelo de ataque innovador al cifrar el registro de arranque maestro (MBR) del disco duro, bloqueando el acceso total al sistema operativo y dejando al usuario sin posibilidad de recuperación directa de sus archivos ni de su entorno de trabajo.

Este informe tiene como objetivo realizar un análisis exhaustivo de Petya, abordando sus técnicas de infección, sus mecanismos de cifrado y evasión, así como sus efectos sobre los sistemas comprometidos. Para ello, se ha seguido una metodología estructurada que combina análisis estático y dinámico, aplicados dentro de un entorno virtualizado y seguro que simula un escenario de ataque real sin poner en riesgo infraestructuras externas. Este enfoque permite observar tanto la arquitectura interna del malware como su comportamiento en tiempo de ejecución, facilitando la extracción de indicadores de compromiso (IoCs) y la identificación de patrones comunes en su operativa.

Además de su valor técnico, este trabajo tiene una finalidad formativa, enmarcada en el desarrollo de competencias en análisis de malware y respuesta ante incidentes. El estudio de Petya y sus variantes, como Mischa, GoldenEye y NotPetya, proporciona una perspectiva evolutiva sobre el ransomware, mostrando cómo estas amenazas han pasado de modelos económicos de extorsión digital a estrategias complejas con posibles fines destructivos o geopolíticos.

Comprender el funcionamiento y la trayectoria de Petya nos permite evaluar su peligrosidad como amenaza aislada, además que refuerza la necesidad de contar con sistemas resilientes, políticas de seguridad proactivas y entornos de análisis éticos y controlados. Este informe pretende contribuir a esa misión, proporcionando un caso práctico de análisis de una de las familias de malware más representativas de la última década.

Objetivos del informe

El presente informe tiene como objetivo principal analizar en profundidad el funcionamiento del malware Petya, una de las variantes de ransomware más representativas y disruptivas de los últimos años. A través de un enfoque metodológico que combina análisis estático y dinámico, se pretende desentrañar su estructura interna, mecanismos de infección, técnicas de cifrado y comportamiento en ejecución, con el fin de comprender su impacto sobre los sistemas afectados y evaluar las posibles estrategias de mitigación.

Este análisis busca también identificar los vectores de ataque empleados por Petya, así como los métodos de persistencia y evasión que le permiten mantenerse activo y oculto dentro del sistema comprometido. Al observar su interacción con el entorno operativo, tanto a nivel de archivos, registro del sistema y red, se podrán extraer indicadores de compromiso (IoCs) útiles para la detección temprana de infecciones similares.

Además, perseguimos un segundo objetivo didáctico: aplicar de manera práctica los conocimientos adquiridos durante el curso de análisis de malware, utilizando herramientas especializadas y entornos controlados de laboratorio.

Contexto del análisis

En los últimos años, el ransomware se ha consolidado como una de las amenazas más relevantes y devastadoras en el panorama de la ciberseguridad. Entre las numerosas variantes que han surgido, Petya destaca por su enfoque particularmente agresivo: en lugar de cifrar archivos individuales, este malware cifra el registro de arranque maestro (MBR) del sistema, impidiendo el arranque normal del equipo y bloqueando el acceso a toda la información del disco.

El análisis de Petya se enmarca en un contexto académico y formativo, dentro de un entorno controlado diseñado para simular situaciones reales de infección. Esta aproximación no solo permite estudiar el comportamiento del malware sin poner en riesgo infraestructuras reales, sino que también ofrece la posibilidad de experimentar con distintas herramientas de análisis y técnicas forenses, replicando así las dinámicas que se aplicarían en un entorno profesional.

La muestra analizada pertenece a una de las versiones más representativas del malware, lo que permite observar características clave como la sobrescritura del MBR, la utilización de técnicas de evasión de análisis y la exigencia de un rescate en criptomonedas. Asimismo, se ha considerado el contexto histórico en el que Petya y sus variantes (NotPetya, Mischa, GoldenEye, etc.) se difundieron, aprovechando campañas de phishing, exploits como EternalBlue o vectores de infección en cadena.

Este análisis se ha realizado con el propósito de comprender de forma integral el ciclo de vida del malware, desde su llegada al sistema hasta sus consecuencias finales. También buscamos reflexionar sobre el impacto real de amenazas de este tipo en organizaciones públicas y privadas, así como en usuarios individuales, poniendo de relieve la necesidad de contar con medidas preventivas, protocolos de respuesta y estrategias de recuperación.

Relevancia de Petya como amenaza

El ransomware Petya marcó un hito en la evolución del malware moderno debido a su enfoque disruptivo y a la sofisticación de sus técnicas. A diferencia de otras amenazas contemporáneas que cifran archivos de usuario, Petya sobresale por atacar directamente el sector de arranque del disco duro, impidiendo que el sistema operativo pueda iniciar y dejando al equipo completamente inoperativo. Esta agresividad convierte a Petya no solo en un mecanismo de extorsión digital, sino también en una herramienta eficaz de sabotaje.

Desde su aparición inicial en 2016, especialmente con la propagación de su variante NotPetya en 2017, el impacto global de este malware fue considerable. Organizaciones públicas, infraestructuras críticas y grandes empresas internacionales se vieron afectadas, generando millones de euros en pérdidas económicas y paralizando operaciones durante días. Casos como los de Maersk, Rosneft o el sistema sanitario ucraniano son prueba de su capacidad destructiva.

Además de su potencia técnica, Petya y sus variantes destacan por el uso de tácticas avanzadas de propagación, como la explotación de vulnerabilidades conocidas como EternalBlue en Windows y el uso de credenciales robadas para moverse lateralmente en redes internas. Estas capacidades convirtieron a Petya en una amenaza altamente eficiente, capaz de extenderse con rapidez incluso en entornos corporativos que contaban con medidas de seguridad tradicionales.

Por otro lado, Petya puso de relieve la delgada línea entre el cibercrimen con fines económicos y el ciberataque con motivaciones políticas o destructivas.

Especialmente en el caso de NotPetya, numerosos analistas coinciden en que su verdadero objetivo no era el lucro, sino causar daños generalizados, utilizando el ransomware como una cortina de humo. Esta característica ha servido como precedente para la aparición de amenazas híbridas, que combinan tácticas de malware financiero con fines de guerra digital.

En suma, la relevancia de Petya no solo radica en sus capacidades técnicas, sino en el profundo impacto que tuvo sobre la percepción y respuesta global frente al ransomware. Su estudio sigue siendo fundamental para comprender cómo evolucionan las amenazas en el ámbito cibernético y por qué es imprescindible adoptar enfoques de seguridad basados en la anticipación, la resiliencia y la respuesta coordinada.

Historia del malware Petya

El ransomware Petya ocupa un lugar destacado en la evolución del malware moderno debido a su enfoque destructivo y su capacidad de propagación masiva. Surgido en 2016, se distinguió rápidamente de otras amenazas por su innovador mecanismo de ataque, que no se limitaba al cifrado de archivos, sino que inutilizaba por completo el sistema al sobrescribir el registro de arranque maestro (MBR). Desde su aparición, Petya ha dado lugar a varias versiones y variantes, como Mischa, GoldenEye y NotPetya, cada una más sofisticada que la anterior y con un impacto exponencialmente mayor a nivel global. El análisis de su historia permite entender tanto su evolución técnica del ransomware como la transformación de herramienta criminal a posible vector de ciberataques con motivaciones geopolíticas. Este recorrido histórico es esencial para contextualizar su peligrosidad y anticipar futuras amenazas de naturaleza similar.

Aparición y evolución

El ransomware Petya hizo su primera aparición documentada en marzo de 2016, y rápidamente llamó la atención de la comunidad de ciberseguridad por su novedoso enfoque: en lugar de cifrar archivos individuales como era común en otros ransomware de la época, Petya sobrescribía el registro de arranque maestro (MBR) del disco duro, bloqueando por completo el acceso al sistema operativo. Este método, más agresivo y destructivo, representó un cambio de paradigma en las técnicas empleadas por el malware extorsivo.

Petya se propagaba principalmente a través de correos electrónicos de phishing con archivos adjuntos maliciosos, normalmente documentos de Word o PDFs que, al ser abiertos, descargaban un archivo ejecutable. Una vez activado, el malware forzaba un reinicio del sistema y en ese momento reemplazaba el MBR con un cargador personalizado que mostraba una pantalla falsa de comprobación de disco, mientras cifraba la tabla de archivos. Finalizado el proceso, el sistema mostraba un mensaje en rojo exigiendo un rescate en bitcoins a cambio de la clave de descifrado.

Posteriormente, aparecieron variantes de Petya que aumentaron su complejidad. Algunas de estas combinaban características de troyano con ransomware, como Mischa, una versión que actuaba como plan B en caso de que Petya no pudiera acceder al MBR. También surgieron otras como GoldenEye, que añadían cifrado

tradicional de archivos además del MBR o utilizaban interfaces gráficas más elaboradas para presentar el mensaje de rescate.

El punto cumbre de esta evolución llegó en junio de 2017, con el brote global de una variante bautizada como NotPetya. Aunque inicialmente parecía otro ataque de ransomware basado en Petya, NotPetya resultó ser un wiper disfrazado: su mecanismo de cifrado era irreversible, lo que hacía imposible recuperar los datos incluso pagando el rescate. Esta variante utilizaba múltiples vectores de propagación, entre ellos el exploit EternalBlue (filtrado por el grupo Shadow Brokers y aprovechado también por WannaCry) y herramientas legítimas como PsExec para moverse lateralmente en redes corporativas.

NotPetya causó estragos en infraestructuras críticas, multinacionales, bancos y gobiernos, especialmente en Ucrania, país del que se sospecha era el objetivo principal. Sin embargo, su propagación no se limitó a una región y afectó a empresas de todo el mundo, evidenciando la capacidad destructiva de los ataques masivos con motivaciones más allá del lucro económico gracias a la globalización y a la interdependencia de las instituciones, tanto públicas como privadas.

La evolución de Petya ilustra cómo un malware puede transformarse desde una amenaza puramente criminal hasta una herramienta potencial de ciberconflicto geopolítico. Su estudio permite comprender mejor las técnicas de cifrado y propagación, además de subrayar la importancia de mantener sistemas actualizados, implementar controles de red robustos y fomentar la concienciación frente a amenazas de ingeniería social avanzada como esta.

Versiones y variantes (Petya, Mischa, GoldenEye, NotPetya)

Desde su aparición en 2016, Petya ha evolucionado en una familia de malware con múltiples variantes, cada una incorporando mejoras técnicas, nuevos vectores de infección y distintas finalidades. Estas versiones comparten un origen común, pero difieren en su comportamiento, alcance destructivo y complejidad, lo que las convierte en un objeto de estudio clave para comprender la evolución del ransomware moderno.

Petya (2016)

La versión original de Petya se distribuía a través de campañas de phishing que incluían enlaces maliciosos o archivos infectados. Una vez ejecutado, Petya sobrescribía el Master Boot Record (MBR) del disco duro, impidiendo que el sistema operativo se iniciara. En lugar de cifrar archivos individualmente, el malware cifraba la tabla del sistema de archivos (MFT) y mostraba una pantalla roja de rescate que imitaba la interfaz de CHKDSK, exigiendo un pago en bitcoins. Este hizo que se convirtiera en una amenaza especialmente destructiva, ya que inutilizaba por completo el sistema afectado.

Mischa (2016)

Mischa surgió como una variante complementaria de Petya. Se activaba en aquellos sistemas en los que el malware original no tenía permisos suficientes para modificar el MBR, por ejemplo, si el usuario no contaba con privilegios de administrador. En ese caso, Mischa cifraba directamente los archivos del sistema, utilizando un enfoque clásico de ransomware. Esta doble estrategia permitía asegurar la eficacia del ataque, independientemente de las restricciones del entorno. Además, Mischa introdujo mejoras en el cifrado y en la interfaz de rescate, haciendo más difícil la recuperación sin el pago del rescate.

GoldenEye (2016–2017)

GoldenEye fue una versión más refinada de Petya, que combinaba las capacidades destructivas del cifrado del MBR con la encriptación individual de archivos, lo que la hacía especialmente efectiva. Se difundió a través de correos electrónicos dirigidos al sector de recursos humanos, utilizando documentos adjuntos infectados con macros maliciosas. Esta versión incluía una interfaz de rescate más elaborada y un cifrado

reforzado, consolidando el modelo de doble extorsión y elevando el impacto potencial sobre organizaciones y empresas.

NotPetya (2017)

NotPetya representa una desviación crítica respecto a sus predecesores. Aunque conservaba la estética y parte del código base de Petya, en realidad se trataba de un wiper disfrazado de ransomware. Aunque mostraba un mensaje de rescate, el proceso de cifrado era irreversible: las claves necesarias para la recuperación no se generaban ni se almacenaban. NotPetya fue responsable de uno de los ataques más devastadores de la década, afectando a grandes corporaciones internacionales, infraestructuras críticas y gobiernos, principalmente en Ucrania. Su propagación fue masiva, gracias al uso de exploits como EternalBlue y herramientas legítimas como PsExec para moverse lateralmente en redes internas. La comunidad internacional de ciberseguridad lo interpretó como un ataque geopolítico encubierto, más que como un acto de cibercrimen tradicional.

Cada una de estas variantes refleja una evolución estratégica en el uso del ransomware: desde ataques con fines meramente económicos hasta operaciones destructivas con implicaciones políticas. El estudio de estas versiones proporciona una visión clara de cómo las amenazas pueden adaptarse y escalar en función de sus objetivos, y destaca la necesidad urgente de adoptar una seguridad proactiva y resiliente.

Impacto global y casos reales

El ransomware Petya y sus variantes posteriores, especialmente NotPetya, han tenido un impacto global sin precedentes, afectando tanto a usuarios individuales como a grandes corporaciones e infraestructuras críticas. La virulencia de su propagación, combinada con la sofisticación técnica y el enfoque destructivo del malware, convirtió a esta familia en uno de los mayores referentes en la historia reciente del cibercrimen.

Uno de los eventos más destacados se produjo en junio de 2017, cuando NotPetya se diseminó de forma masiva a través de una actualización comprometida del software de contabilidad ucraniano M.E.Doc. Aunque el ataque pareció estar dirigido inicialmente contra Ucrania —donde afectó a entidades gubernamentales, bancos y medios de comunicación—, la propagación del malware no tardó en cruzar fronteras, afectando a organizaciones de todo el mundo.

Entre las empresas más afectadas se encuentra Maersk, el gigante danés del transporte marítimo, que experimentó una interrupción completa de sus sistemas de TI, con pérdidas estimadas en 300 millones de dólares. Otro caso notable fue el de la farmacéutica estadounidense Merck, que también sufrió la paralización de sus operaciones globales. En ambos casos, las compañías se vieron obligadas a reconstruir infraestructuras completas desde copias de seguridad y sistemas alternativos, lo que evidenció la gravedad del ataque.

En el sector público, el gobierno ucraniano fue uno de los principales blancos del ataque, con múltiples ministerios, aeropuertos, redes eléctricas y medios de transporte colapsados. Esta situación afectó a miles de ciudadanos, generando un contexto de caos digital sin precedentes en el país.

NotPetya también afectó a la multinacional de logística FedEx, concretamente a su filial europea TNT Express, que reportó retrasos significativos y pérdidas económicas considerables. Otro caso relevante fue el de la empresa de materiales de construcción Saint-Gobain, con sedes en varios países, que tuvo que suspender temporalmente sus actividades debido al ataque.

Estos incidentes revelaron no solo la capacidad técnica del malware, sino también la falta de preparación y resiliencia de muchas organizaciones ante ciberataques a gran escala. Además, el hecho de que NotPetya no incluyera un mecanismo funcional de recuperación tras el pago del rescate alimentó la teoría de que su propósito no era

económico, sino destructivo, lo que refuerza la interpretación de este evento como un acto de ciberguerra encubierta.

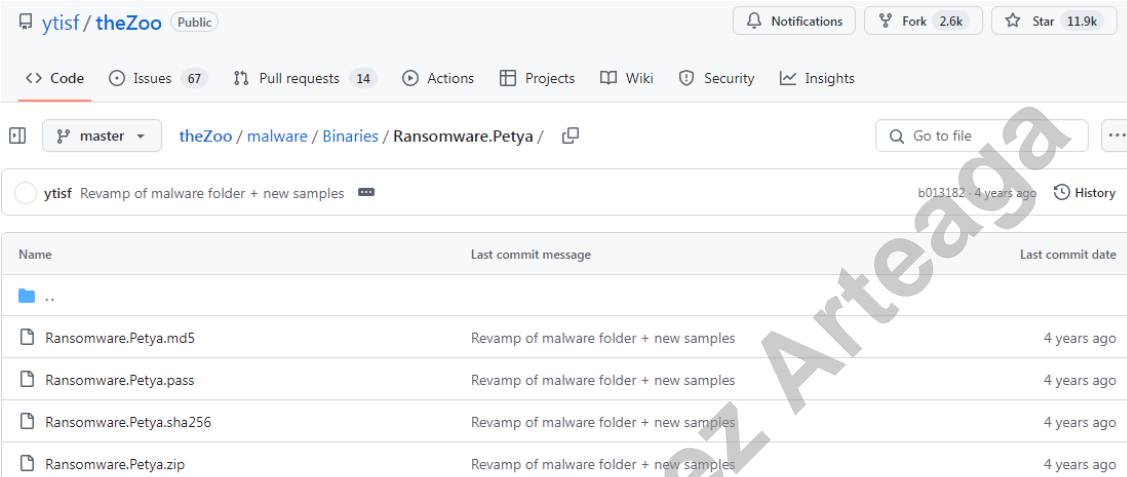
En conjunto, el impacto de Petya y sus variantes no solo se midió en términos económicos o técnicos, sino también en la transformación que provocaron en la conciencia global sobre ciberseguridad. A partir de estos ataques, muchas empresas y gobiernos comenzaron a reforzar sus sistemas de respaldo, implementar políticas de actualización más estrictas y adoptar soluciones de detección de amenazas más sofisticadas.

Nombre	Fecha	Cifrado de archivos	Cifrado MBR	Vector de entrada	Objetivo	Nota destacada
Petya	Mar 2016	No	Sí	Phishing + ejecutable	Económico	Sobrescribe MBR y bloquea acceso total al sistema
Mischa	May 2016	Sí	No	Activado si Petya falla	Económico	Funciona como respaldo cuando no hay privilegios de admin
GoldenEye	Dic 2016	Sí	Sí	Email con CV falso (RRHH)	Económico	Integra cifrado doble y mejor evasión
NotPetya	Jun 2017	No	Sí	M.E.Doc + EternalBlue + PsExec	Destructivo (wiper)	Simula ser ransomware, pero no permite recuperar datos

Informe general de la muestra utilizada

La muestra del ransomware Petya descargada para este análisis ha salido de dos lugares. En primer lugar, hemos descargado un archivo .zip con dos binarios de Github, <https://github.com/ytisf/theZoo>. Después, hemos descargado el archivo ejecutable para Windows .exe de Malware Bazaar, <https://bazaar.abuse.ch>

GitHub TheZoo



ytisf / theZoo Public

Notifications Fork 2.6k Star 11.9k

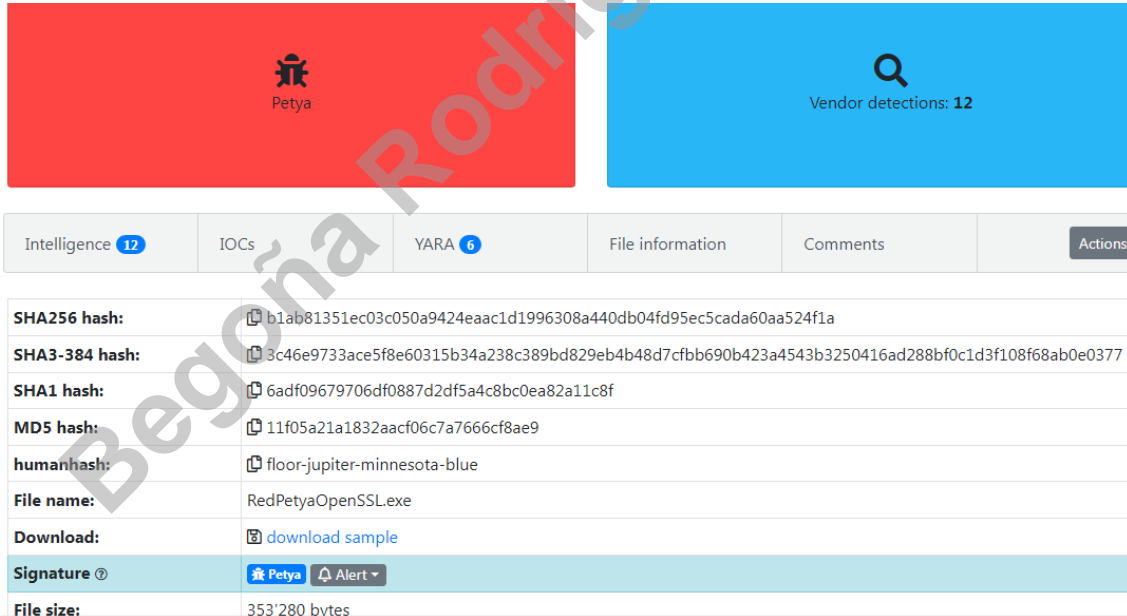
<> Code Issues 67 Pull requests 14 Actions Projects Wiki Security Insights

master theZoo / malware / Binaries / Ransomware.Petya /

ytisf Revamp of malware folder + new samples b013182 · 4 years ago History

Name	Last commit message	Last commit date
..		
Ransomware.Petya.md5	Revamp of malware folder + new samples	4 years ago
Ransomware.Petya.pass	Revamp of malware folder + new samples	4 years ago
Ransomware.Petya.sha256	Revamp of malware folder + new samples	4 years ago
Ransomware.Petya.zip	Revamp of malware folder + new samples	4 years ago

Malware Bazaar



Petya

Vendor detections: 12

Intelligence 12 IOCs YARA 6 File information Comments Actions

SHA256 hash:	b1ab81351ec03c050a9424eaac1d1996308a440db04fd95ec5cada60aa524f1a
SHA3-384 hash:	3c46e9733ace5f8e60315b34a238c389bd829eb4b48d7cfbb690b423a4543b3250416ad288bf0c1d3f108f68ab0e0377
SHA1 hash:	6adf09679706df0887d2df5a4c8bc0ea82a11c8f
MD5 hash:	11f05a21a1832aacf06c7a7666cf8ae9
human hash:	floor-jupiter-minnesota-blue
File name:	RedPetyaOpenSSL.exe
Download:	download sample
Signature ⓘ	Petya
File size:	353'280 bytes

Desde Any.Run, hemos buscado información sobre el malware más extendida, tanto datos generales como el hash del archivo ejecutable con el ramsonware incluido.

General Info

☒ Add for printing

File name:

Ransomware.Petya.zip

Full analysis:

<https://app.any.run/tasks/8a1ab091-0384-469c-93ee-6471680189e1>

Verdict:

Suspicious activity




Analysis date:

January 20, 2025 at 16:16:49

OS:

Windows 10 Professional (build: 19045, 64 bit)

Indicators:

MIME:

application/zip

File info:

Zip archive data, at least v2.0 to extract, compression method=deflate

MD5:

E8FB95EBB7E0DB4C68A32947A74B5FF9

SHA1:

6F93F85342AA3EA7DCBE69CFB55D48E5027B296C

SHA256:

33CA487A65D38BAD82DCCFA0D076BAD071466E4183562D0B1AD1A2E954667FE9

SSDEEP:

12288:h62An+IYWejKM9KlIoyoAWPPpxS8yrST5UvF50VHCJvD3DpNu7NwRUDxuJnU:hJA+BncEoyoJpxS8yrSV0nvHpNu7eQxH

 ANY.RUN is an interactive service which provides full access to the guest system. Information in this report could be distorted by user actions and is provided for user acknowledgement as it is. ANY.RUN does not guarantee maliciousness or safety of the content.

Metodología del análisis

El análisis del ransomware Petya ha sido realizado siguiendo una metodología rigurosa, estructurada en diversas fases que permiten comprender el funcionamiento interno del malware y su comportamiento en entornos reales. Este enfoque combina técnicas de análisis estático y dinámico, apoyadas en herramientas especializadas y desarrolladas dentro de un laboratorio seguro y completamente aislado. A lo largo del proceso, se han tenido en cuenta tanto las buenas prácticas técnicas como los principios éticos fundamentales, garantizando un entorno de investigación responsable, legal y sin riesgo para sistemas externos. La metodología aplicada busca desentrañar las capacidades técnicas de Petya y fortalecer las competencias prácticas en ciberseguridad, además de promover un análisis seguro, replicable y profesional.

Proceso general

El análisis del ransomware Petya se ha llevado a cabo siguiendo una metodología estructurada que combina técnicas de análisis estático y dinámico en un entorno controlado. Esta metodología permite comprender tanto la estructura interna del malware como su comportamiento en ejecución, garantizando al mismo tiempo la seguridad del entorno de análisis y la integridad de los sistemas utilizados.

El proceso comenzó con la preparación del laboratorio de análisis, configurando una máquina virtual con acceso restringido a la red y equipada con herramientas especializadas. Se utilizó una configuración de red en modo host-only para evitar cualquier propagación accidental del malware, y se tomaron instantáneas previas para poder restaurar el sistema a su estado original tras la ejecución de la muestra.

En primer lugar, se realizó un análisis estático, sin ejecutar el archivo malicioso, con el objetivo de obtener información preliminar sobre su naturaleza. Para ello, se examinó los metadatos del ejecutable, hemos introducido los hashes para verificar su presencia en bases de datos de malware conocidas como VirusTotal y hemos extraído cadenas de texto con herramientas como Strings y PE Studio. Este análisis ha permitido la identificación de llamadas sospechosas a funciones del sistema, referencias a direcciones de red y posibles técnicas de evasión.

Después, hemos realizado el análisis dinámico, ejecutando el malware dentro del entorno virtualizado mientras se monitoreaban sus acciones. Hemos observado cambios en el registro de Windows, la creación de nuevos procesos, conexiones a

direcciones IP externas y la manipulación de archivos del sistema. Esta fase permitió identificar los efectos concretos del malware sobre el entorno, así como las técnicas de persistencia y los mecanismos de cifrado que utiliza.

Begoña Rodríguez Arteaga

Consideraciones éticas y de laboratorio seguro

El análisis de malware, especialmente de amenazas avanzadas como Petya, conlleva una serie de responsabilidades éticas y técnicas que deben ser cuidadosamente consideradas para garantizar que la actividad se realice de forma segura, controlada y legal. En este sentido, la implementación de un laboratorio seguro y el cumplimiento de principios éticos fundamentales han sido pilares esenciales del proceso metodológico adoptado.

Desde el punto de vista técnico, se ha trabajado en un entorno completamente aislado, compuesto por una máquina virtual sin conexión directa a internet, configuradas en modo *host-only* para evitar la propagación del malware más allá del laboratorio. Esta segmentación impide que el código malicioso pueda afectar otros sistemas, dispositivos o redes externas, incluso en caso de un fallo de contención. Además, se han deshabilitado todas las funcionalidades de sincronización y acceso compartido entre el sistema anfitrión y la máquina virtual.

Hemos tomado medidas adicionales de seguridad, como el uso de instantáneas previas al análisis, que permiten revertir rápidamente cualquier cambio realizado por el malware. Esta práctica minimiza la posibilidad de contaminación cruzada o errores operativos durante el proceso.

En cuanto a las consideraciones éticas, hemos llevado a cabo un enfoque basado en el principio de responsabilidad. El análisis se ha creado únicamente con fines académicos y formativos, evitando en todo momento la distribución o reproducción del malware más allá del entorno autorizado. Se ha respetado la normativa vigente sobre el uso de software malicioso y no se ha compartido la muestra ni sus derivados en ningún canal público o no controlado.

Hemos promovido una actitud de conciencia profesional y ética frente a las amenazas cibernéticas, entendiendo que el conocimiento técnico adquirido no debe ser utilizado con fines ilícitos o destructivos. El objetivo último del análisis ha sido fortalecer las capacidades de defensa y respuesta ante incidentes, y no replicar las acciones maliciosas que se estudian.

Herramientas utilizadas

El análisis técnico de la muestra petya.exe ha sido llevado a cabo mediante una combinación de herramientas especializadas en análisis estático, dinámico e ingeniería inversa, dentro de un entorno controlado y seguro. Esta metodología responde a las buenas prácticas vistas en el bootcamp y garantiza un abordaje integral de la amenaza.

Herramientas de análisis estático

PEStudio

Herramienta esencial para el análisis sin ejecución del archivo PE. Permite identificar cabeceras, funciones importadas, indicadores sospechosos, rutas .onion, uso de privilegios elevados, y una puntuación de riesgo elevada basada en firmas y heurística. También facilitó el análisis de secciones, niveles de entropía y detección de APIs críticas.

PortEx Analyzer

Utilizada como refuerzo visual al análisis de estructura del binario. Esta herramienta ayuda a representar gráficamente el contenido y segmentación del ejecutable, evidenciando posibles técnicas de empaquetado u ofuscación.

Herramienta de análisis de cadenas (strings)

A través de PEStudio y herramientas nativas de extracción de cadenas, se identificaron más de 4.900 cadenas dentro del ejecutable. Muchas de ellas fueron clasificadas automáticamente como relevantes por su tipología (url-pattern, file, privilege, password, etc.).

Herramientas de análisis dinámico

Any.run

Sandbox interactiva utilizada para ejecutar petya.exe en un entorno Windows virtualizado y monitorizado. Con ella, hemos podido observar en tiempo real los el árbol de procesos generado, el acceso al registro de Windows, la ejecución encadenada de WinRAR, rundll32 y otras herramientas legítimas, así como la posible ejecución de payloads maliciosos. La correlación con técnicas de la matriz MITRE ATT&CK aporta un contexto adicional sobre el comportamiento del malware.

Herramientas de ingeniería inversa

IDA Pro

Software de desensamblado profesional empleado para examinar el código ensamblador de petya.exe.

Begoña Rodríguez Arteaga

Análisis del malware

El análisis de malware es un proceso técnico orientado a identificar, desensamblar y comprender el funcionamiento interno de un software malicioso, con el objetivo de obtener información útil para su detección, mitigación y respuesta. Esta práctica combina técnicas de ingeniería inversa, análisis de comportamiento en tiempo de ejecución y revisión de artefactos generados por el malware. En entornos profesionales, este tipo de análisis es fundamental para desarrollar firmas de detección, reglas YARA, indicadores de compromiso (IoCs) y estrategias de defensa frente a ataques complejos.

En el presente informe, se ha llevado a cabo el análisis técnico del ransomware Petya, utilizando un enfoque mixto que combina análisis estático, con la revisión del binario sin ejecución y el análisis dinámico, que observa el comportamiento del malware con una ejecución controlada.

Análisis estático

El análisis estático constituye una de las fases fundamentales en el estudio de software malicioso, ya que permite examinar las características internas del archivo ejecutable sin necesidad de ejecutarlo, y por ende, sin peligro de infección para el examinador de la muestra. Esta metodología es especialmente valiosa para identificar indicadores de compromiso, estructuras del archivo PE (Portable Executable), cadenas incrustadas, funciones importadas y secciones críticas del binario que pueden revelar su comportamiento potencial.

Aplicando esta técnica al archivo de este ransomware, Petya.exe, se ha llevado a cabo un estudio detallado mediante herramientas como PESTudio, que han permitido identificar múltiples elementos sospechosos desde el primer nivel de inspección: firmas antivirus, privilegios requeridos, uso intensivo de funciones de la API de Windows, referencias a redes anónimas como Tor y librerías relacionadas con operaciones criptográficas. Además, hemos detectado una estructura de secciones y cabeceras altamente consistente con muestras de ransomware sofisticado.

El análisis estático ha sido complementado con la revisión del encabezado DOS, el Rich Header, la cabecera PE y los directorios de importación, lo que ha permitido perfilar el origen y la compilación del binario, además de permitir la suposición de las estrategias de evasión y técnicas de empaquetado utilizadas para dificultar su detección y análisis.

Comenzamos el estudio estático de la muestra del malware Petya con el hash

SHA256:

b1ab81351ec03c050a9424eaac1d1996308a440db04fd95ec5cada60aa524f1a

usando la consola de comandos de nuestra máquina virtual preparada para el análisis de malware, que corre con el sistema operativo Windows 7.

Vamos a utilizar el comando strings para la extracción manual de las cadenas de datos del archivo, para poder obtener pistas sobre la funcionalidad del programa.

Procedemos a la búsqueda con el comando **strings.exe n -20**

C:\Users\master\Downloads\Petya.exe

```
C:\Users\master\Desktop\Análisis Estático>strings.exe n -20 C:\Users\master\Downloads\Petya.exe
```

Esta es la salida de las cadenas de datos del archivo:

```
Administrator: C:\Windows\System32\cmd.exe
.\crypto\engine\eng_init.c
.\crypto\ec\ecp_monf.c
.\crypto\ec\ecp_nist.c
.\crypto\ec\ec2_mult.c
(condition & (condition - 1)) == 0
.\crypto\bn\bn_gf2m.c
.\crypto\bn\bn_gf2m.c
.\crypto\asn1\asn1_utl.c
Stack part of OpenSSL 0.9.8zb 6 Aug 2014
.\crypto\stack\stack.c
.\crypto\buffer\buffer.c
lhash part of OpenSSL 0.9.8zb 6 Aug 2014
.\crypto\lhash\lhash.c
RAND part of OpenSSL 0.9.8zb 6 Aug 2014
@@.\crypto\rand\md_rand.c
You need to read the OpenSSL FAQ, http://www.openssl.org/support/faq.html
.\crypto\engine\eng_lib.c
.\crypto\engine\eng_table.c
.\crypto\ec\ecp_smpl.c
.\crypto\bn\bn_mod.c
.\crypto\bn\bn_gcd.c
.\crypto\bn\bn_sqr.c
.\crypto\bn\bn_div.c
.\crypto\engine\vb_digest.c
.\crypto\rsa\rsa_sign.c
signature has problems, re-make with post SSLey045
SHA1 part of OpenSSL 0.9.8zb 6 Aug 2014
.\crypto\bn\bn_exp.c
.\crypto\bn\bn_rec.c
Microsoft Visual C++ Runtime Library
(Press Retry to debug the application - JIT must be enabled)
For information on how your program can cause an assertion
failure, see the Visual C++ documentation on asserts
<program name unknown>
- Attempt to use MSIL code from this assembly during native code initialization
This indicates a bug in your application. It is most likely the result of calling
g an MSIL-compiled (/clr) function from a native constructor or from DllMain.
- not enough space for locale information
- Attempt to initialize the CRT more than once.
This indicates a bug in your application.
- CRT not initialized
```

```
Administrator: C:\Windows\System32\cmd.exe
CreateToolhelp32Snapshot
Intel Hardware Cryptographic Service Provider
CryptAcquireContextW
.\crypto\bn\bn_mod.c
.\crypto\bn\bn_gcd.c
.\crypto\bn\bn_sqr.c
.\crypto\bn\bn_div.c
.\crypto\engine\vb_digest.c
.\crypto\rsa\rsa_sign.c
signature has problems, re-make with post SSLey045
SHA1 part of OpenSSL 0.9.8zb 6 Aug 2014
.\crypto\bn\bn_exp.c
.\crypto\bn\bn_rec.c
Microsoft Visual C++ Runtime Library
(Press Retry to debug the application - JIT must be enabled)
For information on how your program can cause an assertion
failure, see the Visual C++ documentation on asserts
<program name unknown>
- Attempt to use MSIL code from this assembly during native code initialization
This indicates a bug in your application. It is most likely the result of calling
g an MSIL-compiled (/clr) function from a native constructor or from DllMain.
- not enough space for locale information
- Attempt to initialize the CRT more than once.
This indicates a bug in your application.
- CRT not initialized
```

```
Administrator: C:\Windows\System32\cmd.exe
- not enough space for stdio initialization
- pure virtual function call
- not enough space for _onexit/_atexit table
- unable to open console device
- unexpected heap error
- unexpected multithread lock error
- not enough space for thread data
- abort() has been called
- not enough space for environment
- not enough space for arguments
- floating point support not loaded
!""$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNopqrstuvwxyz[\]^_`abcdefghijklmnopqrstuvwxyz
pqrstuvwxyz[~
GetProcessWindowStation
GetUserObjectInformationW
((((((H
h((((H
H
!""$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNopqrstuvwxyz[\]^_`abcdefghijklmnopqrstuvwxyz
pqrstuvwxyz[~
!""$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNopqrstuvwxyz[\]^_`ABCDEFGHIJKLMN
PQRSTUVWXYZ[~
http://petya3h5tbyuki.onion/
http://petya3h5tbyuki.onion/
lookupPrivilegeValue
```

```
Administrator: C:\Windows\System32\cmd.exe
AdjustTokenPrivileges
QueryPerformanceCounter
EnterCriticalSection
LeaveCriticalSection
GetSystemTimeAsFileTime
UnhandledExceptionFilter
SetUnhandledExceptionFilter
FreeEnvironmentStringsW
GetEnvironmentStringsW
InitializeCriticalSectionAndSpinCount
DeleteCriticalSection
InterlockedIncrement
InterlockedDecrement
IsProcessorFeaturePresent
GetUserObjectInformationW
GetProcessWindowStation
DeregisterEventSource
RegisterEventSourceA
Assertion failed: %s, file %s, line %d
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

[illegible]

```

C:\AdminTools\CI\Windows\Winrm32cmd.exe
U$----- $$$$- $$$$$$U
-$$$$- u$U $$$$-
$$$U u$U u$$$
$$$U u$$$U u$$$
$$$$$U$ $$$$U$$$-
-$$$$$-----
UUU $$$$ $ $ $ $ $ $ $ $UUU
U$----- $$$$$$u$u$$$$ u$$$
$$$UUU -$$$$$u$u$UUU u$u$u$u$
U$-----UUU ----- UUu$U$u$u$u$u$u$
$$$$$-----UUUU UU$u$u$u$u$u$u$u$-----
----- $$$$$$UUU -----
UUUU -----UUUU
U$UUUU$-----UUUU -----$$$$$$$$$$$$UUUU$$$
$$$$$$$$$----- $$$$$$-----
$$$- PRESS ANY KEY! $$$-
123456789AB0CDEFghjKlMNPORSTUVWxyz2abdefghijklmnopqrstuvwxyz
123456789AB0CDEFghijklmnopqrstuvwxyzAB0CDEFghijklMNPORSTUVWxyz
assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
<requestedPrivileges>
<requestedExecutionLevel level="requireAdministrator" uiAccess="false">
/requestedExecutionLevel>
</trustInfo>
</assembly>

```

[illegible][illegible]

```
Administrator: C:\Windows\System32\cmd.exe
3 7 1 0 58505850P3K3X3K3p319
4 4 0444484H4 4K4 4d4p4x4 14
5 5850585<5H5P5T5' 5h515x5
6 6 6 6 6860666P6X6 6h6p6x6
7 7 07074707H7 7K7 7d7p7x7
8 8 480808<8H8P8T8' 8h818x8
9 9 9 9 989098P9K9 9h9p9x9
: ( : 0 : 4 : 0 : H : L : K : ' : d : p : x : | :
: $ : 0 : 8 : < : H : P : T : ' : h : l : x :
: < : ( : < 0 < 0 < P < K < ' < h < p < x < | <
: ( : 0 : 4 : 0 : H : L : K : ' : d : p : x : | :
: $ : 0 : 8 : < : H : P : T : ' : h : l : x :
7 7 7 7 787078P7K7 7h7p7x7
0 0 0004000H0 0K0 0d0p0x0
1 1 110181<1H1P1T1' 1h11x1
2 2 2 2 282020P2K2 2h2p2x2
3 3 0304300H3 3K3 3d3p3x3
4 4 4404484H4 4K4 4d4p4x4
5 5 5 5 585050P5K5 5h5p5t5
6 6 6064606H6 6K6 6d6p6x6
8 8 6 6 688080P8K8 8h8p8x8
9 9 9094909H9 9K9 9d9p9x9
: $ : 0 : 8 : < : H : P : T : ' : h : l : x :
: ( : : 8 : 0 : d : P : K : ' : h : p : t :
```

```
Administrator: C:\Windows\System32\cmd.exe
-=$-0-8<-H-P-T'-h-l-x=
>(>,>8>@D>P>K>\h>p>t>
7(7074707H7L7K77d7k717
0 0004080<000D0H0L0P0T0X0\0'0d0h010p0t0x010
1 141(1.1014181<101D1H11P1T1X1\1'd1h11p1t1x11
2(20242024202422d2p2x212
3 3$30383<3H3P3T3'3h313x3
5(5054505H5L5X5'5d5h515p5t5x515
6 6$60686<6H6P6T6'6h616x6
7 717.7074707H7L7K77d7k717
8(8084808H8L8X8'8d8h818p8t8x818
9 9$90989<9H9P9T9'9h919x9
: (:,:8:@:D:P:X:\h:p:t:
(.0.4.@:H:L:X':d:p:x:l:
< <@<8<<<@<H<P<T<'<h<l<x<
= (==-8=@-D-P-X=-h-p-t=
>(>0>4>@>H>L>X>'>d>p>x>1>
? 7$7(7.7074707?707H7L7K77d7k717?
0 010.0004080<000D0H0L0P0T0X0\0'0d0h010p0t0x010
1(10141014181<101D1H11P1T1X1\1'd1h11p1t1x11
2 2$2(2.20242024202422d2p2x212'2d2h212p2t2x212
3 3(3.3034303<303D3H313P3T3X3\3'd3h313p3t3x313
4 404440444044404D4H4L4P4T4X4\4'd4h414p4t4x414
5 5$50585<5H5P5T5'5h515x5
6 616.6060606P6X6\6h6p6t6x616
```

```
Administrator: C:\Windows\System32\cmd.exe
7(7074707H7L7K77d7k717
8 8$80888<8H8P8T8'8h818x8
9 919.9090909P9X9\9h9p9t9x919
: (:,:0.4:@:D:H:L:P:T:X:\:d:h:l:p:t:x:l:
< <@<(<,<0<4<8<<<@<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
=< (==-0<4<8<<<@<D<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
> >@>(>,>0>4>8><<<@>D>H>L>P>T>X>\>'>d>h>l>p>t>x>1>
? 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
0 000(0.0004080<000D0H0L0P0T0X0\0'0d0h010p0t0x010
1 141(1.1014181<101D1H11P1T1X1\1'd1h11p1t1x11
2 2$2(2.20242024202422d2p2x212'2d2h212p2t2x212
3 3$3(3.3034303<303D3H313P3T3X3\3'd3h313p3t3x313
4 4$4(4.4044484<404D4H4L4P4T4X4\4'd4h414p4t4x414
5 5$5(5.5054585<505D5H515P5T5X5\5'5d5h515p5t5x515
6 6$6(6.6064686<606D6H616P6T6X6\6'6d6h616p6t6x616
7 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
8 8$8(8.8084888<808D8H818P8T8X8\8'8d8h818p8t8x818
9 9$9(9.9094989<909D9H919P9T9X9\9'9d9h919p9t9x919
: (:,:0.4:@:D:H:L:P:T:X:\:d:h:l:p:t:x:l:
< <@<(<,<0<4<8<<<@<D<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
=< (==-0<4<8<<<@<D<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
> >@>(>,>0>4>8><<<@>D>H>L>P>T>X>\>'>d>h>l>p>t>x>1>
? 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
```

```
Administrator: C:\Windows\System32\cmd.exe
? 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
0 000(0.0004080<000D0H0L0P0T0X0\0'0d0h010p0t0x010
1 141(1.1014181<101D1H11P1T1X1\1'd1h11p1t1x11
2 2$2(2.20242024202422d2p2x212'2d2h212p2t2x212
3 3$3(3.3034303<303D3H313P3T3X3\3'd3h313p3t3x313
4 4$4(4.4044484<404D4H4L4P4T4X4\4'd4h414p4t4x414
5 5$5(5.5054585<505D5H515P5T5X5\5'5d5h515p5t5x515
6 6$6(6.6064686<606D6H616P6T6X6\6'6d6h616p6t6x616
7 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
8 8$8(8.8084888<808D8H818P8T8X8\8'8d8h818p8t8x818
9 9$9(9.9094989<909D9H919P9T9X9\9'9d9h919p9t9x919
: (:,:0.4:@:D:H:L:P:T:X:\:d:h:l:p:t:x:l:
< <@<(<,<0<4<8<<<@<D<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
=< (==-0<4<8<<<@<D<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
> >@>(>,>0>4>8><<<@>D>H>L>P>T>X>\>'>d>h>l>p>t>x>1>
? 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
0 000(0.0004080<000D0H0L0P0T0X0\0'0d0h010p0t0x010
1 141(1.1014181<101D1H11P1T1X1\1'd1h11p1t1x11
2 2$2(2.20242024202422d2p2x212'2d2h212p2t2x212
3 3$3(3.3034303<303D3H313P3T3X3\3'd3h313p3t3x313
4 4$4(4.4044484<404D4H4L4P4T4X4\4'd4h414p4t4x414
5 5$5(5.5054585<505D5H515P5T5X5\5'5d5h515p5t5x515
6 6$6(6.6064686<606D6H616P6T6X6\6'6d6h616p6t6x616
7 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
8 8$8(8.8084888<808D8H818P8T8X8\8'8d8h818p8t8x818
9 9$9(9.9094989<909D9H919P9T9X9\9'9d9h919p9t9x919
: (:,:0.4:@:D:H:L:P:T:X:\:d:h:l:p:t:x:l:
< <@<(<,<0<4<8<<<@<D<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
=< (==-0<4<8<<<@<D<H<L<P<T<X<\<'<d<h<l<p<t<x<1<
> >@>(>,>0>4>8><<<@>D>H>L>P>T>X>\>'>d>h>l>p>t>x>1>
? 7$7(7.7074707?707H7L7K77d7k717?7d7h717p7t7x717?
0 0004080<000D0H0L0P0T0X0\0'0d0h010p0t0x010
```

Entre los resultados obtenidos, hemos podido indetificar múltiples rutas internas y referencias a ficheros fuente pertenecientes al paquete criptográfico OpenSSL 0.9.8zb, incluyendo archivos como stack.c, buffer.c, bn_exp.c, rsa_sign.c o md_rand.c. Estas rutas confirman que el malware incorpora funciones criptográficas robustas y probablemente implementa su propia lógica de cifrado a través de bibliotecas bien conocidas. La aparición de estructuras internas de OpenSSL sugiere que Petya cifra partes críticas del sistema, como la Master File Table (MFT), utilizando algoritmos personalizados o adaptados como Salsa20.

Existen varias cadenas asociadas al uso de funciones de la API de Windows, lo que refuerza la hipótesis de que el malware interactúa directamente con el sistema operativo para elevar privilegios, manipular procesos y establecer persistencia. Entre

las funciones detectadas se encuentran `CreateToolhelp32Snapshot`, `CryptAcquireContextW`, `GetProcessWindowStation`, `GetUserObjectInformationW`, `AdjustTokenPrivileges`, `LookupPrivilegeValueA`, y `RegisterEventSourceA`. Estas llamadas permiten al malware acceder al contexto de seguridad del sistema, enumerar procesos activos, y modificar configuraciones sensibles, características comunes en malware diseñado para operar con privilegios de administrador y con un alto grado de evasión.

Además de las llamadas a funciones, se pueden observar mensajes de error y cadenas que apuntan a problemas de inicialización del entorno de ejecución, como "CRT not initialized", "unexpected heap error" o "not enough space for environment". Estas cadenas suelen asociarse a errores deliberadamente provocados por el malware para dificultar su ejecución en entornos de análisis, tales como sandboxes o máquinas virtuales, lo que indica la implementación de mecanismos antianálisis.

Un hallazgo especialmente relevante fue la detección de mensajes de rescate incrustados en el binario, en los que se advierte al usuario que sus discos duros han sido cifrados con un algoritmo de grado militar. El mensaje indica que no es posible restaurar el acceso sin una clave privada y ofrece instrucciones para descargar el navegador Tor y visitar dos sitios .onion: <http://petya3ht5bhuyki.onion/> y <http://petya5koah5t7sv.onion/>. Estas direcciones son típicas de la red Tor y sugieren la existencia de un panel de control oculto donde el atacante solicita el pago del rescate, lo que confirma la funcionalidad ransomware activa de la muestra.

El análisis también reveló cadenas asociadas a la rutina visual que despliega Petya en pantalla tras la infección, incluyendo gráficos en arte ASCII y mensajes como "PRESS ANY KEY". Esta estructura visual, unida a la falsa simulación de una comprobación de disco (CHKDSK is repairing sector), constituye una táctica de ingeniería social cuyo objetivo es engañar al usuario y retrasar su respuesta, mientras el sistema ya ha sido cifrado y bloqueado.

Por último, se identificaron secuencias largas de caracteres no imprimibles, símbolos especiales y cadenas de texto ofuscadas o repetitivas. Estos patrones podrían estar relacionados con el contenido cifrado, bloques padding o estructuras de sobreescritura diseñadas para inutilizar el sector de arranque (MBR) o para dificultar la recuperación forense del sistema. Estas observaciones refuerzan la hipótesis de que la muestra analizada contiene rutinas de bajo nivel capaces de manipular

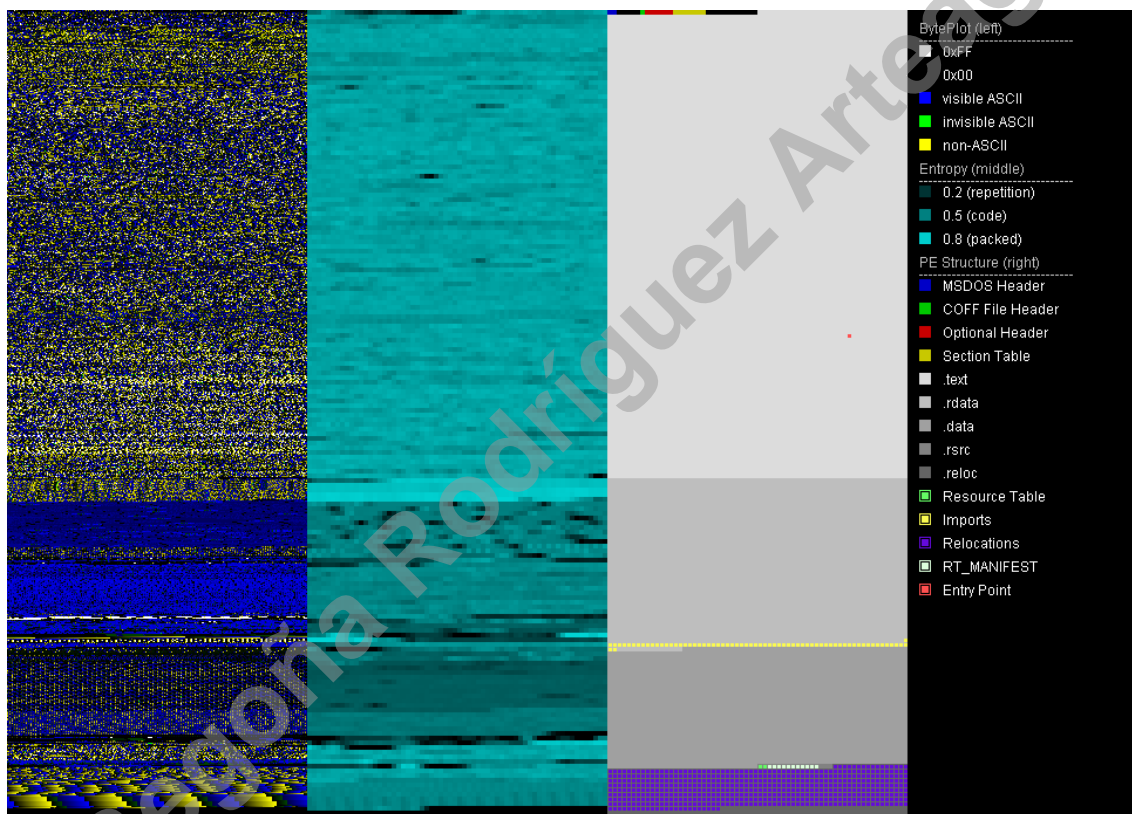
sectores físicos del disco, técnica característica de Petya y sus variantes más destructivas.

Vamos a utilizar PortexAnalyzer para poder crear una imagen fija de la muestra del malware. Utilizamos PortexAnalyzer a través de la consola de comandos de Windows 7 tecleando lo siguiente, **PortexAnalyzer.jar -p radioPetya.png**

C:\Users\master\Downloads\Petya.exe

```
C:\>cd Users\master\Desktop\Análisis Estático\PortexAnalyzer.jar -p radioPetya.png
C:\Users\master\Downloads\Petya.exe_
```

Imagen del malware



Esta herramienta permite representar visualmente el contenido de un archivo PE (Portable Executable) a través de tres perspectivas fundamentales: el BytePlot, el mapa de entropía y la estructura interna del ejecutable (PE layout). Esta aproximación facilita la detección de técnicas de ocultación, compresión, cifrado y manipulación del flujo de ejecución, aspectos clave trabajados durante las prácticas del bootcamp.

En la parte izquierda de la imagen (BytePlot), se observa una combinación predominante de colores azul oscuro, amarillo y azul brillante. Según la leyenda de PortEx Analyzer, estos colores corresponden a caracteres ASCII visibles, no visibles y no ASCII, indicando que el archivo contiene una alta densidad de contenido mixto: instrucciones legibles y bloques codificados. Esta estructura sugiere que el binario ha sido modificado o empaquetado, incorporando técnicas de ofuscación para dificultar su análisis directo, algo que encaja con los patrones estudiados en los ejercicios de cadenas (strings.exe) y detección de empaquetadores.

En el centro, encontramos el mapa de entropía, donde se visualiza una predominancia del color azul claro, equivalente a valores de entropía cercanos a 0.8. Este nivel elevado indica la existencia de bloques comprimidos o cifrados dentro del archivo, lo cual es común en malware que busca ocultar su payload real hasta el momento de la ejecución.

En la parte derecha de la imagen, se detalla la estructura interna del ejecutable PE, donde pueden observarse los encabezados estándar (MS-DOS Header, COFF File Header), así como secciones clásicas del binario como .text, .rdata, .data, .rsrc y .reloc. Cabe destacar que el Entry Point aparece marcado en rojo dentro de una región no habitual, alejada del comienzo de la sección .text. Este detalle es relevante porque podría tratarse de una redirección intencionada al payload cifrado, una técnica habitual en malware empaquetado para evadir la detección estática basada en firmas.

Otro elemento destacable en la estructura PE es la presencia de un bloque RT_MANIFEST y áreas ocupadas por Relocations e Imports, lo que indica que el malware declara metadatos para ejecución privilegiada (ej. manifestos UAC) y realiza cargas dinámicas de librerías o funciones. Este patrón ya fue anticipado en el análisis de strings donde se observaron funciones como AdjustTokenPrivileges y LookupPrivilegeValueA, necesarias para adquirir privilegios elevados y ejecutar código crítico con permisos de sistema.

identificar de forma rápida qué secciones merecen mayor atención en fases posteriores, como el análisis dinámico o de código ensamblador.

Nos vamos a la herramienta PE Studio para poder analizar más de fondo el archivo. Vamos a dar pinceladas de cada apartado que nos aparece en el árbol.

Indicadores

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

file settings about

c:\users\master\downloads\petya.exe

- indicators (44)
- virustotal (58/72)
- dos-header (64 bytes)
- dos-stub (160 bytes)
- rich-header (7)
- file-header (time-stamp)
- optional-header (GUI)
- directories (5)
- sections (99.71%)
- libraries (3)
- imports (89)
- exports (n/a)
- exceptions (n/a)
- ts-callbacks (n/a)
- relocations (8568)
- resources (manifest)
- strings (4926)
- debug (n/a)
- manifest (administrator)
- version (n/a)
- certificate (n/a)
- overlay (n/a)

xml-id	indicator (44)	detail	level
1430	The file references string(s) tagged as blacklist	count: 36	1
1400	The file execution privilege has been found	level: administrator	1
1266	The file imports symbol(s) tagged as blacklist	count: 21	1
1434	The file references a URL pattern	url: http://petya37h5bhyvki.onion/	1
1434	The file references a URL pattern	url: http://petya3koahstf7sv.onion/	1
1321	The time-stamp of the compiler is suspicious	year: 2022	2
1120	The file is scored by virustotal	score: 58/72	3
1019	The file contains a rich-header	status: yes	3
1261	The file imports deprecated function(s)	count: 6	3
1634	The file references a group of API	api: cryptography, count: 8	3
1634	The file references a group of API	api: execution, count: 21	3
1634	The file references a group of API	api: memory, count: 13	3
1634	The file references a group of API	api: windowing, count: 3	3
1634	The file references a group of API	api: network, count: 3	3
1634	The file references a group of API	api: desktop, count: 4	3
1634	The file references a group of API	api: keyboard-and-mouse, count: 1	3
1634	The file references a group of API	api: file, count: 11	3
1634	The file references a group of API	api: diagnostic, count: 6	3
1634	The file references a group of API	api: system-information, count: 6	3
1634	The file references a group of API	api: dynamic-library, count: 8	3
1634	The file references a group of API	api: security, count: 3	3
1634	The file references a group of API	api: console, count: 5	3
1634	The file references a group of API	api: synchronization, count: 6	3
1634	The file references a group of API	api: exception-handling, count: 2	3
1633	The file references a group of hint	hint: dos-message, count: 1	3
1633	The file references a group of hint	hint: file, count: 64	3
1633	The file references a group of hint	hint: size, count: 10	3
1633	The file references a group of hint	hint: password, count: 1	3
1633	The file references a group of hint	hint: utility, count: 2	3
1633	The file references a group of hint	hint: privilege, count: 1	3
1633	The file references a group of hint	hint: url-pattern, count: 2	3
1633	The file references a group of hint	hint: rtti, count: 6	3
1634	The file references a group of API	api: cryptography, count: 1	3
1634	The file references a group of API	api: network, count: 2	3
1269	The file references whitelisted string(s)	count: 34	4
1050	The file uses Control Flow Guard (CFG) as software security defense	status: no	4
1100	The file opts for Data Execution Prevention (DEP) as software security defense	status: yes	4
1102	The file opts for Address Space Layout Randomization (ASLR) as software security defense	status: yes	4
1043	The file contains a Manifest	status: yes	4
1106	The file opts for Stack Buffer Overrun Detection (GS) as software security defense	status: yes	4
1040	The file contains a digital Certificate	status: no	4
1109	The file opts for Code Integrity (CI) as software security defense	status: no	4
1287	The file subsystem has been found	type: GUI	4
1215	The file-ratio of the section(s) has been determined	ratio: 99.71%	4

Los indicadores proporcionados por PESTudio permiten identificar comportamientos sospechosos en archivos ejecutables. En el caso de petya.exe, se identificaron 44 indicadores, varios de nivel crítico. Entre ellos destacan cadenas y símbolos importados asociados a listas negras, el uso de privilegios elevados (nivel administrator), y la aparición de dos direcciones URL con dominio .onion, lo cual indica que el malware podría establecer comunicaciones a través de la red Tor para evadir detección y establecer canales de comando y control. La agrupación de APIs empleadas incluye funciones relacionadas con criptografía, ejecución de procesos, manipulación de memoria y red, todas ellas técnicas típicas utilizadas por malware avanzado.

Detección por Virus Total

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]				
file settings about				
c:\users\master\downloads\petya.exe		engine (72/72)	score (58/72)	date (dd.mm.yyyy)
indicators (44)				age (days)
virustotal (58/72)		ALYac	clean	28.03.2024
dos-header (64 bytes)		AhnLab-V3	clean	18.05.2025
dos-stub (160 bytes)		Baidu	clean	24.04.2025
rich-header (7)		Cynet	clean	16.05.2025
file-header (time-stamp)		DrWeb	clean	18.05.2025
optional-header (GUI)		F-Secure	clean	18.05.2025
directories (5)		Kaspersky	clean	17.05.2025
sections (99.71%)				
libraries (3)				
imports (89)				
exports (n/a)				
exceptions (n/a)				
tls-callbacks (n/a)				
relocations (8568)				
resources (manifest)				
strings (4926)				
debug (n/a)				
manifest (administrator)				
version (n/a)				
certificate (n/a)				
overlay (n/a)				

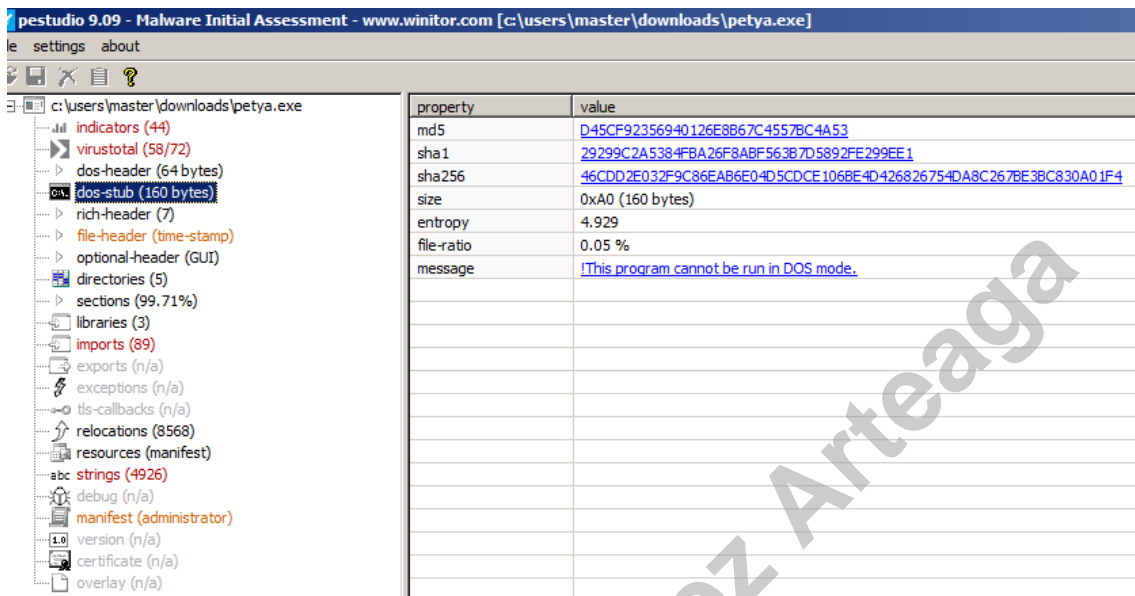
El resultado muestra que 58 de los 72 motores antivirus lo clasifican como malicioso, lo que representa una tasa de detección del 80,5%, altamente significativa. Aunque algunos motores concretos (como ALYac, AhnLab-V3, Baidu, DrWeb, F-Secure o Kaspersky) aparecen como “clean” en esta vista parcial, el elevado número de motores que sí detectan el archivo como malicioso refuerza su perfil de amenaza crítica.

DOS Header

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]	
e settings about	
c:\users\master\downloads\petya.exe	
indicators (44)	property
virustotal (58/72)	md5
dos-header (64 bytes)	sha1
dos-stub (160 bytes)	sha256
rich-header (7)	size
file-header (time-stamp)	entropy
optional-header (GUI)	file-ratio
directories (5)	file-header-offset
sections (99.71%)	
libraries (3)	
imports (89)	
exports (n/a)	
exceptions (n/a)	
tls-callbacks (n/a)	
relocations (8568)	
resources (manifest)	
strings (4926)	
debug (n/a)	
manifest (administrator)	
version (n/a)	
certificate (n/a)	
overlay (n/a)	

El encabezado DOS (DOS Header) forma parte del formato PE y no suele aportar valor directo en términos de detección, pero puede usarse para verificar integridad y formato.

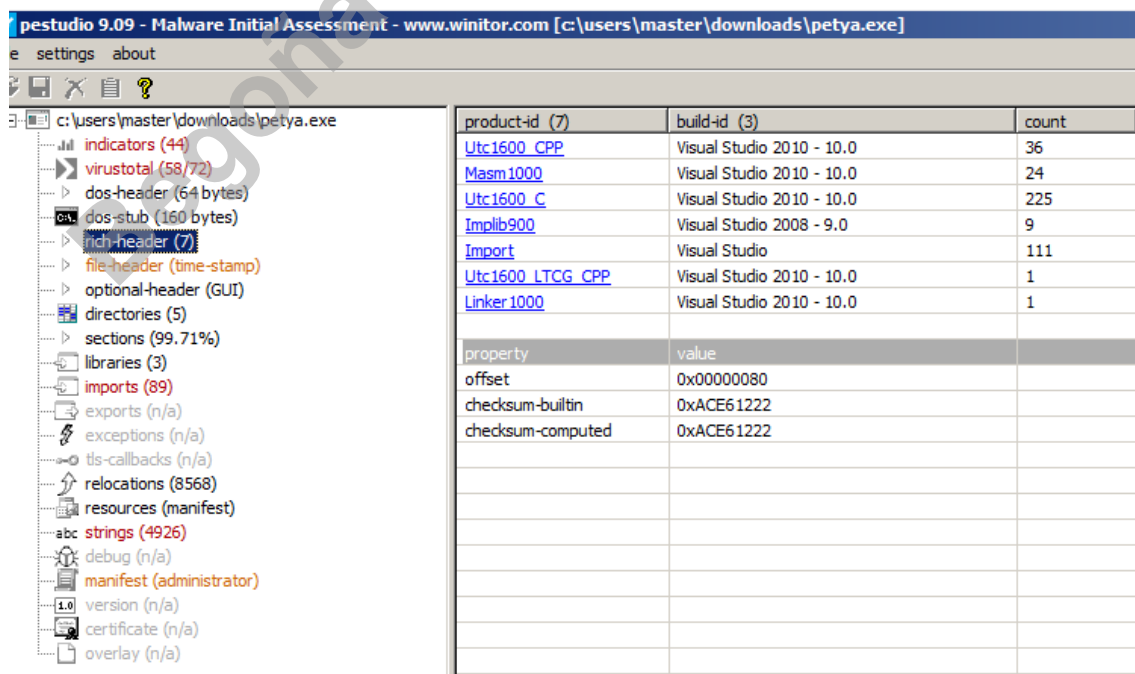
DOS Stub



property	value
md5	D45CF92356940126E8B67C4557BC4A53
sha1	29299C2A5384FBA26F8ABF563B7D5892FE299EE1
sha256	46CDD2E032F9C86EAB6E04D5C0CE106BE4D426826754DA8C267BE3BC830A01F4
size	0xA0 (160 bytes)
entropy	4.929
file-ratio	0.05 %
message	[This program cannot be run in DOS mode.]

En el DOS Stub, se presenta el mensaje “This program cannot be run in DOS mode”, lo cual es esperado. Se ha registrado un nivel de entropía relativamente alto, que podría indicar modificaciones o empaquetado, lo cual también es común en muestras de malware.

Rich Header

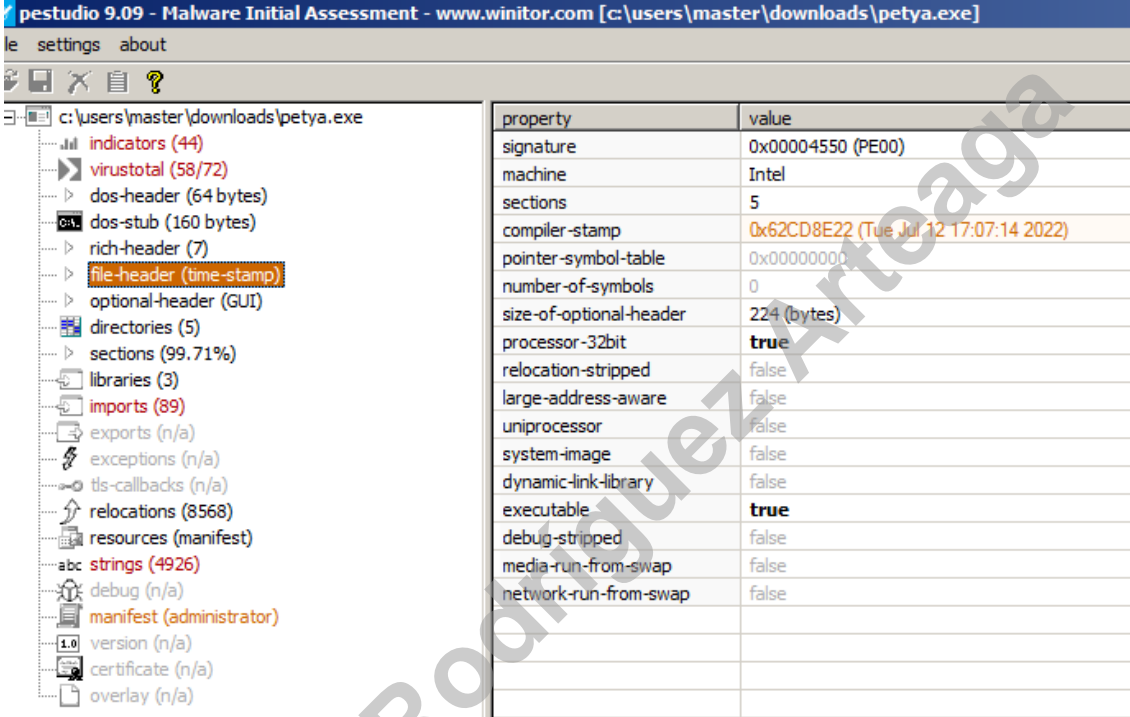


product-id (7)	build-id (3)	count
Utc1600_CPP	Visual Studio 2010 - 10.0	36
Masm1000	Visual Studio 2010 - 10.0	24
Utc1600_C	Visual Studio 2010 - 10.0	225
Implib900	Visual Studio 2008 - 9.0	9
Import	Visual Studio	111
Utc1600_LTCG_CPP	Visual Studio 2010 - 10.0	1
Linker1000	Visual Studio 2010 - 10.0	1

property	value
offset	0x00000080
checksum-builtin	0xACE61222
checksum-computed	0xACE61222

El encabezado Rich indica información sobre los entornos de compilación empleados. En este análisis, aparecen referencias a versiones de Visual Studio y a técnicas de compilación optimizadas (LTCG). Según los apuntes, esto puede sugerir que el ejecutable ha sido generado con herramientas legítimas pero reutilizadas para propósitos maliciosos, y que posiblemente ha sido vinculado estáticamente para dificultar el análisis forense.

File Header



property	value
signature	0x00004550 (PE00)
machine	Intel
sections	5
compiler-stamp	0x62CD8E22 (Tue Jul 12 17:07:14 2022)
pointer-symbol-table	0x00000000
number-of-symbols	0
size-of-optional-header	224 (bytes)
processor-32bit	true
relocation-stripped	false
large-address-aware	false
uniprocessor	false
system-image	false
dynamic-link-library	false
executable	true
debug-stripped	false
media-run-from-swap	false
network-run-from-swap	false

El campo de fecha de compilación (Time Stamp) muestra una fecha de julio de 2022, lo cual no coincide con la cronología original del malware Petya (2016). Esto podría implicar que el binario haya sido recompilado o alterado para confundir herramientas automatizadas. Además, se confirma que se trata de un ejecutable de 32 bits sin atributos de depuración, algo habitual en malware ofuscado.

Optional Header (GUI)

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

e settings about

c:\users\master\downloads\petya.exe

- indicators (44)
- virustotal (58/72)
- dos-header (64 bytes)
- dos-stub (160 bytes)
- rich-header (7)
- file-header (time-stamp)
- optional-header (GUI)**
- directories (5)
- sections (99.71%)
- libraries (3)
- imports (89)
- exports (n/a)
- exceptions (n/a)
- tls-callbacks (n/a)
- relocations (8568)
- resources (manifest)
- strings (4926)
- debug (n/a)
- manifest (administrator)
- version (n/a)
- certificate (n/a)
- overlay (n/a)

property	value
magic	0x010B
entry-point	0x00023A6B (section:.text)
base-of-code	0x00001000 (section:n/a)
base-of-data	0x00033000 (section:.rdata)
image-base	0x00400000
linker-version	10.0
size-of-code	0x00031C00 (203776 bytes)
size-of-initialized-data	148480 (bytes)
size-of-uninitialized-data	0 (bytes)
size-of-image	376832 (bytes)
size-of-headers	1024 (bytes)
size-of-stack-reserve	1048576 (bytes)
size-of-stack-commit	4096 (bytes)
size-of-heap-reserve	1048576 (bytes)
size-of-heap-commit	4096 (bytes)
section-alignment	0x00001000 (4096 bytes)
file-alignment	0x00000200 (512 bytes)
os-version	5.1
image-version	0.0
Win32VersionValue	0x00000000
subsystem	GUI
subsystem-version	5.1
file-checksum	0x000654CB
real-checksum	0x000654CB
LoaderFlags	0x00000000
directories-number	16
address-space-layout-randomization (ASLR)	true
code-integrity	false
data-execution-prevention (DEP)	true
image-isolation	true
structured-exception-handling (SEH)	true
image-bound	false
windows-driver-model (WDM)	false
terminal-server-aware	true
control-flow-guard (CFG)	false

La Optional Header muestra un subsistema GUI y sistema operativo objetivo Windows XP/2003. A pesar de tratarse de un malware, se detecta la activación de medidas de seguridad como ASLR, DEP, SEH y Code Integrity. Esto puede parecer contradictorio, pero como vimos en clase, algunas muestras avanzadas simulan buenas prácticas para evadir detección por heurística de antivirus.

Directories

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

settings about

c:\users\master\downloads\petya.exe

- indicators (44)
- virustotal (58/72)
 - dos-header (64 bytes)
 - dos-stub (160 bytes)
 - rich-header (7)
 - file-header (time-stamp)
 - optional-header (GUI)
 - directories (5)
 - sections (99.71%)
 - libraries (3)
 - imports (89)
 - exports (n/a)
 - exceptions (n/a)
 - tls-callbacks (n/a)
 - relocations (8568)
 - resources (manifest)
 - strings (4926)
 - debug (n/a)
 - manifest (administrator)
 - version (n/a)
 - certificate (n/a)
 - overlay (n/a)

name (15/15)	size (bytes)	location (address)	location (section)	time-stamp
export-table	0x00000000 (0)	0x00000000	n/a	n/a
import-name	0x00000050 (80)	0x000447EC	.rdata	empty
resource	0x000001C0 (448)	0x00055000	.rsrsc	empty
exception	0x00000000 (0)	0x00000000	n/a	n/a
security	0x00000000 (0)	0x00000000	n/a	n/a
relocation	0x00004500 (17664)	0x00056000	.reloc	empty
debug	0x00000000 (0)	0x00000000	n/a	n/a
architecture	0x00000000 (0)	0x00000000	n/a	n/a
global-pointer	0x00000000 (0)	0x00000000	n/a	n/a
thread-storage	0x00000000 (0)	0x00000000	n/a	n/a
load-configuration	0x00000040 (64)	0x00044280	.rdata	empty
bound-import	0x00000000 (0)	0x00000000	n/a	n/a
import-address	0x00000170 (368)	0x00033000	.rdata	empty
delay-loaded	0x00000000 (0)	0x00000000	n/a	n/a
com-runtime	0x00000000 (0)	0x00000000	n/a	n/a

En el análisis de directorios PE existen cinco entradas con contenido válido, incluyendo tablas de importación y relocalización. Un tamaño elevado en la tabla de relocalaciones puede estar vinculado a la capacidad del archivo de ejecutarse en distintas posiciones de memoria, lo que complica la detección por firmas estáticas.

Sections

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

settings about

c:\users\master\downloads\petya.exe

- indicators (44)
- virustotal (58/72)
 - dos-header (64 bytes)
 - dos-stub (160 bytes)
 - rich-header (7)
 - file-header (time-stamp)
 - optional-header (GUI)
 - directories (5)
 - sections (99.71%)
 - libraries (3)
 - imports (89)
 - exports (n/a)
 - exceptions (n/a)
 - tls-callbacks (n/a)
 - relocations (8568)
 - resources (manifest)
 - strings (4926)
 - debug (n/a)
 - manifest (administrator)
 - version (n/a)
 - certificate (n/a)
 - overlay (n/a)

property	value	value	value	value	value
name	.text	.rdata	.data	.rsrsc	.reloc
md5	9598758c740a2a305339...	35670a480ef9521c748c88...	c070f8470586668d463548...	1372309f8684cf0794f7f043...	5c08f85769c86756957b6e8...
entropy	6.658	6.072	5.394	5.073	5.705
file-ratio (99.71%)	57.68 %	21.01 %	14.64 %	0.14 %	6.23 %
raw-address	0x00000400	0x00033000	0x00044200	0x00050C00	0x00050E00
raw-size (352256 bytes)	0x00031C00 (203776 bytes)	0x00012200 (74240 bytes)	0x0000CA00 (51712 bytes)	0x00000200 (512 bytes)	0x00005600 (22016 bytes)
virtual-address	0x00040100	0x00043000	0x00446000	0x00455000	0x00456000
virtual-size (360368 bytes)	0x0003186A (203626 bytes)	0x00012056 (73814 bytes)	0x0000E0EC (60940 bytes)	0x000001C0 (448 bytes)	0x00005424 (21540 bytes)
entry-point	0x00023A6B	-	-	-	-
characteristics	0x60000020	0x40000040	0xC0000040	0x40000040	0x42000040
writeable	-	-	x	-	-
executable	x	-	-	-	-
shareable	-	-	-	-	-
discardable	-	-	-	-	x
initialized-data	-	x	x	x	x
uninitialized-data	-	-	-	-	-
unreadable	-	-	-	-	-
self-modifying	-	-	-	-	-
virtualized	-	-	-	-	-
file	n/a	n/a	n/a	n/a	n/a

El ejecutable contiene cinco secciones. La sección .text presenta una entropía de 6.658, lo que podría indicar cifrado u ofuscación de código. Además, el hecho de que el 99,71% del archivo esté ocupado por datos útiles sugiere empaquetamiento o compresión, lo que dificulta el análisis estático.

Librerías

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

settings about

	library (3)	blacklist (0)	type (1)	imports (89)	description
indicators (44)	kernel32.dll	-	implicit	76	Windows NT BASE API Client DLL
virustotal (58/72)	advapi32.dll	-	implicit	9	Advanced Windows 32 Base API
dos-header (64 bytes)	user32.dll	-	implicit	4	Multi-User Windows USER API Client DLL
dos-stub (160 bytes)					
rich-header (7)					
file-header (time-stamp)					
optional-header (GUI)					
directories (5)					
sections (99.71%)					
libraries (3)					
imports (89)					
exports (n/a)					
exceptions (n/a)					
tls-callbacks (n/a)					
relocations (8568)					
resources (manifest)					
strings (4926)					
debug (n/a)					
manifest (administrator)					
version (n/a)					
certificate (n/a)					
overlay (n/a)					

El malware importa funciones clave desde librerías del sistema como kernel32.dll, advapi32.dll y user32.dll. Estas librerías son ampliamente utilizadas por programas maliciosos para interactuar con el sistema, manipular servicios y realizar tareas de persistencia. Esto coincide con lo estudiado sobre los patrones comunes de comportamiento de malware en sistemas Windows.

Imports

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

settings about

	name (89)	group (12)	type (1)	ordinal (0)	blacklist (21)	anti-debug (0)	undocumented (0)	deprecated (6)	library (3)
indicators (44)	GetDesktopWindow	windowing	implicit	-	x	-	-	-	user32.dll
virustotal (58/72)	GetTickCount	system-information	implicit	-	-	-	-	-	kernel32.dll
dos-header (64 bytes)	QueryPerformanceCounter	system-information	implicit	-	-	-	-	-	kernel32.dll
dos-stub (160 bytes)	GetVersionExA	system-information	implicit	-	-	-	-	x	kernel32.dll
rich-header (7)	IsDebuggerPresent	system-information	implicit	-	-	-	-	-	kernel32.dll
file-header (time-stamp)	IsProcessorFeaturePresent	system-information	implicit	-	-	-	-	-	kernel32.dll
optional-header (GUI)	GetSystemDirectoryA	system-information	implicit	-	-	-	-	-	kernel32.dll
directories (5)	EnterCriticalSection	synchronization	implicit	-	-	-	-	-	kernel32.dll
sections (99.71%)	LeaveCriticalSection	synchronization	implicit	-	-	-	-	-	kernel32.dll
libraries (3)	InitializeCriticalSectionAndSpinCount	synchronization	implicit	-	-	-	-	-	kernel32.dll
imports (89)	ResetCriticalSection	synchronization	implicit	-	-	-	-	-	kernel32.dll
exports (n/a)	InterlockedIncrement	synchronization	implicit	-	-	-	-	-	kernel32.dll
exceptions (n/a)	InterlockedDecrement	synchronization	implicit	-	-	-	-	-	kernel32.dll
tls-callbacks (n/a)	AdjustTokenPrivileges	security	implicit	-	x	-	-	-	advapi32.dll
relocations (8568)	LookupPrivilegeValueA	security	implicit	-	x	-	-	-	advapi32.dll
resources (manifest)	OpenProcessToken	security	implicit	-	x	-	-	-	advapi32.dll
strings (4926)	GetMemorystatus	memory	implicit	-	x	-	-	x	kernel32.dll
debug (n/a)	HeapSetInformation	memory	implicit	-	x	-	-	-	kernel32.dll
manifest (administrator)	HeapFree	memory	implicit	-	-	-	-	-	kernel32.dll
version (n/a)	HeapReAlloc	memory	implicit	-	-	-	-	-	kernel32.dll
certificate (n/a)	HeapAlloc	memory	implicit	-	-	-	-	-	kernel32.dll
overlay (n/a)	HeapCreate	memory	implicit	-	-	-	-	-	kernel32.dll
	GetStringTypeW	memory	implicit	-	-	-	-	x	kernel32.dll
	HeapSize	memory	implicit	-	-	-	-	-	kernel32.dll
	GetProcessHeap	memory	implicit	-	-	-	-	-	kernel32.dll
	GetFileType	file	implicit	-	-	-	-	-	kernel32.dll
	GetSystemTimeAsFileTime	file	implicit	-	-	-	-	-	kernel32.dll
	FlushFileBuffers	file	implicit	-	-	-	-	-	kernel32.dll
	CreateFileW	file	implicit	-	-	-	-	-	kernel32.dll
	SetEndOfFile	file	implicit	-	-	-	-	-	kernel32.dll
	ReadFile	file	implicit	-	-	-	-	-	kernel32.dll
	WriteFile	file	implicit	-	-	-	-	-	kernel32.dll
	SetFilePointerEx	file	implicit	-	-	-	-	-	kernel32.dll
	SetFilePointer	file	implicit	-	-	-	-	-	kernel32.dll
	CreateFileA	file	implicit	-	-	-	-	-	kernel32.dll
	GetCurrentProcessId	execution	implicit	-	x	-	-	-	kernel32.dll
	GetCommandLineA	execution	implicit	-	-	-	-	-	kernel32.dll
	GetStartupInfoW	execution	implicit	-	-	-	-	-	kernel32.dll
	TerminateProcess	execution	implicit	-	x	-	-	-	kernel32.dll
	FreeEnvironmentStringsW	execution	implicit	-	-	-	-	-	kernel32.dll
	GetEnvironmentStringsW	execution	implicit	-	x	-	-	-	kernel32.dll
	GetCurrentThreadId	execution	implicit	-	x	-	-	-	kernel32.dll
	TlsAlloc	execution	implicit	-	-	-	-	-	kernel32.dll
	TlsSetValue	execution	implicit	-	-	-	-	-	kernel32.dll
	TlsGetSetValue	execution	implicit	-	-	-	-	-	kernel32.dll
	TlsFree	execution	implicit	-	-	-	-	-	kernel32.dll

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

settings about

c:\users\master\downloads\petya.exe

name (89)	group (12)	type (1)	ordinal (0)	blacklist (21)	anti-debug (0)	undocumented (0)	deprecated (6)	library (3)
TlsSetValue	execution	implicit	-	-	-	-	-	kernel32.dll
TlsFree	execution	implicit	-	-	-	-	-	kernel32.dll
Sleep	execution	implicit	-	-	-	-	-	kernel32.dll
GetCurrentProcess	execution	implicit	-	-	-	-	-	kernel32.dll
ExitProcess	execution	implicit	-	-	-	-	-	kernel32.dll
UnhandledExceptionFilter	exception-handling	implicit	-	-	-	-	-	kernel32.dll
SetUnhandledExceptionFilter	exception-handling	implicit	-	-	-	-	-	kernel32.dll
FreeLibrary	dynamic-library	implicit	-	-	-	-	-	kernel32.dll
LoadLibraryA	dynamic-library	implicit	-	-	-	-	-	kernel32.dll
GetModuleFileNameW	dynamic-library	implicit	-	x	-	-	-	kernel32.dll
GetModuleHandleW	dynamic-library	implicit	-	-	-	-	-	kernel32.dll
GetModuleHandleA	dynamic-library	implicit	-	x	-	-	-	kernel32.dll
LoadLibraryW	dynamic-library	implicit	-	-	-	-	-	kernel32.dll
GetModuleHandleA	dynamic-library	implicit	-	-	-	-	-	kernel32.dll
GetProcAddress	dynamic-library	implicit	-	-	-	-	-	kernel32.dll
GetLastError	diagnostic	implicit	-	-	-	-	-	kernel32.dll
SetLastError	diagnostic	implicit	-	-	-	-	-	kernel32.dll
RegisterEventSourceA	diagnostic	implicit	-	x	-	-	-	advapi32.dll
ReportEventA	diagnostic	implicit	-	x	-	-	-	advapi32.dll
DeregisterEventSource	diagnostic	implicit	-	x	-	-	-	advapi32.dll
GetProcessWindowStation	desktop	implicit	-	x	-	-	-	user32.dll
GetUserObjectInformationW	desktop	implicit	-	x	-	-	-	user32.dll
CryptGenRandom	cryptography	implicit	-	x	-	-	-	advapi32.dll
CryptAcquireContextA	cryptography	implicit	-	x	-	-	-	advapi32.dll
CryptReleaseContext	cryptography	implicit	-	x	-	-	-	advapi32.dll
GetStdHandle	console	implicit	-	-	-	-	-	kernel32.dll
WriteConsoleW	console	implicit	-	-	-	-	-	kernel32.dll
GetConsoleCP	console	implicit	-	-	-	-	-	kernel32.dll
GetConsoleMode	console	implicit	-	-	-	-	-	kernel32.dll
SetStdHandle	console	implicit	-	-	-	-	-	kernel32.dll
GetVersion	-	implicit	-	-	-	-	x	kernel32.dll
DecodePointer	-	implicit	-	-	-	-	-	kernel32.dll
WideCharToMultiByte	-	implicit	-	-	-	-	-	kernel32.dll
EncodePointer	-	implicit	-	-	-	-	-	kernel32.dll
GetCPInfo	-	implicit	-	-	-	-	-	kernel32.dll
GetACP	-	implicit	-	-	-	-	-	kernel32.dll
GetOEMCP	-	implicit	-	-	-	-	-	kernel32.dll
IsValidCodePage	-	implicit	-	-	-	-	-	kernel32.dll
MultiByteToWideChar	-	implicit	-	-	-	-	-	kernel32.dll
RtlUnwind	-	implicit	-	-	-	-	-	kernel32.dll
LCMapStringW	-	implicit	-	-	-	-	x	kernel32.dll
CloseHandle	-	implicit	-	-	-	-	-	kernel32.dll
DeviceIoControl	-	implicit	-	x	-	-	-	kernel32.dll
SetHandleCount	-	implicit	-	-	-	-	x	kernel32.dll
MessageBoxA	-	implicit	-	-	-	-	-	user32.dll

El archivo petya.exe importa 89 funciones desde librerías críticas del sistema como kernel32.dll, advapi32.dll o user32.dll. Muchas de ellas pertenecen a categorías sensibles como ejecución de procesos, carga dinámica de DLLs y funciones criptográficas. Varias están marcadas como blacklist, lo que indica que son comúnmente usadas por malware. Este tipo de imports refuerza el perfil de petya.exe como ransomware con capacidad de cifrado, evasión y manipulación del sistema.

Relocations

pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\master\downloads\petya.exe]

settings about

c:\users\master\downloads\petya.exe

item (8568)	address	type (2)
0x0000	0x00001000	absolute
0x0000	0x00003000	absolute
0x0000	0x00005000	absolute
0x0000	0x00008000	absolute
0x0000	0x00009000	absolute
0x0000	0x0000A000	absolute
0x0000	0x0000C000	absolute
0x0000	0x0000E000	absolute
0x0000	0x00010000	absolute
0x0000	0x00011000	absolute
0x0000	0x00013000	absolute
0x0000	0x00016000	absolute
0x0000	0x00018000	absolute
0x0000	0x00019000	absolute
0x0000	0x0001A000	absolute
0x0000	0x0001D000	absolute
0x0000	0x0001F000	absolute
0x0000	0x00021000	absolute
0x0000	0x00024000	absolute
0x0000	0x00026000	absolute
0x0000	0x0002A000	absolute

El archivo contiene más de 8.500 entradas de relocalización de tipo absoluta. Este dato, sugiere que el ejecutable ha sido preparado para cargar en distintas ubicaciones de memoria, lo cual puede estar relacionado con técnicas anti-debugging o anti-forensics.

Resources (manifest)

[illegible]

El ejecutable contiene un manifiesto embebido, con estructura XML y entropía normal (4.778), lo que indica que no está ofuscado. Este recurso puede ser utilizado por el malware para declarar privilegios elevados o compatibilidad específica con Windows, algo común en ransomware sofisticado.

Strings

[illegible]

Hemos detectado cadenas de texto que hacen referencia a direcciones .onion, funciones criptográficas, variables relacionadas con contraseñas y privilegios como

SeShutdownPrivilege. Todo ello es típico en el comportamiento de ransomware, cuya finalidad es cifrar el sistema y dificultar su recuperación.

Begoña Rodríguez Arteaga

Análisis dinámico

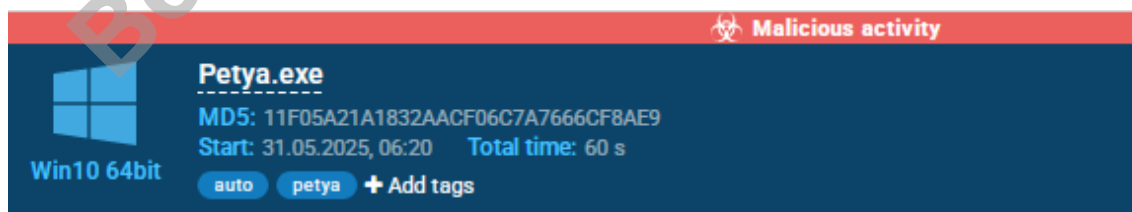
El análisis dinámico permite observar el comportamiento real de una muestra maliciosa durante su ejecución en un entorno controlado. A diferencia del análisis estático, que examina el código sin ejecutarlo, esta técnica facilita la identificación de acciones específicas que realiza el malware en tiempo de ejecución, como creación de procesos, modificaciones en el sistema de archivos o el registro, conexiones de red y alteraciones en servicios del sistema operativo.

Aplicando esta metodología al archivo petya.exe, hemos utilizado la plataforma Any.run para simular la ejecución del malware en un entorno aislado y monitorizado. Este enfoque permite detectar patrones de actividad que podrían pasar desapercibidos en un análisis superficial, como el uso de herramientas legítimas del sistema (por ejemplo, runas.exe, WinRAR.exe o rundll32.exe) con fines maliciosos, una técnica ampliamente documentada en el marco MITRE ATT&CK bajo el concepto de Living off the Land, técnica que aprovecha herramientas y funciones legítimas para realizar acciones maliciosas en la máquina víctima.

El análisis dinámico realizado demuestra cómo petya.exe desencadena una serie de acciones encadenadas, incluyendo la descompresión y ejecución de una DLL maliciosa, interacción con el registro de Windows, y llamadas a procesos críticos. Estos comportamientos confirman su perfil como amenaza activa y evasiva, y proporcionan información clave para su detección y contención.

A continuación, se describen los eventos más relevantes observados durante la ejecución monitorizada de la muestra. Vamos a comenzar el análisis dinámico con la utilización de la herramienta Any.Run.

Esta es la ficha técnica de la muestra utilizada del archivo ejecutable de Petya.exe.



Resumen de las tácticas de acción de Petya



El análisis dinámico permitió mapear las acciones observadas sobre la matriz MITRE ATT&CK, revelando que Petya.exe ejecuta al menos cinco eventos maliciosos distribuidos en cuatro tácticas principales:

Ejecución

El archivo malicioso ha sido lanzado mediante un proceso de ejecución del usuario, a través de la ejecución de un archivo malicioso. Esta técnica implica la necesidad de interacción por parte del usuario, lo que indica que el atacante probablemente se apoya en ingeniería social para lograr la ejecución del binario.

Escalada de privilegios

El binario realiza abuso del mecanismo de control de elevación, UAC, que busca obtener privilegios administrativos en el sistema mediante bypass del control de cuentas de usuario, lo que permite al malware operar con mayor libertad dentro del sistema operativo sin restricciones.

Evasión de defensas

Se repite la misma técnica anterior, ya que el bypass de UAC también cumple la función de evadir mecanismos de protección del sistema, como Windows Defender o restricciones de ejecución.

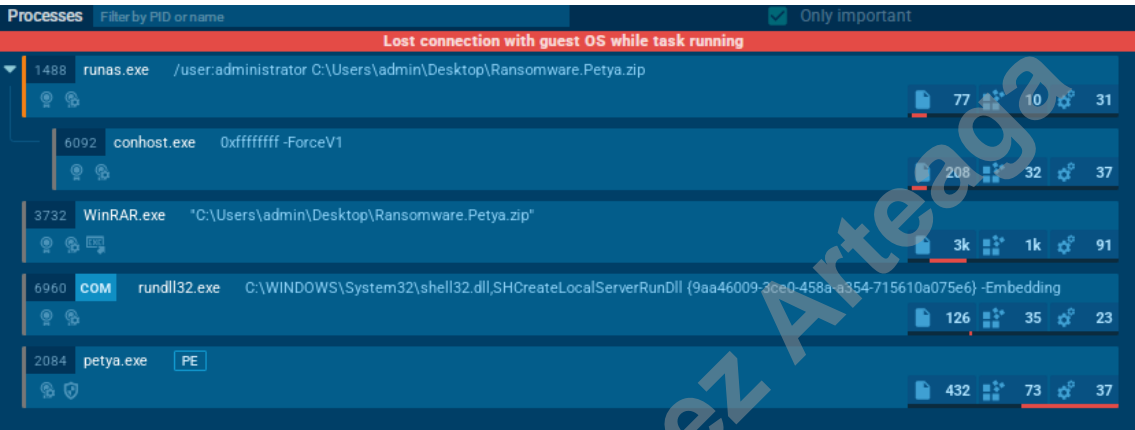
Descubrimiento

El ejecutable realiza consultas de información del sistema (System Information Discovery) y del registro (Query Registry), lo cual permite al malware recolectar datos sobre el entorno operativo. Estas acciones son típicas en fases tempranas del ciclo de ataque para adaptar su comportamiento a la víctima.

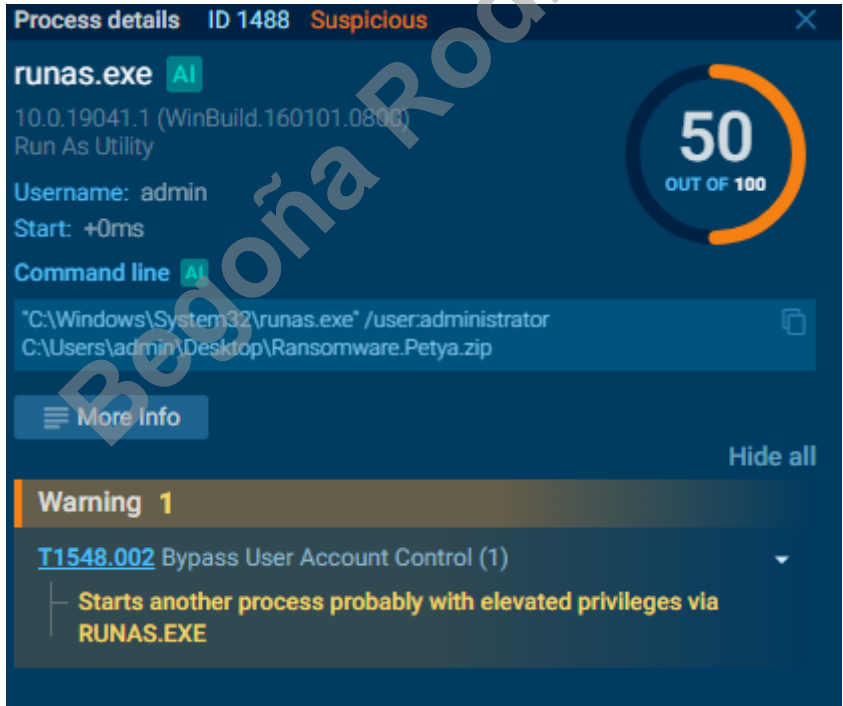
Estos son los procesos que se ejecutan en el sistema.

Processes Add for printing			
Total processes	Monitored processes	Malicious processes	Suspicious processes
134	5	0	1

Cada proceso en detalle.



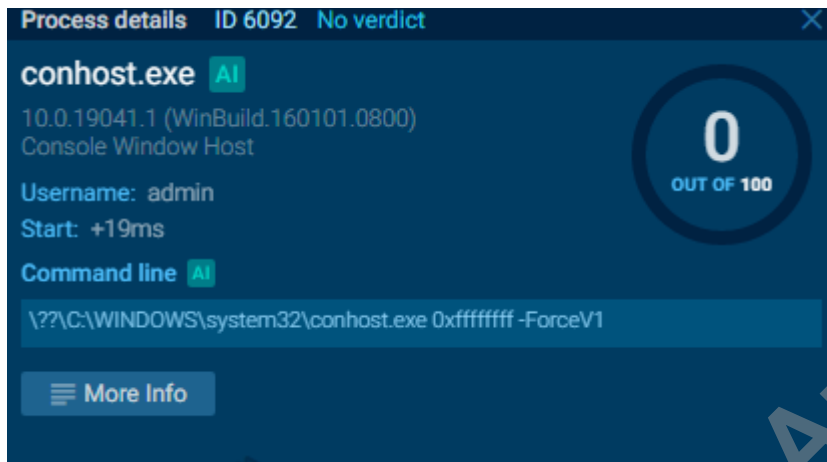
Runas.exe



Lanzado por el usuario admin, ejecuta el archivo Ransomware.Petya.zip con elevación de privilegios mediante la instrucción de las credenciales `runas /user:administrator`.

Se asigna una puntuación de riesgo 50/100 y se cataloga como táctica T1548.002. Esto indica un intento claro de evadir el UAC para ejecutar el contenido con permisos administrativos.

Conhost.exe



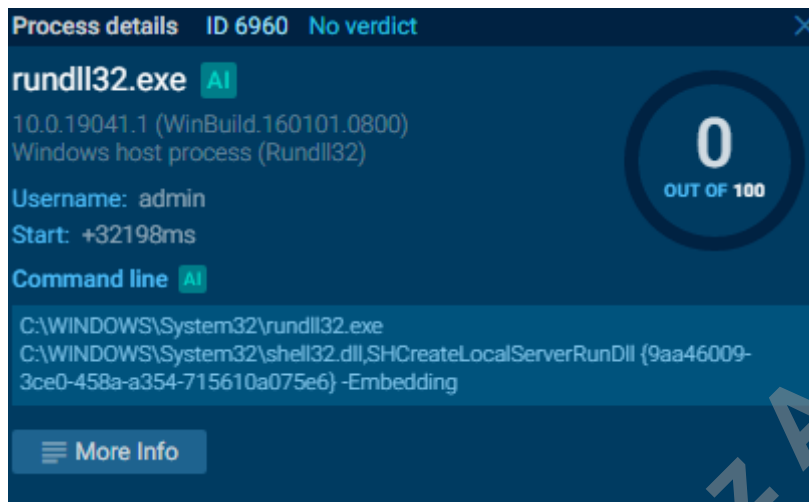
Ejecutado como parte del entorno de consola, no se atribuyen acciones maliciosas, pero forma parte de la cadena de ejecución iniciada por runas.exe. No presenta indicadores de riesgo directo.

Winrar.exe



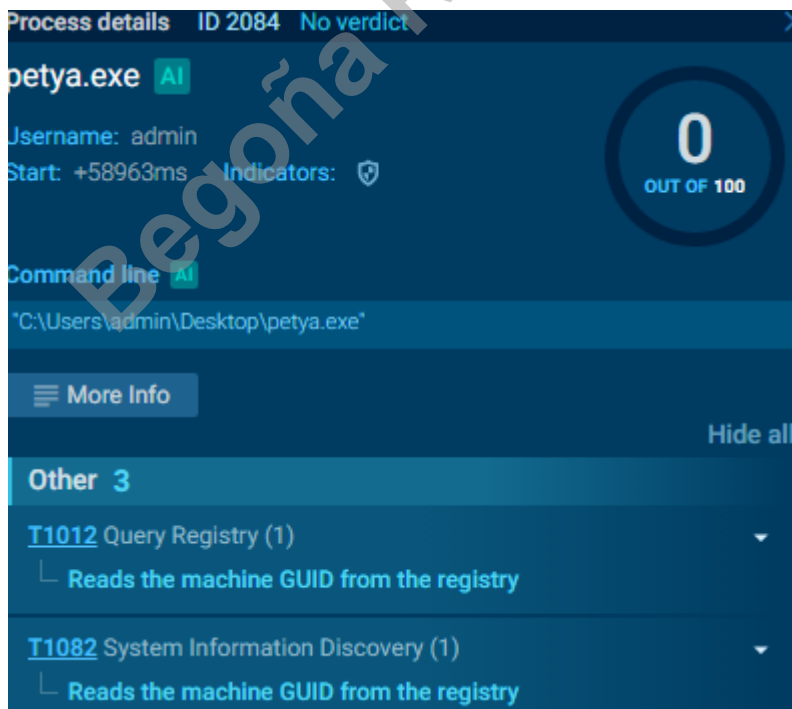
Ejecutado manualmente por el usuario con el fin de descomprimir el archivo ZIP malicioso. No se detecta riesgo directo, pero su uso permite entregar el payload final. Está relacionado con la técnica T1204.002.

Rundll32.exe

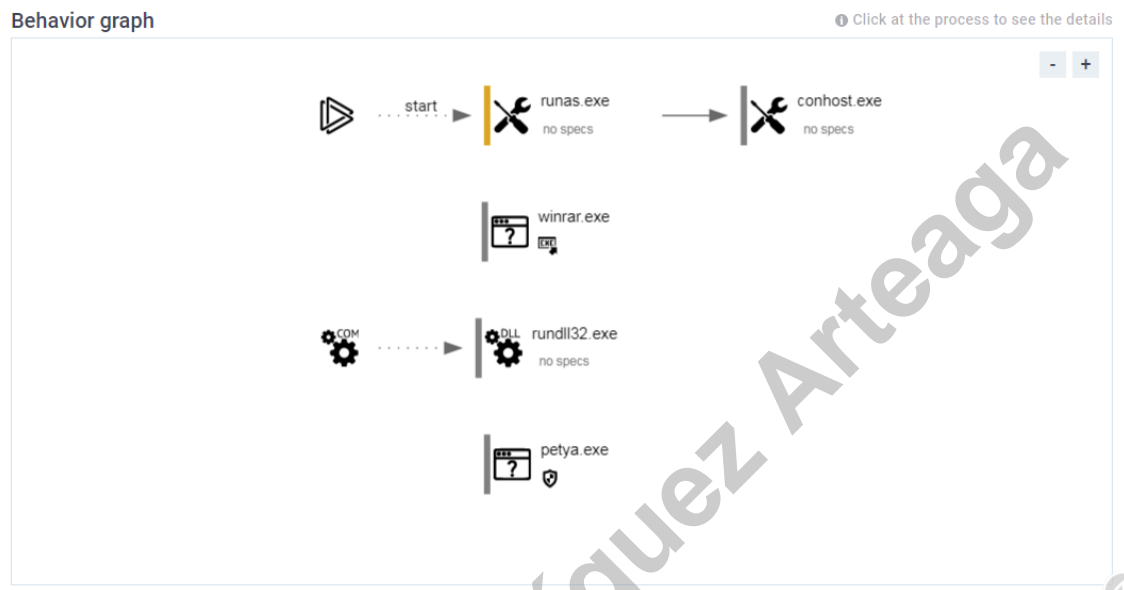


Carga la librería shell32.dll con el comando SHCreateLocalServerRunDll, lo que permite la ejecución de una DLL en contexto privilegiado. Aunque sin alerta, esta acción encaja en técnicas de ejecución indirecta mediante componentes del sistema.

Petya.exe



Ejecutado desde el escritorio del usuario admin, este binario realiza consultas al registro y recolecta información del sistema, según las técnicas T1012 y T1082. No presenta una puntuación elevada de riesgo de forma inmediata, lo cual puede deberse a su comportamiento diferido o en espera de condiciones específicas.



Actividad en el registro de Windows

El malware genera una cantidad significativa de interacciones con el registro: 1886 eventos en total, de los cuales 1851 corresponden a operaciones de lectura, 22 a escritura y 13 a eliminación de valores.

Las entradas modificadas se vinculan principalmente a la aplicación WinRAR, la cual es utilizada como medio para descomprimir el archivo Ransomware.Petya.zip. Las claves modificadas están ubicadas en:

HKEY_CURRENT_USER\SOFTWARE\WinRAR\ArcHistory

HKEY_CURRENT_USER\SOFTWARE\WinRAR\FileList\FileColumnWidths

HKEY_CURRENT_USER\SOFTWARE\WinRAR\DialogEditHistory\ExtrPath

Estas claves reflejan archivos descomprimidos recientemente y preferencias del entorno de usuario, incluyendo el nombre y ubicación del archivo ZIP malicioso. Estas evidencias refuerzan el uso de la herramienta como parte del vector de entrega inicial.

Registry activity

Total events	Read events	Write events	Delete events
1 886	1 851	22	13
Modification events			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: C:\Users\admin\Desktop\preferences.zip</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\ArcHistory</div><div>Name: 3</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: C:\Users\admin\Desktop\chromium_ext.zip</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\ArcHistory</div><div>Name: 2</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: C:\Users\admin\Desktop\omni_23_10_2024_.zip</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\ArcHistory</div><div>Name: 1</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: C:\Users\admin\Desktop\Ransomware.Petya.zip</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\ArcHistory</div><div>Name: 0</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: 120</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\FileList\FileColumnWidths</div><div>Name: name</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: 80</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\FileList\FileColumnWidths</div><div>Name: size</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: 120</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\FileList\FileColumnWidths</div><div>Name: type</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: write</div><div>Value: 100</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\FileList\FileColumnWidths</div><div>Name: mtime</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: delete value</div><div>Value:</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\DialogEditHistory\ExtrPath</div><div>Name: 15</div></div>			
<div><div>(PID) Process: (3732) WinRAR.exe</div><div>Operation: delete value</div><div>Value:</div></div> <div><div>Key: HKEY_CURRENT_USER\SOFTWARE\WinRAR\DialogEditHistory\ExtrPath</div><div>Name: 14</div></div>			

Actividad de la red

Durante el análisis dinámico de petya.exe, se observó un volumen moderado de tráfico de red, registrándose 7 peticiones HTTP(S), 32 conexiones TCP/UDP y 17 consultas DNS. A pesar de no haberse detectado ninguna amenaza directa asociada a estas conexiones según las listas blancas (whitelisted), su análisis resulta esencial

para comprender el comportamiento del archivo en red y evaluar posibles técnicas de comunicación externa o contacto con servidores de comando y control (C2).

Una actividad de red legítima suele estar bien documentada, ser predecible y dirigirse a dominios conocidos. En el contexto de malware, sin embargo, cualquier comportamiento que implique consultas anómalas, conexiones cifradas sin justificación o peticiones a dominios de reputación incierta debe ser considerado sospechoso.

Network activity

HTTP(S) requests 7 TCP/UDP connections 32 DNS requests 17 Threats 0

HTTP requests

PID	Process	Method	HTTP Code	IP	URL	CN	Type	Size	Reputation
4712	MoUsCoreWorker.exe	GET	200	2.16.164.32:80	http://crl.microsoft.com/pki/crl/products/MicRooCerAut2011_2011_03_22.crl	unknown	—	—	whitelisted
4712	MoUsCoreWorker.exe	GET	200	2.23.246.101:80	http://www.microsoft.com/pkiops/crl/MicSecSerCA2011_2011-10-18.crl	unknown	—	—	whitelisted
5064	SearchApp.exe	GET	200	2.17.190.73:80	http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgMCGGUABBTjrydRyt%2BApF3GSPypFHBxR5xt0QU9tPmHxdiuNkHMEWnpYim8S8YCEAl5PUjX4kJarLQcAAso18o%3D	unknown	—	—	whitelisted
1176	svchost.exe	GET	200	2.17.190.73:80	http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgMCGGUABBSAUQYBMQ2awn1Rh6Doh%2FsBYgFV7gQUA95QNVbRTLtm8KPIGxvDI7i90VUCEAJ0LQoXyo4hxe7H%2Fz9DKA%3D	unknown	—	—	whitelisted
6748	SIHClient.exe	GET	200	23.37.237.227:80	http://www.microsoft.com/pkiops/crl/Microsoft%20ECC%20Product%20Root%20Certificate%20Authority%202018.crl	unknown	—	—	whitelisted
6212	backgroundTaskHost.exe	GET	200	2.17.190.73:80	http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgMCGGUABBQ50otx%2Fh0Ztl%2Bz8SiPI7wEWVxDIQUTJUIBjV5uNuSg%2F6%2BkS7QYXjkCEAUZSZEm49Gjh0j13P68w%3D	unknown	—	—	whitelisted
6748	SIHClient.exe	GET	200	23.37.237.227:80	http://www.microsoft.com/pkiops/crl/Microsoft%20ECC%20Update%20Secure%20Server%20CA%202.1.crl	unknown	—	—	whitelisted

En la sección de HTTP requests se evidencia que varios procesos, como MoUsCoreWorker.exe, SearchApp.exe, svchost.exe y SIHClient.exe, realizaron solicitudes GET a URLs relacionadas principalmente con Microsoft y DigiCert. Estas incluyen peticiones a servicios OCSP (Online Certificate Status Protocol), CRL (Certificate Revocation List) y validación de certificados, como es habitual en sistemas Windows cuando se inicia un ejecutable firmado digitalmente.

En particular, se observan múltiples peticiones a direcciones como <http://crl.microsoft.com> y <http://ocsp.digicert.com>, que indican que el sistema está intentando verificar la validez de certificados digitales, algo relacionado con el arranque de procesos nuevos como parte del flujo normal del sistema operativo. Estas peticiones fueron marcadas como seguras (whitelisted), lo que indica que no se observó comportamiento anómalo en este aspecto.

Connections

PID	Process	IP	Domain	ASN	CN	Reputation
—	—	20.73.194.208:443	—	MICROSOFT-CORP-MSN-AS-BLOCK	NL	whitelisted
1596	svchost.exe	20.73.194.208:443	—	MICROSOFT-CORP-MSN-AS-BLOCK	NL	whitelisted
4712	MoUsoCoreWorker.exe	2.16.164.32:80	crl.microsoft.com	Akamai International B.V.	NL	whitelisted
4	System	192.168.100.255:138	—	—	—	whitelisted
4712	MoUsoCoreWorker.exe	2.23.246.101:80	www.microsoft.com	Ooredoo Q.S.C.	QA	whitelisted
4	System	192.168.100.255:137	—	—	—	whitelisted
5064	SearchApp.exe	2.16.204.136:443	www.bing.com	Akamai International B.V.	DE	whitelisted
1176	svchost.exe	40.126.31.73:443	login.live.com	MICROSOFT-CORP-MSN-AS-BLOCK	IE	whitelisted
1176	svchost.exe	2.17.190.73:80	ocsp.digicert.com	AKAMAI-AS	DE	whitelisted
5064	SearchApp.exe	2.17.190.73:80	ocsp.digicert.com	AKAMAI-AS	DE	whitelisted

En cuanto a las conexiones establecidas, el sistema y varios procesos realizaron comunicaciones con direcciones IP asociadas a Microsoft, Akamai, DigiCert y Bing, principalmente a través de los puertos 80 (HTTP) y 443 (HTTPS). Esto incluye conexiones de SearchApp.exe a www.bing.com, o de svchost.exe a login.live.com, lo cual es común durante el uso normal del sistema operativo o de aplicaciones que requieren conectividad a servicios web.

DNS requests

Domain	IP	Reputation
google.com	172.217.18.14	whitelisted
crl.microsoft.com	2.16.164.32 2.16.164.72 2.16.164.24 2.16.164.9 2.16.164.49 2.16.164.114 2.16.164.106 2.16.164.81 2.16.164.18	whitelisted
www.microsoft.com	2.23.246.101 23.37.237.227	whitelisted
www.bing.com	2.16.204.136 2.16.204.153 2.16.204.147 2.16.204.135 2.16.204.151 2.16.204.156 2.16.204.146 2.16.204.148 2.16.204.134	whitelisted
login.live.com	40.126.31.73 40.126.31.71 40.126.31.67 40.126.31.69 20.190.159.64 20.190.159.23 20.190.159.71 20.190.159.73	whitelisted

El análisis DNS muestra resoluciones para dominios ampliamente conocidos como google.com, bing.com, login.live.com, crt.microsoft.com, ocsf.digicert.com y otros dominios de Microsoft. Todos ellos fueron marcados como de confianza (whitelisted), y responden a necesidades típicas del sistema operativo, tales como sincronización, actualizaciones, validación de certificados o autenticación de usuarios.

Begoña Rodríguez Arteaga

Análisis de código

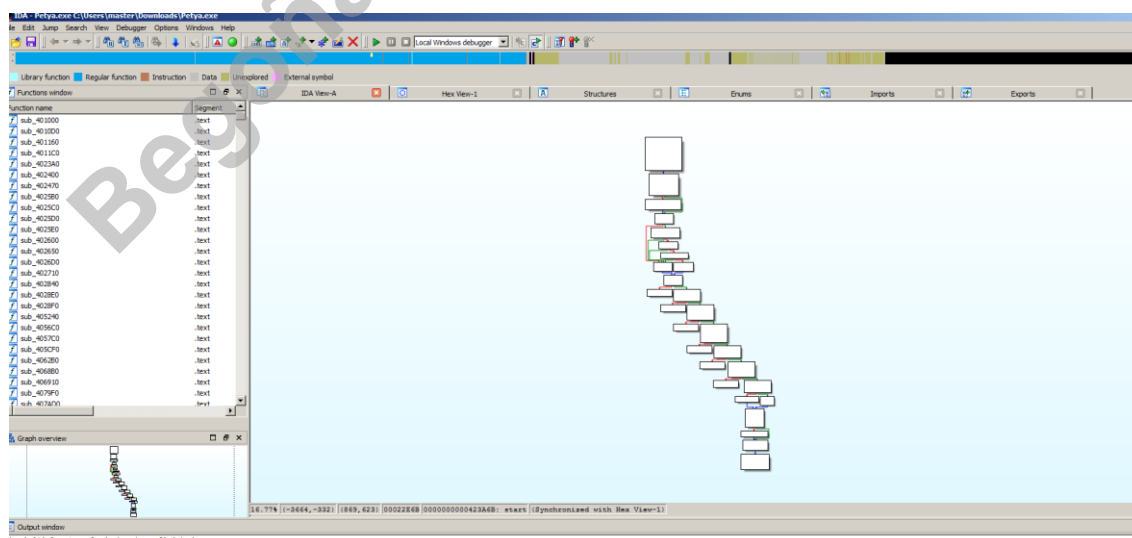
El análisis de código constituye una fase crítica dentro del estudio estático de malware, ya que permite comprender con mayor profundidad la lógica interna del ejecutable, sus rutinas clave, y la forma en que interactúa con el sistema operativo y sus recursos. A través del uso de herramientas como IDA Pro, es posible desensamblar el binario petya.exe y examinar el comportamiento de bajo nivel que escapa al análisis superficial de indicadores, cadenas o cabeceras.

Este tipo de análisis, orientado a la ingeniería inversa, facilita la identificación de técnicas de evasión, funciones de cifrado, control de flujo, verificación de entorno y condiciones previas a la activación del payload.

En el caso de petya.exe, el análisis del código revela una estructura controlada, con múltiples llamadas a funciones de la API de Windows y bifurcaciones condicionales que responden a patrones comunes en familias de ransomware. A continuación, se detallan los hallazgos más relevantes extraídos a través del análisis estático avanzado.

El análisis de código nos ofrece una vista detallada y profunda del malware. Aunque requiere tiempo, atención y conocimientos técnicos, nos permite entender cómo actúa realmente el malware, incluso cuando intenta ocultarse. Esta fase es clave para detectar variantes, desarrollar herramientas defensivas y mejorar nuestras capacidades como analistas de ciberseguridad.

Vamos a realizar el análisis del código tanto con IDA Pro como con Any.Run.



Estructura general del flujo de ejecución

Las imágenes obtenidas de IDA Pro muestran el análisis inicial del flujo de ejecución del malware petya.exe. En la vista gráfica, se aprecia un flujo lineal de ejecución con bifurcaciones condicionales, lo que evidencia una lógica estructurada de toma de decisiones. Esta secuencia, representada como un diagrama de flujo, es clave para comprender cómo se ramifica el código malicioso en función del entorno en el que se ejecuta. Estos bloques permiten identificar funciones sospechosas, rutinas de evasión y validaciones previas al payload.

Función principal y preparación del entorno

En el comienzo del análisis se observa la función de entrada start, donde se realiza una llamada a GetStartupInfoW, una función típica utilizada por ejecutables Windows para recopilar información sobre cómo ha sido lanzado el proceso. Después, se invoca HeapSetInformation, función que puede emplearse para modificar el comportamiento del heap del proceso. Aunque esta API es legítima, puede ser usada por malware para desactivar técnicas de depuración como el heap corruption detection.

Comprobaciones de integridad y entornos

A lo largo del flujo, el binario realiza múltiples comparaciones contra direcciones de memoria específicas como 0x400000, 0x40003C, 0x400018, 0x400074, y 0x4000E8. Estas comparaciones pueden corresponder a chequeos de la cabecera del propio ejecutable en memoria, lo cual sugiere que el malware podría estar intentando verificar su propia integridad o detectar si ha sido modificado (por ejemplo, por un analista o sistema de defensa).

Estas técnicas son representativas del comportamiento anti-debug y anti-tamper, donde el malware valida que está siendo ejecutado en condiciones esperadas antes de continuar con su ejecución maliciosa.

Extracción y uso de parámetros en línea de comandos

Más adelante se observa el uso de GetCommandLineA, una API crítica cuando se desea capturar argumentos de ejecución del malware. En este caso, el programa parece depender de estos parámetros para modular su comportamiento, lo que es característico de familias de malware polimórficas o cargadores (loaders) que requieren instrucciones externas para ejecutar su carga útil.

El uso de parámetros puede indicar flexibilidad operativa del malware, como la selección de objetivos, directorios o el cifrado de unidades específicas.

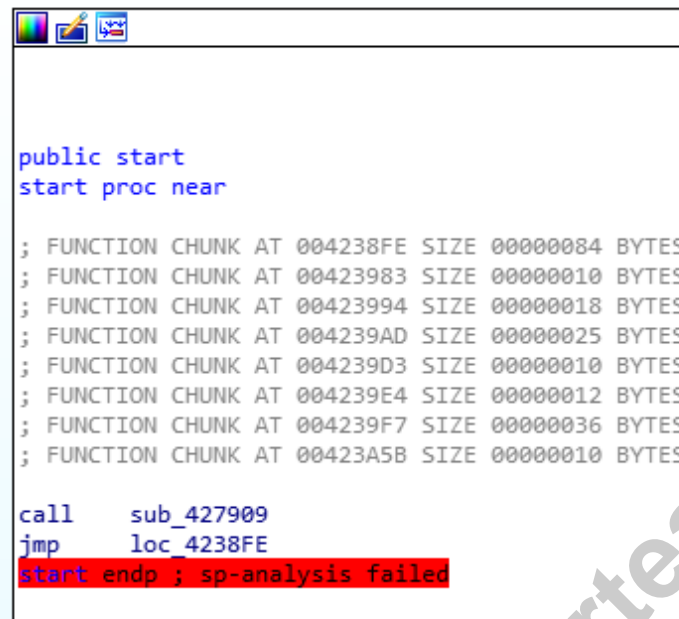
Posible ejecución de payload y finalización controlada

En las fases finales del flujo, se observa la ejecución de una subrutina importante mediante call sub_4011C0, seguida del uso de la instrucción ExitProcess, que finaliza el proceso de forma controlada. Esto es un patrón común en malware que ejecuta su payload en segundo plano o que actúa descargando o invocando otro componente y saliendo silenciosamente para no despertar sospechas.

La forma en que finaliza el binario, junto con las instrucciones de comparación y bifurcación que le preceden, indica una lógica defensiva pensada para evitar ejecución innecesaria si ciertas condiciones no se cumplen. Este tipo de control condicional es habitual en amenazas persistentes avanzadas (APT) y campañas de ransomware como Petya.

El código ensamblador visualizado en IDA Pro revela un binario cuidadosamente estructurado, que realiza múltiples verificaciones de integridad, consulta información del entorno de ejecución y adapta su comportamiento en función de parámetros recibidos por línea de comandos. Estas características están alineadas con las prácticas comunes de evasión, que refuerzan la hipótesis de que petya.exe es una muestra maliciosa avanzada, diseñada para ejecutarse solo en condiciones específicas y maximizar su efectividad una vez desplegado en el sistema operativo víctima.

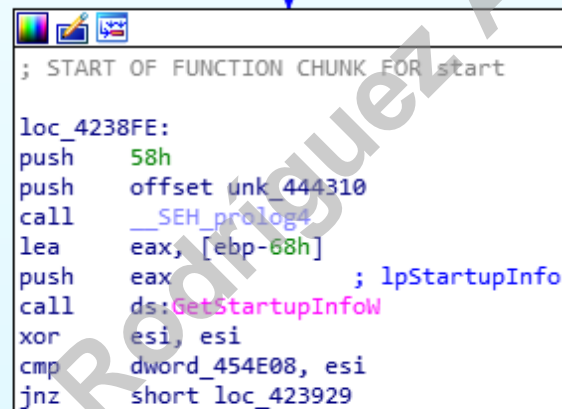
Esquema completo de IDA Pro de Petya.exe



```
public start
start proc near

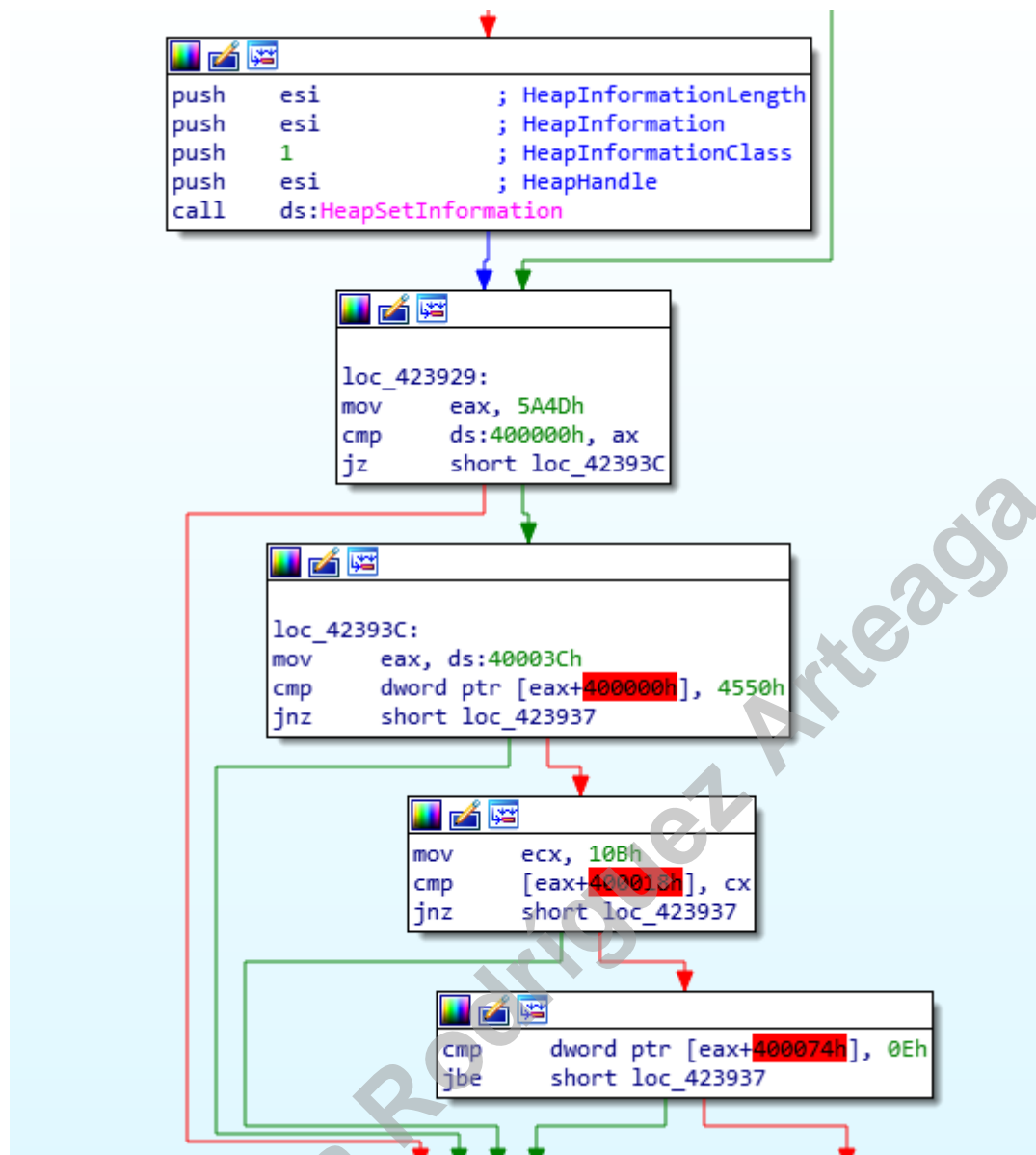
; FUNCTION CHUNK AT 004238FE SIZE 00000084 BYTES
; FUNCTION CHUNK AT 00423983 SIZE 00000010 BYTES
; FUNCTION CHUNK AT 00423994 SIZE 00000018 BYTES
; FUNCTION CHUNK AT 004239AD SIZE 00000025 BYTES
; FUNCTION CHUNK AT 004239D3 SIZE 00000010 BYTES
; FUNCTION CHUNK AT 004239E4 SIZE 00000012 BYTES
; FUNCTION CHUNK AT 004239F7 SIZE 00000036 BYTES
; FUNCTION CHUNK AT 00423A5B SIZE 00000010 BYTES

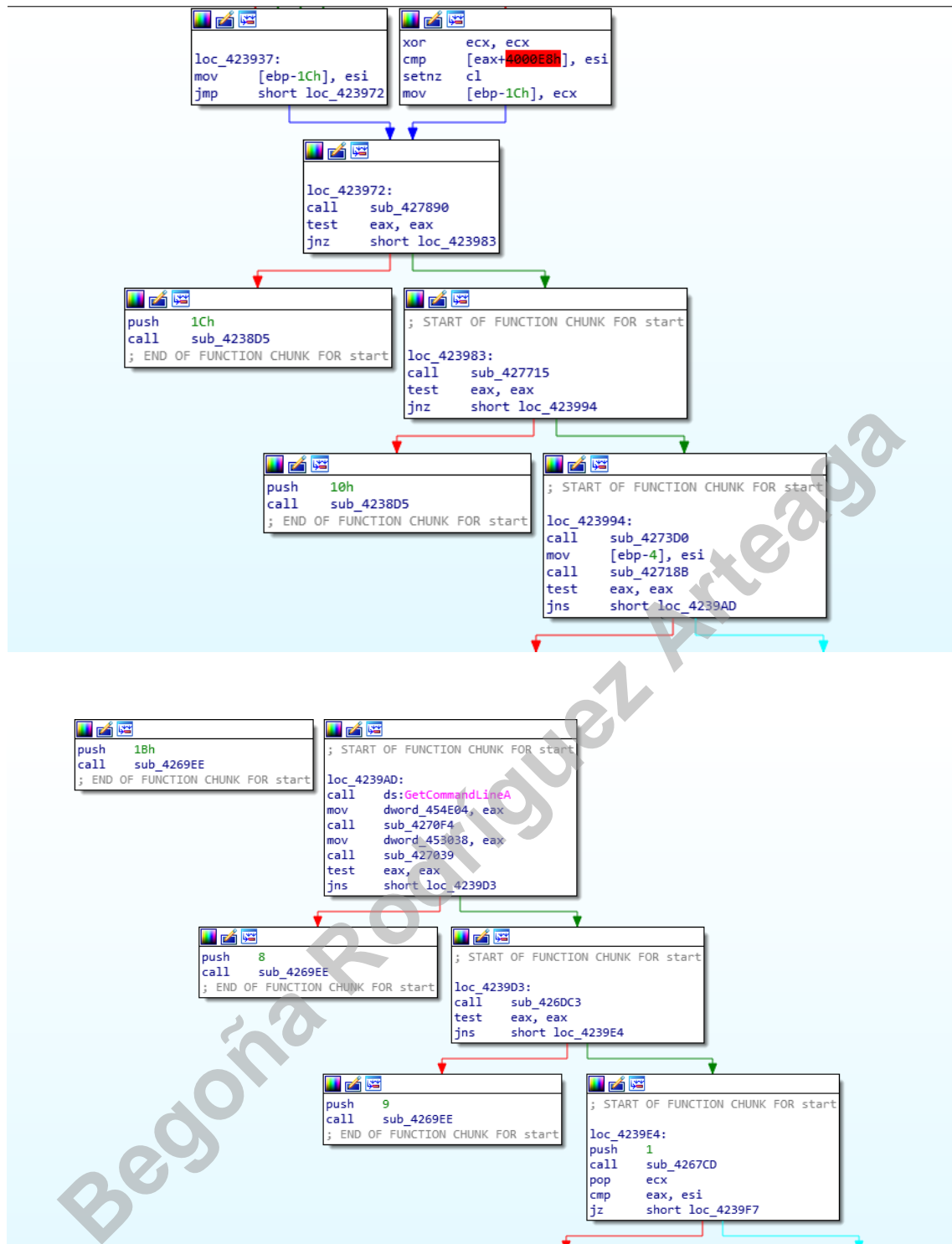
call     sub_427909
jmp      loc_4238FE
start endp ; sp-analysis failed
```

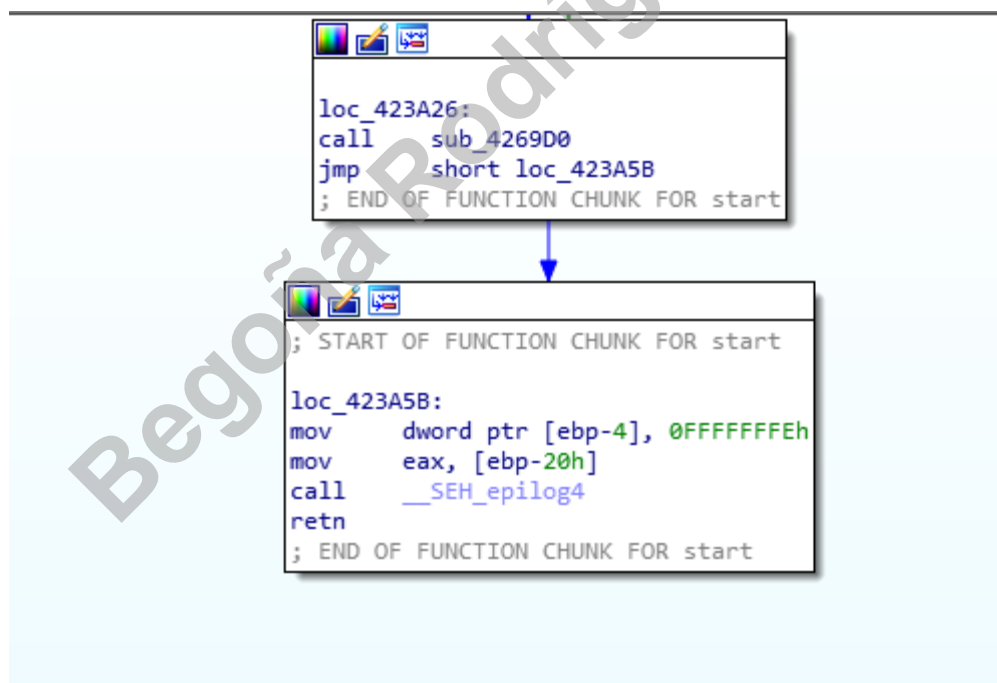
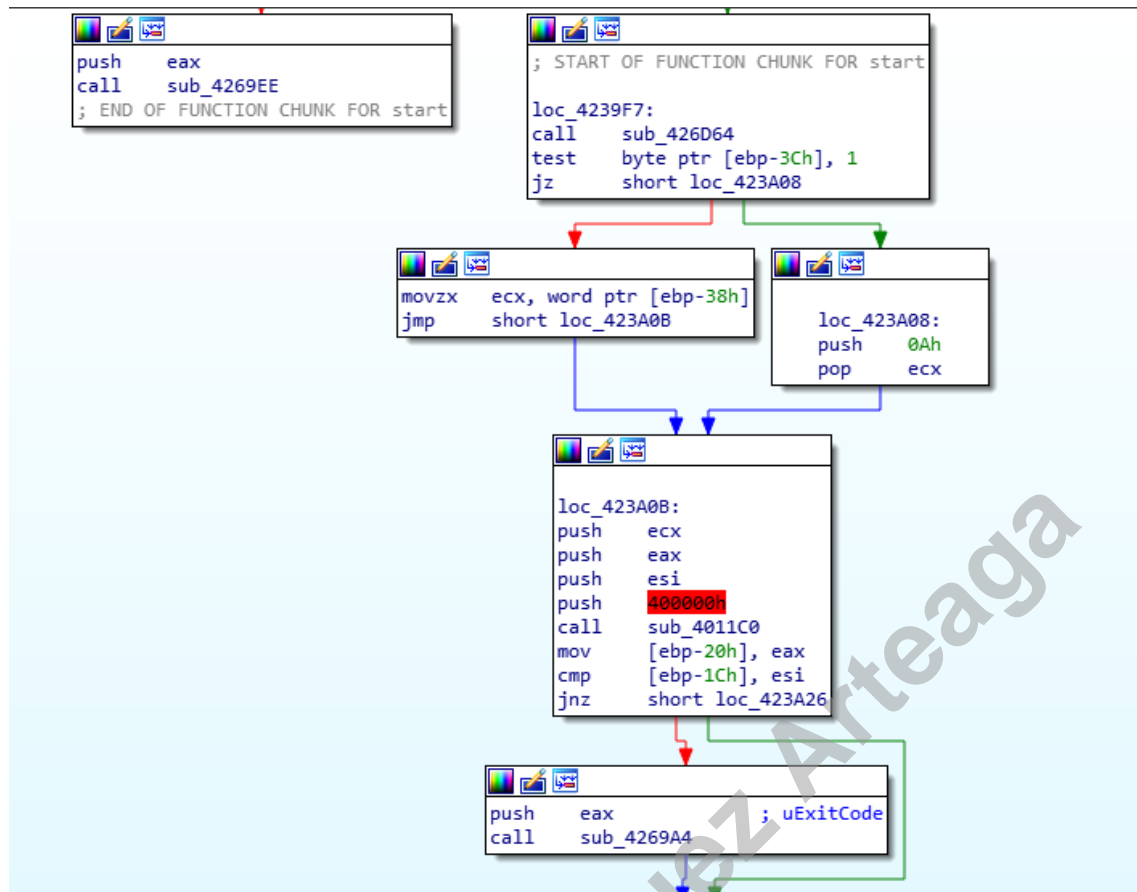


```
; START OF FUNCTION CHUNK FOR start

loc_4238FE:
push     58h
push     offset unk_444310
call     __SEH_prolog4
lea      eax, [ebp-68h]
push     eax ; lpStartupInfo
call     ds:GetStartupInfoW
xor      esi, esi
cmp      dword_454E08, esi
jnz      short loc_423929
```







Conclusiones

El análisis combinado estático y dinámico de la muestra petya.exe ha revelado una serie de características asociadas al comportamiento típico de malware tipo ransomware. A nivel estático, hemos identificado múltiples indicadores de riesgo como elevadas puntuaciones en herramientas antivirus (58/72 en VirusTotal), el uso de privilegios administrativos, la presencia de cadenas sospechosas (como rutas hacia librerías criptográficas y direcciones .onion) y un uso intensivo de funciones críticas del sistema operativo, tales como CreateProcess, HeapSetInformation, y GetCommandLine.

Durante el análisis dinámico, hemos observado la ejecución encadenada de procesos legítimos del sistema (runas.exe, conhost.exe, WinRAR.exe, rundll32.exe) que culminan en la activación del binario malicioso. Este flujo responde a técnicas de evasión como Living off the Land, donde se ejecuta un archivo malicioso escondido tras procesos legítimos del sistema. También, se han detectaron accesos y modificaciones al registro de Windows, lectura de claves del sistema y múltiples conexiones de red utilizadas para validación de certificados o sincronización del sistema.

El análisis en IDA Pro confirmó la presencia de lógica condicional, verificación del entorno, estructuras de control orientadas a evitar ejecución en entornos de análisis, y la posible ejecución de un payload dependiente de ciertos parámetros, lo cual refuerza la naturaleza avanzada y evasiva del código.

Medidas de detección y prevención

Para mitigar y prevenir infecciones como las causadas por Petya, se recomiendan las siguientes medidas, estructuradas en torno a los principios de defensa en profundidad y a las técnicas vistas en el curso como el fortalecimiento de la detección mediante la implementación de soluciones EDR (Endpoint Detection and Response) que permitan detectar comportamientos anómalos en tiempo real, como el uso de runas.exe, rundll32.exe o accesos masivos a claves del registro; además de establecer alertas basadas en técnicas de MITRE ATT&CK como T1059 (Command and Scripting Interpreter), T1547 (Boot or Logon Autostart Execution) y T1486 (Data Encrypted for Impact).

Se requiere una monitorización avanzada del sistema, con la activación y revisión periódica de los logs de Sysmon, especialmente los eventos 1 (ejecución de

procesos), 10 (acceso a claves del registro) y 3 (conexiones de red), con el objetivo de detectar patrones que coincidan con actividades maliciosas sospechosas.

Hay que añadir a estas herramientas de detección, la incorporación de herramientas sandbox de análisis automatizado en entornos cerrados como Any.run para poder realizar un estudio del comportamiento de ejecutables sospechosos antes de permitir su ejecución en los sistemas productivos.

Es necesario el establecimiento de políticas de ejecución que impidan el uso de compresores externos (WinRAR, 7zip) desde rutas no autorizadas y la inhabilitación del archivo ejecutable rundll32.exe salvo en contextos controlados. Esto puede realizarse a través de Windows Defender.

Primordial para evitar la infección maliciosa es la concienciación y sensibilización de los usuarios sobre los riesgos de la ejecución de archivos adjuntos o la realización de descargas de fuentes desconocidas, incluyendo aquellos enmascarados como herramientas legítimas.

Por último, es esencial la creación de planes de contingencia ante ataques maliciosos con el mantenimiento de copias de seguridad cifradas y offline, el desarrollo de procedimientos de respuesta ante incidentes, además del cumplimiento de las prácticas definidas por guías y marcos de estándares de riesgos y de gestión cibernética como la NIST CSF o la ISO 27001.

Bibliografía

[https://es.wikipedia.org/wiki/Petya_\(malware\)](https://es.wikipedia.org/wiki/Petya_(malware))

<https://www.proofpoint.com/es/threat-reference/petya>

<https://www.avast.com/es-es/c-petya>

<https://www.tibagroup.com/es/comercio-internacional/mercado/ciberataque-petya>

J. S. Aidan, H. K. Verma and L. K. Awasthi, "Comprehensive Survey on Petya Ransomware Attack," 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS), Jammu, India, 2017, pp. 122-125, doi: 10.1109/ICNGCIS.2017.30

Alert (TA17-181A) Petya Ransomware, US-CERT (2017). [Online]. <https://www.us-cert.gov/ncas/alerts/TA17-181A>.

O. Solon, A. Hern, Petya' ransomware attack: what is it and how can it be stopped?, The Guardian (2017)

<https://www.theguardian.com/technology/2017/jun/27/petya-ransomware-cyber-attack-who-what-why-how>. Accessed 7 Nov 2017

P. Bedwell, A deep dive into the NotPetya ransomware attack, Lastline (2017)

<https://www.lastline.com/blog/notpetya-ransomware-attack/>.

L. Abrams, Petya Ransomware skips the Files and Encrypts your Hard Drive Instead, BleepingComputer (2016).

<https://www.bleepingcomputer.com/news/security/petya-ransomware-skips-the-files-and-encrypts-your-hard-drive-instead/>.

A. Kharpal, 'Petya' ransomware: All you need to know about the cyberattack and how to tell if you're at risk, CNBC (2017). <https://www.cnbc.com/2017/06/28/petya-ransomware-cyberattack-explained-how-to-tell-if-youre-at-risk-or-been-attacked.html>.

T. Fox-Brewster, 3 Things You Can Do To Stop 'NotPetya' Ransomware Wrecking Your PC, Forbes (2017).

<https://www.forbes.com/sites/thomasbrewster/2017/06/28/three-things-you-can-do-to-stop-notpetya-ransomware-wrecking-your-pc/#6f276e377b05>.

I. Thomson in San Francisco 2017 at 03:19 tweet_btn(), Everything you need to know about the Petya, er, NotPetya nasty trashing PCs worldwide, The Register®—Biting the hand that feeds IT (2017).

https://www.theregister.co.uk/2017/06/28/petya_notpetya_ransomware/.

Symantec Security Response, Petya ransomware outbreak: Here's what you need to know, Symantec (2017). <https://www.symantec.com/connect/blogs/petya-ransomware-outbreak-here-s-what-you-need-know>.

S. Eschweiler, Decrypting NotPetya/Petya: Tools for recovering your MFT after an attack, CrowdStrike (2017). <https://www.crowdstrike.com/blog/decrypting-notpetya-tools-for-recovering-your-mft-after-an-attack/>.

J. Splinters, NotPetya ransomware virus. How to remove? (Uninstall guide), 2-spyware (2017). <https://www.2-spyware.com/remove-notpetya-ransomware-virus.html#data-recovery!>

Patrik, Petya.A/NotPetya virus removal——How to protect computer, My AntiSpyware (2017). <http://www.myantispware.com/2017/06/28/petya-notpetya-virus/>.

CASPAR, Guide to remove NotPetya ransomware permanently, Viruses Removal Pro (2017). <http://provirusesremoval.com/guide-remove-notpetya-ransomware-permanently/>.

P. Paganini, Ransomware: How to recover your encrypted files, the last guide, Security Affairs (2016). <http://securityaffairs.co/wordpress/53438/malware/ransomware-recover-guide.html>.