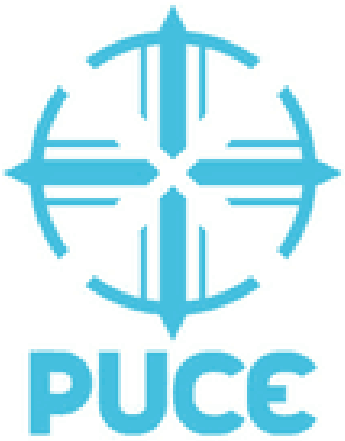
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS



TÍTULO DE LA DISERTACIÓN

ANÁLISIS COMPARATIVO DE HERRAMIENTAS ANALÍTICAS DE MALWARE.
CRITERIO: CAPACIDAD DE OBTENCIÓN DE INDICADORES DE COMPROMISO-IOC

Nombre:

Alberto Mucarsel Sánchez

Quito, 2022

Tabla de contenido

Índice de ilustraciones

Índice de tablas

Resumen:

Los Indicadores de Compromiso o IoC por sus siglas en inglés, son todo tipo de artefactos o patrones que permiten la identificación de cualquier tipo de actividad maliciosa realizada por parte de malware. Debido a esto se han vuelto relevantes dentro del mundo de la ciberseguridad, especialmente a que estos artefactos pueden llegar a ser procesados en los que se conoce como Inteligencia de Ciber Amenazas o CTI por sus siglas en inglés, la cual posibilita la planificación de estrategias en pro de aumentar las capacidades de los equipos de seguridad en realizar detecciones tempranas de una posible infección o ataque.

Dicho esto, el proceso para la obtención de dichos artefactos es algo complicado, además de que gracias a las múltiples técnicas de camuflaje que las muestras modernas de malware pueden llegar a emplear, este proceso se dificulta aún más al no poder detectar un artefacto que logre identificar a la muestra, o que no sea confundida con algún archivo de uso común, y termine provocando lo que se conoce como un falso positivo. Debido a estas razones, los profesionales del campo han optado por el uso de diversas herramientas para realizar la obtención de estos indicadores.

En este trabajo, se analizarán 5 herramientas implementadas dentro del campo de la informática forense, en específico dentro del proceso de análisis de malware. Dichas herramientas serán comparadas según su capacidad para el análisis, identificación, y obtención de los Indicadores de Compromiso, lo cual se llevará a cabo poniendo a prueba cada una de las herramientas al cargar 5 muestras de malware a cada una de estas, simulando un proceso de análisis de malware el cual será tanto de modo estático como dinámico.

El objetivo de este trabajo es obtener la mejor herramienta de tipo open-source que sobresalga en esta comparativa de acuerdo a su capacidad para obtener IoC de una muestra de malware determinada, además de comprender la importancia de dichos artefactos, y los riesgos que representa realizar dichas tareas en ambiente no asegurado.

Dedicatoria:

"Dedico este trabajo a todas aquellas personas que durante todo este viaje de crecimiento y aprendizaje me han enseñado tanto sobre mi carrera y más que nada sobre la vida, ya que sin ellas no sería la persona y el profesional que soy en la actualidad.

A mis padres Alberto y Wilma quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más, gracias por inculcar en mí el ejemplo de esfuerzo y valentía, de no temer las adversidades, y siempre darme ánimos para levantarme y dar lo mejor de mí mismo.

A mi hermano Daniel por su cariño, apoyo incondicional, y especialmente por la infinita paciencia que me tuvo durante mis largas noches de elaboración de trabajos y mis mal genios por el estrés. Por estar conmigo en todo momento, gracias.

A quien en vida fue mi gran amigo, Sebastián, quien siempre me enseñó que la vida no es simplemente como nos la pintan, que hay muchas formas de verla y de aprender sobre ella, y de quien aprendí que siempre podemos hacer de este mundo un mejor lugar con simples y pequeños actos.

Finalmente quiero dedicar esta tesis a todos mis amigos, especialmente a mi hermano del alma, Alfredo, por apoyarme cuando más los necesito, por extender su mano en momentos difíciles y por el amor brindado cada día, de verdad mil gracias, siempre las llevo en mi corazón."

Agradecimiento:

“Mi profundo agradecimiento a la Pontificia Universidad Católica del Ecuador, a toda la Facultad de Ingeniería, a mis profesores, en especial a la Ing. Suyana Arcos, Ing. Javier Cóndor, y Mgtr. Charles Escobar quienes con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional, gracias a cada uno de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

Finalmente quiero expresar mi más grande y sincero agradecimiento al Ing. Edison Mora, quien, con su dirección, conocimiento, y enseñanza permitió el desarrollo de este trabajo.”

Introducción:

Tema

ANÁLISIS COMPARATIVO DE HERRAMIENTAS ANALÍTICAS DE MALWARE[1]. CRITERIO: CAPACIDAD DE OBTENCIÓN DE IOC[2].

Justificación

En la situación que se vive, la capacidad de adaptar nuestro modo de vida a los medios electrónicos que nos permiten estar conectados con nuestra comunidad se ha vuelto crucial para poder salir adelante, y ahora nuestra presencia digital se ha vuelto un recurso valioso, el cual puede convertirse en el objetivo de ciber criminales que buscan adueñarse de nuestra información mediante el uso de diferentes tipos de malware.

Con este antecedente expuesto, se vuelve necesario que los métodos de prevención y estudio de software malicioso se encuentren en constante evolución y que el público en general esté informado sobre cómo estas amenazas pueden afectarlos; según el artículo "*Malware Analysis: An Introduction"* (Distler, 2020) se menciona que la meta de los procesos de análisis de malware es la de ganar conocimiento de cómo funciona una muestra específica de malware, con la finalidad de que, a partir de este entendimiento, poder crear defensas para así proteger la red de una organización. Con lo anteriormente dicho, se pretende realizar el presente trabajo de titulación, el cual se enfocará en realizar un análisis comparativo de las herramientas analíticas de malware con un criterio de análisis sobre su capacidad de obtención de los IOC, mediante un estudio teórico de metodologías de análisis de malware, el establecimiento de entornos seguros de prueba mediante el uso de software de virtualización y laboratorios de prueba de cada herramienta con diferentes muestras que se lleguen a obtener.

Esto permitirá conocer las herramientas en el campo de análisis de malware que aporten a la obtención de IOC. Los resultados de esto derivarán en una conclusión sobre la relevancia de la selección de obtención del IOC como criterio de análisis en este tipo de herramientas. Adicional a esto, mediante los análisis de muestras en entornos seguros, se buscará mostrar al público en general los alcances que estas pueden tener y

[1] Malware: contracción de "Malicious Software". Es cualquier software diseñado intencionalmente para causar daño a un computador, servidor, cliente, o red (Robert Moir, 2009).

[2] IOC: Siglas de "Indicators of Compromise". Artefactos con datos forenses que indican la intrusión en un equipo (Gragido, 2013)

los riesgos que se corre si se llegase a tener un equipo infectado con estos programas maliciosos.

Planteamiento del problema

En la actualidad, la cantidad de malware que se encuentra en el mundo del internet que pretenden infectar a equipos tecnológicos es bastante amplia; muchos proveedores de antivirus tienen en sus bases de datos las diferentes firmas para que sus productos puedan realizar la comparación necesaria, detectar y detener de la manera más rápida el esparcimiento de este software malicioso antes de que pueda llegar a causar daños graves. Los costos que representa la adquisición de dichos productos no están al alcance del público en general, que se inician dentro de esta rama de ciberseguridad.

Dentro de este contexto, dichos especialistas deciden recurrir al uso de varias herramientas para llevar a cabo su labor, pero el hecho de seleccionar aquella que le ayude a alcanzar la mayor eficiencia en su trabajo representa cierta dificultad. Añadido a esto, la obtención de los IOC representa una dificultad variable (Lord, 2017), dependiendo de la complejidad del malware.

En función de esta problemática se plantea la siguiente pregunta principal a resolver en el trabajo de titulación:

- ¿Cuál es la mejor herramienta de análisis de malware, para la obtención del IOC?

Y las siguientes preguntas secundarias:

- ¿Cuáles son las capacidades y eficiencia de las herramientas que se puedan emplear durante un proceso de análisis?
- ¿Qué relevancia representa la obtención de IOC al seleccionar la mejor herramienta de análisis de malware?
- ¿Qué daños pueden ocasionar, si llega a ser infectado un equipo tecnológico por uno de los malware que están bajo prueba?

Objetivos

Objetivo general

Analizar y determinar la mejor herramienta de análisis de malware según la obtención de IOC.

Objetivos específicos

- Comparar las funcionalidades y eficiencia de las 5 herramientas de análisis de malware seleccionadas.
- Identificar los riesgos que se pueden presentar al tener un equipo infectado con las muestras de malware que se han obtenido durante el presente proyecto de titulación.
- Mostrar la relevancia de la selección del criterio IOC en el proceso comparativo de herramientas de análisis de malware.

Capítulo 1: Marco Teórico

Seguridad Informática

En la actualidad se puede evidenciar la revolución tecnológica que a la que se ha llegado con todos los avances que la humanidad ha logrado a lo largo de las últimas décadas. Es una verdad que hoy en día, el internet se ha vuelto parte de los servicios básicos en cada hogar, los planes móviles enfocados a paquetes de datos para navegación han tomado una mayor relevancia en las compañías telefónicas y como principal producto de consumo por parte del mercado, se puede trabajar desde casa con el uso de una VPN para acceder a los sistemas de la oficina sin mucha dificultad, etc.

En un contexto económico, lo que ha potenciado exponencialmente la adopción de la tecnología ha sido el fenómeno de la globalización, el cual ha demandado que todo tipo de industrias busque implementar en sus modelos de negocios una gama variada de soluciones tecnológicas que puedan adaptarse a la nueva realidad comercial en la que vivimos. Con este paso acelerado en la adopción de nuevas tecnologías, la naturaleza en la que la información y comunicaciones planteaban su infraestructura se ha visto modificada de forma significativa. La cantidad y variedad de equipos que ahora se intercomunican y solicitan acceso a las diferentes infraestructuras existentes ha crecido, así como la cantidad de usuarios conectados al internet.

Debido a este aumento, la información transmitida por estos dispositivos a través de los medios disponibles ha sido modificada y ha incrementado en un porcentaje considerablemente alto. Con tal cantidad de información circulando por el internet, aquellas organizaciones encargadas de proveer la infraestructura para establecer comunicaciones deben poner en consideración la seguridad de dicha información, y por lo tanto implementar políticas de lo que se conoce como *seguridad informática.*

Previo a profundizar en esta disciplina y la importancia que esta trae dentro de este trabajo, se revisarán los varios términos que son empleados por profesionales en el campo, acto seguido, se verá su definición, y se nombrarán sus puntos más importantes dentro del contexto de este trabajo.

La *seguridad informática, ciberseguridad* o *seguridad de la Tecnología de la Información (seguridad TI)* son solo algunos de los varios términos utilizados por varios profesionales que trabajan en el área. A pesar de que estos términos tienen

diferencias matizadas entre ellos, fueron lo suficientemente tangibles para tener un significado unificado para una población más amplia (Schatz et al., 2017).

Para establecer los matices de cada uno, primero se hablará sobre lo que considerado como *ciberseguridad*, ya que este término es usualmente señalado como aquel que engloba a todas las demás, o en otras palabras, es aquel que define de manera más general todo lo relacionado con prácticas de seguridad relacionadas a la Informática.

Según el portal del gigante de la seguridad informática, *Kaspersky* (2019) se define a la ciberseguridad de la siguiente manera:

> "La Ciberseguridad es la práctica de la defensa de computadoras, servidores, dispositivos móviles, sistemas electrónicos, redes, y datos ante ataques maliciosos"

Dentro de esta mención, la corporación rusa menciona que a la ciberseguridad se la conoce también como seguridad TI o *seguridad de la información electrónica*, y que estos términos son aplicables en una amplia variedad de contextos.

De acuerdo con la publicación de *Figeroa-Suárez et al*. (2018) donde se presentan varías definiciones, con la finalidad de comparar los términos de *seguridad informática* y *seguridad de la información* para mostrar lo que diferencia el uno del otro, llega a la siguiente conclusión en cuanto a las diferencias entre estos dos conceptos:

- La *Seguridad Informática* es aquella que se encarga de la parte de la infraestructura, es decir, se hace cargo de la protección de las instalaciones informáticas y de la información que se encuentre alojada en medios digitales.
- La *Seguridad de la Información* va un paso más allá y esta busca la protección de la información, independientemente del medio en el que esta se encuentre.

Una vez se ha aclarado esta diferencia entre ambos términos, se puede apreciar que realmente no son términos usados a manera de sinónimos, sino que cada uno representa una idea diferente, sin embargo, ambas disciplinas se encuentran bastante ligadas entre sí.

A partir de aquí, para facilidad de lector y el desarrollo de este trabajo, se usará el término *seguridad informática* como la disciplina dentro de la que esta disertación se encuentra, y se englobará todo lo relacionado a *seguridad de la información* y *ciberseguridad* dentro del mismo.

Vulnerabilidades y ataques en la Seguridad Informática

Una vulnerabilidad es considerada como una falla o debilidad, ya sea que esta se encuentre en el diseño, la implementación, la lógica u operación, o en el control interno de cualquier tipo de sistema. Según el INCIBE[3] (2017) , dentro del contexto de la informática se define una vulnerabilidad de la siguiente manera:

> "Es una debilidad o fallo en un sistema de información que pone en riesgo la seguridad de la información pudiendo permitir que un atacante pueda comprometer la integridad, disponibilidad o confidencialidad de la misma, por lo que es necesario encontrarlas y eliminarlas lo antes posible…"

Pero una vulnerabilidad por sí sola no es la que produce daños a una infraestructura tecnológica, ya que son solamente condiciones y características únicas de cada sistema y son las mismas las que los vuelven susceptibles a amenazas (INCIBE, 2017).

Para comprender la magnitud de los riesgos que se corren a diario en el campo de TI[4] es necesario dar de igual manera una definición del término *amenaza* dentro del contexto informático.

Dentro del glosario de la CSRC[5], proporcionado por la institución gubernamental estadounidense NIST[6], podemos encontrar la siguiente definición de *amenaza:*

> "Cualquier circunstancia o evento con el potencial de impactar de manera adversa operaciones organizacionales (incluyendo la misión, funciones, imagen, o reputación), activos de la organización, o a individuos a través de sistema de información vía acceso no autorizado, destrucción, desclasificación, modificación de información, y/o denegación de servicios. También, el potencial de una fuente de amenaza de exitosamente explotar una particular vulnerabilidad de un sistema de información"

Debido a que existe una amplia variedad de fuentes de amenazas, varios corporativos informáticos y entidades reguladoras, se han dedicado a la clasificación y/u observación de dichas amenazas, y de distintas formas, a lo largo de los años desde el

[3] INCIBE: Siglas de Instituto Nacional de Ciberseguridad de España.
[4] TI: Siglas de Tecnologías de la Información.
[5] CSRC: Siglas en inglés de Computer Security Resource Center.
[6] NIST: Siglas en inglés de National Institute of Standards and Technology.

inicio de sus labores, han presentado las amenazas o ataques más comunes, de manera anual, ya que nos encontramos en un entorno que evoluciona constantemente.

Según un artículo publicado por la compañía multinacional de TI, Hewlett-Packard (2019), se listan cuatro fuentes de amenazas, o para definirlas de mejor manera, ciber ataques: entre los cuales podemos encontrar:

- Malware
- Phishing
- Man In The Middle
- Denegación de servicios

Para dar a entender la severidad de cada uno de estos tipos de ataque o fuentes de amenazas, se los definirá y serán profundizados mediante estadísticas, utilizando reportes de los principales proveedores de productos de protección de TI, la información que se presentará a continuación es mayormente tomada de los reportes referentes al año 2020.

- *Malware*

  El término *malware* es la contracción de las palabras *malicious software*, y es un amplio término que ampara a todo tipo de programas de computadora dañinos que busquen obtener acceso no autorizado a un sistema informático determinado y/o causar daños al mismo.

  Durante el año 2020, la población global se vio forzada a migrar sus medios de trabajo a medios digitales y de forma remota, por lo que el uso de computadores personales, los cuales son potenciales objetivos de criminales informáticos, se elevó proporcionalmente. Debido a este incremento, los malwares que se detectaron durante el último cuarto del año 2020 nos dan las siguientes cifras:

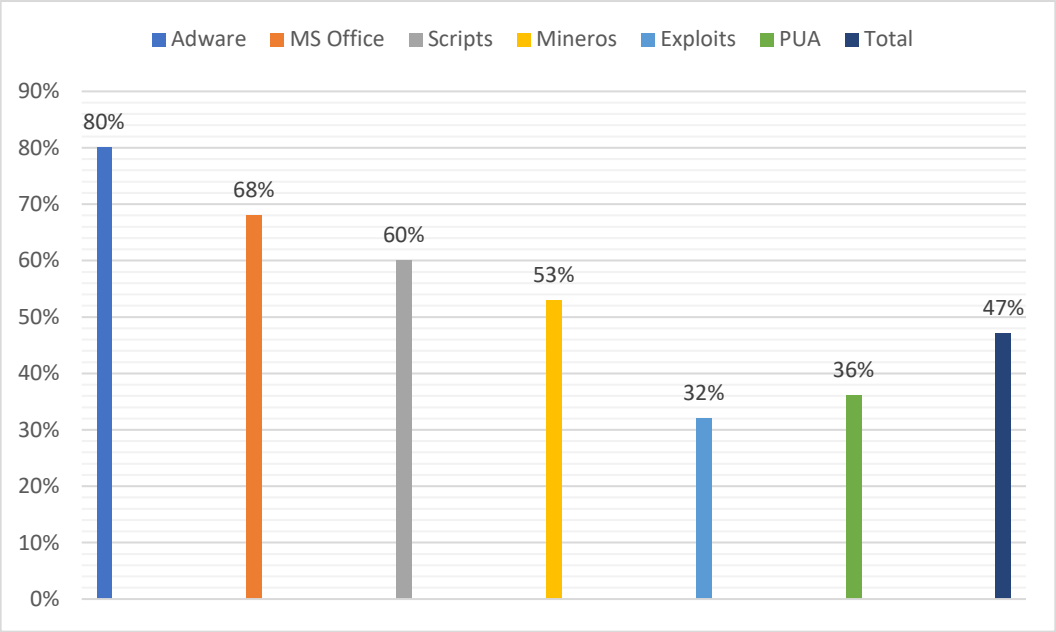


Ilustración a: Incremento de amenazas de malware durante el Q4 2020
Fuente: Avira Protection Labs (2021)

Según la estadística mostrada en la *ilustración a* se puede observar que, a manera global, se ha podido observar un 47% de incremento, eso sí, dentro de todas las amenazas que han llegado a ser detectadas por parte de la compañía Avira.

Por otro lado, el laboratorio de amenazas de McAfee presentó su reporte con un enfoque a la detección de nuevas amenazas de malware, del cual se observa un incremento en varias categorías de malware, en especial se detectó un aumento en malware para productos Microsoft Office, y de forma similar, se registró un incremento particular en volumen en cuanto a ransomware, aproximadamente un 69% entre el cuartil 3 y 4 del año 2020.

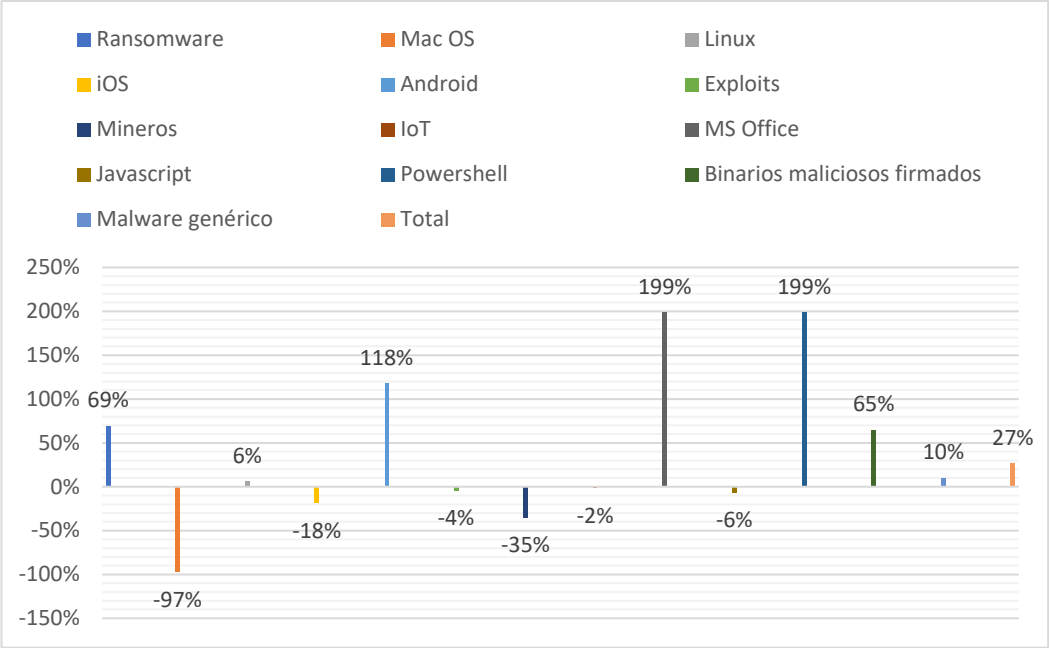


Ilustración b: Incremento de amenazas de malware durante el Q4 2020
Fuente: McAfee Labs (2021)

- *Spam y Phishing*

Según el diccionario de Cambridge, el término *spam* tiene algunos significados, den entre los cuales, dentro del contexto de este trabajo es aquel donde define a *spam* como "emails que son enviados a muchas personas, específicamente emails no deseados" (Cambridge University Press, n.d.).

El spam suele ser uno de los medios por los que se lleva a cabo frecuentemente un tipo de ataque de *ingeniería social*[7] que tiene como objetivo el robo de datos, de forma más específica credenciales de inicios de sesión y números de tarjetas de crédito. Este ataque, el cual se conoce como *Phishing*, sucede cuando el atacante, pretendiendo ser una entidad o persona de confianza de la víctima, engaña a esta para abrir un email o cualquier tipo de mensaje electrónico, y cuando se induce a la víctima a ingresar a un enlace contenido en dicho mensaje, el cual puede contener archivos maliciosos que pueden infectar la computadora y provocar desde robo de información hasta un congelamiento total del sistema (Imperva, 2019).

[7] Ingeniería social: cualquier acto que influencie a una persona a tomar una acción que puede o no ser en su mejor interés

De acuerdo con el portal SecureList, el cual es llevado por la compañía de seguridad de la información, Kaspersky, el reporte en cuando a Spam y Phishing en 2020, que fue presentado en febrero del 2021 nos entrega la siguiente información:

Proporción de Spam en el tráfico de correos electrónicos

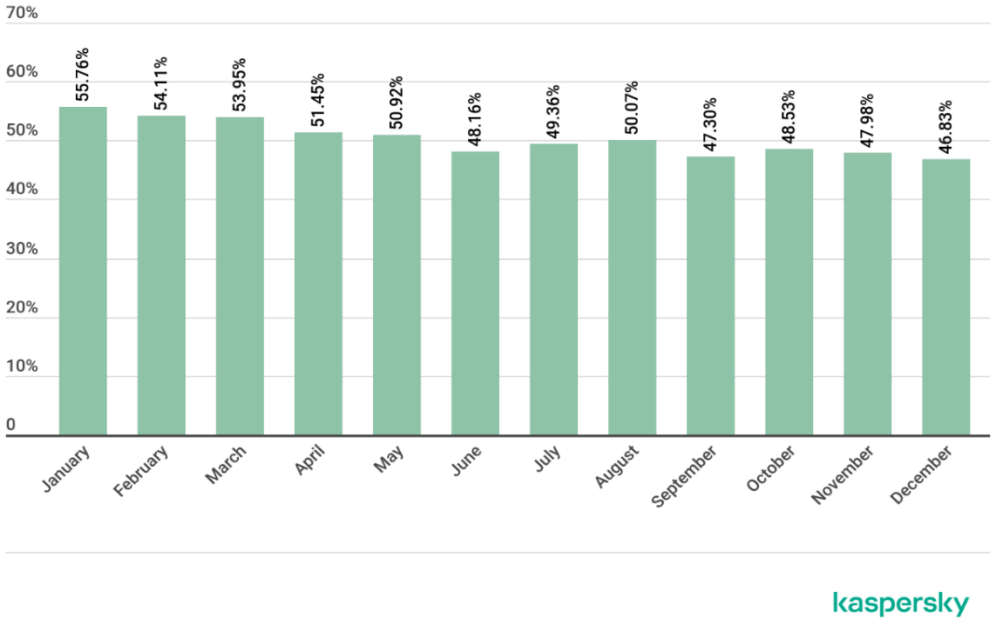


Ilustración c: Porcentajes de Spam en el tráfico de correos electrónicos a lo largo del año 2020. Fuente: SecureList by Kaspersky (Kulikova et al., 2021)

Como se puede observar en *ilustración c*, se encontró al mayor porcentaje (55.76%) de representación del spam dentro del tráfico de correos electrónicos durante el mes de enero y una disminución durante todo el año hasta llegar a su valor más bajo (46.83%) en diciembre. Todo esto se puede deber a la transición global que se tuvo de trabajo en oficial al trabajo remoto, lo que conllevó a un aumento en tráfico legítimo de correos electrónicos.

- *Man In The Middle (MITM)*

  El ataque MITM, el cual traducido en español significa “El hombre en el medio” consiste principalmente de 3 participantes, los 2 puntos involucrados en establecer una comunicación, y el atacante. Este ataque es aquel en el que el atacante se vuelve secretamente un intermediario en la comunicación entre 2 puntos, lo que le posibilita espiar e inclusive alterar la comunicación entre los

2 involucrados (ACSC, 2020), quienes piensan que se encuentran en una comunicación directa.

Debido a su misma naturaleza, su detección se vuelve complicada y, en consecuencia, la contabilización de este se vuelve una tarea sumamente compleja, mas esto no significa que este tipo de ataques no estén presentes dentro del ciberespacio.

Según el portal del proveedor de antivirus Norton (Chivers, 2020) Se lista 7 formas en las que se llevan a cabo los ataques MITM:

1. IP spoofing
2. DNS spoofing
3. HTTPS spoofing
4. SSL hijacking
5. Email hijacking
6. Wi-Fi eavesdropping
7. Robo de cookies del navegador

- *Denegación de Servicios*

  Los ataques DoS[8] y DDoS[9] pueden considerarse como los más conocidos entre la comunidad informática y el público en general. Para entrar en el contexto, se define como un ataque DoS al ciberataque en el cual el atacante busca que una máquina, servicio, o recurso de red determinado deje de estar disponible para los usuarios que se supone van a utilizarlos, ya sea de forma temporal o de manera indefinida al interrumpir dichos servicios(Cyber and Infrastructure Security Agency, 2018). La manera más conocida en la que se ejecuta este tipo de ataques es enviando numerosas solicitudes al objetivo con la finalidad de sobrecargarlo y así prevenir que pueda atender solicitudes legítimas. El ataque DDoS es un paso más allá, ya que ahora el tráfico vano y numeroso que busca hacer que un servicio caiga o se interrumpa, viene desde diferentes fuentes.

  Previamente se mencionó que este tipo de ataque es bastante conocido por el público en general, esto se debe a que la consecuencia directa de estos ataques, como fue explicado, es que un servicio deje de estar disponible en un momento en el que está planificado que siga funcionando con normalidad. A lo largo de

[8] DoS: Siglas en inglés de “Denial of Services”
[9] DDoS: Siglas en inglés de “Distributed Denial of Services”

los años se han podido registrar varios sitios web, videojuegos multijugador, entidades bancarias, y muchos otros tipos de servicios en la web que sufrieron una "caída repentina" y se mostraron fuera de servicio. Un ejemplo reciente de esto al momento de la elaboración de este trabajo es el caso del popular videojuego multijugador PUBG MOBILE, el cual fue reportado por su equipo una vez que se reportaron problemas por parte de los jugadores al no poder acceder a partidas en línea.

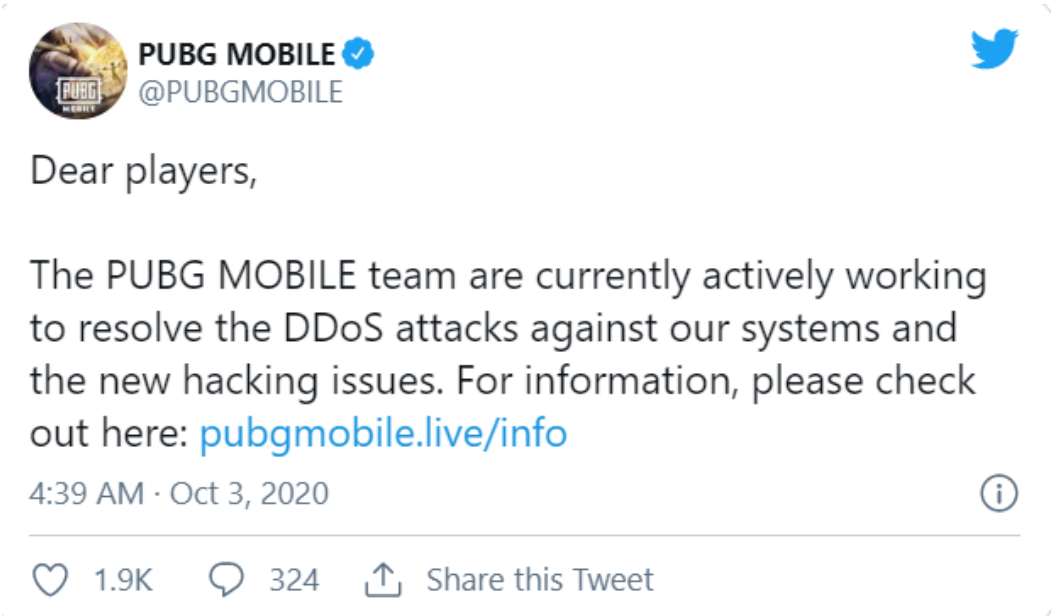


Ilustración d: Reporte de ataque DDoS en la Red Social Twitter por parte de PUBG Mobile team
Fuente: SecureList by Kaspersky (Kupreev et al., 2021)

Según el reporte del último cuarto del 2020 elaborado por personal de Kaspersky se tiene que, a comparación del año 2019, en el tercer cuarto del 2020 el incremento en cuanto al total de ataques DDoS fue del 61.90% mientras que contrario a lo que se pronosticaba, en el último cuarto del 2020 se vio una reducción de aproximadamente un 50%. En cuanto a los nuevos ataques DDoS inteligentes, se puede observar en la *ilustración e* que el incremento de estos tanto en el tercer como último cuarto del año 2020 no fue tanto a comparación del año 2019, tan solo hubo un 34.62% y un 19.23% en el tercer y último cuarto del año 2020, respectivamente.

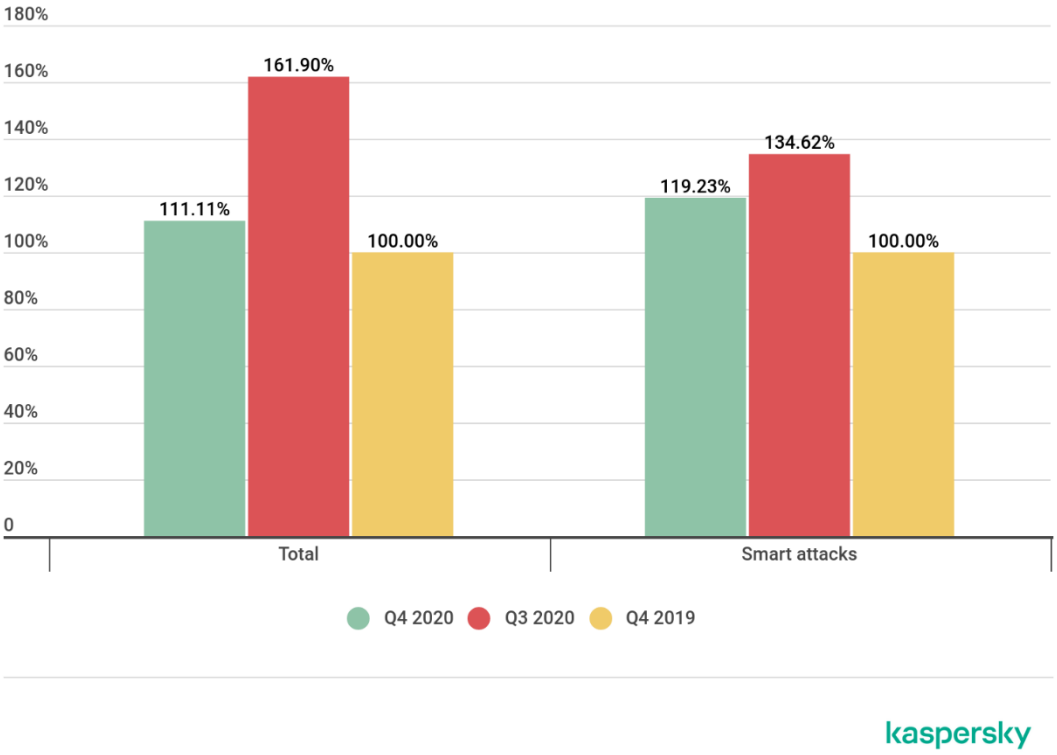


Ilustración e: Comparación de cantidad de ataque DDoS entre el Q3/Q4 2020 y Q4 2019, la data de Q4 2019 se toma como el 100%
Fuente: SecureList by Kaspersky (Kupreev et al., 2021)

Estas cifras llegan a ser alarmantes ya que dentro del mismo reporte se nos informa que en lo que ellos concluyen como los días "más tranquilos" la cantidad de estos ataques, de forma diaria llegaba a ser uno y que, en el día más activo, el cual fue el 31 de diciembre del 2020, se llegaron a registrar 1349 ataques.

Las vulnerabilidades dan lugar a que las amenazas informáticas tengan un medio por el cual acceder a un sistema de forma no autorizada, manipular el sistema para que ejecute código malicioso, inhabilite recursos del equipo afectado, etc. El medio por el cual se logra aprovechar una vulnerabilidad es conocido como *exploit*[10].

Por lo general, aquellos que buscan aprovecharse de un exploit preparan programas que contienen malware que es hecho específicamente para una vulnerabilidad dada, y es debido a esto que cada vez aparecen de forma masiva numerosos tipos y variantes de

[10] Exploit: Herramienta o ataque que toma ventaja de vulnerabilidades en aplicaciones, redes, sistemas operativos o hardware.

malware, por lo que, dentro de este trabajo, se procederá a tratar netamente sobre malware.

Malware

El término malware nace de la unión de dos palabras empleados en el término *malicious software.* Como definición se tiene que, se considera como malware a todo software diseñado, de forma intencionada, para causar daños a cualquier tipo de dispositivo electrónico o una red de estos (*Defining Malware: FAQ | Microsoft Docs*, n.d.). Esta definición es la que se ha venido adoptando en la actualidad, mas, la primera vez que se definió el término fue en una publicación del US-CERT(2005), la cual dice:

> "Programación (código, scripts, contenido activo y otro software) diseñada para interrumpir o negar operaciones, recopilar información que lleve a la pérdida de privacidad o explotación, obtener acceso no autorizado a los recursos del sistema y otros comportamientos abusivos"

Tipos de Malware

Actualmente se utiliza al término malware como una especie de sombrilla que ampara a numerosos tipos de programas maliciosos, cada uno con características distintivas, pero que, según la definición de malware, la esencia de su objetivo continúa siendo la misma. Dentro de un consenso por la comunidad, se puede clasificar bajo la sombrilla del término "malware" a un gran número de softwares maliciosos, y cada entidad se ha encargado de incluir, o excluir ciertos tipos que se definían bajo el amplio concepto de malware.

La compañía de ciberseguridad Kaspersky incluye en su clasificación (2016) a los siguientes tipos:

- Virus
- Ransomware
- Troyanos
- Spyware
- Gusanos

Otra clasificación elaborada por la empresa de ciberseguridad norteamericana, CrowdStrike, adiciona más tipos de software malicioso que puede definirse bajo la sombrilla de malware, su clasificación(Baker, 2021) es la siguiente:

- Ransomware
- *Fileless malware*
- Spyware
- Adware
- Troyanos
- Gusanos
- Rootkits
- Keyloggers
- Bots / Zombies
- Malware móvil

Para no dejar términos sin definirse, dentro de esta clasificación se presenta el *fileless malware,* el cual es un tipo de software malicioso que, a diferencia del malware tradicional, no instala nada en el dispositivo víctima, sino que realiza cambios a archivos nativos del sistema operativo, como en el caso de Windows, cambios mediante PowerShell o WMIC[11].

En el caso de *malware móvil,* viene a ser una especie de pequeña sombrilla, ya que en este tipo constan todo tipo de malware que va dirigido a afectar dispositivos móviles, y en este se pueden hallar de igual forma troyanos, ransomware, entre otros.

La clasificación elaborada por la compañía británica Sophos, dedicada a productos de seguridad tanto de software como hardware, la cual fue presentada en su sitio web *Naked Security* (Ducklin, 2019), tiene considerados a los siguientes tipos de software maliciosos como malware:

- Keyloggers
- *Data Stealers*
- *RAM Scrapers*

[11] WMIC: Siglas de "Windows Management Instrumentation Console". Interfaz de línea de comandos para WMI (Windows Management Instrumentation) la cual es una tecnología de Microsoft para la gestión de ambientes operativos.

- Bots
- Troyanos bancarios
- R.A.T.[12]
- Ransomware

Dentro del mercado de la seguridad de la información, existen una amplia gama de compañías que se dedican a la detección y clasificación de este tipo de programas, por lo que existen muchas más clasificaciones que pueden tanto coincidir, aumentar o disminuir tipos en sus listados, sin embargo, dentro de este trabajo únicamente se han tomado las que se han considerado más relevantes.

Cada uno de estos tipos de programas maliciosos tiene sus rasgos distintivos y modos de operación en cuanto se trata a infección de dispositivos informáticos. Tras una breve revisión entre los diferentes listados presentados, a partir de este punto se trabajarán con aquellos tipos de malware que he considerado como los más relevantes, los cuales son:

- Ransomware
- Troyanos
- Gusanos
- Bots
- Spyware

A continuación, se revisará la definición de cada uno y las características que nos ayudarán más adelante para comprender la dificultad en su tratamiento y el nivel de amenaza que cada uno de estos tipos de malware pueden llegar a representar.

- *Ransomware*
  El término proviene de la combinación de dos palabras, "*ransom malware"* (Malwarebytes, 2020a). Este tipo de malware es aquel que nace a partir de años de estudio en el campo de la *criptovirología*[13]. Este software

[12] R.A.T.: Siglas en inglés de "Remote Access Tools" o "Remote Administration Tools". Programas para el acceso remoto a una computadora u otro dispositivo conectado al internet o a una red local (Kaspersky, 2018).

[13] Criptovirología: Rama de la informática que se encarga del estudio del uso de la criptografía implementada en la creación de malware (Young & Yung, 1996).

malicioso emplea de forma dañina técnicas de encriptado para tener como un *rehén*, a todo el sistema de la víctima, y se debe pagar un *rescate* para poder recuperar el acceso a este.

Ilustración f: Captura de pantalla de un ataque de ransomware WannaCry en Windows 8
Fuente: SecureList by Kaspersky (GReAT [Global Research & Analysis Team], 2017)

Este tipo de malware generalmente está diseñado para esparcirse a lo largo de una red y marcar como objetivo a archivos de servidores y bases de datos, con el fin de llegar a paralizar rápidamente a una organización(Mcafee, 2020).

- *Troyanos*

  Los programas maliciosos que se encuentren ocultos o sean disfrazados como software legitimo entran en la clasificación de malware troyano (Kaspersky, 2020) . Esta definición se deriva de la famosa historia de la antigua Grecia sobre el Caballo de Troya.

  Este tipo de malware por lo general son esparcidos mediante técnicas de ingeniería social, ya que, a diferencia de los virus y gusanos, este no se auto replica ni se ejecuta automáticamente, se requiere que la víctima sea engañada y lo ejecute por sí misma. Una vez se instale la pieza de software

disfrazado, el troyano procederá a ejecutar la acción para la que fue diseñada(Johansen, 2020).

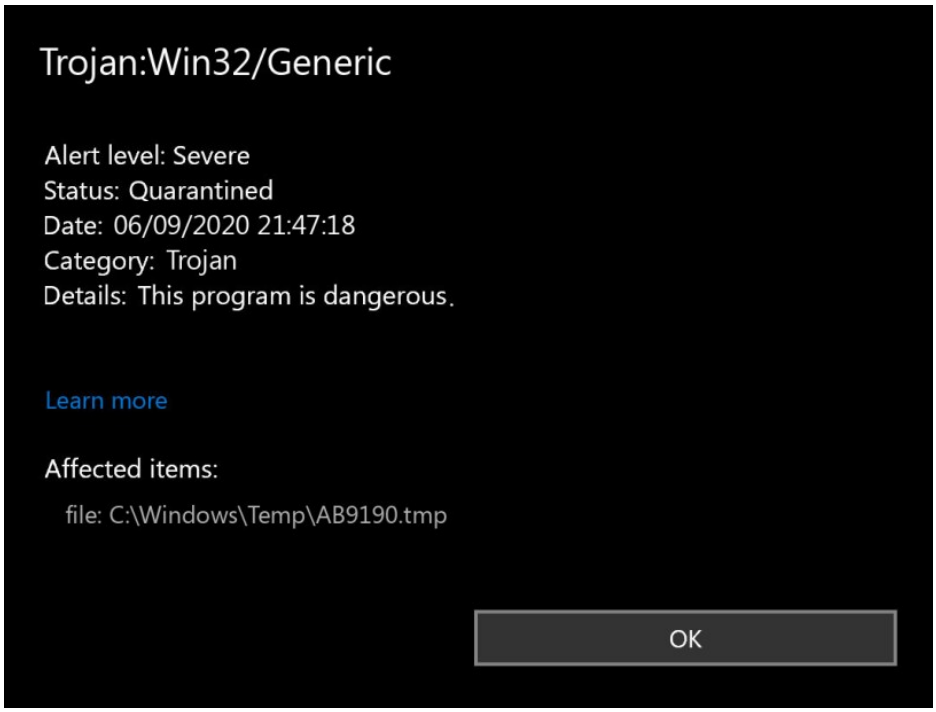


Ilustración g: Captura de pantalla de detección del malware Troyan.Win32.Generic por parte de Windows Defender
Fuente: How to Fix.guide (Woodham, 2019)

Dentro de lo que se conocen como troyanos, se genera una subclasificación de estos, esto se debe a que más que ser solo un tipo de software malicioso, por su propia naturaleza, se vuelve una técnica para la infección de sistemas, y se puede a llegar a combinar con el resto de los tipos de malware. Además de esto, también se pueden considerar diferentes clases de troyanos basadas en el tipo de acción que pueden llegar a realizar en el ordenador (Kaspersky, 2021a):

- Backdoor
- Exploit
- Rootkit
- Troyano Bancario
- Troyano DDoS
- Troyano Downloader
- Troyano Dropper
- Troyano fakeAV
- Troyano GameThief

- Troyano IM
- Troyano Ransom
- Troyano SMS
- Troyano Spy
- Troyano Mailfinder

- *Gusanos*

  Se denomina *gusano de computadora* a un tipo de malware que esparce copias de sí mismo a través de diferentes dispositivos. Este tipo de software malicioso cuenta con la capacidad de auto replicación sin la necesidad de ninguna interacción humana (NortonLifeLock, 2019).

Ilustración h: Captura de pantalla de forma de esparcimiento del gusano ILOVEYOU
Fuente: F-Secure (2021)

Los gusanos pueden llegar a ser transmitidos vía vulnerabilidades de software, mediante el uso correcto de exploits. Otra forma de infección inicial es que los gusanos lleguen dentro de un troyano en la forma de correo electrónico no deseado o mensajes instantáneos. Tal es el ejemplo del gusano *ILOVEYOU,* el cual una vez instalado en sistemas operativos

Windows, se replicaba así mismo usando el directorio de contactos de Outlook de la víctima y enviándoles una copia a todos, el correo electrónico se mostraba como si se hubiera mandado una carta de amor como archivo adjunto, tal como se puede ver en la *ilustración h.*

Una vez los gusanos son abiertos y descargados, al terminar su instalación, el gusano realiza la acción para la que fue diseñada, infectando el equipo y esparciéndose por la red, sin el conocimiento del usuario.

- *Bots*

  Un bot es una aplicación o programa o proceso que se creó para realizar tareas repetitivas, interactuar con servicios en la red, etc. Otros términos con los que se los conoce son *spiders, rastreadores* y *bots web* (NortonLifeLock, 2018). Lastimosamente esta misma característica para ayudar en la automatización de tareas, ha sido aprovechada por los cibercriminales y por lo general los bots adoptan la forma de malware.

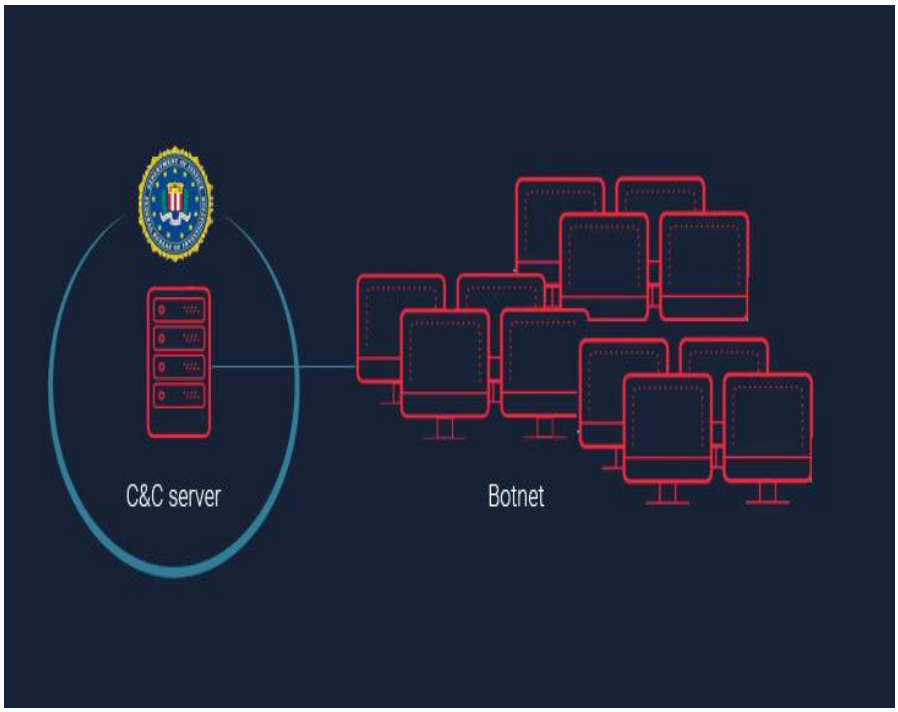


Ilustración i: Arquitectura de una botnet
Fuente: The Hacker News (Khandelwal, 2018)

Los bots maliciosos se definen como malware que tiene, al igual que un gusano, la capacidad de auto propagación, con la característica que la acción para la cual es diseñada es la de infectar a su host y este lo conecto

con un conjunto de servidores centralizados. En este caso, el servidos actúa como un *centro de control y comando* para lo que viene a ser una red de bots o como se abrevia en inglés, una *botnet*, la cual es una red de dispositivos comprometidos.

Además de establecer la conexión con el servidor C&C[14] los bots maliciosos también realizan acciones como (Sophos, 2019):

- Registrar el tecleo de cada letra
- Robar contraseñas
- Robar información financiera y personal
- Enviar campañas de phishing en forma de spam

- *Spyware*

  El spyware es igual un término que alberga dentro de sí a varios tipos de programas maliciosos. Dentro del glosario de términos de la compañía ESET (2017) se define a *spyware* de la siguiente manera:

  > "Término genérico para un rango de malware subrepticio[15] tales como keyloggers, R.A.T.s, y troyanos backdoors, especialmente aquellos que permiten la vigilancia remota de contraseñas y otros datos sensibles"

  Este tipo de malware se encarga de monitorear y recolectar de forma constante los movimientos que el usuario realiza dentro del dispositivo, sea los sitios web que visite, los archivos que descargue, información de métodos de pago, tráfico de correos electrónicos, etc. (Malwarebytes, 2020b).

[14] C&C: comando y control
[15] Subrepticio: *adj.* Que se hace o toma ocultamente y a escondidas.

Ilustración j: Comparación de modo de operación entre virus y spyware
Fuente: Avast Academy (Seguin, 2020)

El Spyware se caracteriza por no buscar llamar la atención dentro del equipo que ha logrado infectar. Este malware no busca seguir infectando una red de dispositivos, a diferencia de, por ejemplo, un virus, sino que buscar permanecer completamente oculto para realizar la vigilancia y recolección de información.

La cantidad de malware que se encuentra en el ciberespacio es incalculable, y varios expertos de ciberseguridad se han dedicado al estudio de estos programas maliciosos para que software antivirus y de detección estén mejor preparados para evitar que el software malicioso logre su cometido, con lo que dentro del campo de la seguridad de la información aparece esta subrama conocida como *Análisis de malware.*

Análisis de malware

El origen de todo se remonta al año 1966 en un artículo publicado por John von Neumann bajo el título "*Theory of self-reproducing automata*" en el cual, el referente de la Informática presenta la naturaleza por la cual en la actualidad los virus informáticos son caracterizados, la capacidad de auto replicación. Con la teoría presentada pronto nacería la necesidad de comprender dichos comportamientos.

Desde la aparición de los virus y el nacimiento del malware, en la década de los 80, con el descubrimiento del virus "*Elk Cloner"* en una computadora Mac en 1982, y la creación del primer malware para PC conocido como "*Brain.A"*(Regan, 2020), conllevaron a que se fortalezca la necesidad de poder comprender el comportamiento de estos programas, con lo que se comenzaron con las prácticas de análisis de malware.

Se entiende por análisis de malware al procedimiento para llegar a determinar cuál es la funcionalidad, origen y potencial impacto que puede llegar a tener una muestra de malware determinada, de cualquier tipo existente.

Para los autores *Rami Sihwail, Khairuddin Omar*, y *K.A.Z. Ariffin* en su trabajo titulado *"A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis"* (2018) definen que el objetivo de la práctica de análisis de malware sobre múltiples archivos maliciosos, es el de conseguir un mejor entendimiento sobre varios aspectos del malware, tal como su comportamiento, evolución sobre el tiempo y sus objetivos seleccionados.

Partiendo de ese concepto, el análisis de malware actualmente llega a ser implementado dentro de diferentes escenarios, por lo cual, a continuación, se detallarán los casos de uso más frecuentes en los cuales se decide ejecutar un análisis de malware.

Casos de uso

Los casos de uso para el análisis de malware, detallados por parte de la compañía de ciberseguridad *CrowdStrike* (Baker, 2020), son los siguientes:

- *Detección de Malware*

  En el caso de buscar la detección de amenazas en tiempo real, gracias a las más recientes implementaciones de análisis profundo de comportamiento, identificación de código compartido, funcionalidades o infraestructuras maliciosas, la tarea de su detección se ha vuelto relativamente sencilla.

  Adicionalmente en este apartado, un resultado que se obtiene del análisis de malware es la extracción de los conocidos Indicadores de Compromiso (IOC por sus siglas en inglés), los cuales son utilizados para alimentar a

programas SIEM para que estos ayuden en la detección automática de amenazas.

- *Alertas de amenazas y clasificación*

  Las soluciones de análisis de malware presentes en el mercado proveen con alerta de mejor calidad de forma temprana dentro de ciclo de vida de un ataque.
- *Respuesta a Incidentes*

  Los equipos de Respuesta a Incidentes se centran en dar un análisis de la causa principal de un ataque, determinar su impacto, y lograr con éxito las fases de remediación y recuperación. Con el análisis de malware se pueden apoyar para cumplir con estos objetivos.
- *Caza de Amenazas (Threat Hunting)*

  La ejecución de análisis de malware puede llegar a dejar al descubierto comportamiento o artefactos que los cazadores de amenazas podrían reutilizar con la finalidad de encontrar actividad similar, y de esta manera con la información recabada y reutilizada, ser capaces de detectar amenazas similares a partir de los artefactos descubiertos.
- *Investigación de malware*

  Este caso de uso es empleado más por académicos o investigadores de la industria, cuya meta es la obtención de un mayor entendimiento en cuanto a las técnicas, exploits y herramientas de vanguardia utilizadas por los creadores de malware, los cuales denominan como *adversarios.*

Etapas del análisis

Las 4 etapas del análisis de malware, presentadas por la Institución SANS[16] (Zeltser, 2021), la cual es un referente en el campo de la ciberseguridad, son las siguientes:

1. *Análisis completamente automatizado*

[16] SANS Institute: El Instituto SANS (SysAdmin Audit, Networking and Security) es una institución con ánimo de lucro fundada en 1989, cuyos objetivos son los de reunir información sobre todo lo referente a la Seguridad Informática y, ofrecer capacitación y certificación en el ámbito de la Seguridad Informática.

Se debe correr/ejecutar al archivo sospechoso dentro de un ambiente automatizado de análisis (mejor conocidos como "*sandbox"*) para obtener un informe sobre todas sus acciones.

2. *Análisis de propiedades estáticas*

   Se debe analizar la metada y todo tipo de detalles que se encuentren embebidos en el archivo, sin ejecutarlo. Esto se realiza con la finalidad de observar áreas dentro del archivo para examinar a profundidad en próximas etapas.

3. *Análisis de comportamiento interactivo*

   Se debe ejecutar al archivo en un ambiente de pruebas aislado, del cual se tenga control total. Se debe jugar con todas las combinaciones de configuraciones de dicho ambiente en una serie de experimentos iterativos con el fin de estudiar el comportamiento del sujeto de estudio.

4. *Inversión manual de código*

   Se considera a esta etapa como la más compleja. En esta fase se debe examinar el código que contiene el archivo, por lo general se lo realiza con la ayuda de herramientas desensambladoras (*disassemblers)* y depuradores (*debuggers*) de código, con la meta de aprender sobre las capacidades clave del archivo sospechoso e ir completando el reporte general con lo encontrado en fases previas.

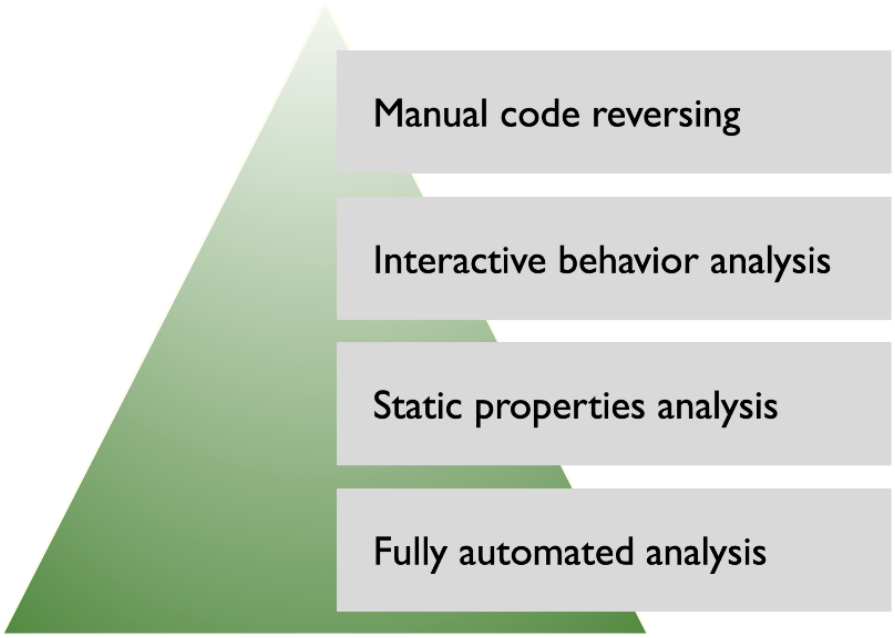


Ilustración 11: Pirámide de las etapas del análisis de malware
Fuente: SANS Institute (Zeltser, 2021)

Cada una de estas fases se vuelve más compleja que su antecesora, siendo la fase 4 la más compleja de todas tal como se puede observar en la pirámide de la *ilustración 11*, por lo que cada una de estas requieren determinados procesos y herramientas para llevarse a cabo de forma óptima. Debido a que en ciertas fases se debe tener al archivo en ejecución y en otras se contrasta esto al analizarlo sin ejecutarlo, se han generado 3 grandes tipos de análisis de malware los cuales son: estático, dinámico, e híbrido.

Tipos de análisis de malware

Como se mencionó en el punto anterior, cada una de las etapas de la práctica del análisis de malware requieren de diferentes técnicas y herramientas para ser llevadas a cabo de forma exitosa. Por la naturaleza de cada una de las etapas, surgen los tipos de análisis de malware los cuales serán explicados a continuación:

- *Análisis Estático*
  Según *Rami Sihwail et al.* (2018) esta técnica se refiere al análisis de los archivos Ejecutables Portables (PE files) sin ejecutarlos en lo absoluto. Esto se debe a que, por lo general, malware utiliza empaquetadores binarios para evitar ser analizado durante su ejecución.

  En el mismo artículo se menciona que todo este proceso se lo realiza con la finalidad de observar anomalías o patrones extraños en las siguientes características dentro del código examinado: llamadas a APIs[17], grafos de flujo de control (CFG)[18], cadenas de caracteres, Opcodes[19], N-gramas[20], tamaño del archivo, puertos, peticiones HTTP, entre otras.

  Todo lo mencionado previamente son características presentes en todo tipo de software, por lo que realizar un análisis estático ayuda a entender qué es lo que el archivo está destinado a realizar una vez se ejecute. Claro que esto puede cambiar, debido a las técnicas actuales tales como la evasión de firmas la cual ayuda a que los malwares activos no sean detectados por los motores de detección de los antivirus, y es debido a este último punto que se requiere de otra técnica la cual es el análisis dinámico.

- *Análisis Dinámico*
  En el artículo titulado "*Dynamic Malware Analysis in the Modern Era—A State of the Art Survey*" (Or-Meir et al., 2019) se menciona que el análisis dinámico es aquel que se refiere al desarrollo del análisis de un programa, código o script sospechoso, al ejecutarlo y observando cada una de sus

[17] API: Siglas en inglés de Application Programming Interface. Es un software intermediario que permite que otros 2 programas sean capaces de comunicarse entre sí.
[18] CFG: Siglas en inglés de Control Flow Graph. Un CFG es un grafo dirigido que muestra el flujo de control de un programa.
[19] Opcode: es la primera parte del lenguaje de máquina, el cual identifica qué operación debe ser ejecutada por el CPU.
[20] N-gramas: subsecuencias contiguas de una secuencia o cadena de caracteres de longitud N.

acciones. Las acciones observables pueden ir desde el más bajo nivel, el cual le corresponde al código binario, hasta aquellas que se efectúen sobre el sistema entero, como, por ejemplo, cambios en el registro del sistema o el sistema de archivos.

Como fue dicho previamente, la finalidad de esta técnica es la de exponer toda actividad maliciosa llevada a cabo por el programa durante su ejecución, sin comprometer la seguridad de la plataforma en la cual se realiza el análisis.

A diferencia del análisis estático, el análisis dinámico soluciona el problema de las técnicas de evasión que muchos malwares puedan tener integrados, tales como devolver un código de ensamblaje totalmente diferente al legítimo debido a que detectó que estaba siendo desensamblado para su análisis.

- *Análisis Híbrido*
  Esta aproximación es aquella en la que se busca realizar un análisis más a profundidad, al combinar tanto el análisis estático como el dinámico. Para enfatizar, el análisis estático es totalmente inútil cuando el programa analizado viene protegido con técnicas avanzadas de camuflaje (Kabakus & Dogru, 2018).

  Al utilizar esta aproximación se obtiene el beneficio máximo, mediante la obtención de indicadores obtenidos en el análisis estático y al romper los límites del mismo gracias a la detección del comportamiento intencionado del archivo analizado gracias al análisis dinámico.

Todos estos análisis requieren de alguna referencia para saber si se tiene a un archivo malicioso bajo examinación o no; hasta este punto, se han mencionado cadenas de caracteres, puertos, peticiones HTTP, firmas, entre otros objetos, los cuales hacen de referencia para la detección de malware. Todos estos objetos se los conocen como Indicadores de Compromiso, los cuales serán explicados en la siguiente sección.

Indicadores de Compromiso (IOC)

Para la detección de archivos maliciosos se requieren tanto de varias técnicas como marcos de trabajo (frameworks) y referencias para llevar esta tarea de forma eficiente, una de estas referencias son los índices de compromiso. Según (Lee, 2020) se los puede definir de la siguiente manera:

> "artefactos forenses que pueden identificar compromisos en el sistema o infecciones de malware"

Los IOC comprenden un vasto rango de objetos que pueden ser determinados como tal, como por ejemplo direcciones IP, hashes MD5, archivos de caché, entre otros. Es por

esto por lo que, para tener un panorama más claro sobre estos artefactos se los clasifica de la siguiente forma.

Clasificación

Para preservar una organización limpia en cuanto a los IOC se los clasifican de la siguiente manera (Lejonqvist & Larsson, 2018):

Por su presencia o atributos específicos:

- *Atómicos*
  Aquellos artefactos que ya no pueden ser descompuestos en artefactos más pequeños, como tal son ya una unidad. Ejm: dirección IP.
- *Computados*
  Cualquier artefacto resultado de cualquier cálculo de datos. Ejm: hash MD5, expresión regular.
- *Conductual*
  Estos son la propia firma del ataque. Pueden llegar a estar compuestos de varios IOC atómicos y computados. Ejm: Tráfico de red de entrante y saliente, creación y manipulación de archivos.

Otra clasificación seria la siguiente, esta se clasifica basada en su entorno:

- *Basado en el Host*
- *Basado en la Red*

En este caso podrán existir casos en los que ciertos artefactos pertenezcan a ambas, esto se debe considerar como una correlación (Lejonqvist & Larsson, 2018).

Formatos

Debido a que aún no existe un formato estándar de IOC aceptado por la comunidad, la mayoría de las veces se los trabaja en el formato XML, sim embargo, existen varios frameworks con los cuales se trabajan los IOC, entre los más reconocidos se encuentran:

- Yara
- STIX
- OpenIOC

La importancia que estos artefactos representan dentro de la práctica del análisis de malware es alta, y es debido a esto que este aspecto de los IOC será analizado a mayor detalle en el Capítulo 3.

Con la base sentada de todo el aspecto teórico y técnico de este trabajo, la siguiente sección trata sobre la selección de las muestras y herramientas que fungirán como sujetos de estudio dentro de este análisis.

Capítulo 2: Selección

Selección de muestras de malware

Para la selección de las muestras que serán utilizadas se tomará en cuenta el impacto que cada una de las muestras detalladas a continuación han provocado, sea este en cuanto a innovación de técnicas de infección, pérdidas provocadas a sus víctimas, velocidad de infección, entre otros. Cabe recalcar que cada uno de estos programas maliciosos son generados "a la medida" por varios grupos cibercriminales, por lo que, en ciertos casos la información disponible sobre la muestra podrá llegar a ser escasa o nula.

Las muestras seleccionadas para este trabajo serán 5, las cuales se someterán a los diferentes tipos de análisis de malware con las distintas herramientas que se detallarán a continuación.

Con este preámbulo, se detallan a continuación las muestras seleccionadas:

WannaCry

Como se describe en el portal de Kaspersky (2021b) WannaCry es un espécimen de ransomware de cifrado. Este software malicioso es considerado como un criptogusano, lo que quiere decir que además de encriptar la data de su víctima, posee características de un programa gusano, lo que quiere decir que puede auto propagarse. Para poder recuperar la información, las víctimas deben pagar un rescate de aproximadamente $600, y lo deben hacer mediante el uso de bitcoin.

WannaCry sacaba beneficio de una vulnerabilidad en su sistema operativo víctima, Microsoft Windows, conocida como EternalBlue[21], la cual afectaba al servicio para compartir archivos entre máquinas Windows.

Los ataques de WannaCry comenzaron el 12 de mayo de 2017, y este brote de ataques tuvo una duración de 3 días, periodo en el cual, provocó un resultado de más de 200000 víctimas alrededor de 150 países, que sufrieron daños evaluados en altas cantidades de dinero.

Los creadores de este ransomware, y perpetradores de los primeros ataques son conocidos bajo el nombre de *Lazarus Group*, esto se dio a conocer mediante un comunicado por parte de Microsoft en conjunto con Facebook (Smith, 2017), en el cual se habla sobre la labor que ambas organizaciones llevaron a cabo para detener dichos ataques y proteger a sus clientes.

PoisonFrog

De acuerdo a un reporte presentado en SecureList (GReAT [Global Research & Analysis Team], 2019) de Kaspersky encontró este malware de tipo backdoor durante un análisis relacionado con una filtración del grupo OilRig, utilizando su herramienta de reglas YARA, la cual fue desarrollada en el 2019.

[21] EternalBlue: exploit desarrollado por la National Security Agency (NSA) de los Estados Unidos. Este exploit aprovecha una vulnerabilidad en la implementación de Microsoft del protocolo SMB, lo que provocaba que no se manejen adecuadamente paquetes de datos enviados por atacantes de forma remota, lo que permitía que se ejecutase código de forma arbitraria en la computadora víctima.

Lo que se pudo obtener de la muestra fue un archivo ejecutable, el cual fue escrito inicialmente en C#, el cual de por sí no es relevante, sino la funcionalidad que este presenta, la cual se encarga de generar un script de PowerShell, el cual es que contiene los comandos necesarios para generar un backdoor mediante el protocolo HTTP, a este script fue el que se lo bautizó como PoisonFrog.

Quasar RAT

La muestra mencionada en este punto no proviene de un exploit determinado, sino que nace desde un programa legítimo. Quasar es una herramienta de administración remota, o RAT por sus siglas en inglés, de tipo open-source la cual está escrita en el lenguaje de programación C#, destinada para ser usada en sistemas operativos Microsoft Windows. Este aplicativo fue desarrollo por el usuario de Github bajo el nombre de "MaxXor".

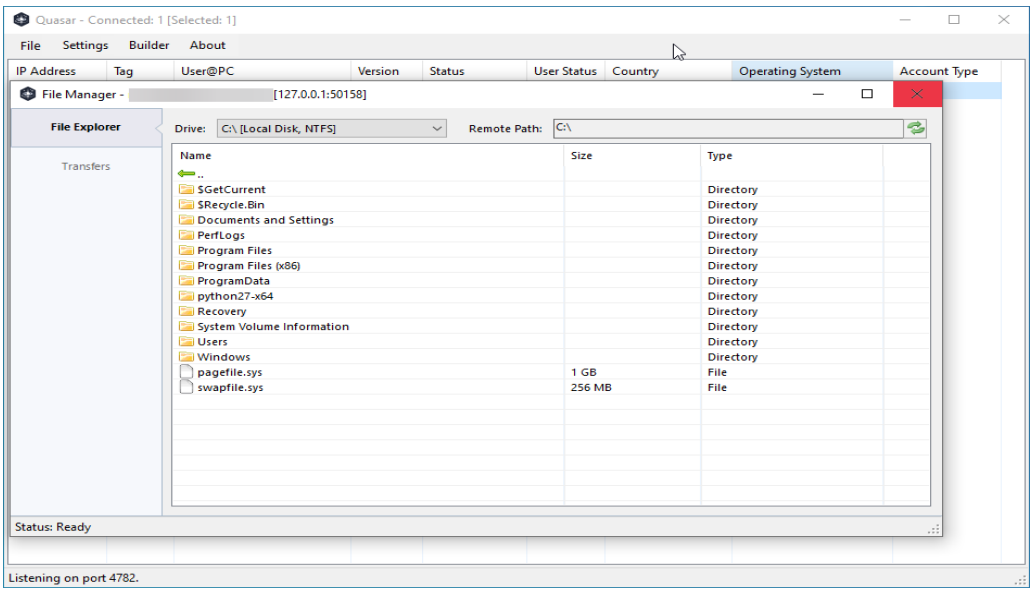


Ilustración 12: Captura de pantalla de QUASAR
Fuente: Github (GitHub - Quasar/Quasar: Remote Administration Tool for Windows, n.d.)

Dentro del archivo README.md del repositorio públicamente disponible podemos ver que menciona lo siguiente(MaxXor, 2020) :

> *"El uso varía desde la asistencia al usuario hasta el trabajo administrativo diario y la supervisión de los empleados. Al proporcionar una alta estabilidad y una interfaz de usuario fácil de usar, Quasar es la solución de administración remota perfecta para usted."*

A pesar de lo señalado, entidades dedicadas a la ciberdefensa encontraron dentro de sus estudios y análisis que varios actores criminales optaron por implementar esta herramienta en sus múltiples campañas de infección. El proveedor de productos de ciberseguridad, Cyware, dentro de uno de sus productos conocido como Cyware

Social, se reportaron varios casos en los Quasar fue utilizado por actores criminales (Cyware Hacker News, 2019), por ejemplo:

- Enero del 2017, la compañía Palo Alto observó que campañas dirigidas a entidades gubernamentales del Medio Oriente realizadas por el grupo DustSky localizado en Gaza, buscaba instalar el descargador Downeks, el cual una vez listo, se encargaba de soltar Quasar para así tener acceso a las máquinas de las víctimas.

- Enero del 2018, ataques dirigidos al Ministerio de Defensa Ucraniano fueron analizados, encontrando que la carga maliciosa fue un malware personalizado bajo el nombre VERMIN y Quasar. El programa malicioso se distribuía mediante documentos de siñuelo y este a su vez se encargaba de implementar el RAT.

Bad Rabbit

Dentro de la epidemia de ransomware que se desató en el año 2017, junto con el ataque masivo de WannaCry y Petya, se evidenció un tercer malware empleado a gran escala, bajo el nombre de Bad Rabbit.

Dentro del periodo en el que este programa malicioso fue utilizado inicialmente, se reportaron a varios medios rusos infectados, tales como Interfax news agency y el portal Fontanka.ru.

Según el equipo de investigadores de Kaspersky (Perekalin, 2017), gracias a la información recaba de estos ataques, se determinó que este malware era parte de una campaña de ataques *drive-by*, con este término refiriéndose a una técnica en la que la víctima, de manera no intencionada descarga el malware en su dispositivo, dejándolo vulnerable ante un ataque.

Otro resultado obtenido por el grupo de investigadores es que este nuevo malware empleaba código ya antes visto en la epidemia de Petya. Debido a la naturaleza del ataque, se consideró a esta muestra infecciosa como un malware de tipo troyano que aprovechaba de atacar al WMIC de los dispositivos de las víctimas para lograr su movimiento lateral dentro de las redes de las empresas atacadas. A diferencia de Petya, BadRabbit no es un malware que elimine los archivos, sino que su carga es ransomware puro, es decir, su única función es la de encriptar toda la información de un equipo. Dicho esto, dado el caso en el que un dispositivo víctima no llegue a que su disco duro no llegue a ser encriptado en su totalidad por BadRabbit, es posible recuperar la información del mismo, bajo ciertas condiciones.

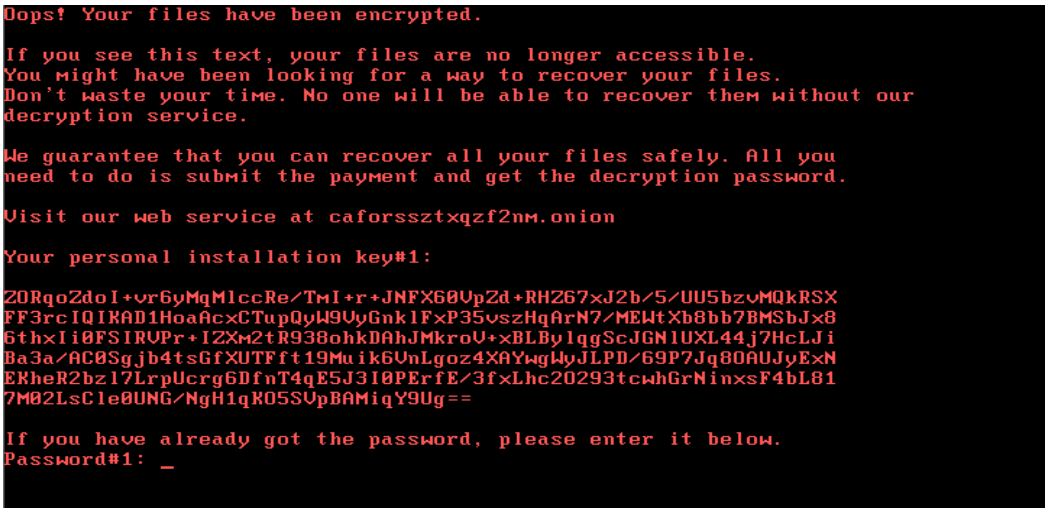


Ilustración 13: Captura de pantalla de equipo infectado con el ransomware BadRabbit
Fuente: WeLiveSecurity by ESET (Léveillé, 2017)

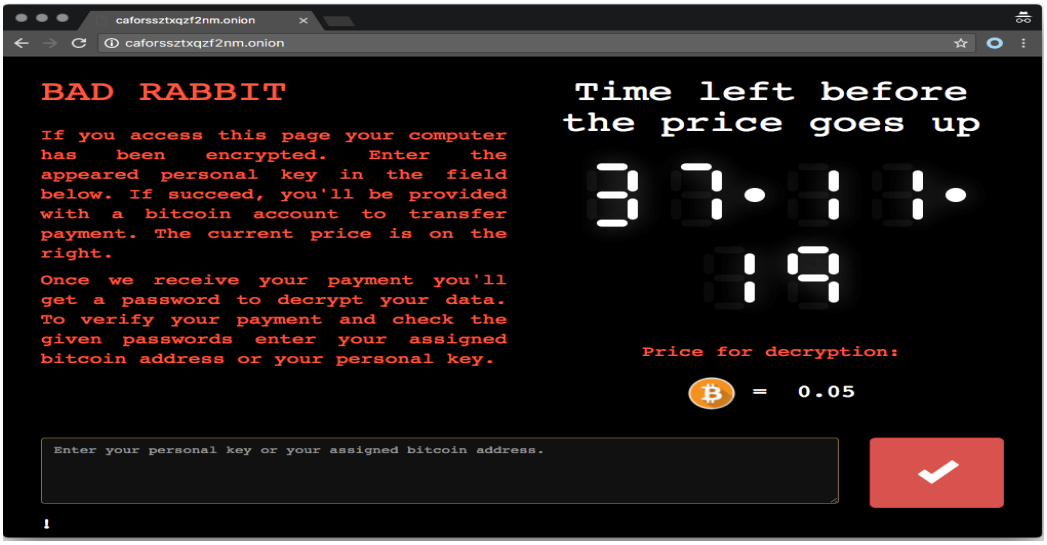


Ilustración 14: Captura de pantalla de dominio de pagos del ransomware BadRabbit
Fuente: WeLiveSecurity by ESET (Léveillé, 2017)

Glimpse

Esta muestra de malware es similar a la antes mencionada PoisonFrog, y de igual manera resulta está relacionada con el grupo criminal denominado APT34. Glimpse es un malware que destaca por su habilidad evasiva a las herramientas de detección de amenazas.

Este código malicioso fue escrito en PowerShell, y se ejecuta mediante un script de Visual Basic. Lo que diferencia a esta muestra con PoisonFrong es que utiliza el modo texto como una alternativa al tipo de registro de recurso DNS para comunicarse con su centro de mando, y esto le permite realizar tareas con muchas menos transacciones de instrucciones. Sumado a esto, este virus no depende de librerías DNS que existen ya en el framework de Microsoft, conocido como .NET, sino que elabora su propia librería y realiza las peticiones DNS de forma manual y las comunica directamente a su centro de mando.

Una vez se han expuesto las muestras seleccionadas para el análisis practico que forma parte del presente trabajo, es necesario ahora hablar sobre las herramientas seleccionadas para llevar a cabo dicha tarea.

Selección de herramientas de análisis

En esta sección se detallarán cada una de las 5 herramientas que fueron seleccionadas para ser evaluadas durante el desarrollo de este trabajo. Dado a que el proceso completo de análisis de malware requiere realizar diferentes observaciones de las muestras bajo cada una de las diferentes modalidades de análisis, se buscó tener una variedad entre las herramientas listadas a continuación para lograr cubrir ampliamente todo el espectro del proceso de análisis.

GHidra

Desarrollado por la National Security Agency (NSA) de los Estados Unidos, GHidra es un framework para ingeniería inversa de software. Este framework incluye un conjunto de herramientas de análisis de software de alta gama con todas las funciones que permiten a los usuarios analizar el código compilado en una variedad de plataformas, incluidas Windows, macOS y Linux.

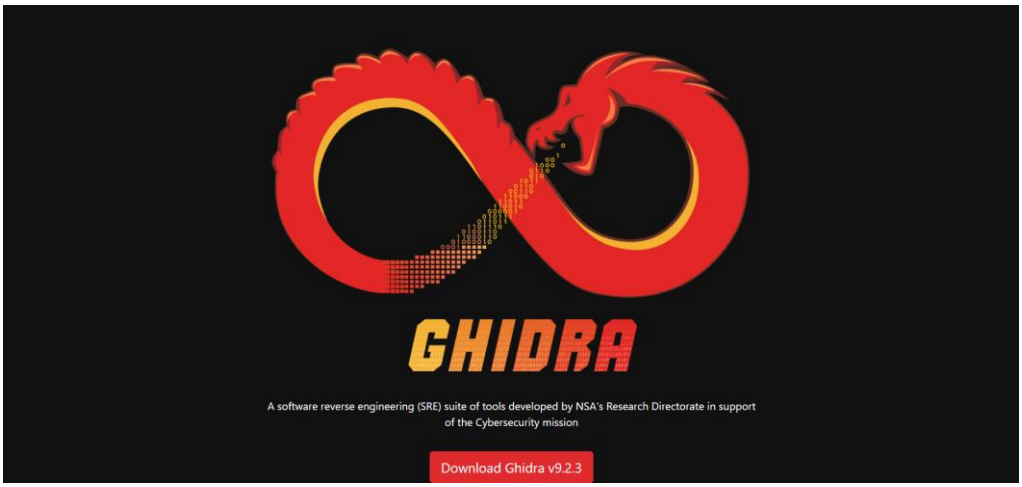


Ilustración 15: Landing page de ghidra-sre.org

Las capacidades con las que GHidra cuenta incluyen desmontaje, ensamblaje, descompilación, creación de gráficos y secuencias de comandos, junto con cientos de otras funciones. También admite una amplia variedad de conjuntos de instrucciones de procesador y formatos ejecutables y se puede ejecutar tanto en modo interactivo como automatizado con el usuario. Los usuarios también pueden desarrollar sus propios componentes de extensión de Ghidra y/o scripts utilizando Java o Python.

El proyecto, hoy en día se encuentra mantenido por la NSA y se encuentra disponible para su descarga desde su página web, ghidra-sre.org, y su código fuente

se encuentra disponible en su repositorio en GitHub, github.com/NationalSecurityAgency/ghidra

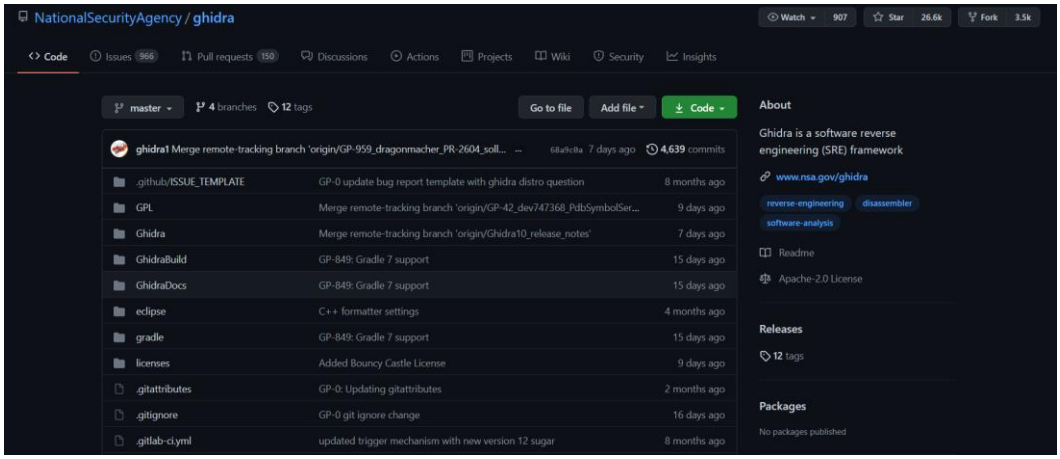


Ilustración 16: Repositorio en GitHub de GHidra
Fuente: GitHub (NSA, 2019)

Cutter

Cutter es una plataforma de ingeniería inversa open-source gratuita. Según la documentación disponible, Cutter cuenta con varias características para el análisis de malware, entre las cuales la más importante es que cuenta con un descompilador, y un desensamblador con la finalidad de que la plataforma analiza binarios e intenta generar representaciones de alto nivel del código máquina contenido en el archivo binario analizado (Cutter Dev Team, 2020).

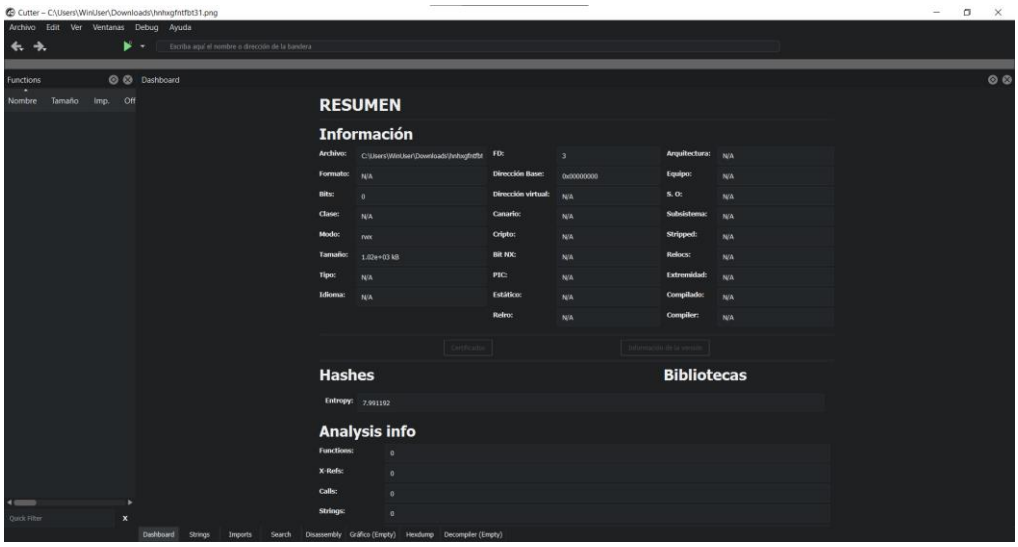


Ilustración 17: Captura de pantalla de Cutter tras análisis incial

De igual manera esta plataforma cuenta con su propio depurador y modo emulación con el fin de analizar de mejor forma los archivos binarios. Fuera de esto, esta plataforma cuenta con varios plugins que pueden ser integrados desde otras herramientas, como la anteriormente nombrada *GHidra.*

Cutter es altamente apoyado por la comunidad, teniendo un repositorio albergado en la plataforma de control de versionamiento *Github*, contando con varias

solicitudes de cambios abiertas recientemente y cuenta con altas calificaciones y copias del mismo repositorio.

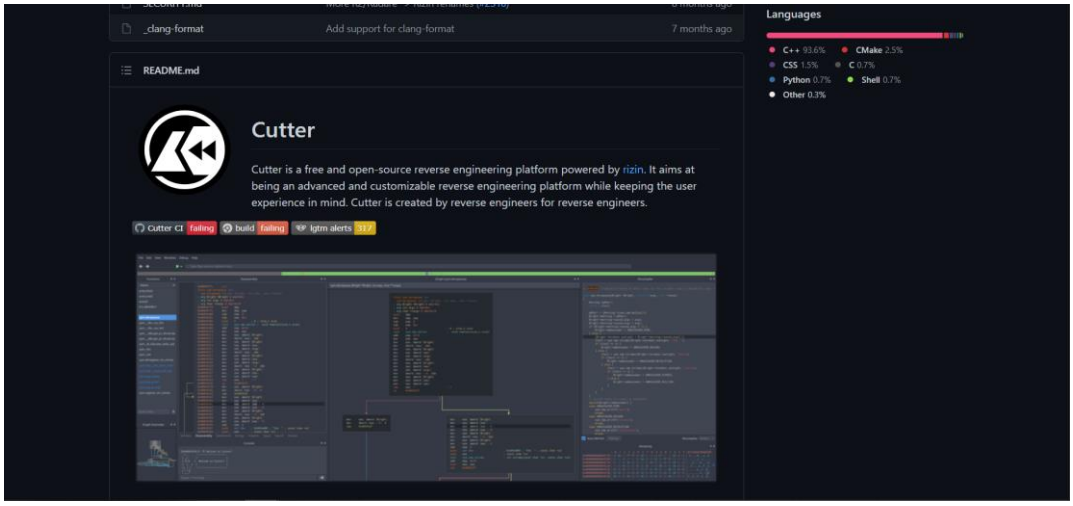


Ilustración 18: Repositorio en GitHub de Cutter
Fuente: GitHub (GitHub - Rizinorg/Cutter: Free and Open Source Reverse Engineering Platform Powered by Rizin, n.d.)

Intezer Analyze

Intezer Analyze es una plataforma web de análisis de malware que permite interactuar con muestras extremadamente peligrosas sin riesgo alguno gracias a sus varios servicios ofertados.

Según la propia compañía Intezer (2022), comentan que han logrado desarrollar el primer "sistema inmunológico cibernético contra código malicioso".

Esta plataforma cuenta con varias herramientas a la disposición del usuario, tales como vaciado de memoria, búsqueda de muestras ya analizadas, pruebas de endpoints, integraciones con otras herramientas, etc.

Intezer Analyze permite el análisis de muestras de malware mediante la carga del archivo bajo sospecha como por su hash, sea este en formato SHA256, SHA1 o MD5. Una de las ventajas de esta herramienta es que cuenta con algo conocido como análisis genético, el cual se encarga de estudiar todos los elementos que la muestra maliciosa a reutilizado de otras muestras previamente analizadas, los cuales incluyen al código fuente, cadenas de caracteres, funcionalidades, entre otros.

La herramienta utilizada análisis estático y dinámico mediante su entorno sandbox para su ejecución, extracción de código, además de implementar diferentes artefactos tales como IOC (Maltego Technologies, 2021).

Los archivos que soporta para su análisis son los siguientes:

- Archivos ejecutables de Windows.
- Archivos DLL.
- Documentos PDF.
- Documentos de la suite de Microsoft Office.

- Archivos comprimidos que contengan un archivo.
- Aplicaciones de Android.
- Archivos ejecutables de Linux.
- Scripts

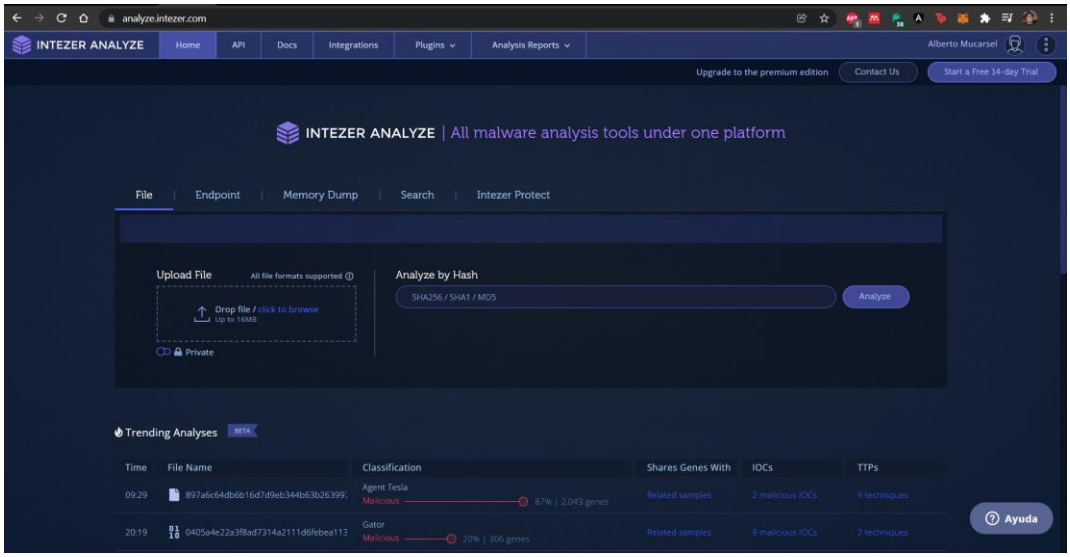


Ilustración 19: Página de inicio de Intezer Analyze
Fuente: Intezer Analyze (*Intezer Analyze – All-In-One Malware Analysis Platform*, n.d.)

Intezer Analyze provee de 2 tipos de cuentas para sus usuarios, gratuita y pagada. Entre las varias ventajas que ofrece una cuenta pagada, la que resulta más importante dentro del contexto de este trabajo es el de la capacidad de descargar las muestras de los análisis que se han llevado a cabo en la plataforma, por lo que esta plataforma actuará de forma simultánea como herramienta de análisis y repositorio de muestras, ya que como se puede observar en la *figura 19*, se puede hacer uso de un periodo de prueba de 14 días para acceder a las diferentes funciones pagadas.

VirusTotal

VirusTotal nació como un sitio web creado por la compañía de seguridad española "Hispasec Sistemas", la cual fue adquirida por Google en el 2012, y más tarde, en el 2018, fue cedida a una empresa subsidiaria de Google, llamada Chronicle Security.

VirusTotal es un servicio gratuito de análisis de archivos y URLs con el fin de encontrar virus, gusanos, troyanos y otros tipos de contenido malicioso.

El modo de operación de VirusTotal es bastante sencillo, el usuario final tiene la opción de subir, sea un archivo desde su máquina o adjuntar una URL mediante las diferentes vías que VirusTotal proporciona, sea mediante su página web, extensiones de navegador, programas de escritorio, o mediante el uso de API[22]. Una vez se sube el objeto de estudio, la herramienta pasa a este objeto por sus más de 70 escáneres antivirus y servicios de listado de bloqueos de URLs.

[22] API: Application Programming Interface.

La particularidad que tiene VirusTotal es que, una vez terminado el análisis, se comparte un reporte al usuario final con la información esencial obtenida durante el estudio, y dichos resultados son a su vez compartidos con sus compañeros examinadores (VirusTotal, 2021), lo que se puede entender que, una vez se envíe cualquier tipo de archivo sospechoso a VirusTotal, se estará contribuyendo a un vasto número de herramientas y plataformas dedicadas a mejorar la seguridad informática en el planeta.

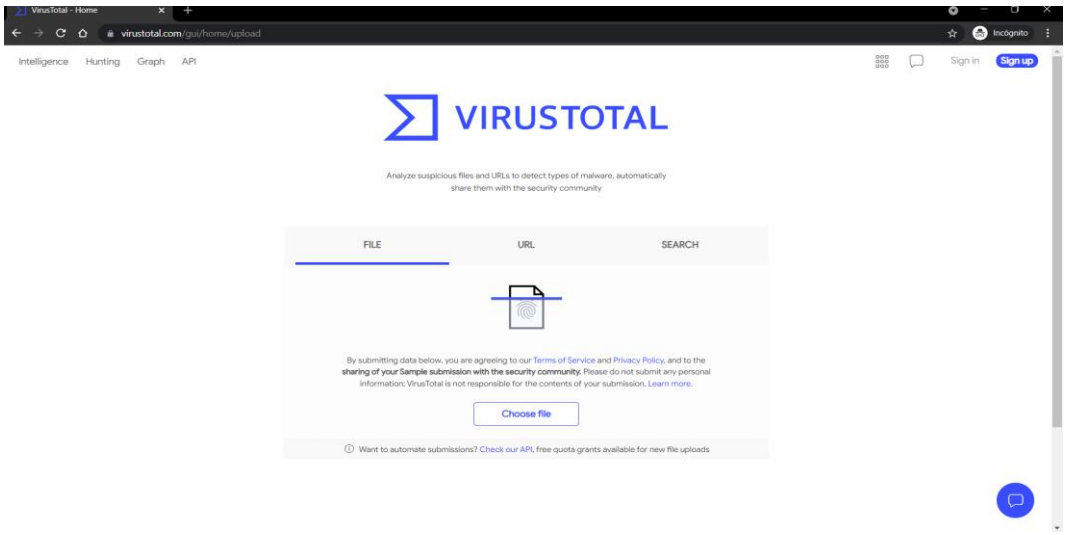


Ilustración 20: Página de inicio de virustotal.com

Volatility

Volatility es un framework de tipo open source desarrollado por Volatility Foundation, una organización sin fines de lucro fundada en el 2007.

De acuerdo con la documentación del repositorio de Volatility se describe a este framework como una colección de herramientas implementadas en el lenguaje de programación Python, con el fin de lograr la extracción de artefactos digitales desde muestras de memoria volátil, en este caso, muestras de memoria RAM. Es pertinente aclarar que este conjunto de herramientas no proporciona la obtención de las muestras como tal, solamente para el análisis de estas, una vez han sido obtenidas mediante cualquier solución disponible.

Volatility soporta el análisis de varias imágenes de memoria de acuerdo al sistema operativo. Entre ellos se tiene soporte para Windows, desde XP hasta ciertas versiones de Windows 10.

Lo atractivo de esta herramienta es que es desarrollada para ser ejecutada mediante consola, de esa manera se pueden aprovechar de mejor manera sus bondades al ahorrar recursos de procesamiento en una interfaz gráfica, sumando a este ahorro, también se debe tomar en cuenta la velocidad de ejecución con la que cuenta al ser implementado en Python, el cual es un lenguaje interpretado.

Obtención de muestras de malware

Las muestras de malware serán obtenidas desde diferentes repositorios disponibles en la web. A continuación, se dará una breve introducción a cada uno de ellos.

theZoo

El repositorio conocido como “The Zoo”, el cual se encuentra disponible en la plataforma Github, y en su propia página web, thezoo.morirt.com. El proyecto es mantenido por el usuario *ytisf* y tiene su fecha de creación como el 19 de junio de 2014.

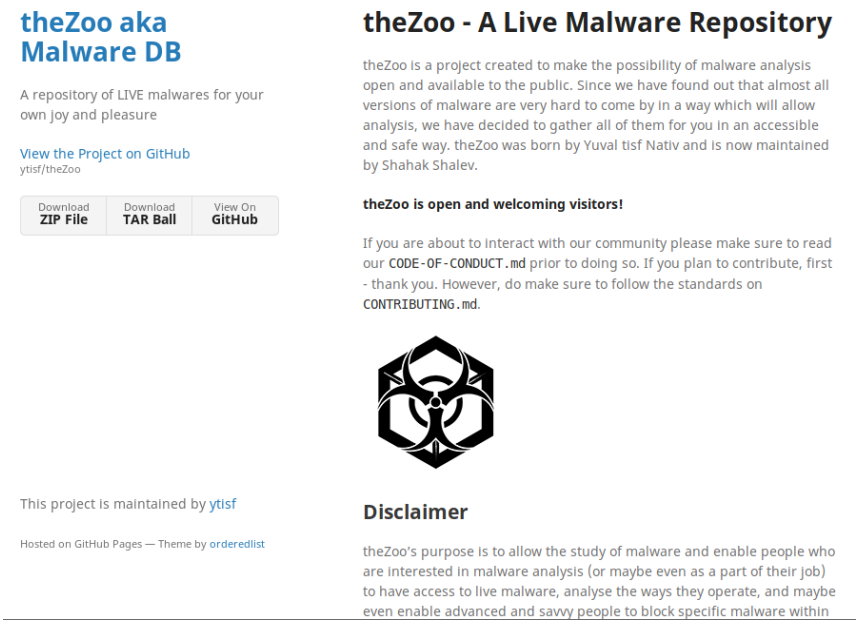


Ilustración 21: Pagina inicial de theZoo

Este proyecto se encuentra mantenido por la comunidad, y da la facilidad de descárgalo todo como un script de Python, en cual dentro de su página principal entrega las respectivas instrucciones para su instalación y uso.

Como parte de un protocolo que se lleva por la comunidad de ciberseguridad, el anuncio en su archivo README.md de The Zoo hace un *descargo de responsabilidad* (disclaimer) de que este proyecto es con fines educativos y aquellos que opten por utilizarlo no lo hagan con fines maliciosos, además de enfatizar que en este proyecto se contienen muestras de malware activas, por lo que, para evitar una infección y esparcimiento involuntario, se encuentran todos encriptados.

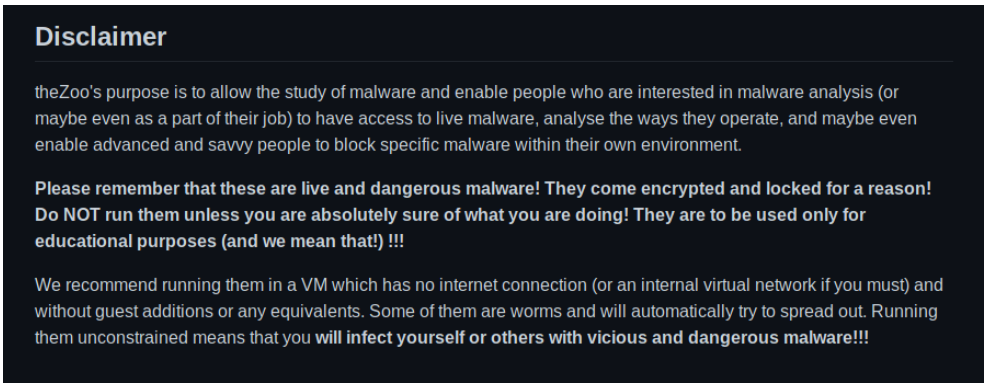


Ilustración 22: Disclaimer dentro del repositorio de theZoo en GitHub

Las muestras serán descargadas a conveniencia de forma directa desde el directorio */malwares/Binaries*, esto con la finalidad de solo manipular las muestras mencionadas en este trabajo y minimizar el riesgo de infección compleja de controlar en los entornos de prueba.

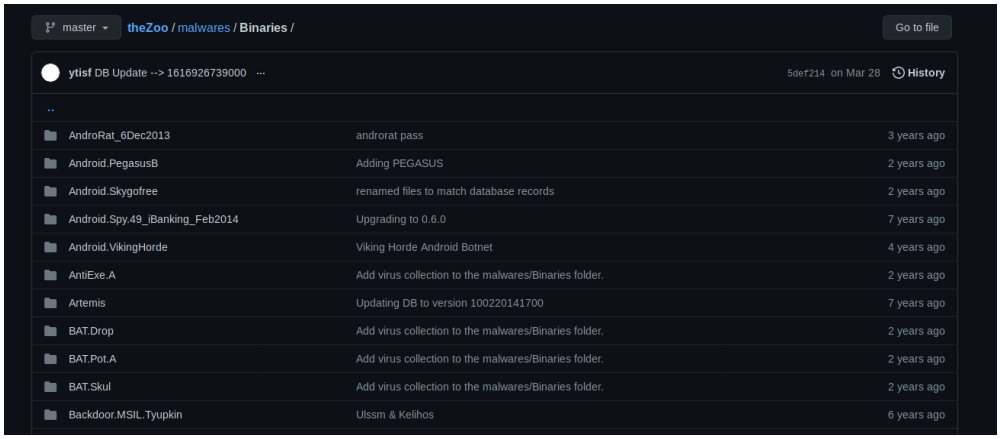


Ilustración 23: Muestra del contenido de muestras albergado en theZoo

MalwareBazaar

MalwareBazaar es un proyecto similar a theZoo, tiene como objetivo la recolección y compartir muestras de malware, para así ayudar a investigadores de seguridad de TI como a analistas de amenazas a cumplir con la protección contra ciber amenazas a sus clientes (Abuse.ch, 2020).

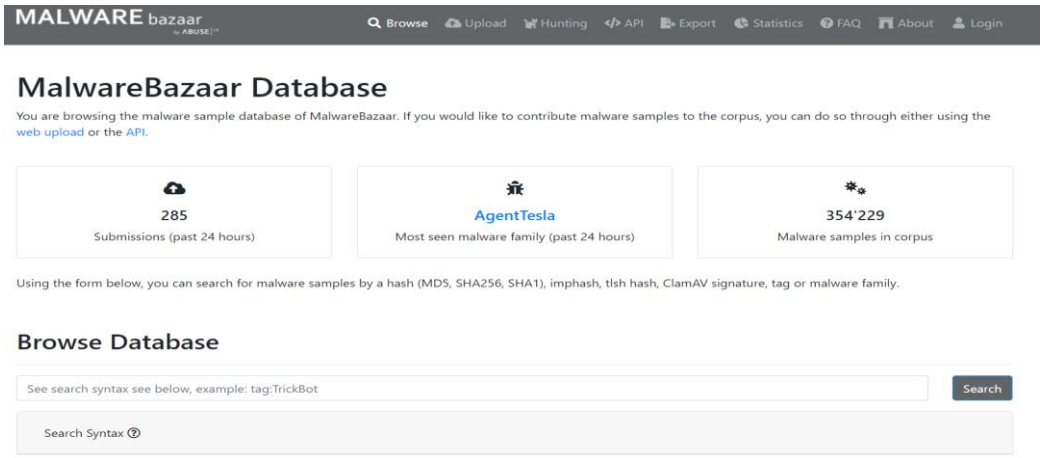


Ilustración 24: Página inicial de bazaar.abuse.ch

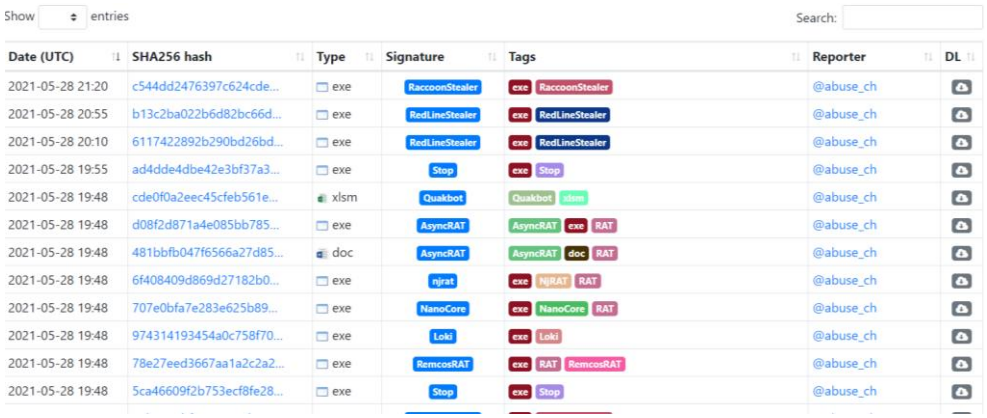


Ilustración 25: Listado de muestras almacenadas en MalwareBazaar

El repositorio fue desarrollado por *abuse.ch*, una iniciativa de investigación desarrollada por el Institute for Cibersecurity and Engineering (ICE) el cual es albergado en la Bern University of Applied Sciences (BFH) en Suiza . El proyecto *abuse.ch* comenzó con una persona suiza que buscaba hacer el bien en la web, y ha escalado hasta el punto de trabajar con grandes fabricantes de la industria. Este proyecto no solo oferta el repositorio de muestras de malware sino muchos otros como se puede observar en la *ilustración x*, tal como *Feodo Tracker* el cual se dedica a rastrear infraestructuras de botnets, otro como *I Got Phished*, se dedica a notificar a dueños de dominios si credenciales de correos electrónicos relacionadas a sus dominios de correo han sido vulneradas.

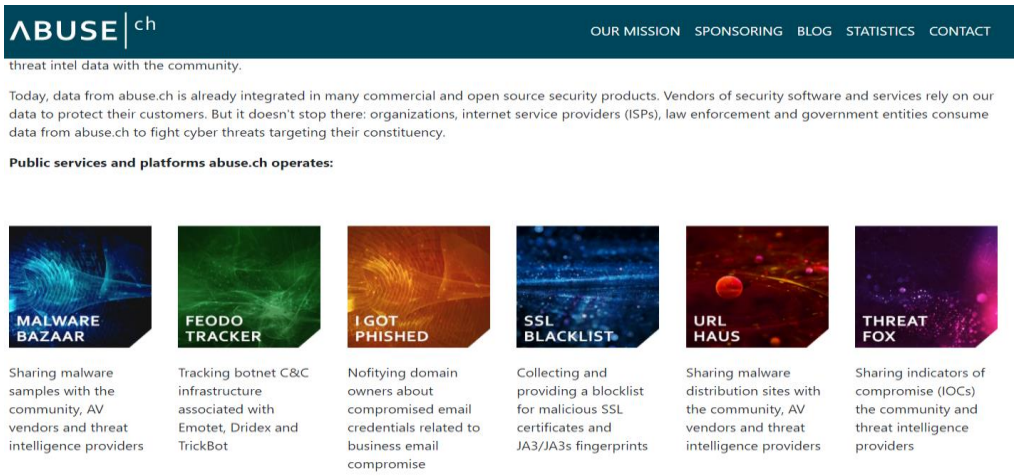


Ilustración 26: Listado de productos de Abuse.ch

Malware-sample-library

Este repositorio alojado en GitHub fue creado el 24 de octubre del 2018, y se encuentra mantenido actualmente por el usuario Mustafa Kaan Demirhan, que utiliza el nombre de usuario *mstfknn.* Dentro de este repositorio se encuentra filtraciones de muestras de malwares de ciertos actores criminales tales como APT34, el cual es un grupo hacker patrocinado por el país de Irán.

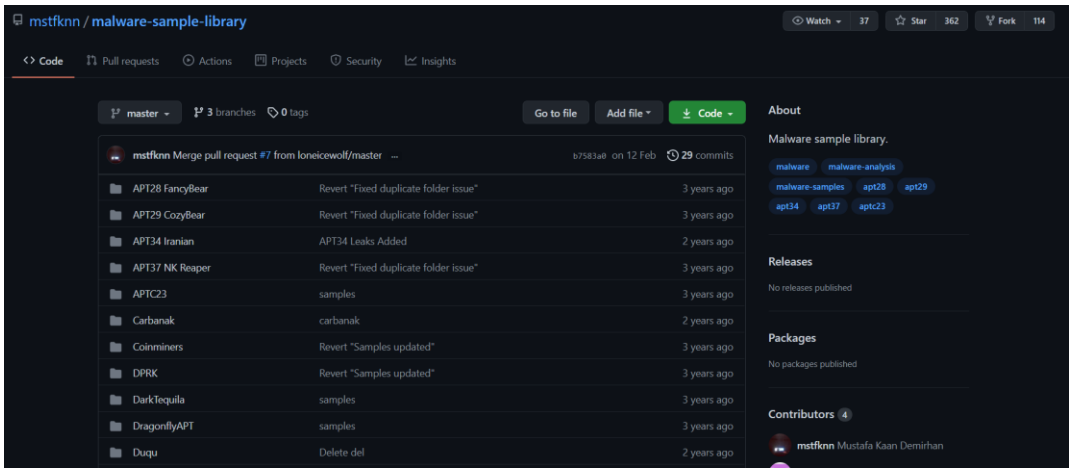


Ilustración 27: Repositorio en GitHub de Malware-sample-library

Con los elementos tratados anteriormente, la siguiente sección del presente trabajo se tratará del punto central del mismo, el cual es la importancia que los IOC tendrán dentro del proceso de análisis de malware de las muestras expuestas con el uso de las herramientas mencionadas.

Capítulo 3: Aplicación

Importancia de los IOC

En este punto de la historia en el cual estamos ante el advenimiento de una nueva era tecnológica, se ha llegado a presenciar un incremento sin precedentes en la cantidad de ciber ataques detectados, los cuales han logrado evolucionar y llevarse a cabo de maneras mucho más sofisticadas. Los actores maliciosos cuentan con un amplio set de tácticas y herramientas para llevar a cabo sus campañas de ataque contra sus víctimas y ser capaces de llevar a cabo su objetivo sin importar cualquiera que pueda llegar a ser. El llegar a comprender al atacante se ha transformado un proceso altamente complicado y sumamente relevante, ya que, una vez se logre obtener información del atacante sobre la que se pueda actuar, esta misma se puede ser empleada para adaptar las defensas de redes informáticas de manera automatizada para llegar a proteger a la red de posibles amenazas.

Para llegar a comprender a los actores maliciosos, los expertos en el área de ciber defensa han concentrado sus esfuerzos en refinar lo que se conoce como *Inteligencia de Ciber Amenazas* (C.T.I. por sus siglas en inglés) la cual es el resultado derivado de la recolección, evaluación y análisis de la *Información de Ciber Amenazas* (CIS, 2018). Con la obtención del CTI, lo que se conoce como la guerra informática ha ido tomando un camino diferente. En otros tiempos sin este análisis de la vasta información que los analistas recolectan y comparan, se tenía presente un alto nivel de incertidumbre al momento en el que se detectaba un ataque *hecho a la medida* de la víctima debido a que esto no tenía precedente alguno y complicaba el proceso de respuesta a incidentes.

Gracias a que el CTI se enfoca en las capacidades, motivaciones y metas del actor malicioso o adversario y cómo estas pueden llegar a llevarse a cabo (Ramsdale et al., 2020), se ha logrado mejorar notablemente el proceso tanto en el levantamiento de ciber defensas, así como en la informática forense. Dentro de esto la unidad de información dentro del CTI se conocen como artefactos, los cuales son objetos del sistema asociados con el ataque. Tal vez esa pequeña definición pueda escucharse familiar, y eso se puede deber a que es la idea principal tras el tema principal de este trabajo y de la cual se ha estado hablando hasta este punto, esos artefactos son los IOC.

Los IOC son la unidad básica con la trabaja la CTI, en pocas palabras, los artefactos sencillos que se han venido mencionado resultan ser elementos fundamentales dentro del campo de la seguridad informática a lo largo de muchas de sus subramas.

Con todo lo expuesto anteriormente, se expone que los IOC llegan a convertirse en objetos preciados para los profesionales de la ciberseguridad, y en el caso específico que nos compete en este trabajo, toman mayor importancia para los analistas y académicos forenses.

El estar bajo el constante proceso de monitoreo y obtención de los IOC ayuda a que las diferentes organizaciones mejoren a capacidad de detección y respuesta antes incidentes de ciberseguridad. De forma más concreta, la constante recolección y comparación de estos indicadores (la cual gracias a las nuevas tecnologías puede efectuarse en tiempo real) permite la identificación temprana de brechas de seguridad que hayan logrado pasar desapercibidas por otras herramientas de defensa, y claramente

estos artefactos llegan proporcionan los suficientes recursos para la elaboración del análisis forense de dichos incidentes.

A pesar de que estos objetos pueden llegar a escucharse como evidencias más que explicitas de cualquier táctica maliciosa, la realidad es que cuentan ciertos aspectos que pueden llegar a ser contraproducentes en ciertos casos, por lo que a continuación veremos a mayor detalle las características, ventajas y desventajas con las que cuentan los IOC.

Características

Según Shengíng Zhou et al.(2018) los Indicadores de Compromiso son artefactos forenses que son usados como señales de que un sistema se encuentra infectado con una pieza particular de malware. Según Zhou, los IOC son en realidad la combinación de varios objetos tales como: firmas de virus, direcciones IP, nombres de dominio o URLs de botnets, hashes MD5 de archivos de ataque, peticiones HTTP inusuales, etc.

La razón por la cual se llega a definir a los IOC como un conjunto de estos objetos mencionados y no a cada uno como un indicador de por sí, es debido a que por sí solos no son suficiente evidencia para lograr la identificación de posibles actividades sospechosas dentro del sistema (Chauhan & Bansal, 2021).

Existen varios tipos de IOC por lo que varios expertos de seguridad han optado por ayudar en la estandarización de estos artefactos, y por esto se han llegado a formar grandes comunidades que, hasta la fecha, se encuentran contribuyendo a estos proyectos para el beneficio de todos en el campo. Existen 3 grandes protocolos estandarizados que pueden ser utilizados para que un indicador sea documentando, estos son:

- Yara
- STIX
- OpenIOC

Tal como se mencionó en el Capítulo 1, los formatos de los IOC juegan un papel importante dentro de la práctica de la informática forense y detección de amenazas, ya que de esta manera es como la comunidad se asegura de poder compartir la así llamada inteligencia de manera más ordenada y eficiente.

YARA

Para continuar de manera correcta, primero es necesario aclarar que YARA es un acrónimo en ingles el cual significa *Yet Another Ridiculous Acronym.* YARA en realidad es una herramienta la cual, según su propia documentación dicta que fue diseñada para “ayudar a los investigadores de malware a identificar y clasificar muestras de malware” (Alvarez, 2020). Esta herramienta a como resultado una descripción del objeto analizado, basándose en patrones textuales o binarios. Cada una de estas descripciones, las cuales se las conoce de manera más popular como *reglas YARA*, están compuestas de un conjunto de cadenas de caracteres y una expresión booleana que determina su lógica.

La manera en la que una regla YARA funciona es bastante sencilla, se podría decir que funciona como un fragmento de un lenguaje de programación. Estas reglas

trabajan al definir un determinado número de variables las cuales contienen patrones encontrados en la muestra de malware de la cual se escribe la respectiva regla. Tras esto se establece una o varias condiciones que por lo general se encargan de evaluar la presencia de los patrones previamente definidos. Dependiendo de cómo fue establecida la condición si todos o varios puntos se cumple, entonces esta regla YARA puede identificar exitosamente a una muestra de malware.

```
rule ExampleRule
{
    strings:
        $my_text_string = "text here"
        $my_hex_string = { E2 34 A1 C8 23 FB }

    condition:
        $my_text_string or $my_hex_string
}
```

Ilustración 28: Ejemplo de una regla YARA básica
Fuente: Yara Official Documentation (Alvarez, 2020)

Tal como se puede ver en la *figura x*, la estructura básica de una regla YARA no es compleja, sin embargo, existen otros elementos que pueden ser definidos (Fox, 2021) que pueden ayudar a proveer de mejor contexto a quien opte por utilizar estas reglas.

- ***Metadata:***
  La metadata como tal no afecta directamente al funcionamiento de la regla, esta sirve más para proveer más información sobre la regla misma. Entre los campos que se pueden definir aquí se encuentran:
  - *Autor*
  - *Fecha*
  - *Versión*
  - *Referencia*
  - *Descripción*
  - *Hash*

- ***Cadenas de caracteres:***
  Dado a que lo más común que puede ser hallado en muestra de malware son cadenas de caracteres únicas, estas se vuelven el artefacto idóneo para usarse en la construcción de una regla YARA. Para definir a una cadena, esta debe ser declara dentro como una variable.
  Adicionalmente de su declaración, se pueden añadir modificadores después de la cadena declarada con la finalidad de afinar la búsqueda:
  - *"fullword"*
    Sirve para que se busque por el término exacto.
  - *"wide"*
    Este modificador permite que se comparen cadenas en formato Unicode que pueden estar separadas por bytes nulos.
  - *"wide ascii"*

Este modificador permite que una regla compare cadenas que se encuentren en formato Unicode o ASCII.

- *"nocase"*
  Este modificador permite que se busque a la palabra sin importar el caso.

- ***Condiciones:***
  Las cadenas de caracteres definen el qué se va a buscar con una regla YARA, mientras que en este elemento se definirá todos los requerimientos que se deben cumplir para que la regla YARA determine si ha dado con un resultado positivo. En este apartado se pueden definir un sinnúmero de condiciones que pueden ayudar a dar con una muestra determinada de malware, tal como revisar el tamaño de un archivo, si este es igual, menor o mayor a determinado tamaño, revisar la cabecera de archivo analizado, etc.
  Lo anteriormente mencionado son criterios de condiciones que pueden ayudar a la búsqueda principal, la cual es la presencia de las cadenas de caracteres que se hayan definido previamente dentro de la regla.

- ***Imports:***
  Como se trata de una pieza de código, se pueden implementar librerías externas para añadir criterios con mayor extensión a la búsqueda de la regla.

STIX

Structured Threat Information Expression (STIX) es un lenguaje estructurado para la describir CTI con la finalidad de que esta pueda llegar a ser compartida, almacenada y analizada de una forma consistente.

Este lenguaje representa a sus objetos dentro del formato JSON lo cual lo convierte en una forma universal de transmitir la información contenida por el internet sin ningún problema. Además de esto también cuentan con su propia librería para ser implementada en el lenguaje de programación Python, lo que da el potencial de poder generar varios de estos objetos de manera más rápida mediante la facilidad de creaciones de scripts que nos da Python.

```
{
    "type": "campaign",
    "id": "campaign--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
    "spec_version": "2.1",
    "created": "2016-04-06T20:03:00.000Z",
    "modified": "2016-04-06T20:03:23.000Z",
    "name": "Green Group Attacks Against Finance",
    "description": "Campaign by Green Group against targets in the financial services sector."
}
```

Ilustración 29: Ejemplo de construcción clave básica en STIX 1.0
Fuente: STIX Official Documentation (OASIS, 2019)

Dentro del primer artículo escrito por su creador, Sean Barnum (2014), habla de que, dentro de este lenguaje, en un contexto de alto nivel, se pueden definir 9

construcciones clave y sus relaciones. con las que este lenguaje trabaja los cuales son:

- ***Observables***
- ***Indicadores***
- ***Procedimientos, técnicas y tácticas del adversario***
- ***Objetivos de explotación***
- ***Curso de acción***
- ***Campañas***
- ***Actor de amenaza***
- ***Reportes***

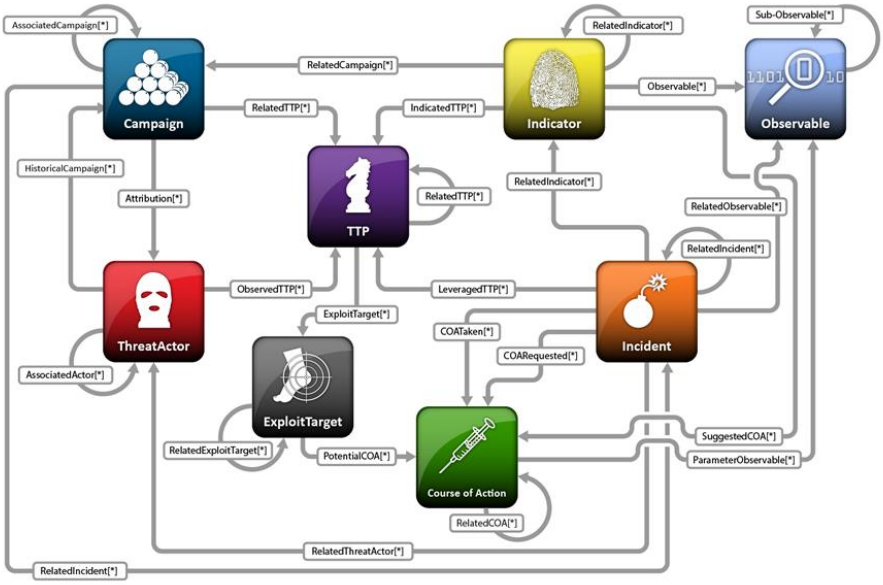


Ilustración 30: Relaciones de construcciones clave en STIX 1.0
Fuente: STIX 1.0 Whitepaper (Barnum, 2014)

Cada una de las construcciones se relaciona con otra tal como se observa en la *figura x* para así lograr una correlación que ayude a la detección de intrusiones e infecciones de forma más precisa. Tras la liberación de la última versión de STIX la cual es la 2.1 y ahora mantenida por la organización OASIS Open, las construcciones clave pasaron de ser 8, a ser 18, y ahora se las conoce oficialmente como los *Objetos de Dominio de STIX (STO por sus siglas en inglés)* los cuales son:

- ***Patrón de ataque***
- ***Campaña***
- ***Curso de acción***
- ***Agrupamiento***
- ***Identidad***
- ***Indicador***
- ***Infraestructura***
- ***Conjunto de intrusiones***
- ***Ubicación***

- ***Malware***
- ***Análisis de malware***
- ***Nota***
- ***Data observada***
- ***Opinión***
- ***Reporte***
- ***Actor de amenaza***
- ***Herramienta***
- ***Vulnerabilidad***

Dado a que este lenguaje es bastante extenso no se hablará a profundidad de cada uno de sus STO actuales, si el lector desea profundizar puede usar como referencia la documentación de STIX la cual se encuentra en el siguiente enlace: https://oasis-open.github.io/cti-documentation/

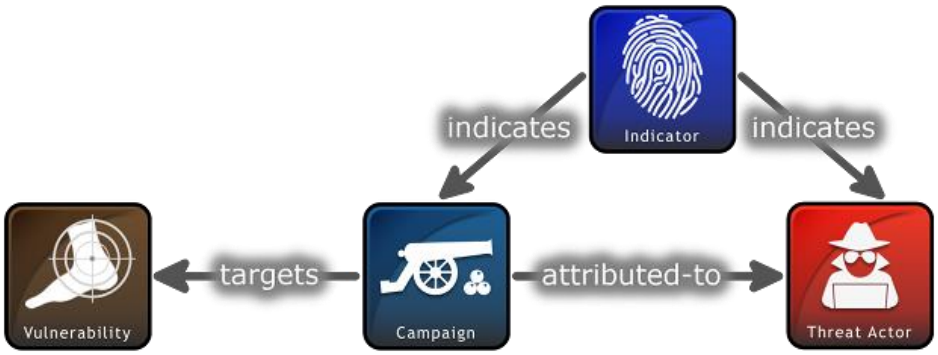


Ilustración 31: Relación básica entre STOs en STIX 2.1
Fuente: STIX Official Documentation (OASIS, 2019)

OpenIOC

OpenIOC es una herramienta que habilita al usuario a gestionar sus propios IOC. Esta iniciativa ha sido generada por la empresa MANDIANT la cual es perteneciente a la marca internacional FireEye, la cual ha sido parte del grupo de pioneros dentro de la definición y el establecimiento del concepto de los IOC. Esta herramienta es de tipo open-source la cual se reparte bajo una licencia libre de topo Apache 2.

El lenguaje en el cual OpenIOC maneja a los artefactos forenses se basa en XML (Extensible Markup Language) el cual provee un formato estándar altamente usado para codificación de información en un formato entendible para una máquina que luego permitirá implementar diversos métodos para comunicar esta misma información. El autor Baldric Mathews, en su artículo titulado *"Sophisticated Indicators for the Modern Threat Landscape: An Introduction to OpenIOC"* (2017) menciona que el hecho de implementar XML dentro de este formato para el manejo de IOC provee varios beneficios para quien lo consuma, por ejemplo, su esquema base es extensible con conjuntos de indicadores en caso de ser necesitados. Otro beneficio de este formato es que, gracias a que emplea XML, existe una facilidad para la creación de programas utilitarios para realizar la conversión de artefactos

escritos en OpenIOC hacia otros formatos que pueden aportar o verse beneficiados de la información contenida en el IOC.

Este framework cuenta con dos herramientas para la gestión directa de los IOC, las cuales son IOC Editor & IOC Finder. De forma sencilla, la primera herramienta se encarga de la gestión, edición y manipulación de los IOC en el formato XML, mientras que la segunda permite la recolección de la información del sistema anfitrión y reporta se llegase a encontrar la presencia de IOCs dentro del mismo. Actualmente estas herramientas son solamente compatibles con ciertas distribuciones de sistemas operativos Windows, por lo que dentro de la industria de la ciberseguridad se lo acusa de una falta de flexibilidad, y por lo que no llega a ser adoptado de buena manera.

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="72669174-dd77-4a4e-82ed-99a96784f36e"
last-modified="2012-01-05T02:49:14" xmlns="http://schemas.mandiant.com/2010/ioc">
  <short_description>DUQU (METHODOLOGY)</short_description>
  <description>Indicator for the duqu trojan. The initial duqu driver will decode and
inject a dll (marked as .pnf) into a system process (usually services.exe). The injected
dll contains another dll encoded within it's resource section which it will inject into
other processes as identified within its encoded configuruation file (another .pnf file).
This second injected dll is responsible for all backdoor/C2 communication.</description>
  <authored_by>MANDIANT</authored_by>
  <authored_date>2011-10-21T16:13:31</authored_date>
  <links>
    <link rel="caveat">Methodology</link>
  </links>
  <definition>
    <Indicator operator="OR" id="9fd46693-ee1c-4d31-b732-35bf952651e3">
      <Indicator operator="AND" id="e4deb0af-7558-498e-b953-6e70ec694767">
        <IndicatorItem id="d5b29cfe-8599-498a-b805-326273fe10c5" condition="contains">
          <Context document="FileItem"
search="FileItem/PEInfo/DigitalSignature/CertificateSubject" type="mir" />
          <Content type="string">C-Media Electronics Incorporation</Content>
        </IndicatorItem>
        <IndicatorItem id="1ca2947c-0b26-409c-93d2-28f6b364bc0b" condition="contains">
          <Context document="FileItem" search="FileItem/FileName" type="mir" />
          <Content type="string">cmi4432.sys</Content>
        </IndicatorItem>
      </Indicator>
```

Ilustración 32: Ejemplo de IOC escrito en XML
Fuente: INCIBE-CERT (López, 2016)

Ventajas y desventajas

Una vez se han expuesto los diferentes formatos en los cuales un IOC puede ser escrito y utilizado, es momento de analizar a manera general, las ventajas y desventajas que estos artefactos traen consigo.

Ventajas

- *Existen varios formatos para su documentación*
  Tal como se ha visto dentro de este capítulo, los IOC tienen varios formatos en los cuales pueden ser escritos e implementados. Gracias a esto, la obtención de CTI de mayor calidad se ha visto en incremento gracias a que los diversos formatos cuentan con herramientas para la conversión en otros mediante diferentes herramientas, lo cual ayuda a que se lleguen a complementar en determinados casos, logrando así un mejor indicador de compromiso con más información.

- *Habilita la detección en tiempo real de amenazas*

Como resultado de sus varios formatos, se han creado diversas herramientas capaces de leer de forma sencilla un IOC. Varios IDS[23] e IPS[24] utilizan los varios artefactos que tienen a su disponibilidad, sean estos obtenidos desde el Internet o generados dentro de su propio entorno, para realizar comparaciones en tiempo real con el tráfico de red que vigilan, y de esta manera optimizar la detección de amenazas gracias a que los indicadores ayudan a que los sistemas sepan qué exactamente puede ser una amenaza dentro del tráfico analizado y puedan realizar su tarea de defender o notificar sobre un posible intento de intrusión no autorizado.

- *Permiten conocer el nivel de protección que se encuentra implementada*
  Con la detección en tiempo real de estos indicadores, una organización es capaz de obtener el nivel de protección que tienen implementada. En determinados casos en la que se presenta una recurrente detección de uno o varios indicadores dentro de un patrón, los equipos de seguridad podrán trazar y lleva a cabo planes de mejoras defensivas ante futuros ataques al cubrir sus presentes vulnerabilidades gracias a estas detecciones recurrentes.

Desventajas

- *Pueden producirse falsos positivos*
  Dado a que varios archivos infecciosos pueden venir encubiertos dentro de archivos inofensivos, la obtención de las firmas de dichos archivos, en el caso de que no actúen como un caballo de Troya, podrían llegar a levantar falsas alarmas. Lastimosamente, la validación de esto no puede realizarse de forma automática ya que esto se debe realizar analizando el contexto de cada alarma para calificarla como una verdadera alerta o una falsa alarma.

- *Procedimientos no están totalmente automatizados*
  Como ejemplo se puede tomar la desventaja anteriormente mencionada. Para realizar la validación de falsos positivos, se debe analizar el contexto en el cual se presenta cada alarma, cosa que hasta ahora no se ha logrado enseñar a que una máquina sepa entender de forma precisa, por lo que, esta tarea se la realiza de forma manual por parte de equipos de seguridad que cuentan con expertos analistas los cuales dependen de su experiencia para llegar a discernir entre estas alarmas.

[23] IDS: Intrusion Detection System
[24] IPS: Intrusion Prevention System

Capítulo 4: Prototipos de pruebas

Levantamiento de entorno de pruebas

Llegada a esta fase del presente trabajo, se requieren de ciertos elementos para iniciar su desarrollo práctico. El componente principal para el análisis tanto de las muestras maliciosas como del desempeño de las herramientas que las analizarán es un entorno de pruebas seguro; el uso de máquinas virtuales ayuda en la prevención de infecciones no deseadas durante este trabajo, por lo que el entorno de pruebas está constituido por 2 máquinas virtuales.

La primera será la máquina en la cual se liberarán las muestras maliciosas, que por definición de objetivos de este trabajo se encuentra implementada con el sistema operativo Windows 10, mientras que la otra máquina se encuentra utilizando una distribución de Linux conocida como Kali Linux, en su versión 2020.1, la cual es una distribución de sistema operativo dedicada a los aspectos de la ciberseguridad, por consiguiente, este S.O. ya viene precargado con varias herramientas que serán de utilidad en el análisis de malware, además de esto, he seleccionado este S.O. como sistema principal para ejecutar los análisis ya que he obtenido bastante experiencia como autodidacta usando dicha distribución.

Instalación de máquinas virtuales

Comenzaremos por la instalación de la máquina virtual en la cual se realizarán las infecciones controladas, que a partir de ahora será referida como "máquina víctima". Para realizar la instalación de la máquina víctima necesitaremos descargar los archivos necesarios para levantarla, afortunadamente Microsoft cuenta con una máquina virtual lista para cada uno de los diferentes softwares de virtualización, que en este caso se cuenta con el software VMware.

Nos podemos dirigir hacia el siguiente enlace https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/, el cual nos llevará a una página de Microsoft donde podemos descargar la máquina virtual empaquetada con Windows 10 Enterprise.

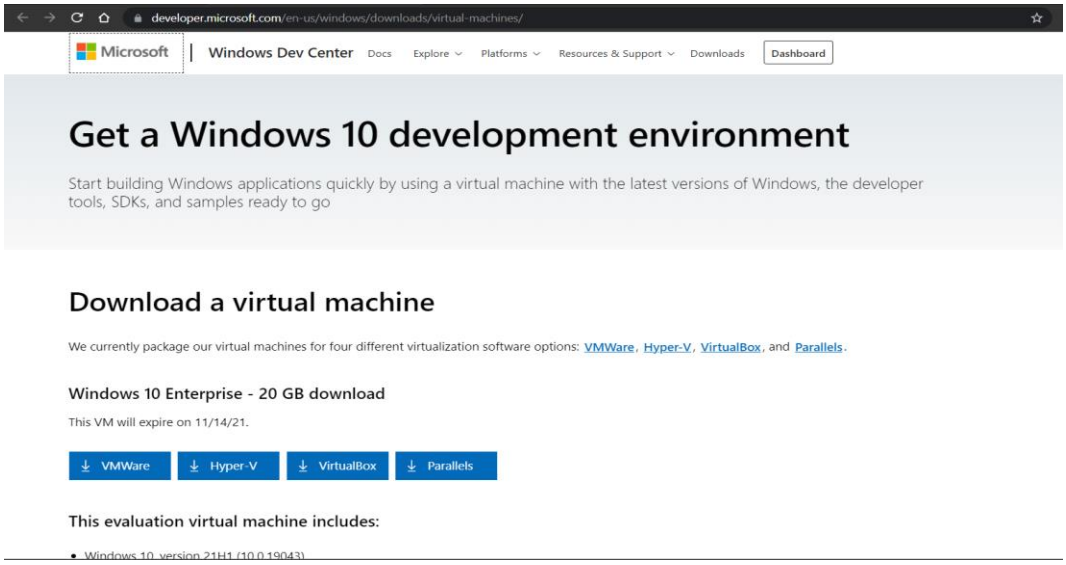


Ilustración 33: Página de descargar de VM con Windows 10

Una vez se ha descargado el archivo, podremos observar lo siguiente dentro del archivo comprimido tal como se puede ver en la *figura x.*

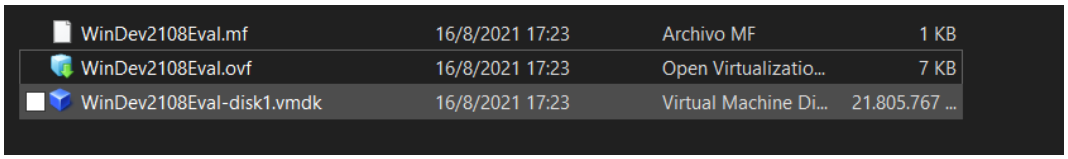


Ilustración 34: Archivos descargados para VMWare

Procedemos a abrir VMware Workstation, y dentro de la pestaña “Archivo” le damos a la opción “Abrir” y buscamos estos archivos recién descargados, y procuramos seleccionar aquel con la extensión *.ovf* e inmediatamente, aparecerá una barra de progreso sobre la importación del archivo hacia nuestro entorno de virtualización.

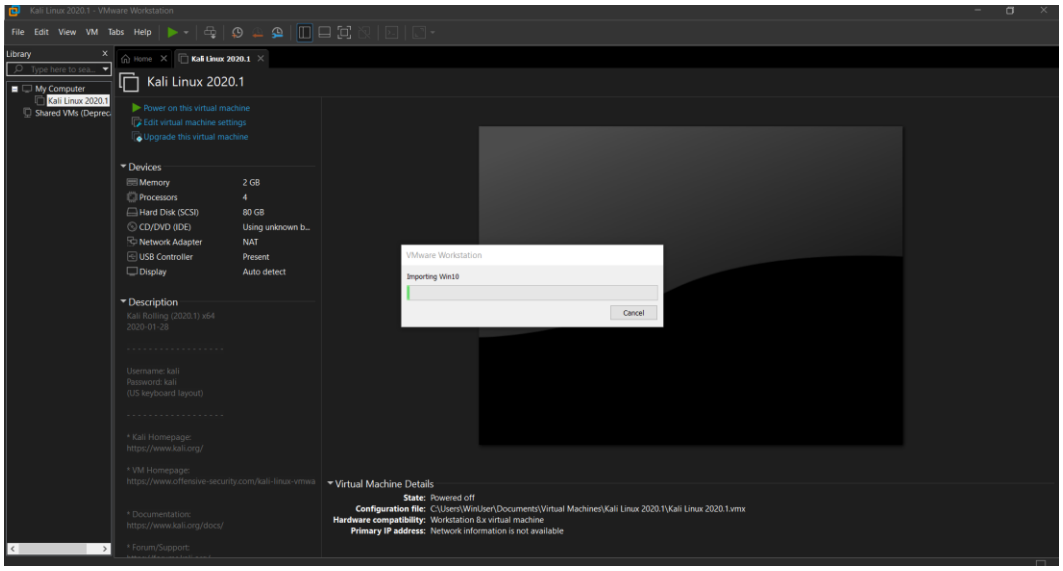


Ilustración 35: Proceso de importación de VM en VMWare

Una vez termina este proceso podemos observar a la máquina virtual configurada y lista para realizar su configuración inicial.

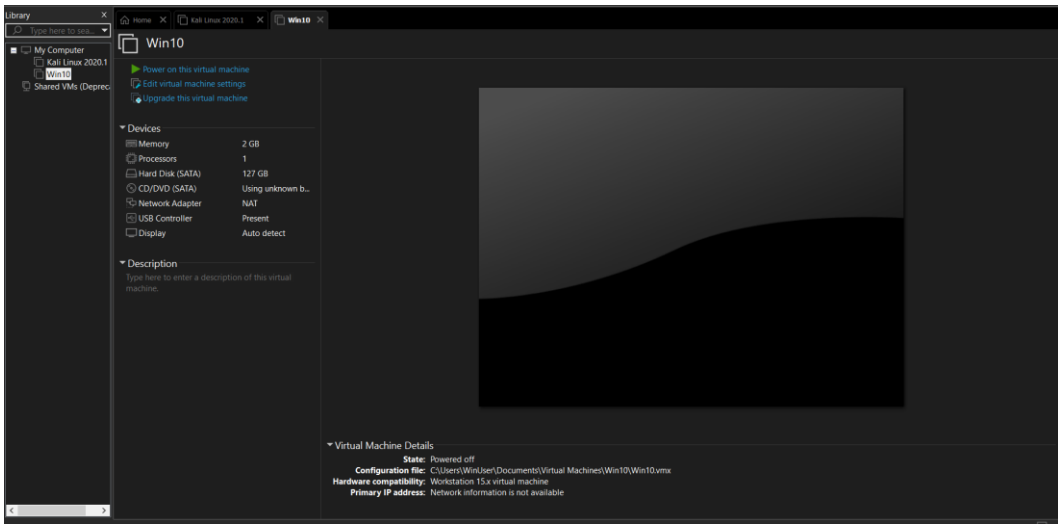


Ilustración 36: Instalación de VM completa

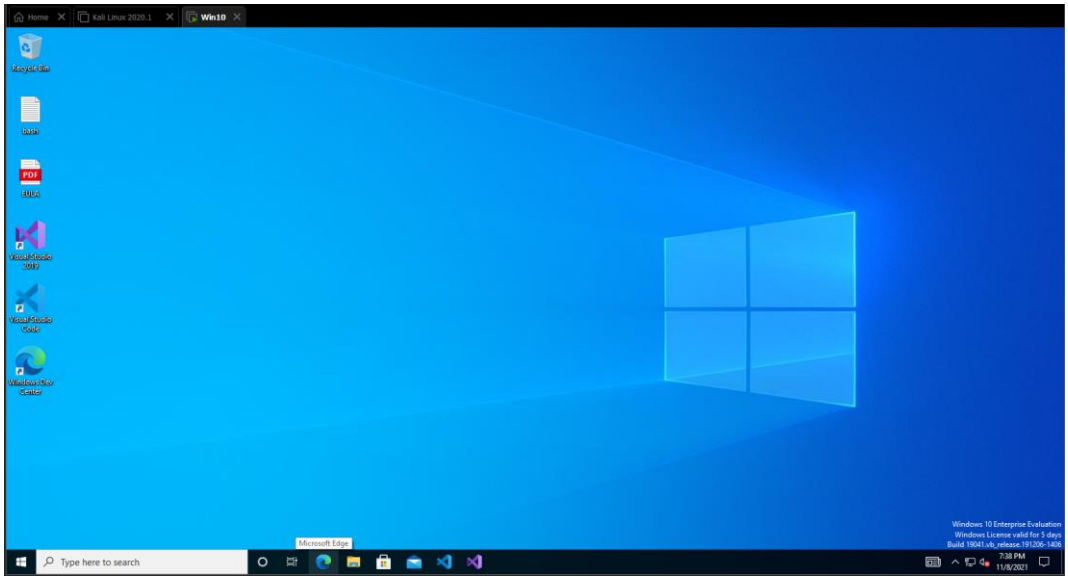


Ilustración 37: VM levantada y en ejecución

Dado a que las piezas modernas de malware cuentan ya con maneras de detectar que se encuentran dentro de un ambiente de análisis y así complicar de mayor manera la tarea de estudiarlas, el entorno virtual deberá simular lo más posible un entorno real, por lo que esta máquina contará con una asignación de recursos mayores a los que usualmente se suelen asignar a este tipo de máquinas virtuales que por lo general es el mínimo requerido para su correcto funcionamiento.

De manera adicional a la asignación de recursos, otro elemento para disfrazar el entorno virtualizado es el de la instalación de software de terceros ya que los programas maliciosos se encargan de ver ciertos programas para poder discernir si es un entorno de análisis o si se encuentra dentro de una computadora real que será su nueva víctima. Algunos de los programas instalados en la máquina víctima son: Adobe Reader, Google Chrome, aTube Catcher, etc. Todos estos fueron seleccionados de manera arbitraria con la finalidad de hacer parecer el entorno virtual lo más posible a mi entorno anfitrión.

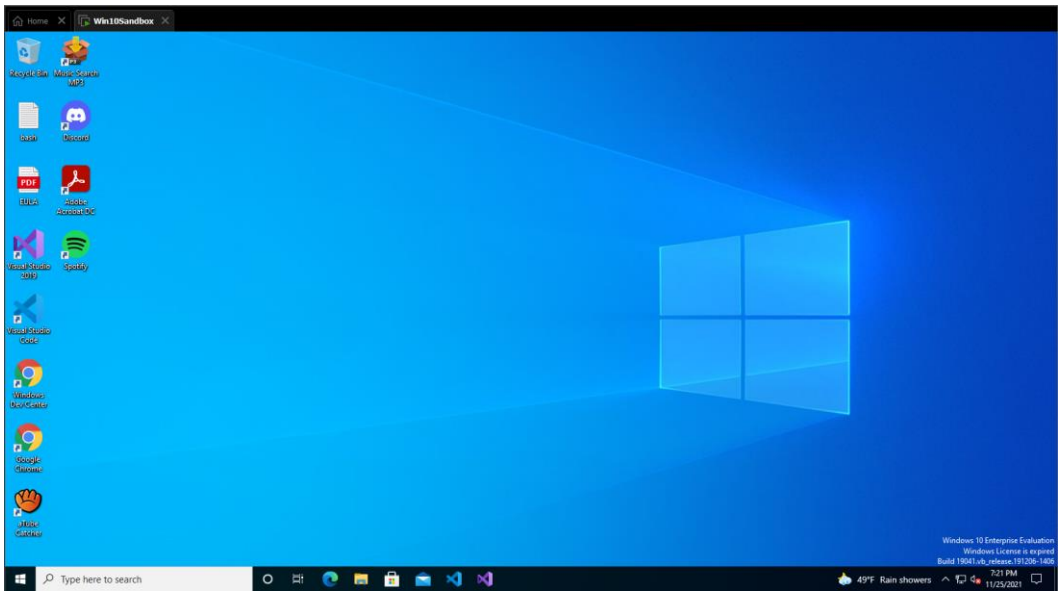


Ilustración 38: VM principal con programas básicos instalados

Otra cosa que puede ser utilizada para determinar si un programa malicioso reconoce si está dentro de un entorno virtualizado es si encuentra lo que son las herramientas de virtualización dentro de la máquina virtual, por lo que se recomienda omitir la instalación de estas.

El paso final por realizarse para contar con la máquina víctima lista es la de deshabilitar todos los servicios de protección que se encuentran corriendo por defecto, tal como lo es Windows Defender.

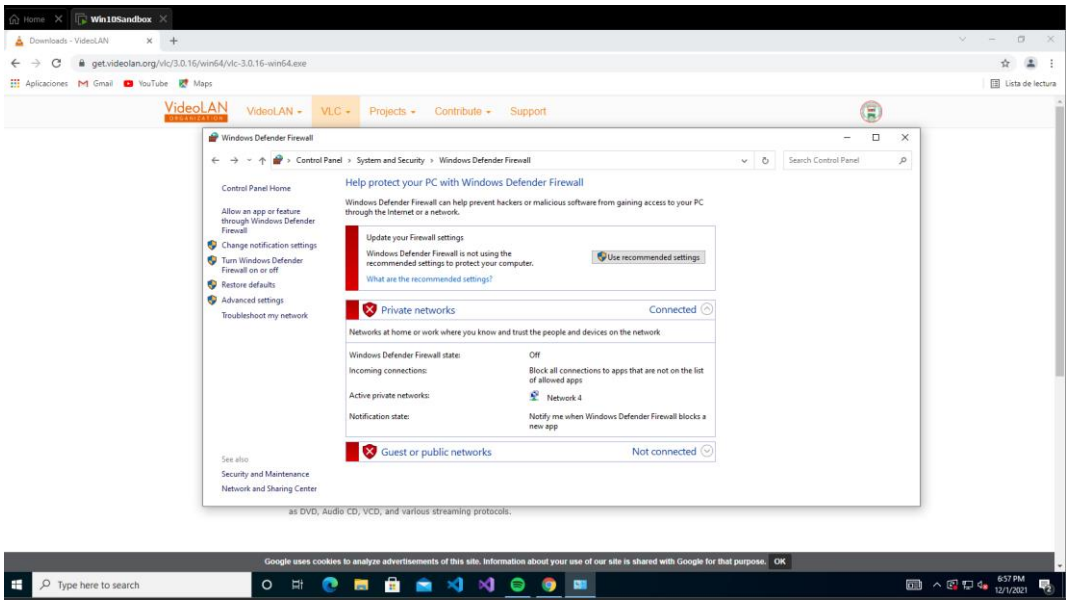


Ilustración 39: Proceso de desactivación de Windows Defender

Una vez se ha levantado la máquina víctima, se debe configurar a la máquina principal que actuará como centro de mando en este desarrollo. En este caso yo ya cuento con una máquina virtual Kali Linux con la que he venido trabajando dentro de diferentes actividades a lo largo de los últimos años. Por lo que explicaré los

puntos más importantes en caso de buscar el levantamiento desde cero de esta máquina principal. Afortunadamente, el equipo de Kali ha generado varias opciones para poder contar con este sistema operativo de manera bastante sencilla que facilitan abismalmente el proceso de instalación de este.

El primer paso para el levantamiento del entorno principal es la obtención de la obtención del instalador de Kali. Dentro de su portal existen varias opciones para la obtención del sistema operativo, entre las cuales están, la imagen de disco (.iso) máquinas virtuales listas para usarse, versión móvil, entre otros.

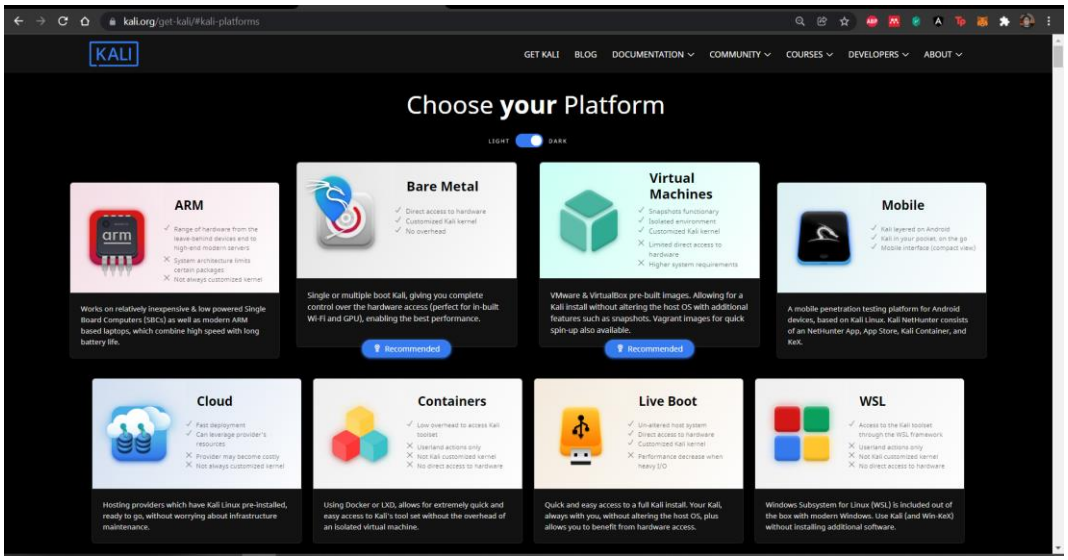


Ilustración 40: Página web de descargas de Kali Linux

En el caso de este trabajo, la opción a seleccionarse será la de máquina virtual, y se escogerá la opción para VMware de 64 bits, la cual ya se encuentra configurada y con ajustes por defecto que pueden ser modificados de acuerdo a las necesidades del usuario.

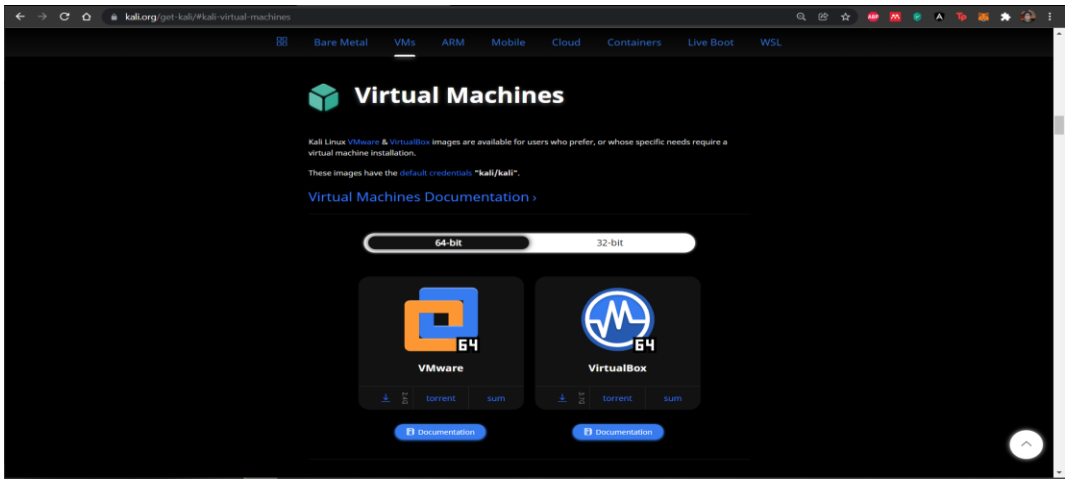


Ilustración 41: Opciones de descarga de VM con Kali Linux

Una vez se ha descargado la máquina virtual, nos dirigimos al software de virtualización de nuestra preferencia, en mi caso, será VMware Workstation, y seleccionamos la opción “Open”. Una vez ahí, navegamos hacia donde los archivos previamente descargados se encuentran, y seleccionamos aquel con la extensión “.ovf” y una vez hecho esto, el virtualizador comenzará el proceso de importación de esta máquina virtual, tal y como se llevó a cabo cuando se levantó la máquina virtual con Windows 10.

Tras estos sencillos pasos, la máquina principal está lista dentro del software de virtualización y el usuario ahora es capaz de modificar los recursos asignados a la misma. Tras realizar la asignación de recursos que se requiera, la máquina principal está lista para su uso.
En el caso de requerir tener Kali Linux como sistema operativo principal, o realizar la instalación de su versión móvil, puede referirse a la documentación oficial de Kali la cual contiene instrucciones para cada una de las instalaciones, los cuales se encuentran en el siguiente enlace: https://www.kali.org/docs/

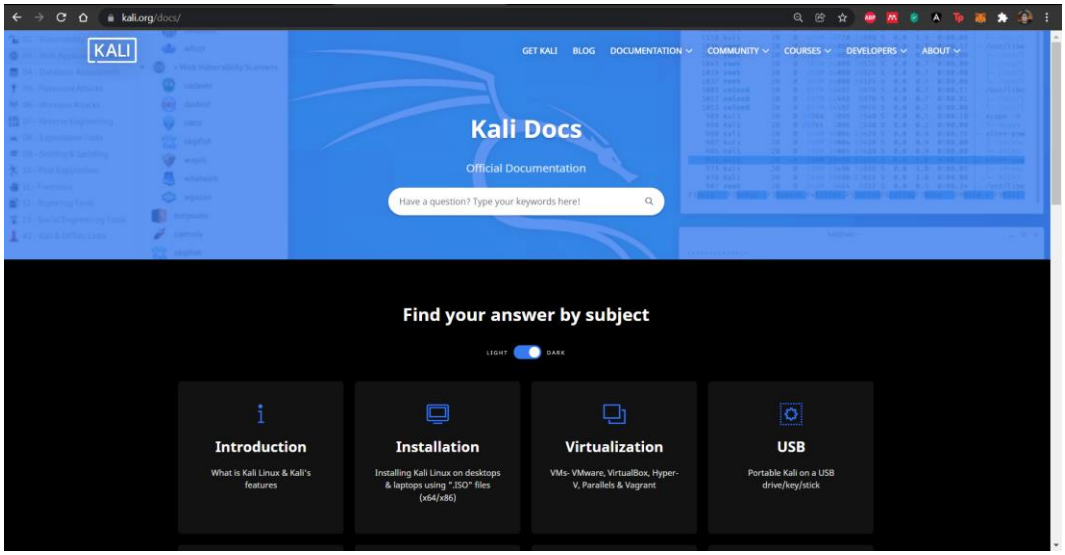


Ilustración 42: Página de inicio de la documentación de Kali Linux

Con los entornos levantados, el siguiente paso es la descarga de las muestras de malware, la cual se llevará a cabo visitando los diferentes repositorios de muestras que fueron listados durante el capítulo 3.

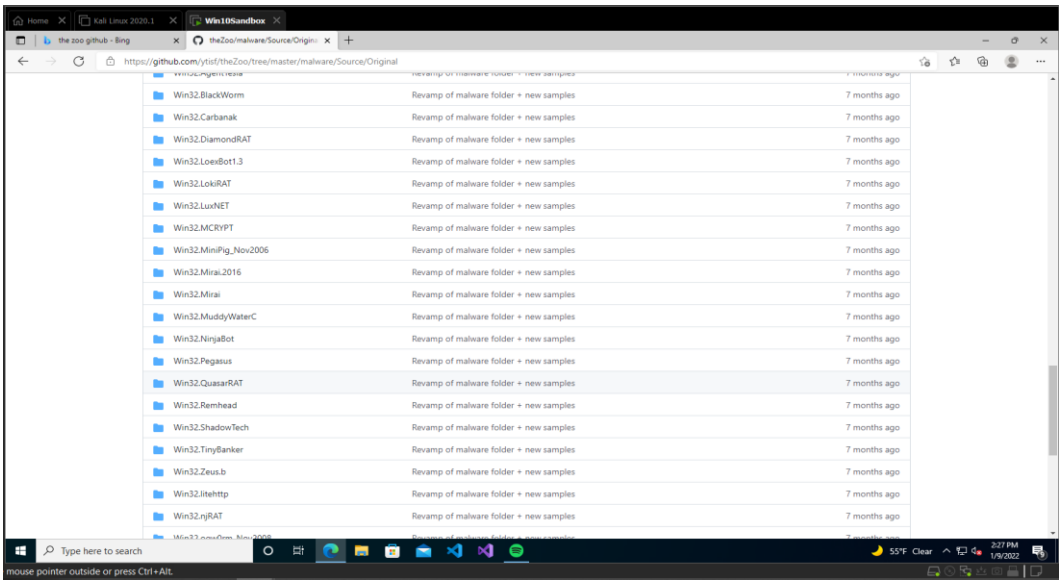


Ilustración 43: Verificación de disponibilidad de muestras dentro de TheZoo

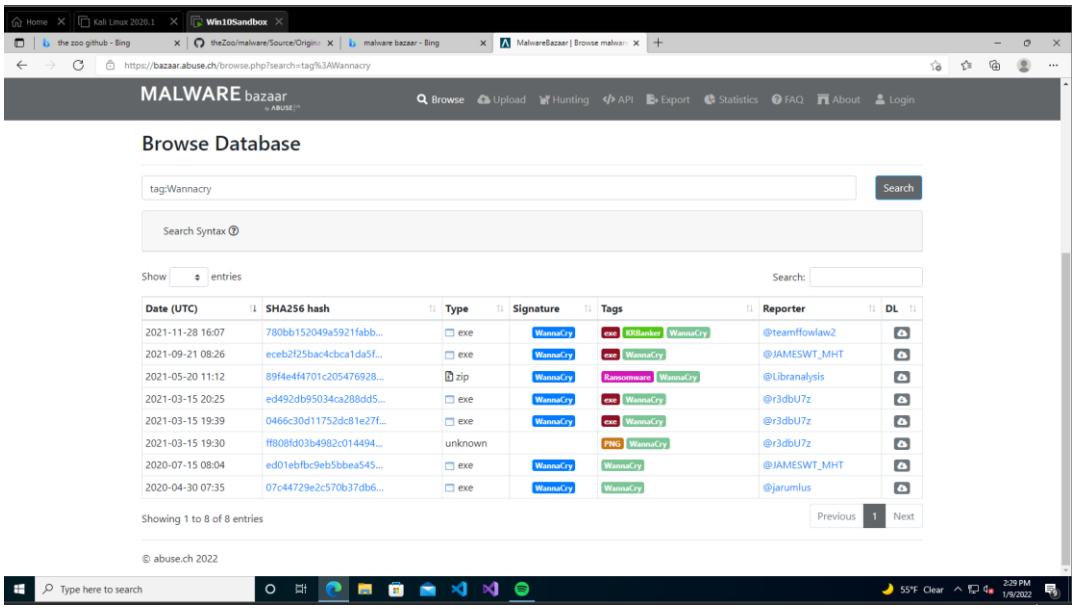


Ilustración 44: Verificación de disponibilidad de muestras dentro de MalwareBazaar

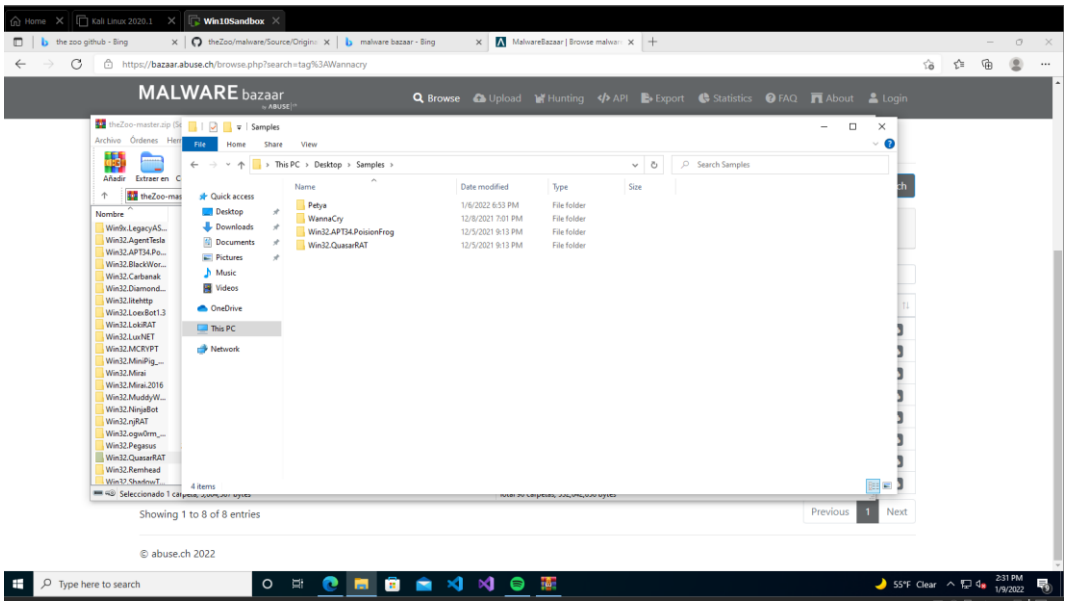


Ilustración 45: Muestras descargadas dentro de la máquina víctima

Una vez que ya se tienen ambas máquinas configuradas, es momento de aislar el entorno de pruebas, con lo que, en este punto, lo más seguro es aislar a la máquina víctima de cualquier tipo de conexión exterior, lo que significa impedir que esta posea conexión a internet.

Para evitar que la máquina víctima tenga cualquier tipo de conexión de internet, se deberá hacer una configuración de red dentro de VMware. Para realizar esto, es posible dirigirse hacia las configuraciones de la máquina virtual, se selecciona el adaptador de red, y dado el caso de este software de virtualización, lo que opté por realizar es deseleccionar la opción “ conectar durante el encendido” y además de esto, cambiar el tipo de conexión a “Host-only”.

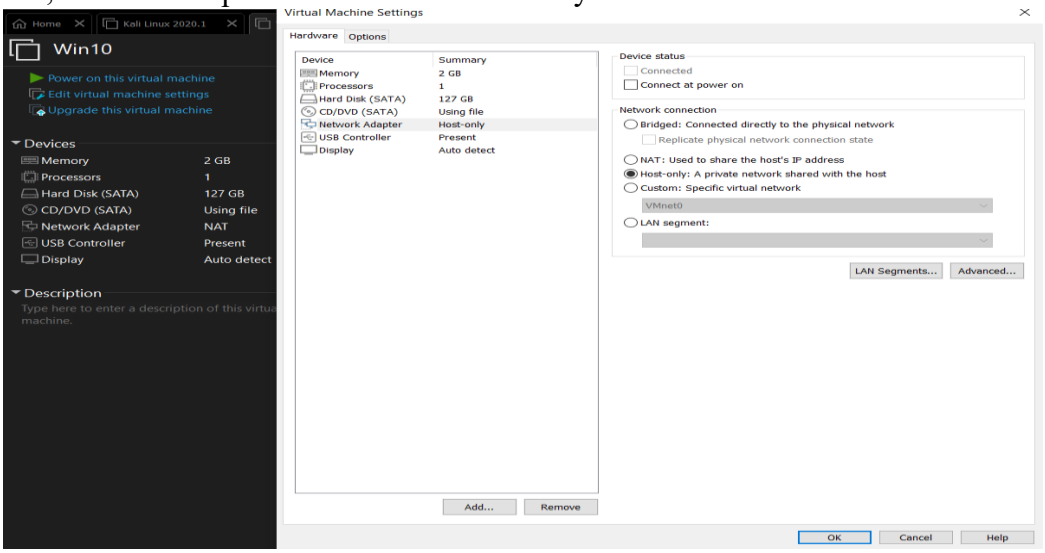


Ilustración 46: configuraciones de seguridad para la VM víctima

Finalmente, para tener un sistema completamente aislado de la máquina anfitrión, la última configuración pendiente de realizarse es la de deshabilitar la opción de

compartir el portapapeles y la opción de carpetas compartidas. Una vez hecho esto, la máquina víctima estará totalmente aislada del entorno anfitrión y el riesgo de una infección en dicho entorno se ve minimizado.

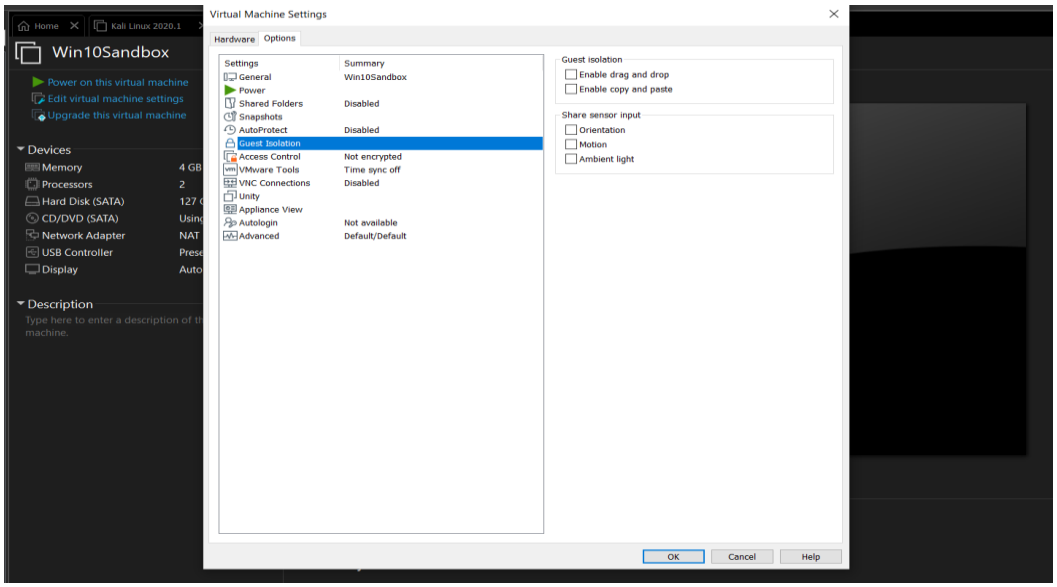


Ilustración 47: Configuraciones de entorno compartido para la VM víctima

Tras el aislamiento de la máquina víctima, se debe levantar una conexión entre la máquina principal y la máquina víctima, con la única finalidad de que se puedan enviar archivos entre sí, para evitar que la maquina anfitrión sea la mensajera y pueda llegar a infectarse de forma accidental. Para lograr esto se genera una subred con un número limitado de direcciones IP, y se coloca a ambas máquinas dentro de dicha subred. Como paso final para realizar la transferencia, opté por realizarlas mediante una conexión SSH por lo que se procedí a realizar la activación del servicio y abrir el puerto 22 para que se pueda realizar la conexión.

Implementación de herramientas

Una vez la máquina principal se encuentra funcionando correctamente, el siguiente paso es la implementación de las herramientas de análisis que serán usadas dentro del análisis de malware. Como aclaración, en las siguientes subsecciones no se encuentran listadas las herramientas Intezer Analyze ni VirusTotal, ya que esta es un aplicativo web accesible desde cualquier navegador; el resto de las herramientas requieren que se efectúen ciertas configuraciones para ser utilizadas en un ambiente local.

Instalación y configuración de Cutter

Para la obtención de la primera herramienta podemos dirigirnos al enlace https://cutter.re/, el cual nos llevará a la página principal del proyecto Cutter.

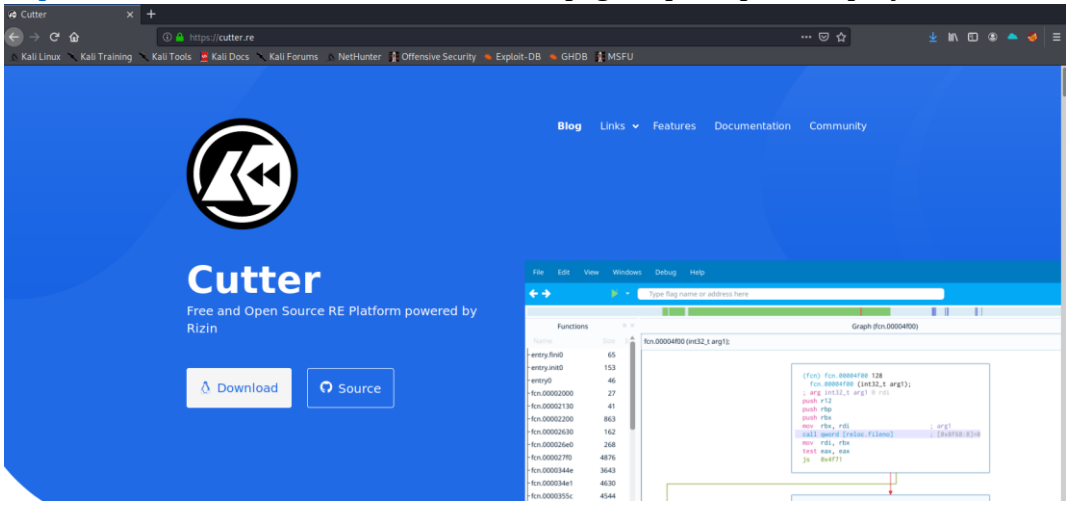


Ilustración 48: Página de inicio de Cutter

Una vez en el sitio web podemos directamente descargar el archivo necesario e inmediatamente tendremos una ventana emergente la cual nos preguntará si queremos guardar el archivo, el cual tiene un peso de 122 MB.

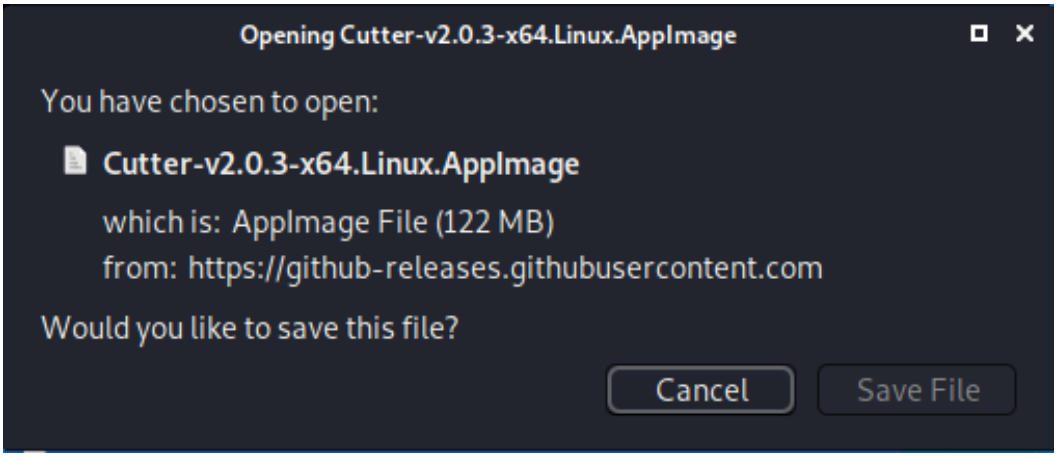


Ilustración 49: Descarga de archivo ejecutable de Cutter

Una vez tengamos listo el archivo, podemos guiarnos con la documentación oficial de Cutter, la cual nos indica que el archivo recientemente descargado debe convertirse en un archivo ejecutable, por lo tanto, requiere de permisos adicionales los cuales son otorgado mediante el comando *chmod +x <nombre_archivo>* tal como se puede observar en la *figura x*. Como resultado podemos ver que en el momento que los archivos dentro del directorio correspondiente son listados, el archivo que nos interesa se encuentra resaltado de color verde, lo que significa que ahora es un archivo ejecutable.

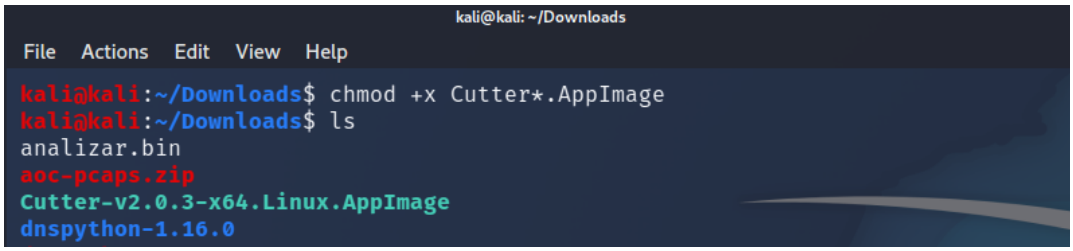


Ilustración 50: Otorgando permisos de ejecución a archivo de Cutter descargado

Con esto podemos ejecutar el comando *./Cutter-v2.0.3-x64.Linux.AppImage*, lo que ejecutará al archivo, y nos permitirá realizar la configuración inicial, la cual consiste de la selección de apariencia e idioma.

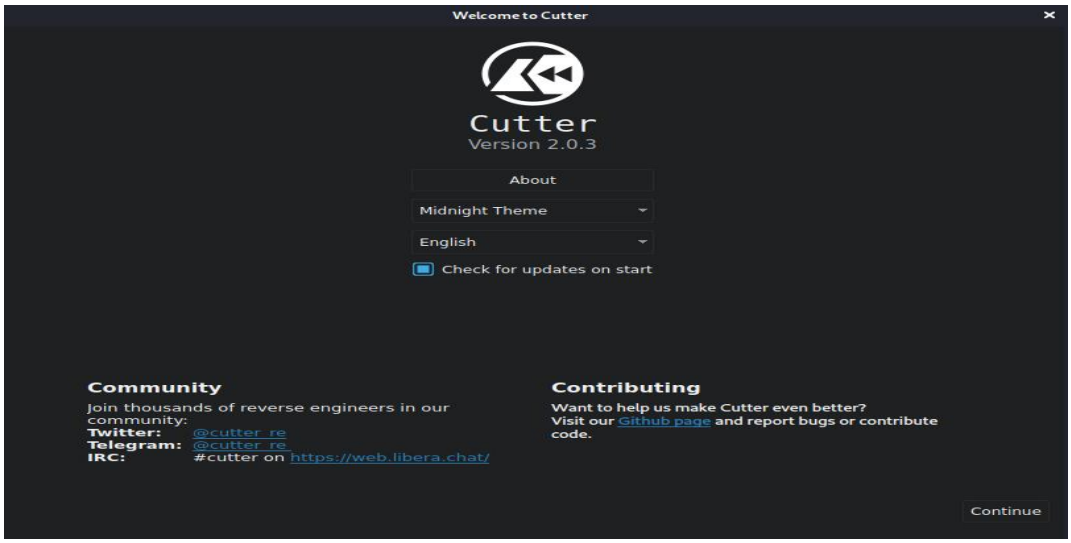


Ilustración 51: Pantalla inicial de Cutter

Una vez se han configurado con las preferencias deseadas, al dar continuar, la herramienta está lista para seleccionar el archivo a ser analizado y así poder darle uso a la herramienta.

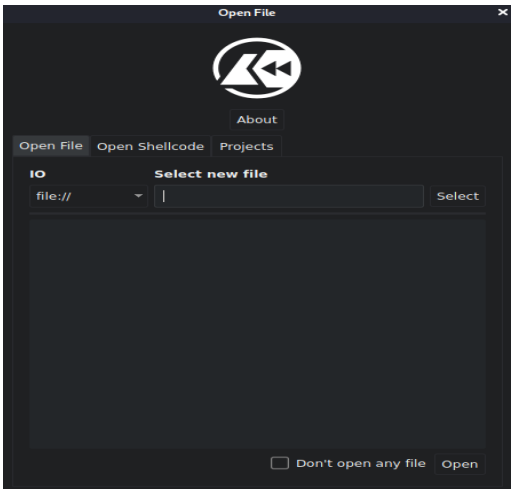


Ilustración 52: Selección de archivos desde Cutter

Instalación y configuración de Ghidra

Para la instalación de Ghidra, se puede descargar la última versión disponible la cual es la 10.0.4, directamente desde su repositorio público en Github, y tal las instrucciones señaladas en su documentación, simplemente se requiere extraer los archivos descargados y ejecutarlo mediante un comando de consola *./ghidraRun*,el cual inmediatamente ejecutará el programa con sus componentes base y nos mostrará la ventana de GHidra Help en conjunto con el explorador de proyectos.

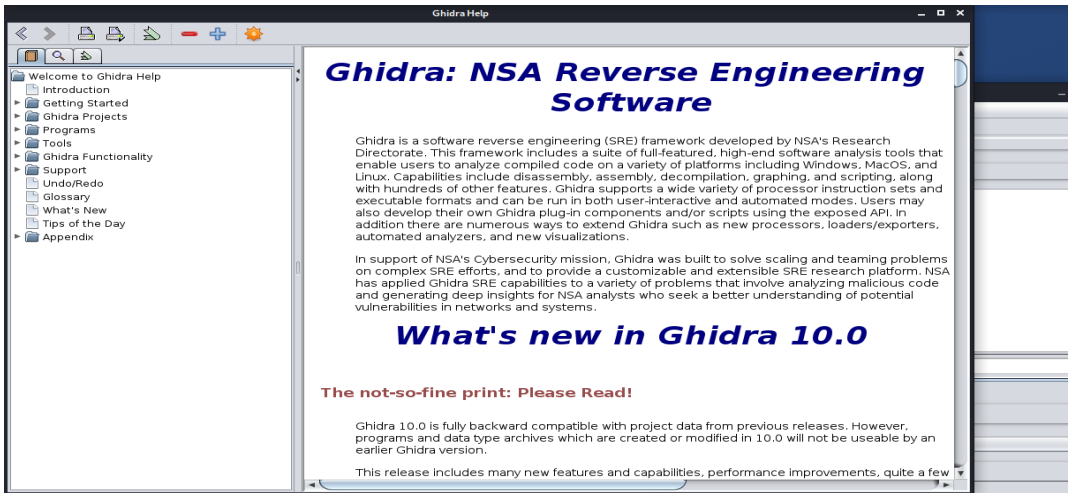


Ilustración 53: Pantalla inicial de Ghidra

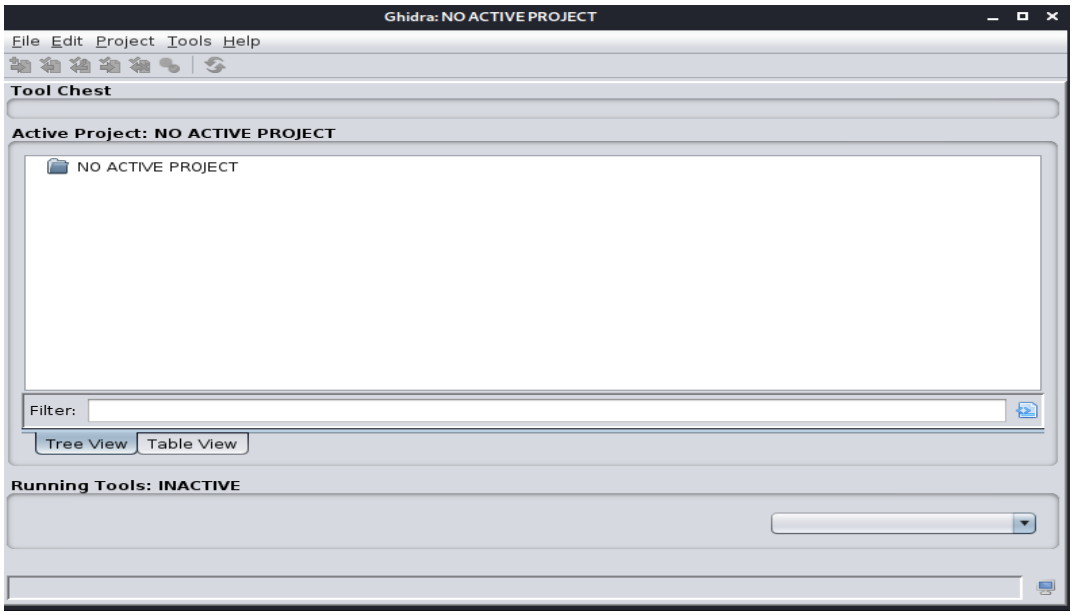


Ilustración 54: Pantalla de listado de proyectos activos en Ghidra

Instalación y configuración de Volatility

Para la instalación de Volatility, se debe optar cual versión se desea usar, ya que en febrero del 2020 Volatility Foundation liberó Volatility3 la cual es una migración

de si primera versión escrita con Python2 a la siguiente versión de este lenguaje la cual es Python3. Para el desarrollo de este trabajo, opté por indagar la nueva versión de la herramienta.

Los pasos de instalación se encuentran dentro de su repositorio público albergado en GitHub al cual se puede acceder mediante el siguiente enlace https://github.com/volatilityfoundation/volatility3. Para hacer uso de programas albergado en esta plataforma de versionamiento, el primer pasó será la clonación del repositorio.

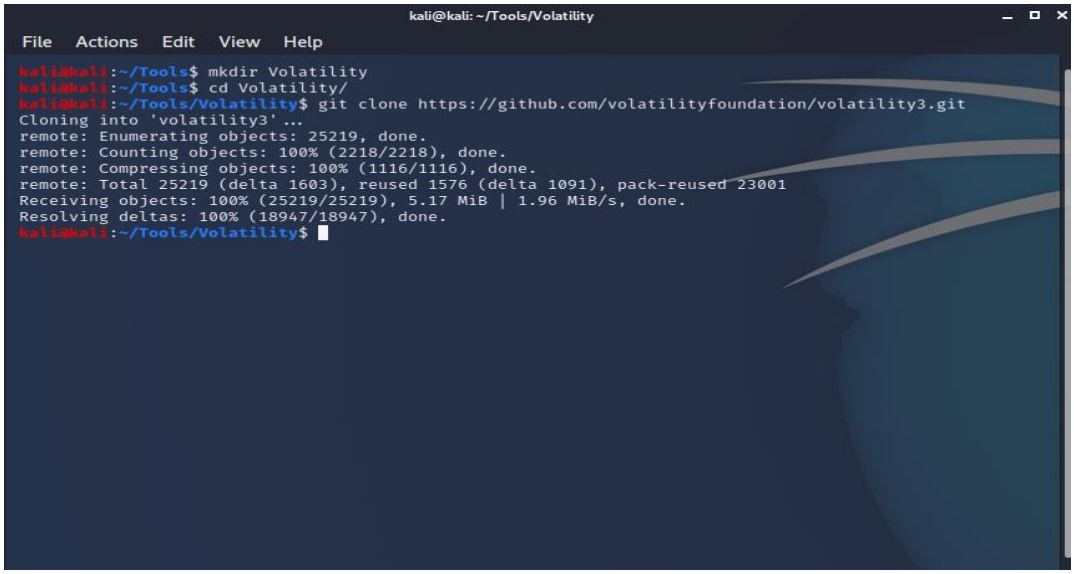


Ilustración 55: Clonación de Volatility dentro de la VM principal

Una vez ya se cuenta con el repositorio clonado dentro de nuestra máquina, se procede a continuar con la guía proporcionada dentro del repositorio, por lo que a continuación se deben instalar todos los requerimientos para su correcto funcionamiento. Dentro de las instrucciones se nos permite escoger 3 formas de instalación, 2 de ellas instalarán los requerimientos mínimos para un funcionamiento básico, pero en este trabajo se optó por la instalación completa.

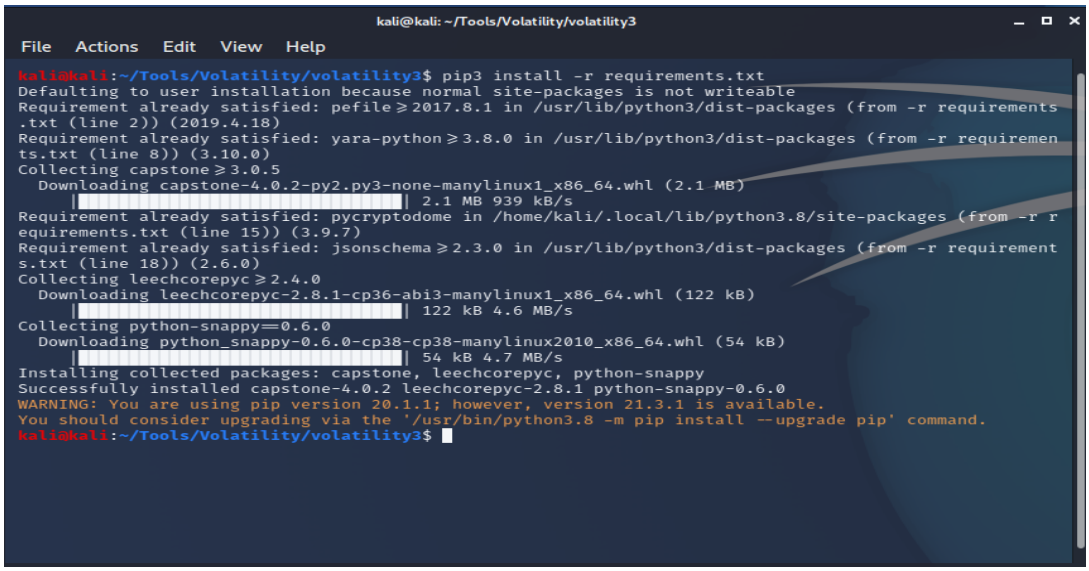


Ilustración 56: Instalación de requerimientos para ejecutar Volatility

Con todos los requerimientos instalados, se puede comprobar que Volatility puede ser usado mediante el comando *python3 vol.py -h* el cual, si la instalación se ejecutó exitosamente, nos devolverá la lista de argumentos y opciones que podemos ejecutar dentro de Volatility, tal como se ve en la *figura 57.*

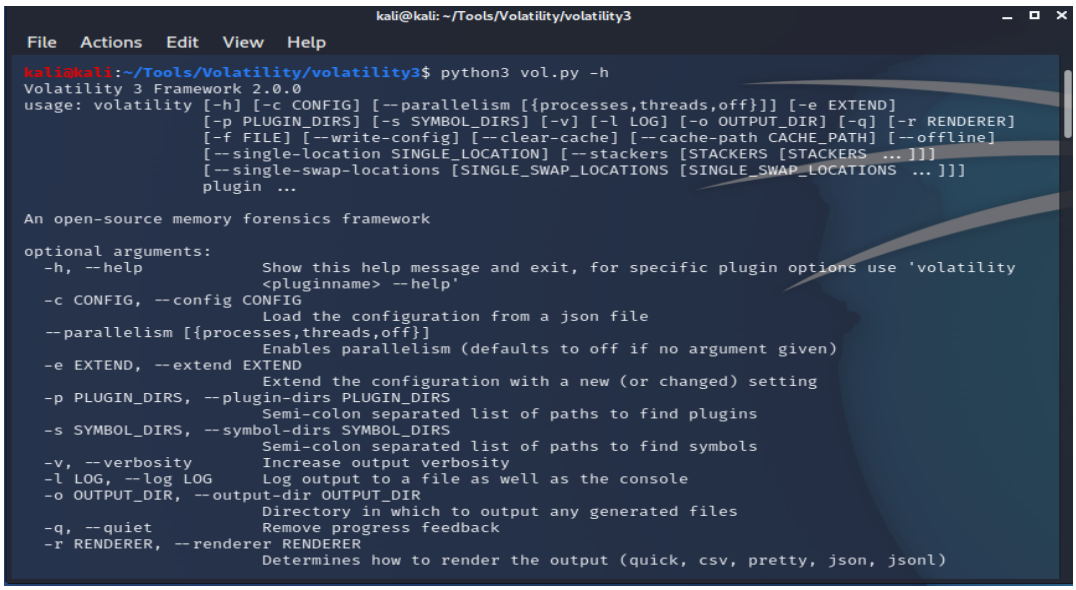


Ilustración 57: guía de argumentos opcionales para la ejecución de Volatility

Procedimiento de infección controlada

El procedimiento que se llevará a cabo para realizar la observación del comportamiento de cada una de las muestras será el siguiente:

1. Tras tener el entorno víctima levantado y equipado, el primer paso será tomar un “snapshot” de la máquina virtual, lo que significa que se guardará el estado actual de la máquina virtual.

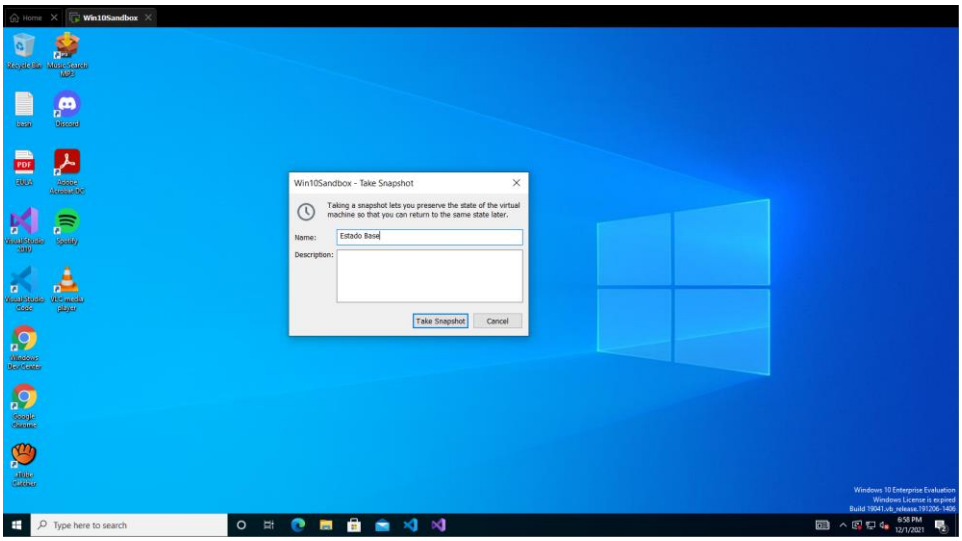


Ilustración 58: Toma del snapshot

2. Se selecciona la muestra a examinarse e intencionalmente se ejecutará el archivo malicioso con el fin de poder ver su comportamiento.

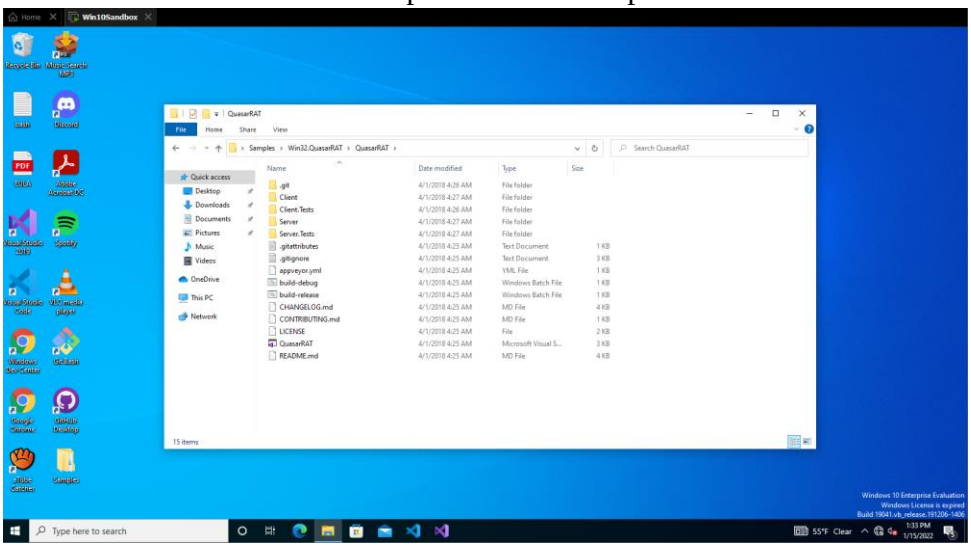


Ilustración 59: Selección e inspección de muestra de malware

3. Con el malware habiendo infectado el entorno de pruebas, se procederá a realizar la toma de muestras o directamente el análisis del entorno según corresponda.

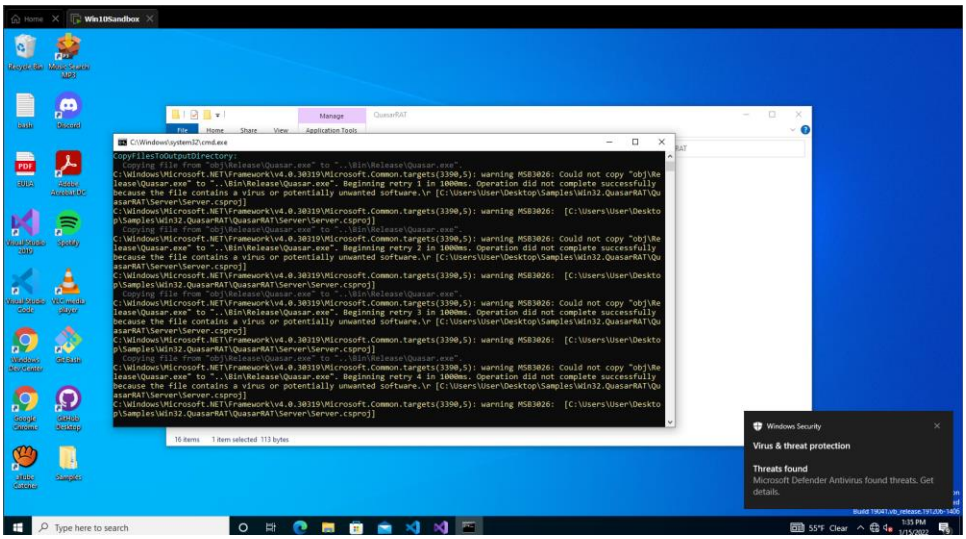


Ilustración 60: Proceso de infección intencional dentro de la VM víctima

4. Una vez se ha terminado el proceso con la muestra seleccionada, se procede a reestablecer a la máquina víctima a su estado previo a la infección, el cual ahora es posible gracias al snapshot tomado durante el paso 1.

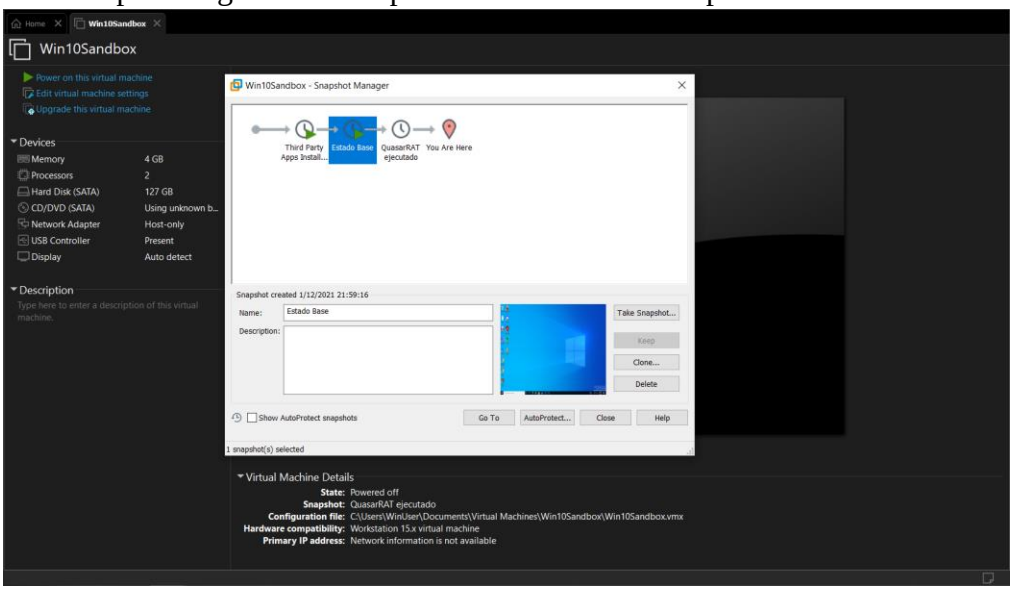


Ilustración 61: Selección de snapshot a reestablecer

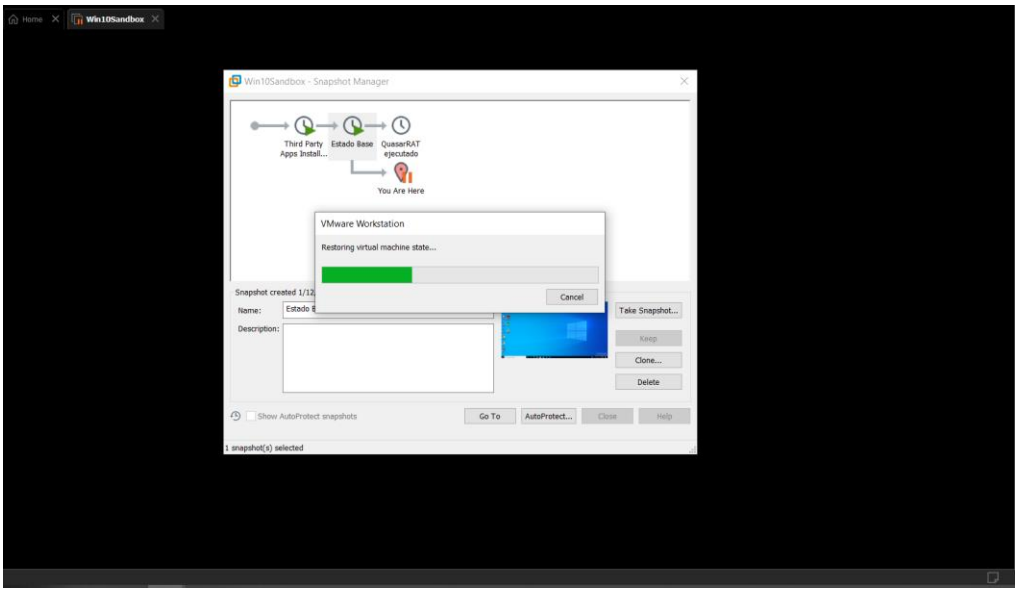


Ilustración 62: Proceso de restablecimiento de snapshot seleccionado

5. Se puede repetir el proceso desde el paso 2.

Explicación de modalidad de análisis

Para realizar el análisis de las muestras, lo he divido en 2 fases, las cuales son: estática y dinámica.

Durante la fase estática, el análisis está centrado en observar los archivos maliciosos previo a ser ejecutados, por lo que, en esta sección, haré uso de las herramientas que cuenten con desensambladores y descompiladores, y de ser que la muestra cuenta con algún archivo de código fuente, la lectura de este será mediante un editor de texto, el cual, en este caso, por familiaridad, seleccioné al editor Visual Studio Code. De igual manera en ciertos casos será necesaria la extracción de cadenas de caracteres o desofuscar código de forma manual; para cualquier efecto necesario, los dos lenguajes de programación que se usarán para intentar realizar dichas tareas serán Python y JavaScript.

Una vez terminada esta fase, en la fase dinámica, ejecutaré cada una de las muestras maliciosas, utilizando los pasos detallados en la Sección 4.3. Durante la ejecución de los programas maliciosos, las herramientas que se encargan de analizar el comportamiento y procesos en ejecución serán utilizadas a lo largo de esta fase.

En ambas fases, iré observando la capacidad que tienen estas herramientas para llegar a obtener IOC tras un proceso de análisis de cada una de las muestras, ya que, de igual manera, cada muestra seleccionada tiene su peculiaridad lo que puede derivar en la dificultad que está presente en ser analizada.

Sin mayor preámbulo, a continuación, se presentarán los resultados obtenidos tras el análisis de cada una de las muestras listadas en la Sección 2.1.

Análisis de herramientas

Tras los diferentes procesos de análisis de cada una de las muestras de código malicioso, que pueden ser hallados dentro de la sección de ANEXOS, se han evidenciado las diferencias entre cada una de las herramientas implementadas. A continuación, se detallará de forma resumida los resultados obtenidos en cuanto a las capacidades y/o funcionalidades corroboradas por mi parte durante los análisis de las muestras de malware, tiempos de respuesta, y capacidades de obtención de los IOC de cada una de las herramientas.

Tabla 1: Tabla comparativa de las herramientas tras análisis de muestras

Herramienta	***Tipo de herramienta***	***Funcionalidades***	***Tiempo de respuesta (promedio)***	***Tipo de obtención de IOC***
Ghidra	Disassembler	Descompilador	10”-15”	Manual
Cutter	Disassembler	Descompilador, depurador	10”-15”	Semiautomático
Volatility	Framework para análisis de volcados de memoria	Inspección y extracción de artefactos de memoria. Análisis de volcados de memoria, snapshots de máquinas virtuales, etc.	30”-2’ (con un volcado de memoria de 4GB)	Semiautomático
VirusTotal	Plataforma para análisis de archivos y URLs	Análisis simultáneo en múltiples motores de antivirus y servicios de listado de URLs y/o dominios bloqueados, análisis dinámico automatizado.	5”-10”	Automática
Intezer Analyze	Plataforma para análisis completo de archivos, URLs, endpoints y volcados de memoria	Análisis genético de muestras, ejecución dinámica, obtención de TTPs realizados por la muestra, conexión con VirusTotal.	20”-1’	Automática

Tras los diferentes análisis realizados a las 5 muestras de código malicioso, se obtuvieron los resultados señalados en la *tabla 1*, los cuales profundizaré en breve.

Para que la comparativa sea justa entre todos los aplicativos, se listaron cada una de las funcionalidades que presta cada uno de estos, con la finalidad de observar cuál de estos tiene una mejor gama de capacidades a comparación de los demás para así poder clasificar a la herramienta como "completa" dependiendo de si cuenta con las suficientes funcionalidades que requiere un informático forense.

En el caso de Ghidra, la herramienta de ingeniería inversa desarrollada y mantenida por la NSA de Estados Unidos, sus tiempos de respuesta pertenecen al intervalo más corto de la comparativa ya que su descompilador es realmente potente, pero lastimosamente contaba con ciertos inconvenientes con determinadas muestras. Las configuraciones por defecto de Ghidra especialmente para la memoria dinámica o como se lo conoce en inglés, heap memory, la cual realiza un alojamiento de espacio determinado de forma dinámica dentro de la memoria RAM y esta solo es liberada si se le indica que puede ser liberada o si el aplicativo concluye su proceso. En este caso, durante el análisis del archivo ejecutable para la muestra Glimpse, Ghidra no pudo procesar correctamente la muestra ya que lastimosamente se obtenía un error en tiempo de ejecución señalando que el tamaño de heap era insuficiente, por lo que se debió modificar el ejecutable de Ghidra para que sea capaz de analizar la muestra.

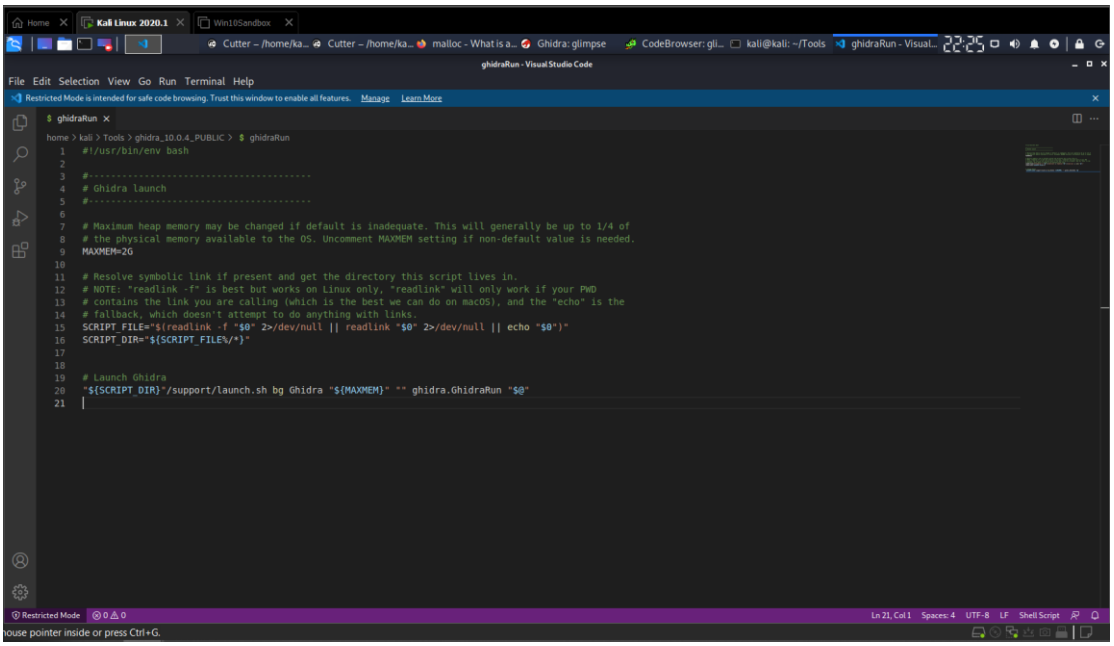


Ilustración 63: Modificación de configuración de Ghidra

Fuera de eso, la herramienta se comporta de buena manera, ofreciendo un sinnúmero de análisis que se pueden realizar al archivo seleccionado. De igual manera, se pueden ejecutar una amplia variedad de scripts precargados dependiendo de las necesidades del analista. Sin embargo, la herramienta no es la mejor en cuanto a la obtención de IOC por sí misma. Ghidra es una herramienta robusta para utilizar por parte de un usuario experimentado que reconoce patrones y es capaz de analizar una muestra

diseccionada a detalle. La herramienta como tal no es capaz de analizar y devolver un listado de los IOC de la muestra inspeccionada. Las bondades de la herramienta es que es capaz de devolver el código ensamblador y hasta cierto punto el código fuente de la muestra, las librerías que esta utiliza, las funciones, los sectores de memoria a los que accede, lo que le permite al usuario experimentado ir analizando cada uno de los elementos y realizar una obtención manual de los IOC para ser implementadas en herramientas de detección posteriormente.

Con lo anteriormente expuesto, se puede determinar que Ghidra es una herramienta para ser utilizada en un estudio minucioso de una muestra de malware que permitirá observar casi todos sus elementos, como si se pusiera una muestra de ADN bajo un microscopio, pero la clasificación de qué elementos pueden ser considerados IOC, quedará a criterio de quién utilice la herramienta para identificar los IOC.

En el caso de Cutter, la herramienta tiene bastantes similitudes con Ghidra, pero esta puede ser considerada como una versión más liviana de un descompilador, ya que no cuenta con scripts precargados y su preprocesamiento no es tan completo con el de la herramienta previamente mencionada. Permite la observación de todas las cadenas de caracteres que puede hallar dentro del archivo examinado, da un resumen de la muestra, tal como se evidencia en la *figura 64.* Se puede observar que nos comenta el lenguaje en el que se escribió, el formato del archivo, los bits de este, la arquitectura para la que fue compilada, el sistema operativo para el que fue desarrollado, etc.

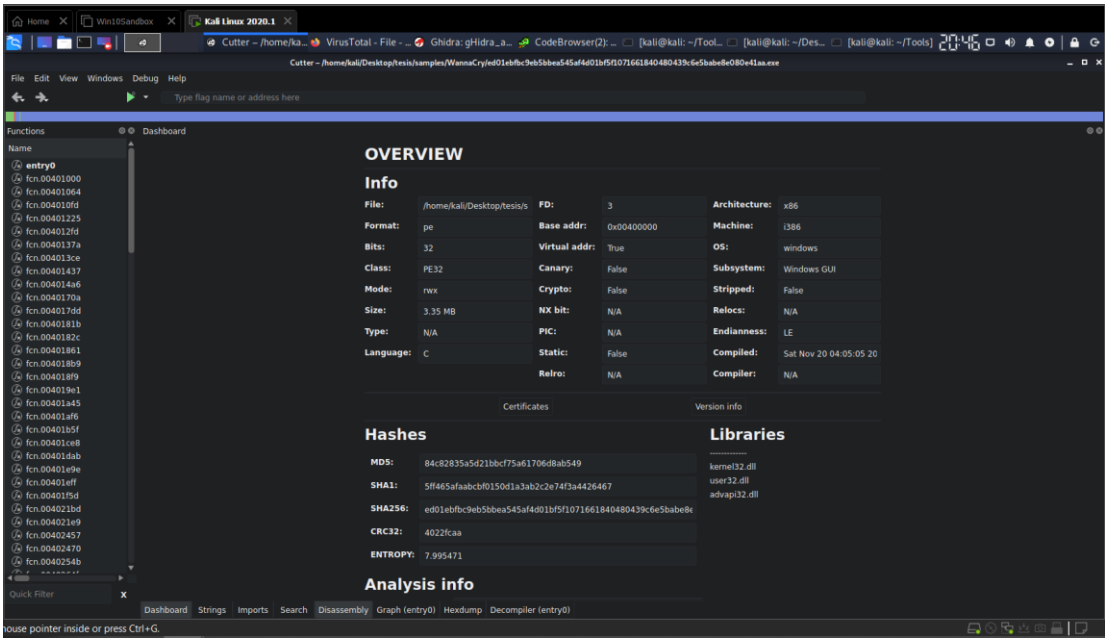


Ilustración 64: Resumen de información sobre muestra WannaCry

El tiempo de respuesta es de igual manera rápido, y provee una respuesta gráfica la cual es altamente apreciada desde el punto de vista de un usuario poco experimentado. Lo que diferencia a Cutter de Ghidra durante este trabajo fue la diferencia en capacidad de obtener lo que el archivo examinado importaba para realizar su cometido, y no solo eso, sino que era capaz de señalar aquellas importaciones que resultaban inseguras, según la herramienta, tal como se puede ver en la *figura 65*. A manera general, Cutter

tiene cierta mejora dentro de la obtención de IOC dentro del análisis ya que gracias a su preprocesamiento puede detectar ciertos artefactos anómalos o inseguros dentro del archivo analizado. Además de que gracias a su interfaz gráfica la herramienta facilita la tarea del analista al momento de realizar un análisis estático breve.

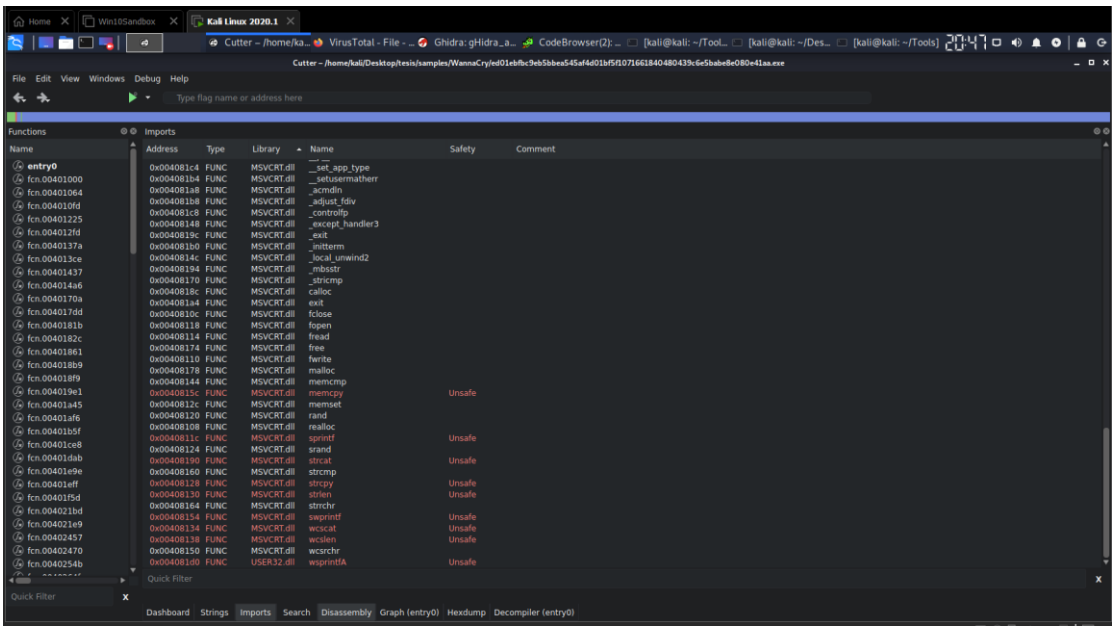


Ilustración 65: Detección de módulos inseguros importados en muestra WannaCry

Dentro de las herramientas que fueron capaces de obtener por sí solas los IOC nos encontramos con VirusTotal y Intezer Analyze. Ambas herramientas se clasifican como plataformas para realizar un análisis de malware completo dando al analista varias opciones y tipos de resultados que aceleran el proceso de la obtención de IOC para su posterior uso en técnicas y herramientas de detección.

La ventaja con la que ambas herramientas cuentan es que estas son plataformas web, por lo tanto, esto posibilita su uso desde cualquier tipo de dispositivo que cuente con una conexión a Internet. Fuera de la ventaja que poseen en conjunto, cada una de estas herramientas cuentan con sus características particulares.

En el caso de VirusTotal, la plataforma se encarga de realizar la inspección de las muestras y/o URLs enviadas a análisis con varios servicios de listas de bloqueos de URL o dominios, y aproximadamente 70 motores de antivirus, tales como Kaspersky, Avast, McAfee, etc. La herramienta cuenta con varias opciones para la carga de muestras, que comprenden a su página web, extensiones de navegador, aplicaciones de escritorio, y una API.

Además de su característica de compartir resultados con sus colaboradores comerciales, una vez que VirusTotal terminal el análisis de una muestra, esta se vuelve disponible para toda la web, lo que quiere decir que cualquiera con la URL de los resultados de una determinada prueba puede acceder a estos y revisarlos por sí mismo, si el lector lo desea confirmar, dentro de los anexos se encuentran las URL de cada una de las muestras subidas a VirusTotal, las cuales ya están disponibles para su observación.

Gracias a esta disponibilidad al público, dentro de la plataforma, sus miembros son capaces de dejar comentarios en los resultados y se tiene la opción de votar en si la muestra de resultado observado es maliciosa o no, con esto se logra un mayor entendimiento y entrenamiento en la comunidad para diferenciar una muestra maliciosa de un falso positivo.

Los resultados son entregados al usuario dentro de una página que nos muestra el resultado de la votación de la comunidad, el número de motores de antivirus que han marcado a la muestra como maliciosa, y la descripción a detalle dentro de 5 pestañas las cuales son: *Detección, Detalles, Relaciones, Comportamiento, & Comunidad.*

Dentro de la pestaña denominada como *Detección*, se nos entrega el listado de todos los motores de antivirus que fueron utilizados durante el análisis de la muestra, la respuesta de cada uno la cual puede variar de motor a motor, ya que algunos dan directamente el nombre que cada uno le ha dado a la amenaza que detectan en la muestra. A partir de la pestaña *Detalles*, se vuelve posible la comparación de esta herramienta con las demás en cuanto a la obtención de IOC. Dentro de esta pestaña se da la información del archivo y/o URL como tal, su hash en los diferentes formatos, el tipo de archivo, su tamaño, fechas de creación, registro de la primera vez que se la vio libre, además de la fecha de la primera carga de la muestra a la plataforma, la fecha más reciente, y la última fecha en la que se realizó su análisis.

Algo a recalcar, que a comparación de las herramientas como Ghidra o Cutter, VirusTotal nos provee de la información general del archivo, pero este no nos dice cuál es el sistema operativo en el que se puede ejecutar la muestra. Otro apartado que llama la atención dentro de esta pestaña es aquel bajo el nombre de “Imports” el cual, de similar manera que Cutter, nos muestra lo que la muestra importa para realizar sus diferentes funciones, pero la diferencia en esta herramienta es que no nos señala aquellas importaciones que se las puede considerar como inseguras, pero es capaz de darnos un listado de todas las funciones disponibles dentro de cada una de las librerías listadas dentro de este apartado, tal como se puede observar en la *figura 66*.

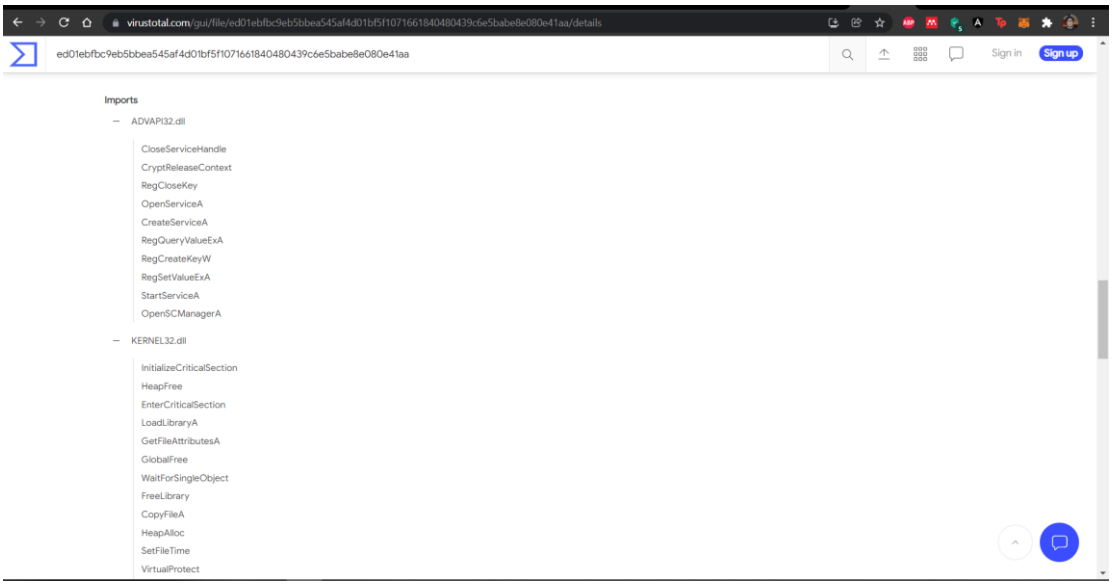


Ilustración 66: Análisis de WannaCry en VirusTotal. Listado de funciones de librerías

Pasando a la siguiente pestaña, Relaciones, se nos entrega todo un listado de recursos que se encuentran relacionados con la muestra analizada. En este apartado podemos observar dominios contactados por la muestra, direcciones IP particulares y el país de pertenencia de esta. Se nos dan resultados bajo el nombre de "Padres de ejecución" donde se refiere a todos los archivos que durante el análisis dinámico fueron los que generaron al archivo en estudio y de igual manera nos muestra la cantidad de motores de antivirus que han detectado a estos archivos padres. Similar a esta sección tenemos la siguiente bajo el nombre de "Padres de recursos ejecutables portables" en donde se enumeran todos los archivos de tipo ejecutable que han sido subidos para observación y dentro de los cuales la muestra correspondiente a nuestro análisis estaba siendo transportada.

Otros apartados similares que se nos entrega dentro de esta pestaña son los que se consideran como "Archivos soltados" los cuales se refiere a aquellos archivos creados por la muestra durante su ejecución, "Hijos de recursos ejecutables portables" en la cual hace referencia a los archivos que, en este caso, viajaban dentro de la muestra de nuestro análisis. Tras determinado tiempo, dentro de esta pestaña se muestra una última sección que se nombra como "Grafo de resumen" el cual nos presenta con una estructura tipo grafo resumiendo todo lo encontrado en esta pestaña tal como se puede observar en la *figura 67*.

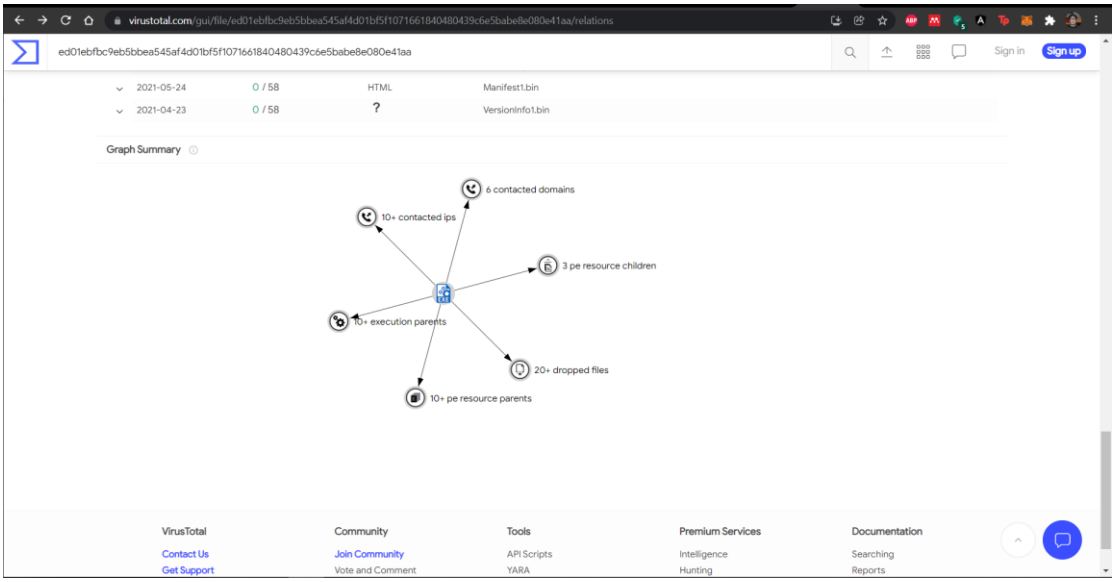


Ilustración 67: Análisis de WannaCry en VirusTotal. Grafo de resumen

En la siguiente pestaña, nombrada *Comportamiento*, se nos devuelve el resultado de un análisis dinámico dentro de múltiples entornos de tipo sandbox en los cuales la plataforma ejecutó y observó la conducta de la muestra cargada para análisis. Cada uno de estos sandbox devuelve todo el comportamiento observado durante la ejecución de la muestra, y cada uno es diferente por lo que cada uno devuelve un listado con diferentes seccionas, mas en la mayoría de las secciones que coinciden en sus resultados comparten información entre sandbox, lo que ayuda a corroborar la fiabilidad del comportamiento observado de la muestra analizada.

Entre las secciones que se observan con mayor frecuencia se encuentras aquellas que nos presentan las acciones realizadas en el sistema de archivos, en el cual se puede observar los archivos que abrió, escribió, modificó o eliminó la muestra dentro del entorno de estudio. Otra sección que se repite en los reportes de los entornos de estudio son las acciones realizadas en el registro de Windows, el cual es una especie de base de datos que se encarga de almacenar todas las configuraciones de nivel base del sistema operativo y para todas las aplicaciones que decidan usar dicho registro.

Una de las secciones más destacables dentro de esta pestaña es la de acciones realizadas a procesos y servicios, en este apartado su pueden observar los procesos generados durante el estudio y de igual manera se puede observar si la muestra logró acceder a una terminal de comandos y de ser así, los comandos ejecutados en la misma. Finalmente, otra sección frecuente es aquellas de acciones relacionadas a mecanismos de sincronización y señales, en especial la creación de los denominados *mutex* (mutuamente excluyentes) los cuales son condiciones que facilitan a que los mecanismos de sincronización para el acceso a la memoria compartida entre los diferentes procesos dentro de un S.O. funcionen correctamente.

La pestaña final, *Comunidad*, contiene información proporcionada por la comunidad de usuarios de VirusTotal, entre los cuales se encuentra un listado de los usuarios en los cuales, dentro de todo su proceso de estudio también se ve involucrada nuestra muestra, además de esto, los usuarios pueden dejar comentarios relacionados al análisis, dejar comentarios de valor para otros analistas, etc.

Tal como se puede observar, la plataforma VirusTotal proporciona un análisis a fondo de las muestras de malware otorgando un sinnúmero de artefactos que pueden ser considerados como IOC. Sus herramientas internas y su principal característica para la detección de una muestra maliciosa mediante el análisis simultaneo en varios motores de antivirus la vuelven una opción recomendable para realizar un análisis completo a cualquier archivo bajo sospecha. Los únicos limites encontrados dentro de esta herramienta es que esta no señala explícitamente entre todos los artefactos cuales pueden ser considerados como IOC propios de una muestra de malware y cuales son simplemente artefactos encontrados en cualquier tipo de software, lo cual es considerado una desventaja ya que si estos artefactos entregados directamente por la plataforma y utilizados para realizar la escritura de una regla YARA personalizada, es probable que llegase a darse un falso positivo.

Sumado a eso, tras una breve búsqueda de las limitaciones en cuanto a cantidad de muestras que permite subir al día, la cual dio como resultado que la plataforma es gratuita para usuarios finales sin fines comerciales y no cuenta con límite fuera de un máximo de 500 MB para la muestra a subirse, y para su API pública, la cual son aproximadamente 10 peticiones diarias. Se encontró que la comunidad, especialmente la más experimentada en el campo de la seguridad informática, no está a favor de realizar la carga de muestras, especialmente de ransomware a esta plataforma. Esto debido a que la comunidad comenta que, en varios casos, los ataques de ransomware suelen ser realizados a la medida y, por lo tanto, dentro de sus archivos puede ya contener información sensible que, al ser subida para análisis se vuelve pública y esto amplifica el impacto del incidente ya que esto provoca el esparcimiento no intencional

de la información, para los diferentes fabricantes de antivirus y la comunidad de VirusTotal.

Otra desventaja de esta plataforma es que, tal y como la misma plataforma lo señala dentro de su sección de preguntas frecuentes (VirusTotal, 2017), dice lo siguiente:

> *"VirusTotal simplemente agrega la salida de diferentes proveedores de antivirus y escáneres de URL, no produce ningún veredicto propio. Como tal, si está experimentando un problema de falso positivo, debe notificar el problema a la empresa que produce la detección errónea, ellos son los únicos que pueden solucionar el problema."*

Por lo tanto, se deduce que VirusTotal es un recopilatorio de resultados de las múltiples herramientas que emplea de manera interna, y es posible que es por esa razón que la plataforma no señala de forma explícita qué IOC pueden ser fiables para su uso, y esto es una desventaja dentro de esta comparativa.

Por otro lado, Intezer Analyze, a diferencia de VirusTotal, cuenta con dos tipos de cuenta tanto para el uso de su API como el de la plataforma web, los usuarios gratuitos o "community", y los usuarios premium, en donde la principal diferencia entre estos usuarios son los límites de análisis por mes, tamaño máximo por archivo, acceso a plugins premium de integración de otras herramientas, etc. Otro punto a favor de Intezer es que no realiza un recopilatorio de resultados y se los devuelve al usuario. Intezer cuenta con sus propios métodos para realizar las diferentes tareas dentro del proceso de análisis de malware lo cual le da una mayor fiabilidad sobre VirusTotal. Sin embargo, dentro de su documentación oficial se indica que, en el caso de análisis realizados por usuarios en modo gratuito, su reporte se volverá disponible para toda la comunidad y a su vez este será compartido con VirusTotal.

Entre las funcionalidades disponibles para un usuario gratuito se encuentra el análisis de archivos, de URLs, y la búsqueda de análisis por cadena de caracteres o por familia de muestras de malware.

Dentro de la funcionalidad que nos compete, la cual es el análisis de archivos, se puede observar que la plataforma realiza una amplia cantidad de comparativas para devolver resultados fidedignos, los cuales son presentados bajo 4 pestañas principales, las cuales son Análisis genético, *TTPs* (siglas en inglés de Tactics, Techniques and Procedures), *IOCs*, y *Comportamiento*.

Dentro del primer apartado, se nos presentan 4 subpestañas las cuales son *resumen genético, muestras relacionadas, código, cadenas de caracteres, y capacidades*. En general, en esta sección a lo largo de las diferentes subsecciones, se nos da toda la información en cuanto a todos los elementos que la muestra bajo observación a reutilizado de otros programas de cualquier índole. La información presentada dentro de *resumen genético* es un listado de los genes de las familias de código de la muestra, lo que se refiere a todas las familias de código, sean estas pertenecientes a un grupo malicioso, confiable o librerías de uso común, que fueron identificadas dentro del archivo analizado.

En este listado se nos describe tanto a la familia de malware identificada, el porcentaje de genes compartidos , el número de genes de código extraídos, las cadenas de caracteres similares con las familias, y los genes comunes, los cuales hacen referencias a genes de código que pueden aparecer tanto en programas maliciosos como confiables. Además de la identificación genética también se presenta la meta data del archivo observado, la cual incluye los varios campos que ya se han presentado en las diferentes herramientas anteriormente vistas, con la diferencia en que se agregan 2 campos, uno que hace referencia al resultado de VirusTotal para ese archivo, y otro bajo el nombre de *Target Machine*, donde se detalla a qué tipo de procesador y arquitectura el archivo está elaborado para que sea ejecutado. Dentro del mismo contexto de la genética de la muestra, la subpestaña "muestras relacionadas" presenta una lista de varias muestras que comparten código con la muestra correspondiente a nuestro análisis.

La finalidad de esto puede ser la identificación de similitudes entre diferentes muestras para identificar si se puede tratar de un ataque dirigido, sin mencionar que se puede llegar a concluir la naturaleza, propósito y posible punto de origen de la muestra analizada. Dentro de este apartado se nos presentan las familias de código relacionadas y por cada una las muestras en sí relacionadas, con el veredicto de su propio análisis, la fecha en que fue analizada por primera vez en la plataforma y el porcentaje de genes reutilizados.

Otro punto que aventaja a esta herramienta frente a las anteriores es lo que se presenta dentro de la sección *"código"*, ya que en este caso se nos presenta el código ensamblador organizado por cada una de las familias relacionadas, y dado el caso de que se presenten genes compartidos, que estos pueden ser bloques específicos de código, se nos mostrarán señalados en rojo, tal como se puede observar en la *figura 68*.

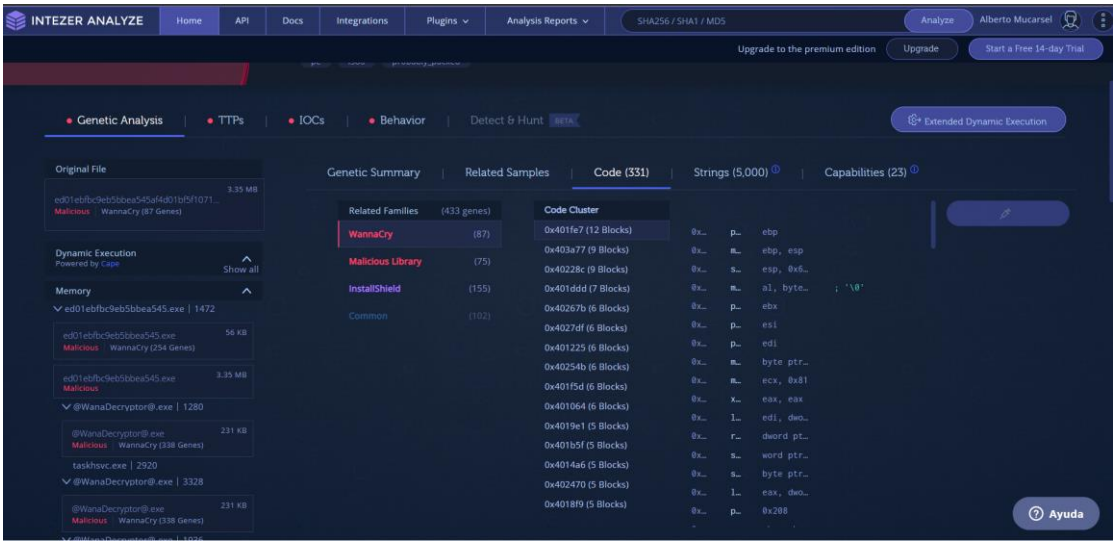


Ilustración 68: Fragmentos de código inseguros detectados

Dentro del apartado de "cadenas de caracteres" se presentan todas las cadenas de caracteres que se han encontrado dentro de la muestra, pero existe la novedad de que de igual manera en la que el usuario puede usar como punto pivote a genes de código entre muestras relacionadas, se puede realizar lo mismo con las cadenas de caracteres, esto se debe a que Intezer realiza la extracción de las cadenas del archivo examinado y las compara con su base de datos para encontrar cualquier coincidencia con análisis

realizados previamente tanto a programas confiables como maliciosos. Esta extracción y comparativa ayuda a una detección inmediata de posibles IOCs basados en cadenas de caracteres, además de que se puede tomar a estas cadenas como puntos pivote con muestras relacionadas que contienen a la misma cadena, se puede llegar a obtener un contexto más amplio de la amenaza de la muestra y que aportará significativamente cuando el IOC derivado de esta sea transformado en CTI de alta fiabilidad.

Otro de los resultados únicos dentro de esta comparativa son los que se muestran bajo "capacidades" ya que se nos presenta una tabla con un análisis del comportamiento de la muestra sospechosa bajo los estándares de detección del framework MITRE ATT&CK[25] y se listan todas las técnicas detectadas dentro de la muestra. La manera en la que este particular resultado se obtiene es mediante el uso de reglas desarrolladas por la plataforma las cuales están basadas en el análisis genético de código como a su vez está implementando las funcionalidades de la librería open-source CAPA, la cual según su descripción en su repositorio disponible en la plataforma Github se describe como una librería que se puede correr contra archivos ejecutables o de scripts de consola y esta se encarga de decirle al usuario qué cree que hace el archivo analizado (Mandiant, 2022).

Dentro del listado de las tácticas encontradas se puede apreciar una breve descripción de la capacidad de la táctica, la categoría bajo la cual Intezer la ha clasificado y finalmente una columna bajo el nombre de "Encontrada en Codigo de" la cual nos brinda otro punto pivote con las muestras relacionadas, y con esta similitud de funcionalidades se vuelve más sencilla la identificación de herramientas utilizadas entre diferentes muestras, y esto a su vez puede proveer un punto de vista para determinar si este comportamiento detectado es considerado de vital importancia para que se cumpla el objetivo del malware, lo cual considero sumamente benéfico ya que muchas veces y tal y como se mencionó en las ventajas y desventajas de los IOC, se pueden llegar a producir falsos positivos por el uso de IOC de forma textual, pero con estos resultados se nos da una vista clara al contexto de la muestra lo que definitivamente puede ayudar a clasificar de mejor manera al artefacto relacionado con la táctica detectada.

Tras la revisión de la primera pestaña, la cual el aquella que presenta la mayor cantidad de información obtenida en el estudio de la muestra, la siguiente pestaña, *TTPs*, nos permite observar las tácticas que están siendo implementadas por la muestra al momento de realizar su trabajo. Similar a la última subpestaña *capacidades*, se nos presenta una tabla llenada con las tácticas detectadas según los lineamientos del framework MITRE ATT&CK, las diferencias en este caso es que ahora se nos presenta a mayor detalle las técnicas utilizadas, proporcionando una descripción más concreta, la severidad de la acción, y los detalles de cada actividad que fue detectada a un nivel más técnico, mostrando que configuraciones son responsables de la creación, modificación, o eliminación de los diferentes archivos o configuraciones del sistema operativo de la víctima.

[25] MITRE ATT&CK framework: MITRE Adversarial Tactics, Techniques, and Common Knowledge. Es una base de conocimiento y modelo para el comportamiento cibernético de un adversario.

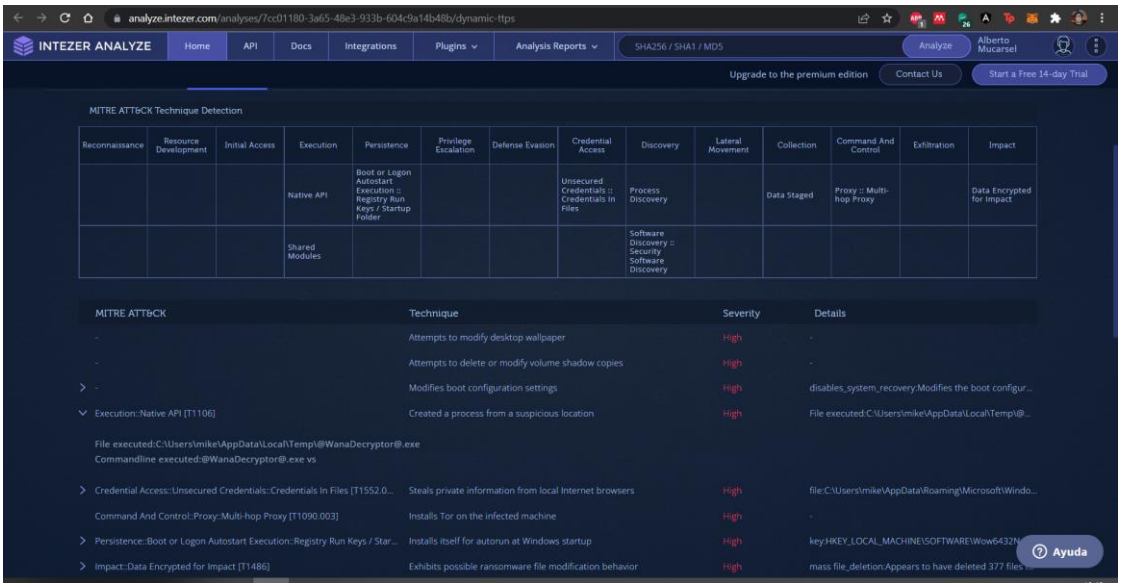


Ilustración 69: Análisis de WannaCry en Intezer. Resultado TTPs

La siguiente pestaña, *IOCs*, que es la que más llamativa se vuelve dentro del contexto de este trabajo, nos entrega directamente los artefactos hallados en la muestra que pueden servir para su identificación a futuro. En este apartado podemos observar a los IOC clasificados por su tipo. En la mayoría de los casos se nos presentan los IOC de red, y los IOC de archivos. Lo más destacable de esta extracción de artefactos es que adicionalmente de la información requerida para identificar a los artefactos, es que nos dan una clasificación de estos, la cual muestra si un IOC es utilizable para identificar a un archivo malicioso, o si forma parte de un archivo de confianza.

Esta clasificación que nos proporciona Intezer sobre cada artefacto, inmediatamente la posiciona por encima de las herramientas previamente detalladas. La razón de esto es que sus procesos internos nos otorgan ya los IOC, tal como lo hace VirusTotal, pero VirusTotal solo nos entregaba un recopilatorio de resultados sin una clasificación clara, y como mencioné previamente esto puede ocasionar falsos positivos. En cambio, con la clasificación de Intezer las probabilidades de que los artefactos sean utilizados en procesos de CTI y detección de amenazas y se ocasione un falso positivo se ven disminuidas en considerable manera, ya que al saber los IOC de confianza, se podría elaborar una serie de reglas que pueda detectar solamente aquellos artefactos maliciosos, o inclusive para una regla más a detalle, se podría especificar que se busque la presencia de la combinación exacta de artefactos de confianza y maliciosos, con lo que las posibles tácticas de camuflaje de una muestra de malware se verían invalidadas.

Finalmente, dentro de *Comportamiento*, Intezer proporciona capturas de pantalla de la ejecución y el comportamiento observado de la muestra dentro de su entorno tipo sandbox. Tras esto se nos presenta el árbol de procesos donde se nos da el paso a paso de todas las actividades realizadas por la muestra dentro de la máquina donde se está ejecutando, y cada paso contiene la ruta, los comandos y su identificador de procesos. Como con VirusTotal se nos proporciona las demás actividades realizadas por el archivo estudiado, tal como su actividad de red, de servicios, de archivos y de existir, cualquier actividad relacionada con los registros del sistema.

Tras la descripción de los resultados obtenidos, Intezer Analyze se muestra como la herramienta más completa para la ejecución de un análisis de malware con la obtención de IOC de forma directa. Esta herramienta se puede considerar como una plataforma

completa y como se define en el mundo del software, como una herramienta *Plug and Play*, ya que lo único que se necesita es subir la muestra a analizarse y es toda la intervención que se debe realizar por parte del/la analista para la obtención de resultados completos los cuales han demostrado tener mayor credibilidad que un recopilatorio de resultados sin una clasificación clara de los mismos.

Uno de los puntos que puede poner en desventaja a la herramienta, es la falta de granularidad en la entrega de los IoC que han sido detectados dentro de un análisis, esto debido a la clasificación que da Intezer, la cual no entra a mayor detalle como por ejemplo si dentro de los IoC de archivo se encuentre algo relacionado a la modificación de algún archivo de configuración o no ya que, dentro de los análisis ejecutados la etiqueta que bajo el campo *tipo* en cada IoC solamente varía entre *archivo principal* y *archivo dejado*. A diferencia de la herramienta previa, que nos entrega un recopilatorio que a pesar de no tener una clasificación crucial de si es malicioso o no el indicador, sí no entrega a mayor precisión las acciones que podrían ser utilizadas para fabricar reglas de detección.

La siguiente herramienta para analizar es el framework de Volatility y, dado a su amplia gama de plugins y diferentes funcionalidades, este análisis se centrará únicamente en las funcionalidades básicas y aquellas de nivel intermedio que sea requeridas de usar durante la ejecución de los análisis.

Tras las configuraciones realizadas para aislar las máquinas utilizadas dentro de la máquina anfitrión, la infección controlada puede llevarse a cabo con los riesgos minimizados. Con estas medidas ahora tras los diferentes análisis que no requirieron la ejecución directa de las muestras es posible realizar la infección para poner en práctica los análisis con Volatility ya que este requiere de un volcado de memoria de una máquina, y preferiblemente si este volcado se obtiene mientras la infección está viva.

En el caso de realizar la infección controlada de WannaCry y BadRabbit dentro del entorno víctima que corre Windows 10 se encontró con un percance. De acuerdo con información oficial de Microsoft (Ganacharya, 2018), los sistemas Windows 10 cuentan ya con las mejoras necesarias incorporadas para que infecciones de ransomware derivados de Petya no puedan darse, tal y como son las 2 muestras previamente mencionadas. Todo esto se debe a que en Windows 10 ya se cuenta con guarda del flujo de control del kernel e integridad de código del kernel, entre otras defensas, las cuales imposibilitan realizar el proceso de infección controlada para la posterior obtención del volcado de memoria. Por lo que, a partir de este punto, el análisis de Volatility se elaborará a partir de la infección de la muestra Glimpse.

		PLATFORM DEFENSES AGAINST WANNACRY			
		Windows 7	Windows 10		
Arrival and installation	Trojan dropped into vulnerable machines via SMBv1 exploit CVE-2017-0145 (EternalBlue)	**Windows AppLocker** **Microsoft Security Essentials** **System Center Endpoint Protection**	**Windows 10 built-in exploit mitigation** (KASLR, NX HAL, and PAGE POOL) **Windows Defender Application Control** **Windows AppLocker** **Windows Defender AV** **Windows Defender Application Guard**	**Windows Defender ATP** detects malicious behavior across the entire attack kill chain	**Windows 10 S** is a configuration of Windows 10 that locks down devices, sealing common ransomware entry points
Payload	File encryption		**Controlled folder access** feature in Windows Defender Exploit Guard		
Lateral movement	Lateral movement SMBv1 exploit CVE-2017-0145 (EternalBlue)		**Windows 10 built-in exploit mitigation** (KASLR, NX HAL, and PAGE POOL) **Windows Defender Application Control**		

Ilustración 70: Defensas incorporadas contra WannaCry en Windows 7 y Windows 10
Fuente: Microsoft Security Blog (Ganacharya, 2018)

		PLATFORM DEFENSES AGAINST PETYA			
		Windows 7	Windows 10		
Arrival and installation	Malicious software download from compromised software supply-chain	**Windows AppLocker** **Microsoft Security Essentials** **System Center Endpoint Protection**	**Windows AppLocker** **Windows Defender Application Control** **Windows Defender Application Guard** **Windows Defender AV**	**Windows Defender ATP** detects malicious behavior across the entire attack kill chain	**Windows 10 S** is a configuration of Windows 10 that locks down devices, sealing common ransomware entry points
Payload	Master boot record (MBR) tampering		**Windows Defender System Guard (Secure Boot)** stops modified MBR from being loaded at boot time		
	Credential theft		**Credential Guard**		
	File encryption		**Controlled folder access** feature in Windows Defender Exploit Guard		
Lateral movement	Lateral movement using stolen credentials to gain access to machines in the network and using tools to remotely execute ransomware		**Windows AppLocker** **Windows Defender Application Control** **Windows Hello with Credential Guard**		
	Lateral movement using SMBv1 exploits CVE-2017-0145 (EternalBlue) and CVE-2017-0145 (EternalRomance)		**Windows 10 built-in exploit mitigation** (KASLR, NX HAL, and PAGE POOL) **Windows Defender Application Control**		

Ilustración 71: Defensas incorporadas contra Petya en Windows 7 y Windows 10
Fuente: Microsoft Security Blog (Ganacharya, 2018)

La obtención del volcado de memoria se realizará con la herramienta WinPMem la cual permite generar archivos .raw con el volcado de memoria al momento de su ejecución, y este archivo resultante será aquel que se envíe a la máquina principal para ejecutar el análisis de los procesos en memoria.

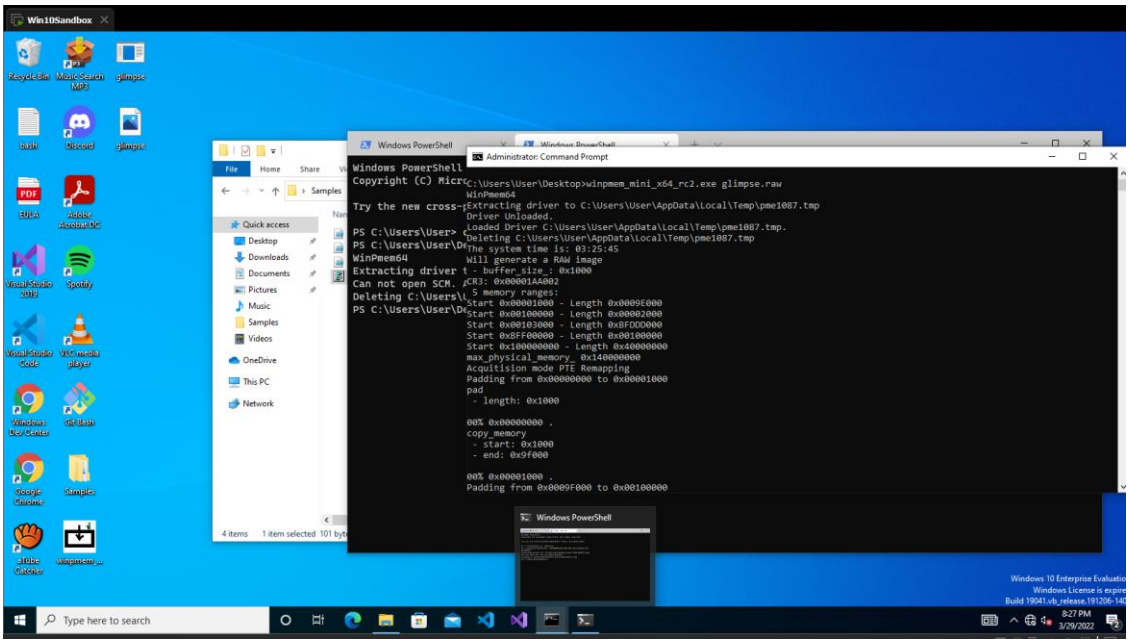


Ilustración 72: Obtención de volcado de memoria tras infección controlada

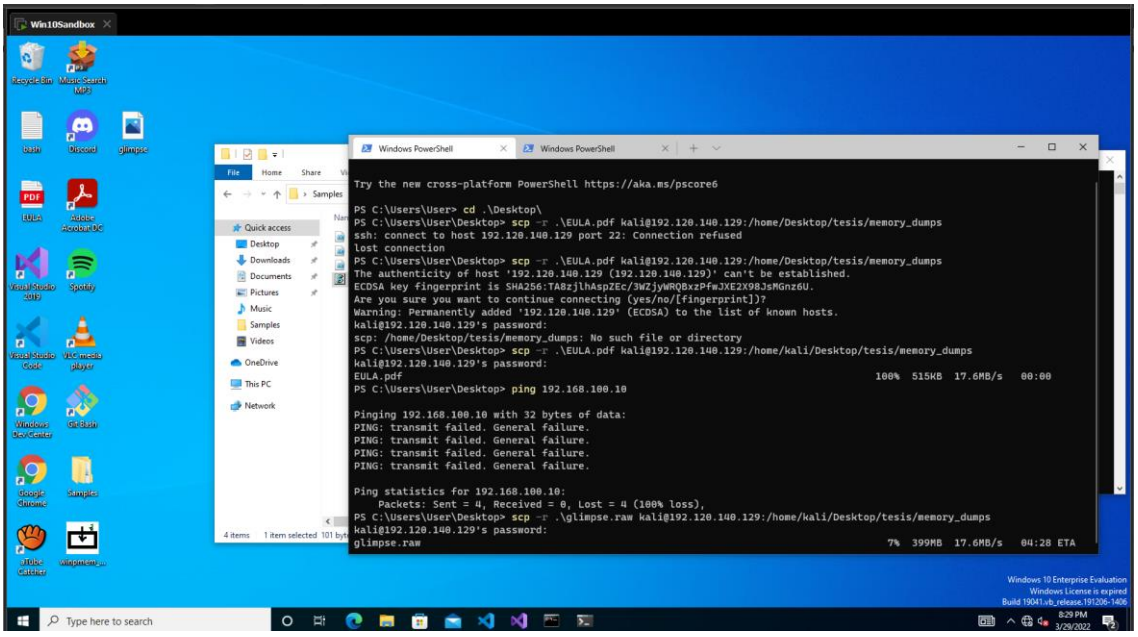


Ilustración 73: Envío de volcado de memoria a máquina Kali para análisis

Una vez se ha recibido el volcado de memoria, se puede examinar el archivo para realizar la búsqueda de IoC con la ayuda del framework en turno. Dado a que volatility tiene varios plugins a ser utilizados, nos enfocaremos en aquellos disponibles para Windows.

Uno de los primeros plugins que se pueden ejecutar para obtener la información básica es *windows.info.Info* el cual ayuda a identificar el kernel y la versión del S.O. detectado dentro del volcado de memoria.

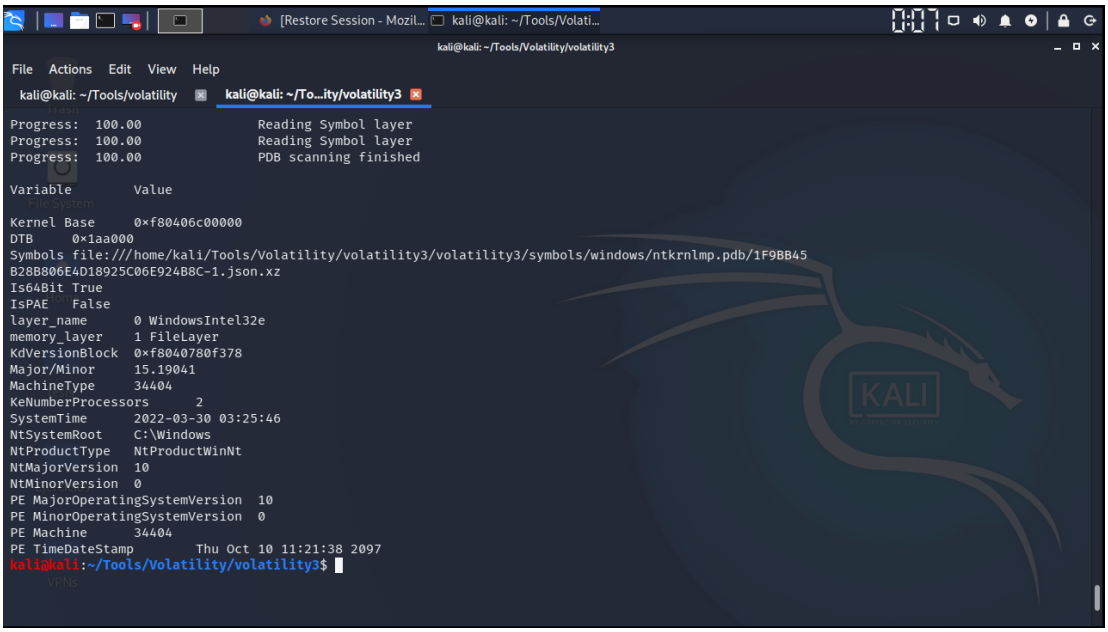


Ilustración 74: información general sobre el volcado de memoria

Al igual que las primeras herramientas listadas en este análisis, Volatility requiere de cierta experticia para realizar la extracción de IoC, y este proceso debe ser realizado por la persona a cargo del análisis de la muestra, ya que la herramienta como tal no es capaz de extraer dichos artefactos.

Por ejemplo, Volatility nos permite realizar la extracción y listado de todos los módulos cargados dentro del kernel hasta el momento de la obtención del volcado de memoria utilizando el plugin *windows.modules.Modules*.

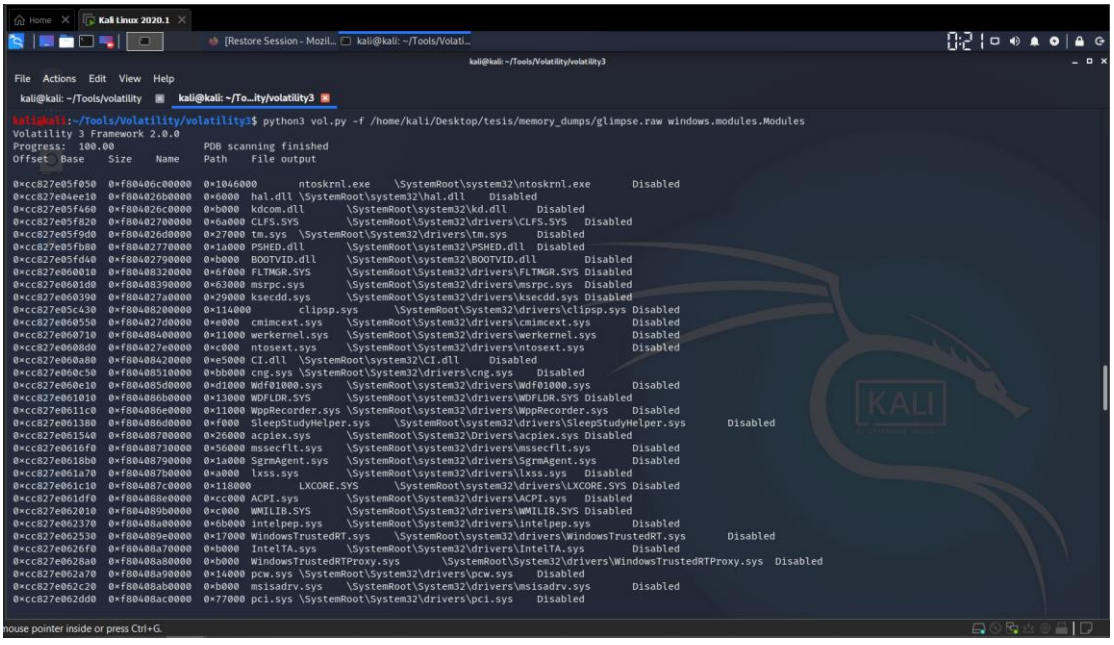


Ilustración 75: Resultados de extracción de información de plugin windows.modules.Modules

Tal como se observa en la figura anterior se puede observar la base en memoria, tamaño, nombre, ruta y resultado, lo cual resulta en un listado amplio de módulos a ser revisados a detalle, y cabe recalcar que este es el volcado de una máquina que cuenta únicamente con 4GB asignados para memoria RAM, por lo que es seguro concluir que

el tamaño de este listado es directamente proporcional al tamaño del volcado de memoria.

En caso de querer analizar este apartado a más detalle, existe otro plugin integrado en el framework que nos permite revisar todas las librerías dinámicas usadas hasta el momento, y para completar esta extracción de información específica se puede hacer uso del plugin *windows.dlllist.DllList*.

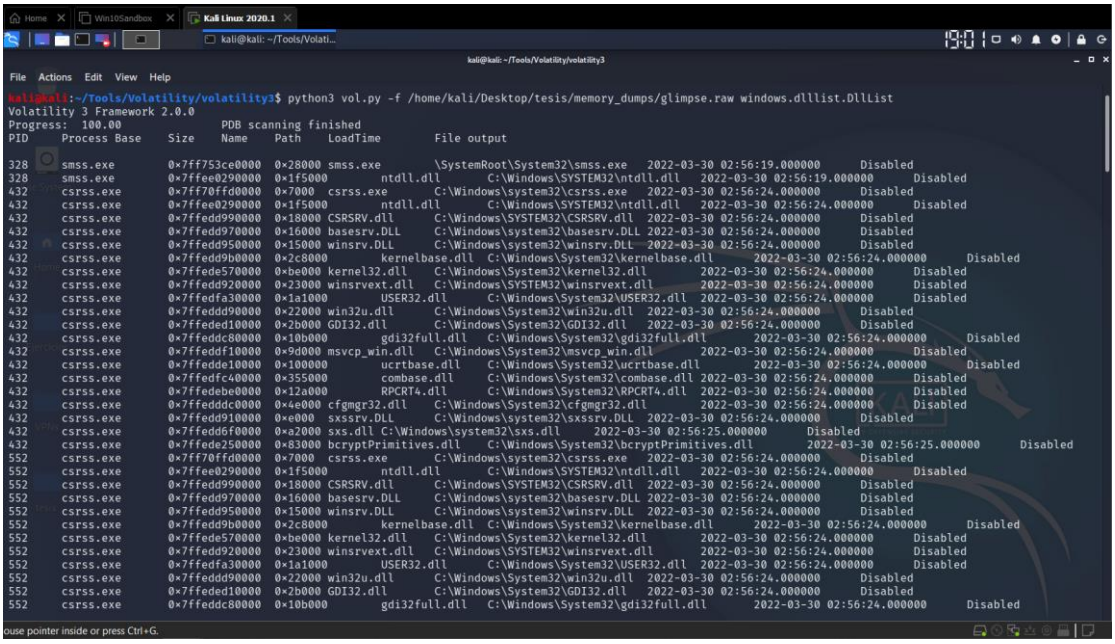


Ilustración 76: Resultados de extracción de información de plugin windows.dlllist.DllList

El resultado se asemeja al de la figura 70 pero como mencioné anteriormente, este plugin nos extrae esta información con mayor detalle, por ejemplo, nos otorga el PID de cada una de las librerías cargadas, el nombre del proceso como tal, y el tiempo de carga. Con el tratamiento apropiado de preprocesamiento de datos, se pueden llegar a extraer las librerías que tengan una presencia no esperada, pero este proceso es independiente de cada analista y la manera en la que busque realizarlo, ya que Volatility no cuenta con dichas capacidades integradas.

Finalmente, otro de los plugins que considero relevantes en este análisis es *windows.cmdline.CmdLine*, el cual nos da el registro de todos los comandos que fueron ejecutados hasta el momento dentro de la consola de nuestra máquina víctima. Quiero recalcar que la obtención de esta información es sumamente importante, por lo que algunos virus contienen técnicas ofensivas en las que logran ejecutar comandos directamente en la consola sin que el usuario se percate de esto ya que no se abre una ventana la cual podría alertar inmediatamente al usuario.

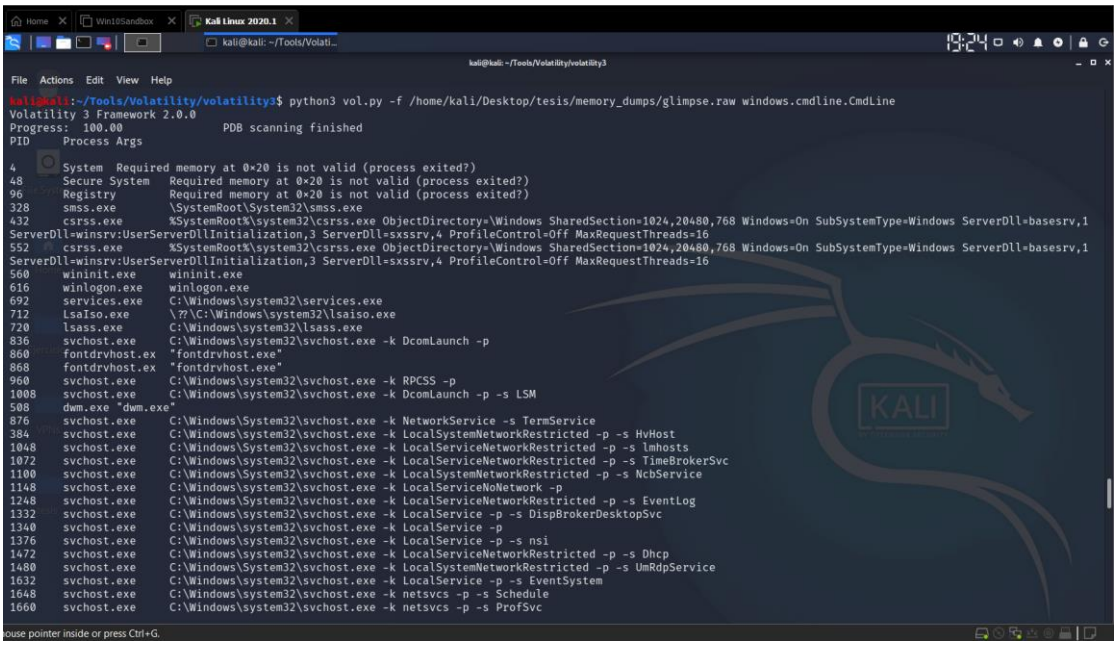


Ilustración 77: Resultados de extracción de información de plugin windows.cmdline.CmdLine

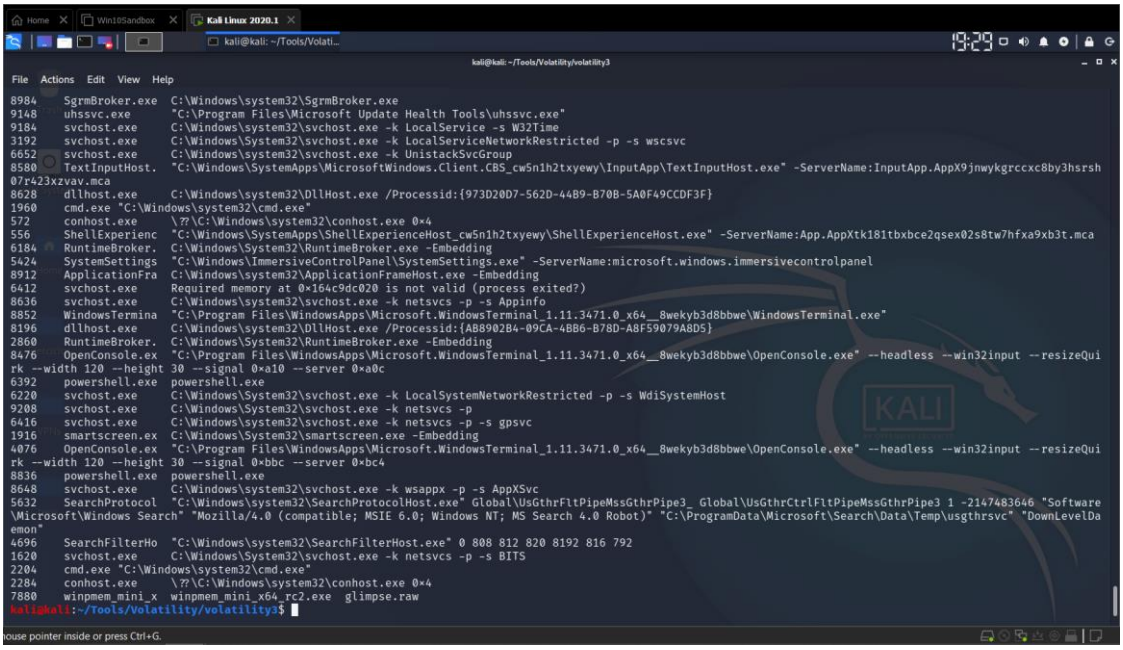


Ilustración 78: obtención de comandos ejecutados previos a obtención de volcado de memoria

Para un analista que presta atención al detalle y conoce cómo filtrar esta información de manera correcta, la obtención de IoC con ayuda de Volatility se vuelve una tarea sencilla, como por ejemplo, si se observa la línea final en la figura , se puede observar claramente el comando que ejecuté dentro de la máquina infectada para conseguir el volcado de memoria, y como este ejemplo pueden hallarse muchos artefactos más que puedan ser de ayuda para la creación de condiciones o reglas destinadas a asistir en la detección de futuras infecciones o intentos de este.

Lo que pone a Volatility por delante de las herramientas como Cutter y Ghidra es que los plugins con los que el framework cuenta, permiten trabajar de una manera más limpia y directa con ciertos segmentos del volcado de memoria donde se puede realizar una examinación más rigurosa y evitar la navegación entre cierta cantidad de información no requerida durante el estudio del malware.

Por desgracia, al igual que las herramientas para realizar análisis estático, la herramienta no es capaz de obtener como tal IoC a partir del archivo que se le proporciona. Se puede concluir que todos estos programas tienen más el objetivo de servir como una ayuda al analista para hacer una especie de disección del archivo observado, y que no se deje nada sin estar a la vista, para que, con esto, la persona experta sepa encontrar las anomalías y procesarlas de manera correcta para que estas sean consideradas como IoC.

Conclusiones y recomendaciones

Conclusiones

Tras los diversos análisis realizados durante la parte práctica de este trabajo se puede concluir que:

La mejor herramienta para análisis de malware según su obtención de IOC es Intezer Analyze.

Tras las diversas pruebas realizadas tanto de forma estática como dinámica, la mejor herramienta que demostró mayor utilidad para llevar a cabo el análisis de muestras es Intezer Analyze gracias a sus múltiples funciones tales como su análisis genético que permite observar rasgos similares entre muestras previamente analizadas y en particular sus algoritmos de análisis e identificación de falsos positivos, lo que facilitó la identificación de la muestra y los IOC que podrían ser implementados para una futura identificación mediante el uso de reglas YARA o herramientas propias con las respectivas pautas si así se lo desea. A pesar de otra de las herramientas como por ejemplo Ghidra, demuestra un acceso directo a todo el código malicioso y permite diseccionarlo con mayor precisión, no da las pautas necesarias para determinar de entre todo contenido qué realmente puede ser utilizado como IOC. Otro aspecto que se llegó a concluir es que estas herramientas suelen ser de mayor utilidad para el usuario cuando se usan en conjunto ya que cada una se encuentra enfocada en un área específica del análisis de malware y esto puede llegar a acelerar el proceso de análisis de ser utilizadas de forma correcta.

Cada una de las herramientas puede llegar a complementarse dentro un proceso de análisis de malware a profundidad

Tras la implementación y el uso de cada una de las herramientas dentro de este trabajo, se puede concluir que ninguna de las herramientas pueda llegar a ser la única usada dentro de un análisis de malware que se lleve a profundidad, como por ejemplo Intezer, nos da una amplia cantidad de información sobre la muestra, y dentro de este contexto nos permite obtener IoC con una mayor confiabilidad, pero la desventaja sería en el caso de que dentro de este proceso se busque entender el comportamiento en sí de la muestra, para lo cual se requeriría efectuar primero un análisis estático observando su código y ver qué es lo que busca realizar el archivo malicioso si fuese ejecutado. Y de igual manera, a pesar de que se pueda entender las acciones que va a llevar a cabo un archivo malicioso tras analizar su código, por ejemplo, utilizando Ghidra, no se tiene la certeza de cómo va a afectar a la máquina víctima, por lo que será necesaria su ejecución en un entorno aislado y seguro, como por ejemplo dentro de las plataformas VirusTotal o Intezer que cuentan con un apartado dedicado a la ejecución de muestras dentro de sus entornos tipo sandbox.

Un dispositivo infectado con código malicioso puede llegar a tener su información robada o perdida permanentemente.

Una vez que cada una de las muestras fueron ejecutadas dentro del entorno víctima, se pudo observar el comportamiento de cada una de ellas como de su alcance en cuanto a

daños a la información almacenada en la máquina. Uno de los ejemplos más riesgosos es fue el de WannaCry que al haber sido ejecutado, toda la información almacenada dentro de la máquina víctima fue inmediatamente encriptada y nos solicitada realizar el pago de un rescate para poder realizar la desencriptación de los datos. De haber ocurrido en un entorno no controlado, la información hubiera tenido una alta probabilidad de haberse considerado perdida de manera permanente y esto como consecuencia de no contar con los debidos respaldos para realizar la recuperación de la información. Con este preámbulo, se puede llegar a concluir que los riesgos que un equipo infectado corre son diversos y pueden ir desde el robo de su información, su manipulación a distancia para fines maliciosos, y pueden llegar hasta la pérdida total de su información.

La obtención de IoC como criterio de comparación puede resultar sumamente útil dependiendo del objetivo a lograrse durante el análisis de malware

Dentro del contexto de este trabajo, este criterio de comparación es útil hasta cierto punto, ya que, desde cierta perspectiva, la comparación entre herramientas que se consideran como plataformas completas versus herramientas dedicadas a la disección del código fuente puede verse en cierta medida disparejo. De igual manera y como se mencionó previamente, depende el propósito del análisis para poder establecer un criterio comparativo adecuado. Cada una de estas herramientas está dedicada a aportar dentro de ciertas fases de todo el proceso que conlleva el análisis de malware, por lo que se concluye que este criterio puede llegar a ser útil siempre y cuando el objetivo final sea una obtención directa de estos artefactos.

Recomendaciones

Finalmente, para cerrar este trabajo, estas son algunas de las recomendaciones para quienes hayan leído este trabajo y se encuentren interesados en incursionar en el campo de la informática forense, en específico al análisis de malware.

Intezer Analyze es la plataforma recomendada para realizar un análisis de malware completo de manera estándar

Todas las funcionalidades revisadas de la plataforma Intezer Analyze durante este trabajo han demostrado que la herramienta es una opción óptima para realizar un análisis de malware de forma completa para todas las muestras que se encuentran merodeando a través de internet. Con Intezer se puede identificar rápidamente todo lo relacionado a las muestras que se encuentren, inclusive es altamente probable que llegue a identificar rasgos sospechosos en nuevas muestras de malware gracias a su funcionalidad de análisis genético, pero se deberá tomar en cuenta que nuevas muestras siempre deben ser analizadas a fondo de forma personalizada hasta entender su comportamiento correctamente.

No limitarse a utilizar una sola herramienta al momento de realizar análisis de muestras.

Tras la clasificación de varias de estas herramientas, se volvió evidente que cada una de estas se encarga de una sección dentro de todo el proceso de análisis de una muestra maliciosa de código, sea este su ejecución, su búsqueda de cadenas de caracteres, revertir una compilación, etc. Dado esto, se notó un mejor resultado cuando se usaron todas las herramientas listadas a una misma muestra. Por lo tanto, es recomendable que al momento de buscar realizar el análisis de una muestra de malware, se debería contar con un conjunto de herramientas focalizadas que puedan complementarse para así llegar a tener una mejor cobertura de análisis durante la examinación de una muestra maliciosa.

Mantenerse al día con todo tipo de actualizaciones y medidas de seguridad dentro de los diferentes dispositivos electrónicos.

Con lo observado durante la ejecución intencional de las muestras de código malicioso dentro de la máquina víctima, es recomendable que se esté al día con las actualizaciones que el proveedor de nuestro sistema operativo vaya publicando de manera constante, ya que esto siempre significa actualizaciones de medidas de seguridad, parches de vulnerabilidades, etc. De igual manera, otra recomendación es la de tener siempre buenas prácticas de seguridad informática para así prevenir una infección que pueda llegar a derivar en el robo de información o la pérdida de esta, y esto se puede hacer instalando un programa antivirus de confianza, un administrador de contraseñas, de preferencia que este sea ajeno al navegador de su preferencia, y evitar la ejecución de archivos que resulten sospechosos.

Determinar el objetivo final de un proceso de análisis de malware para realizar una comparativa entre las herramientas que serán usadas

Previo a realizar un análisis de malware, se recomienda que la persona a realizarla determine de forma concreta la meta de dicho proceso, con el fin de que pueda disminuir el rango de las herramientas disponibles que pueden realizar tareas de informática forense. Con un objetivo claro, la comparativa entre herramientas puede volverse más pareja y se puede evitar incluir herramientas que no vayan a aportar para el objetivo final.

ANEXOS

Anexo A: Obtención y organización de muestras en entorno principal para análisis estático

Se extraen las muestras de los diferentes repositorios listados en el trabajo, por ejemplo, theZoo se puede ejecutar como un programa de línea de comandos gracias a que está escrito en Python y nos permite extraer nuestras muestras desde la consola con unos cuantos comandos.

Ilustración 79: Obtención de muestras mediante ejecución de TheZoo

Otras muestras no estaban disponibles en theZoo por lo que se descargó la siguiente opción, y este es simplemente un repositorio de muestras, por lo que la extracción de muestras en *malware-sample-library* se realizó de forma manual.

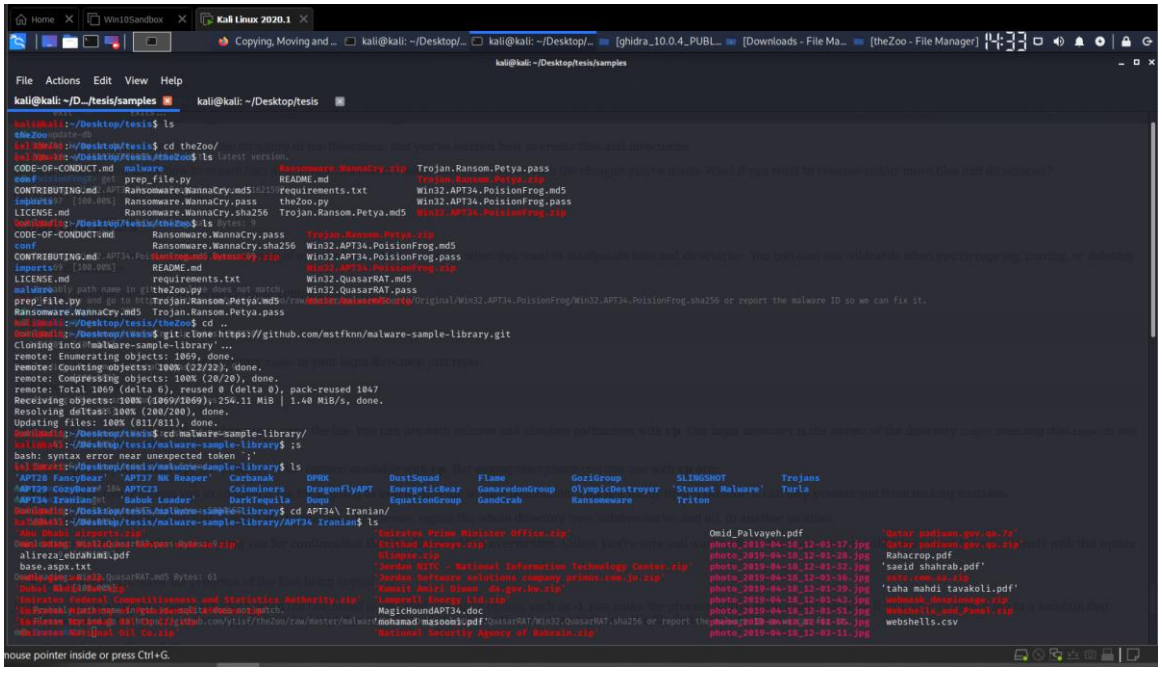


Ilustración 80: Obtención de muestras del repositorio malware-sample-library

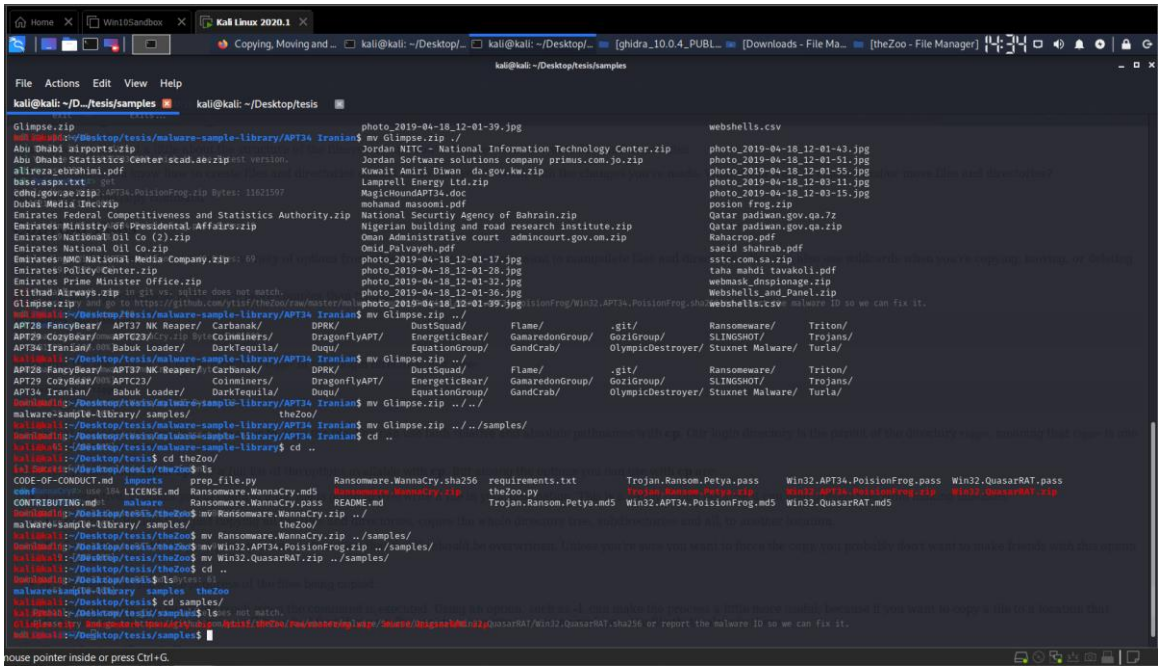


Ilustración 81: Extracción de PoisonFrog y QuasarRAT

Una vez obtenidas todas las muestras, se las organizó a todas bajo un mismo directorio por facilidad.

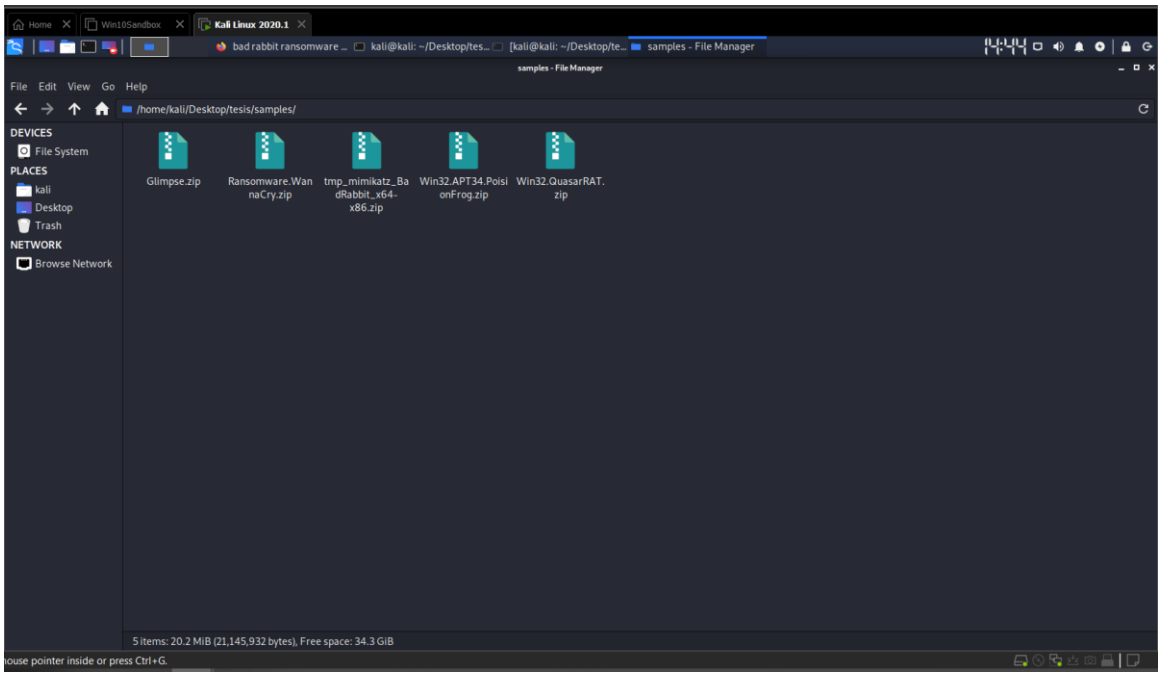


Ilustración 82: Muestras descargadas

Anexo B: Análisis realizado de la muestra de WannaCry

El archivo ejecutable de la muestra WannaCry es sometido a análisis estático con *Ghidra*

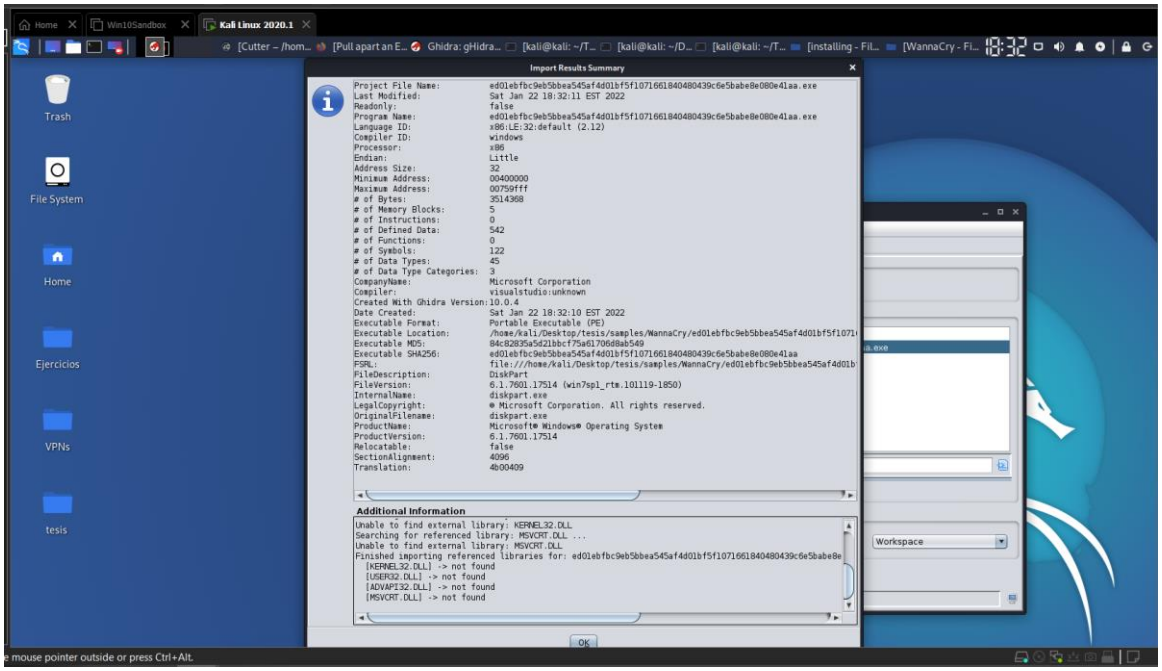


Ilustración 83: Resumen de resultados de análisis de WannaCry en Ghidra

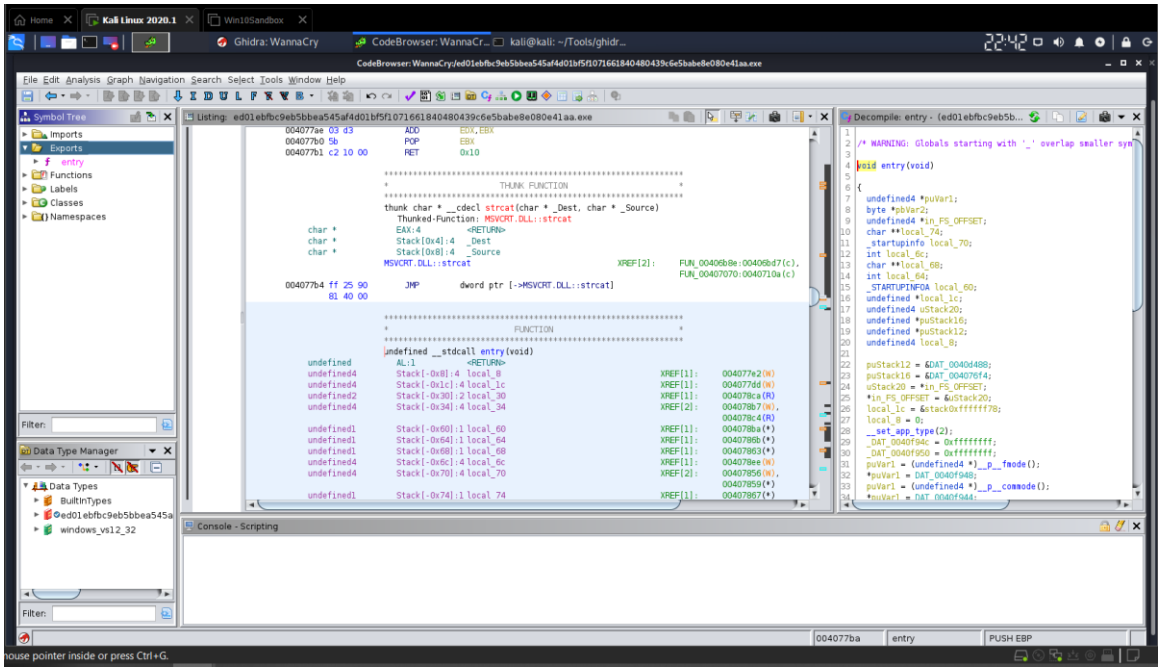


Ilustración 84: Análisis de entry point de WannaCry

Tras *Ghidra*, se realiza otro análisis estático de WannaCry en *Cutter*

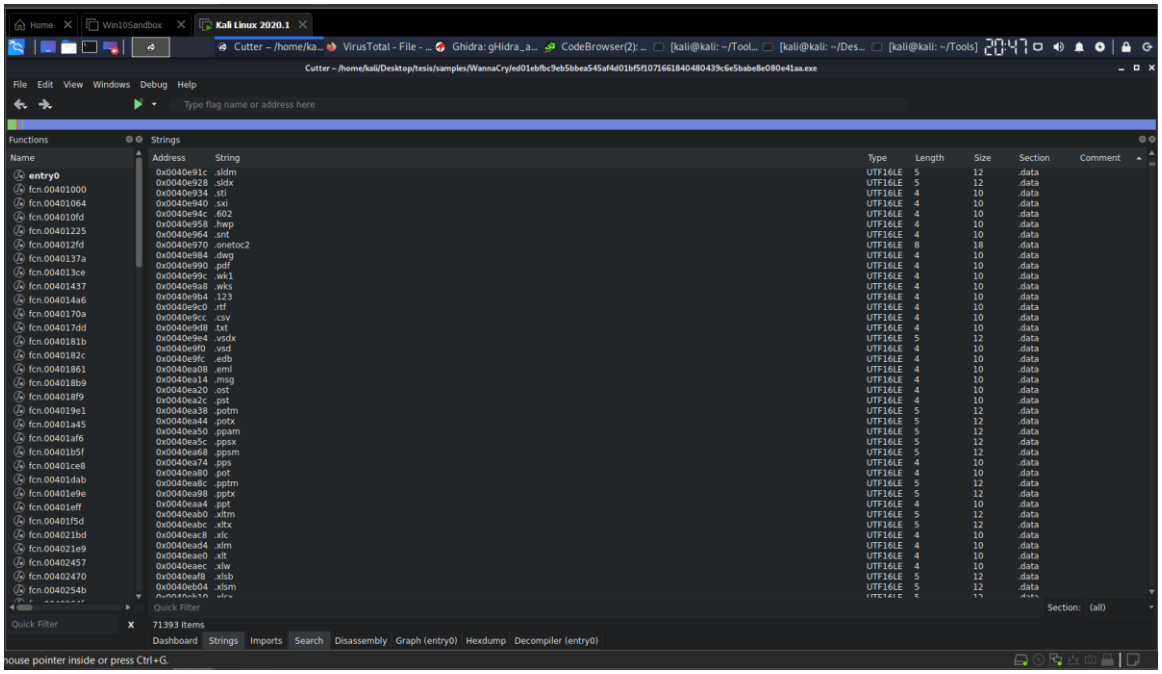


Ilustración 85: Extracción de strings de WannaCry en Cutter

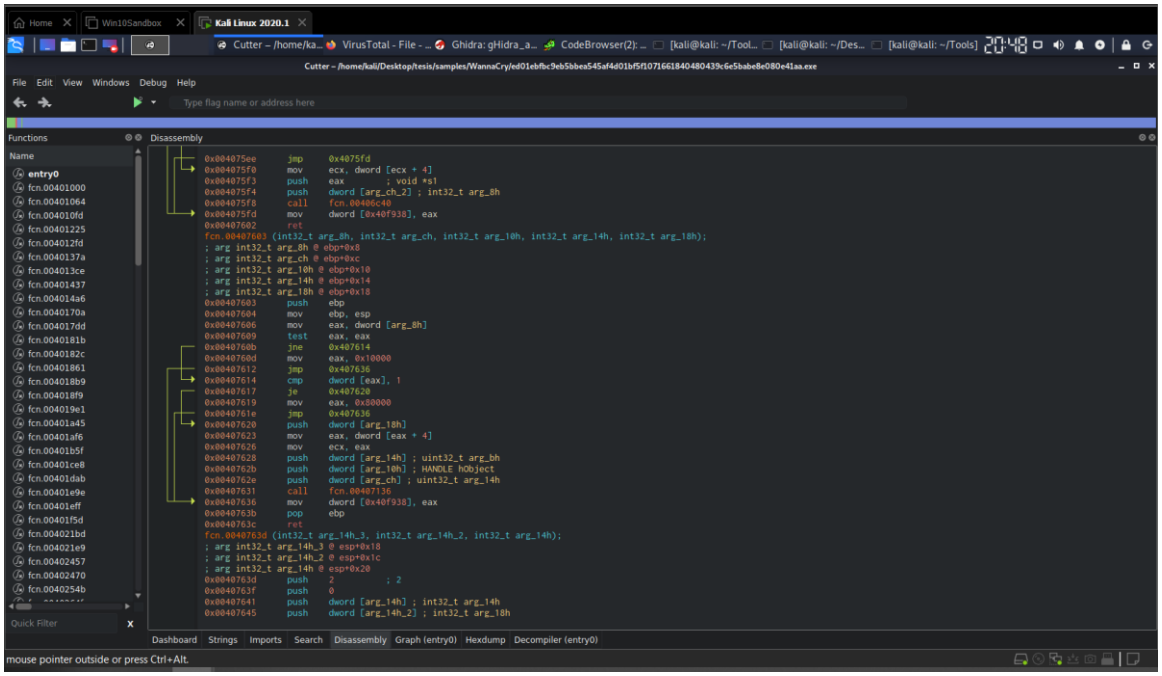


Ilustración 86: Dissasembly code de WannaCry en Cutter

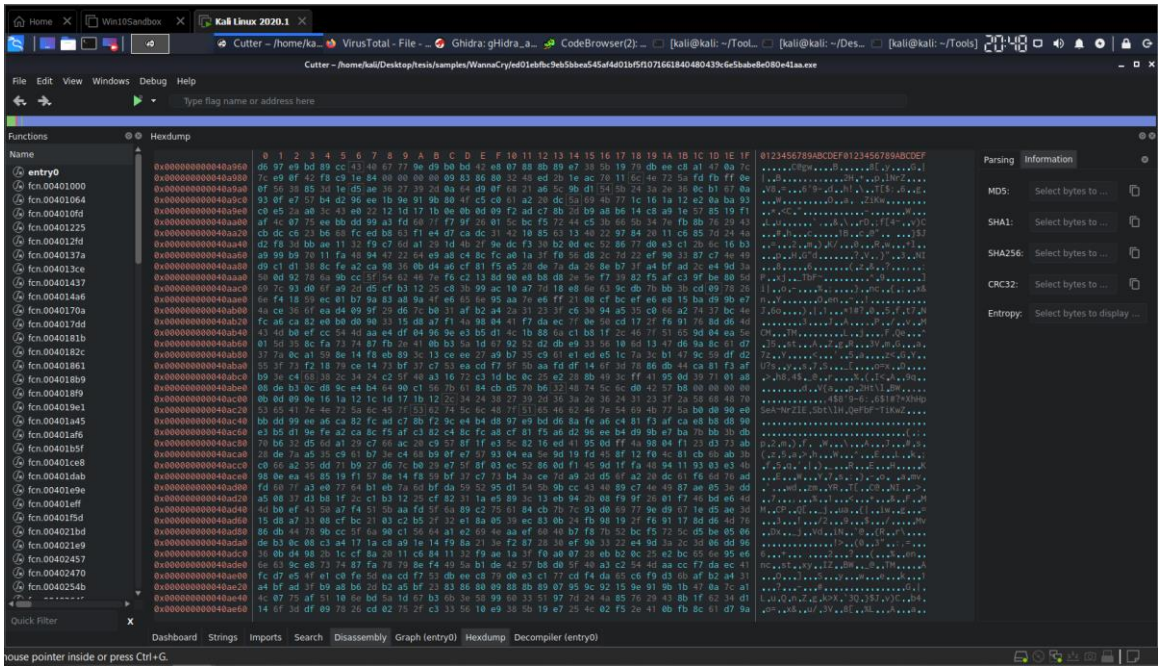


Ilustración 87: Vista hexadecimal de código de WannaCry en Cutter

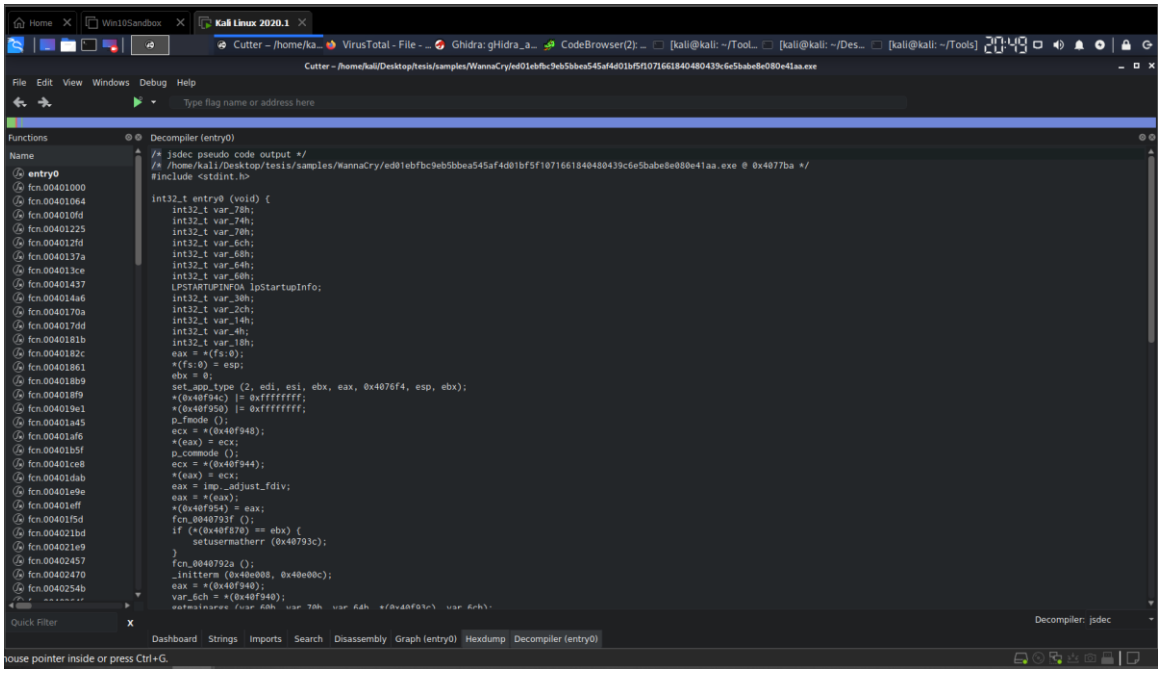


Ilustración 88: Código descompilado de WannaCry en Cutter

El archivo ejecutable con la muestra de WannaCry es subido a la plataforma *VirusTotal* para un análisis dinámico y la URL pública de su resultado es: https://www.virustotal.com/gui/file/ed01ebfbc9eb5bbea545af4d01bf5f107166184048043 9c6e5babe8e080e41aa/relations

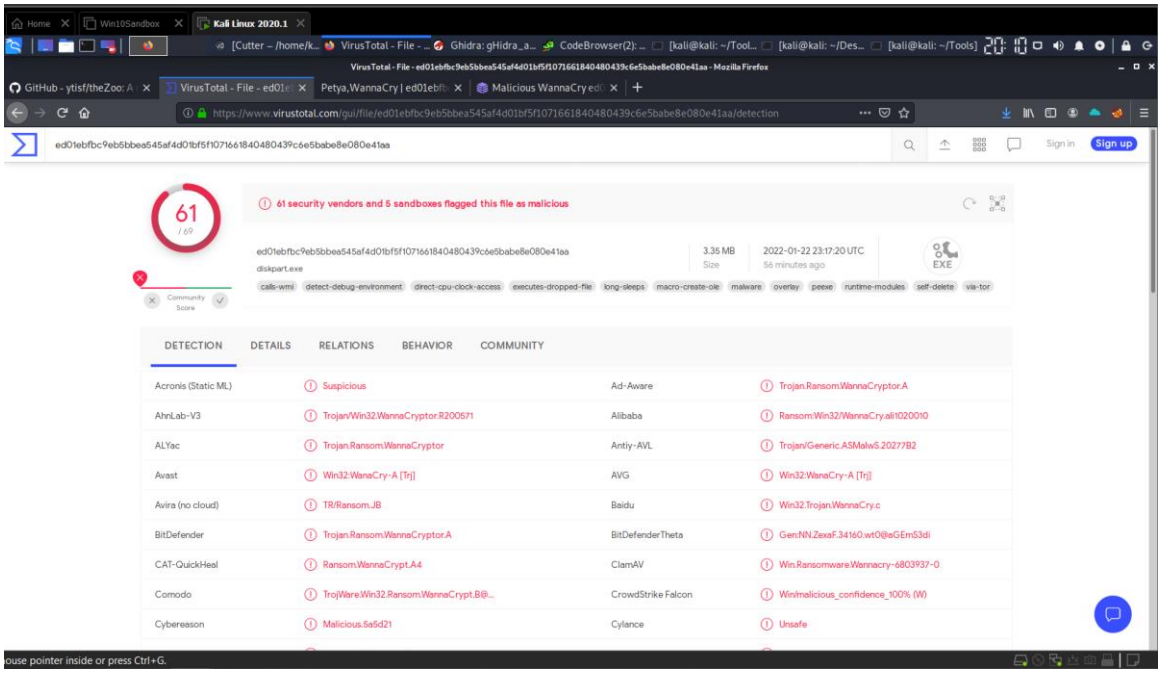


Ilustración 89: Listado de antivirus que detectaron la muestra de WannaCry

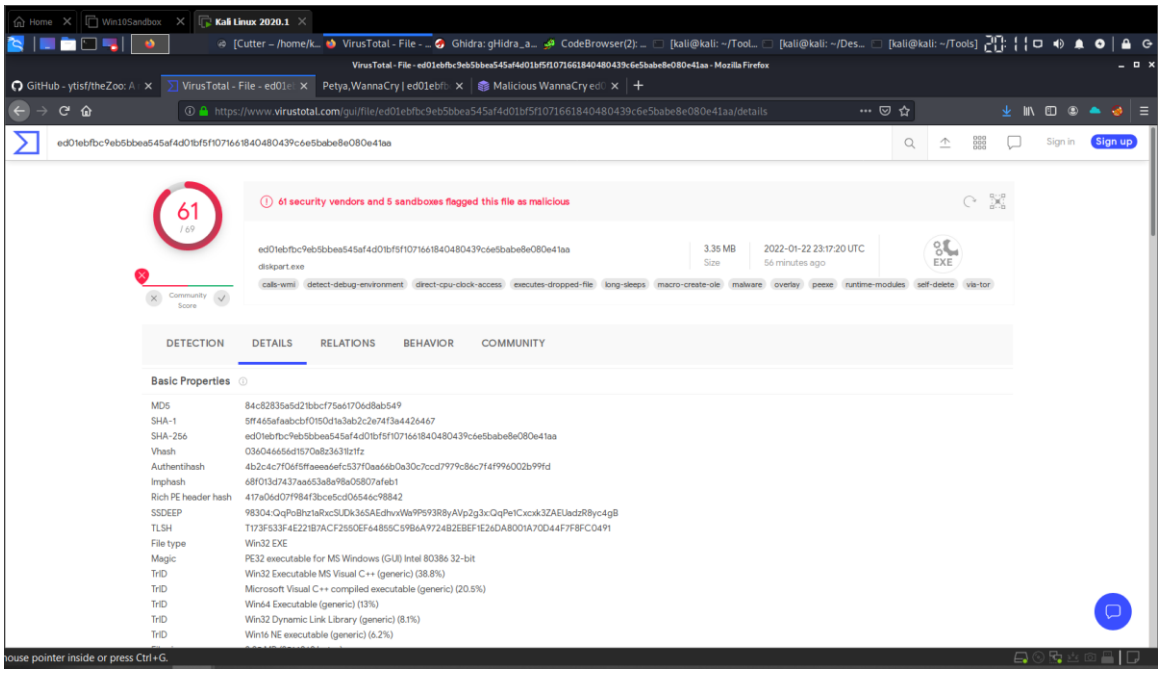


Ilustración 90: Detalles sobre la muestra WannaCry en VirusTotal

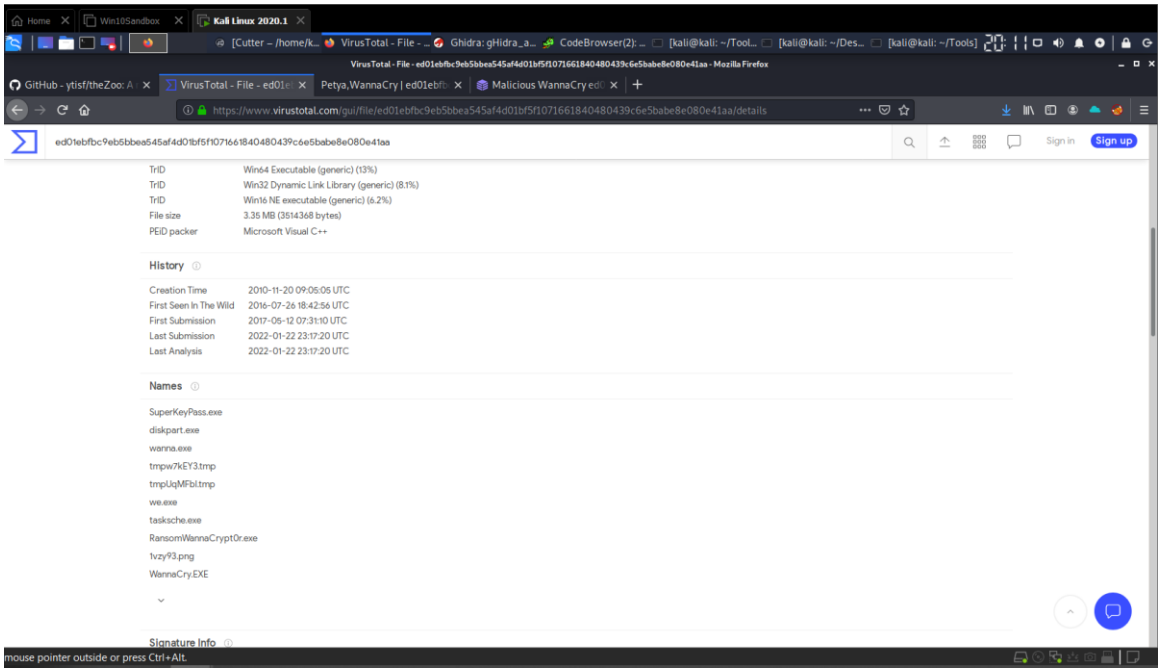


Ilustración 91: Historia y nombres de WannaCry en VirusTotal

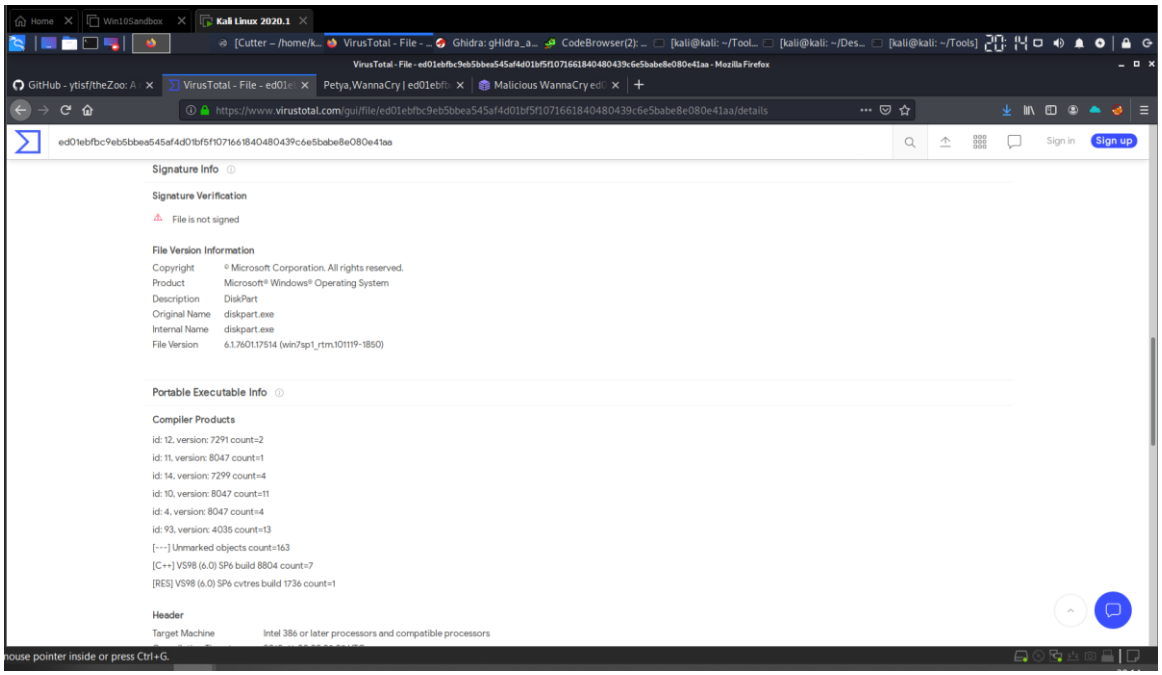


Ilustración 92: verificación de firma, información de la versión del archivo e información de archivo ejecutable de WannaCry en VirusTotal

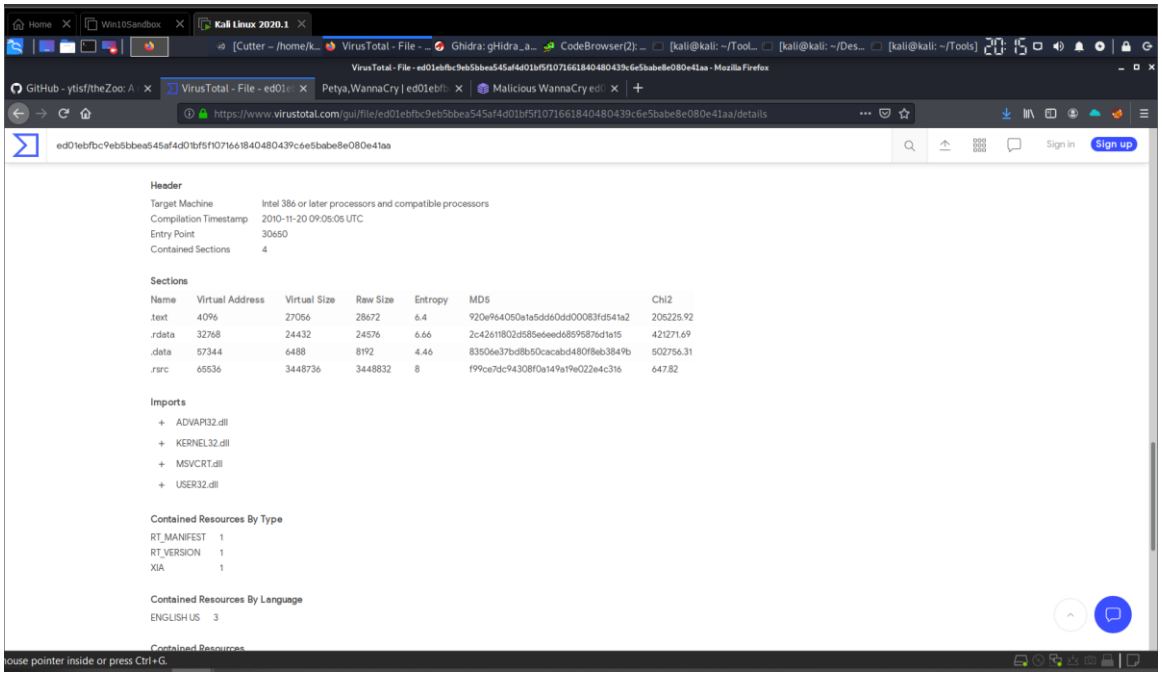


Ilustración 93: Headers, secciones y librerías importadas de WannaCry en VirusTotal

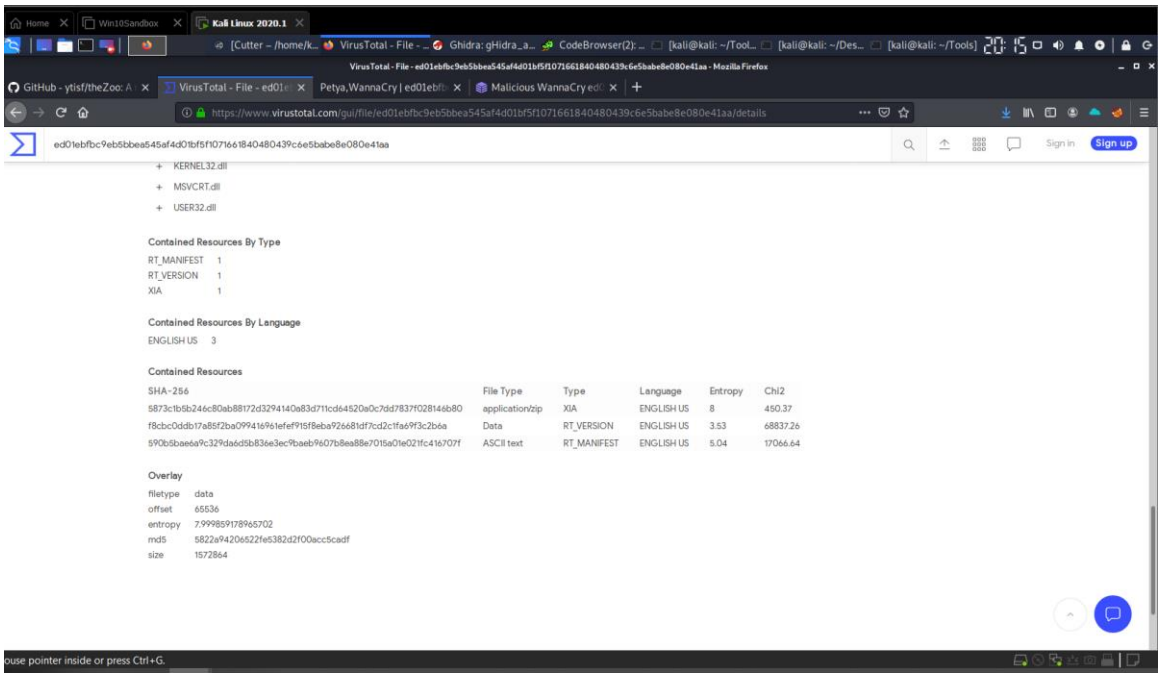


Ilustración 94: Recursos detectados dentro de WannaCry en VirusTotal

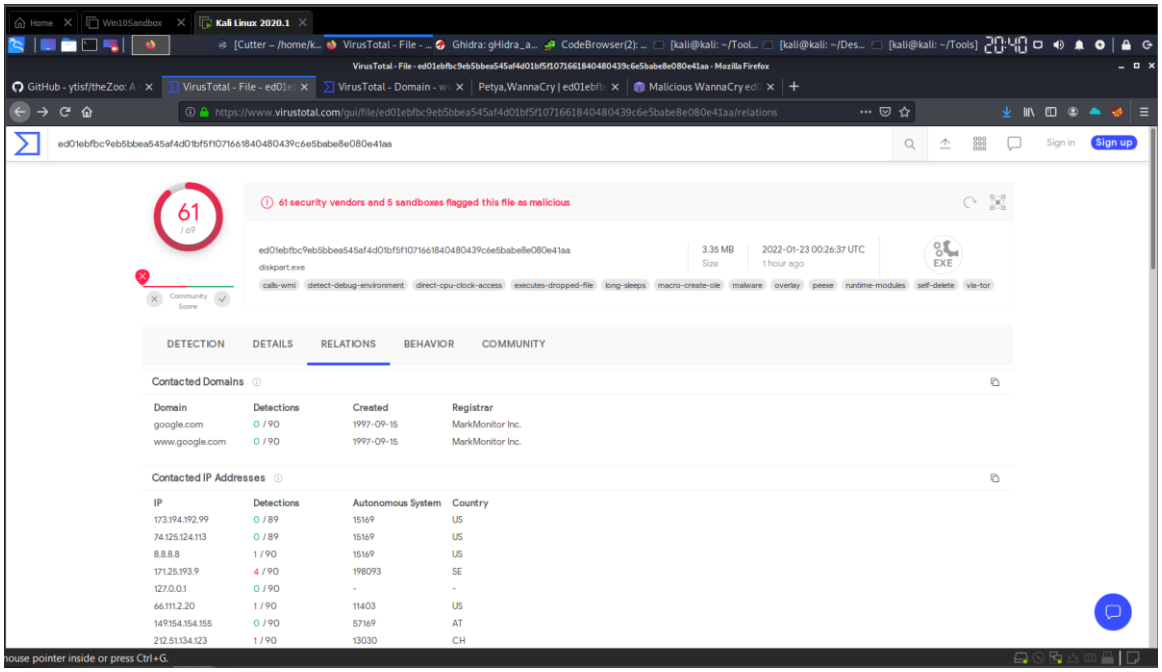


Ilustración 95: Relaciones detectadas de WannaCry en VirusTotal

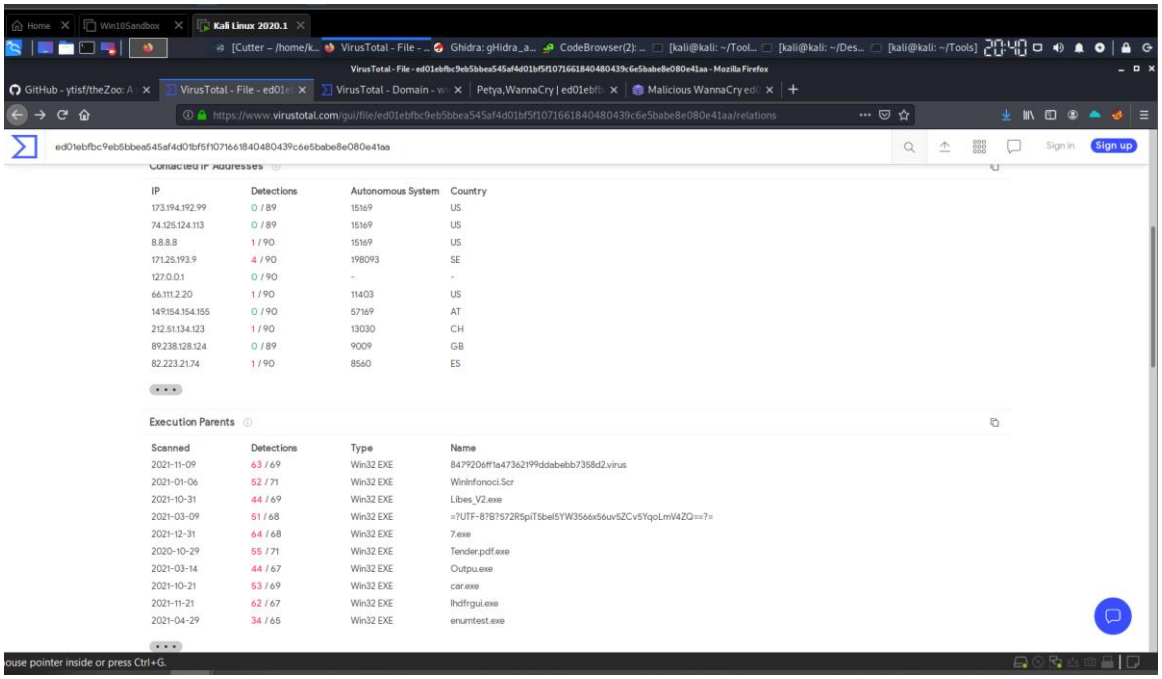


Ilustración 96:Archivos de ejecución padres de WannaCry en VirusTotal

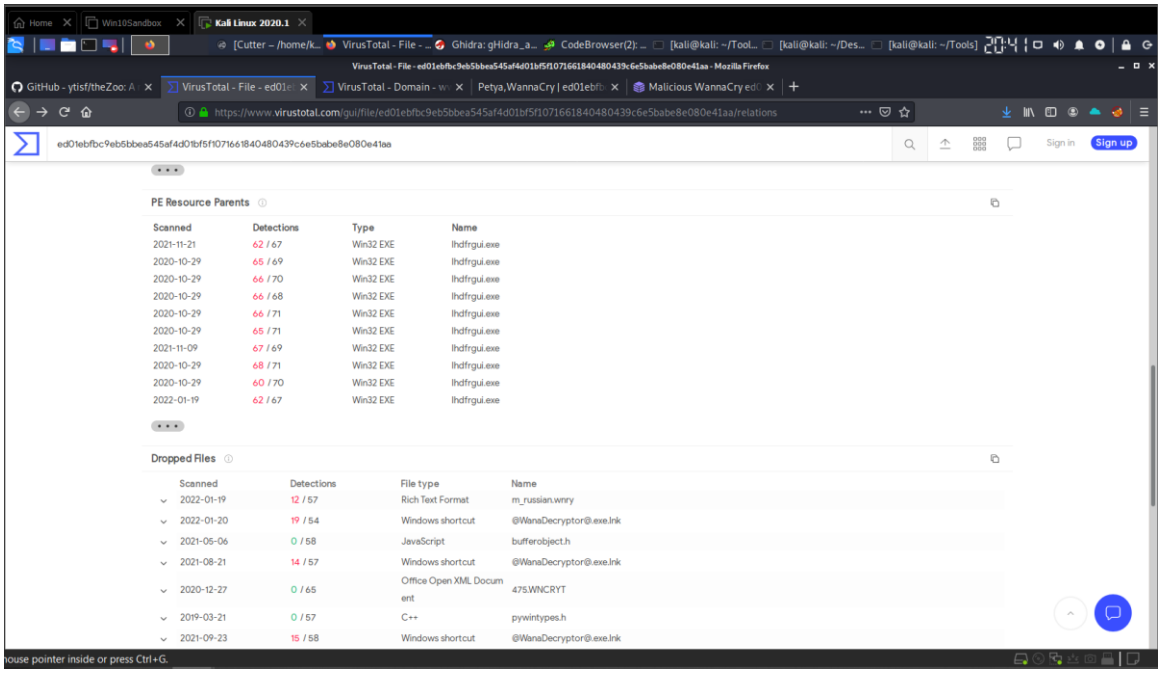


Ilustración 97: Archivos ejecutables portables por WannaCry

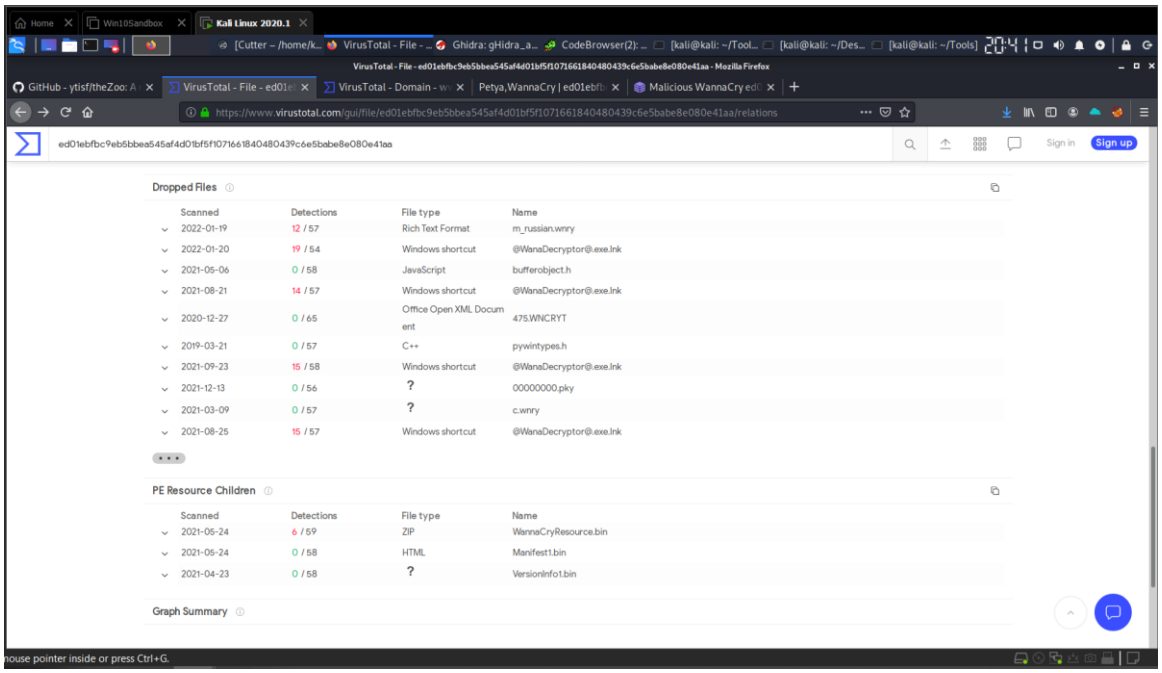


Ilustración 98: Archivos dejados y archivos ejecutables portables hijos de WannaCry

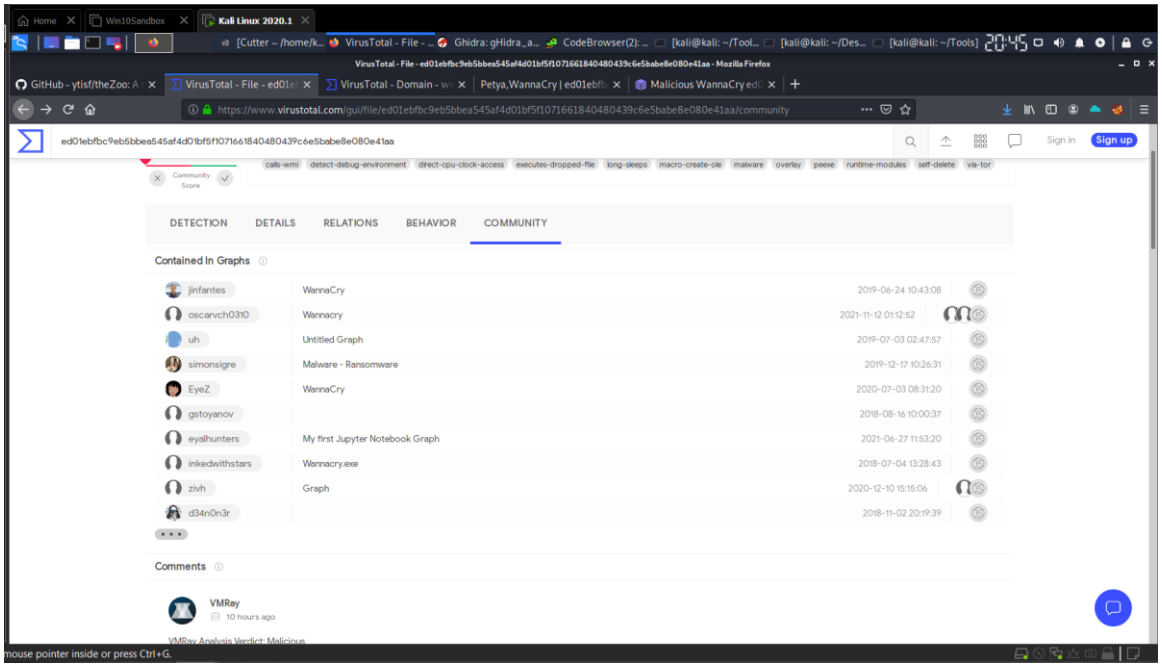


Ilustración 99: Aportes de la comunidad de VirusTotal sobre WannaCry

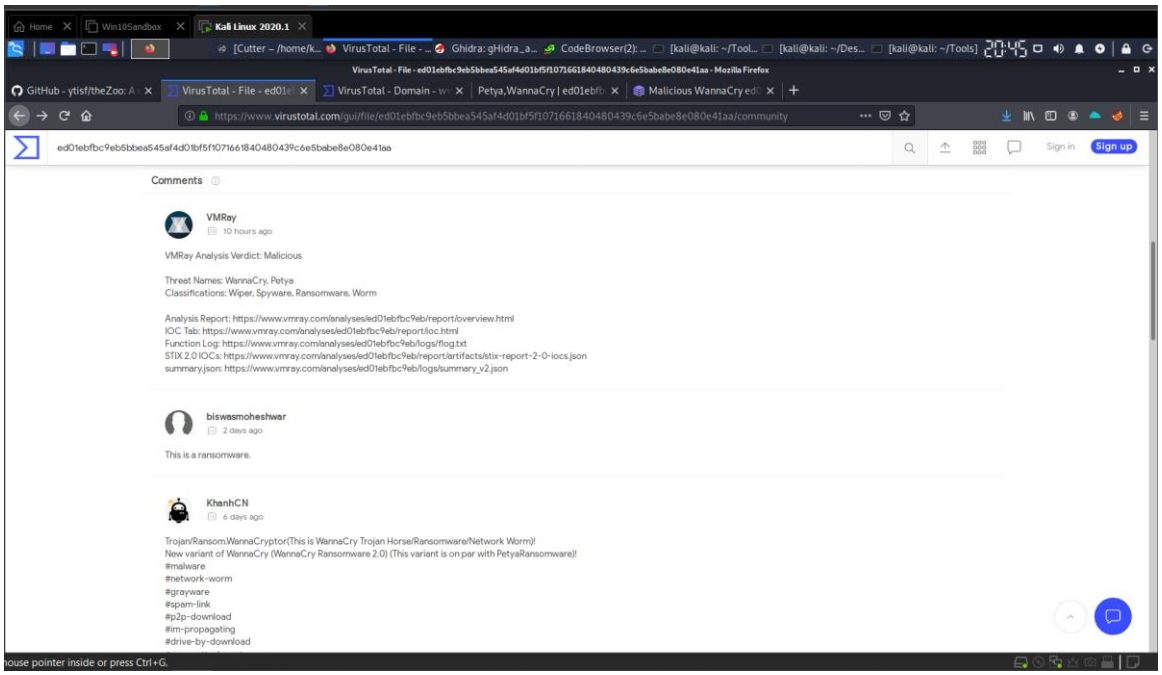


Ilustración 100: Comentarios de la comunidad sobre WannaCry en VirusTotal

El archivo ejecutable de WannaCry es subido a la plataforma *Intezer Analyze* para realizar un análisis dinámico y el análisis genético, proceso propio de la plataforma

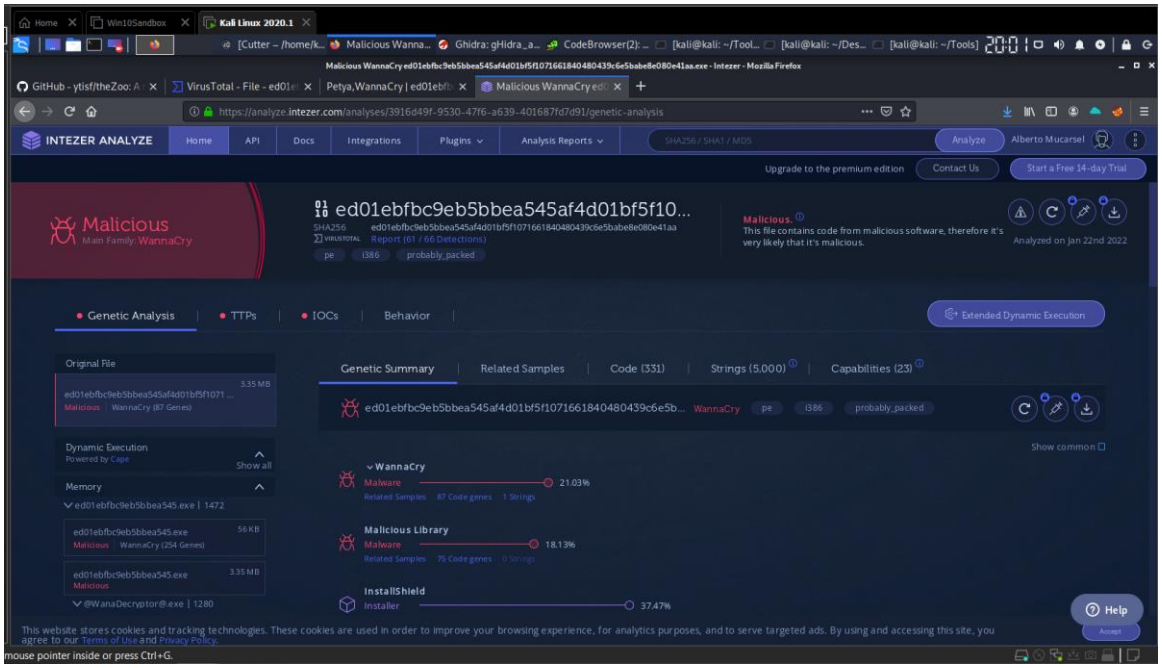


Ilustración 101: Resumen de análisis genético de WannaCry en Intezer

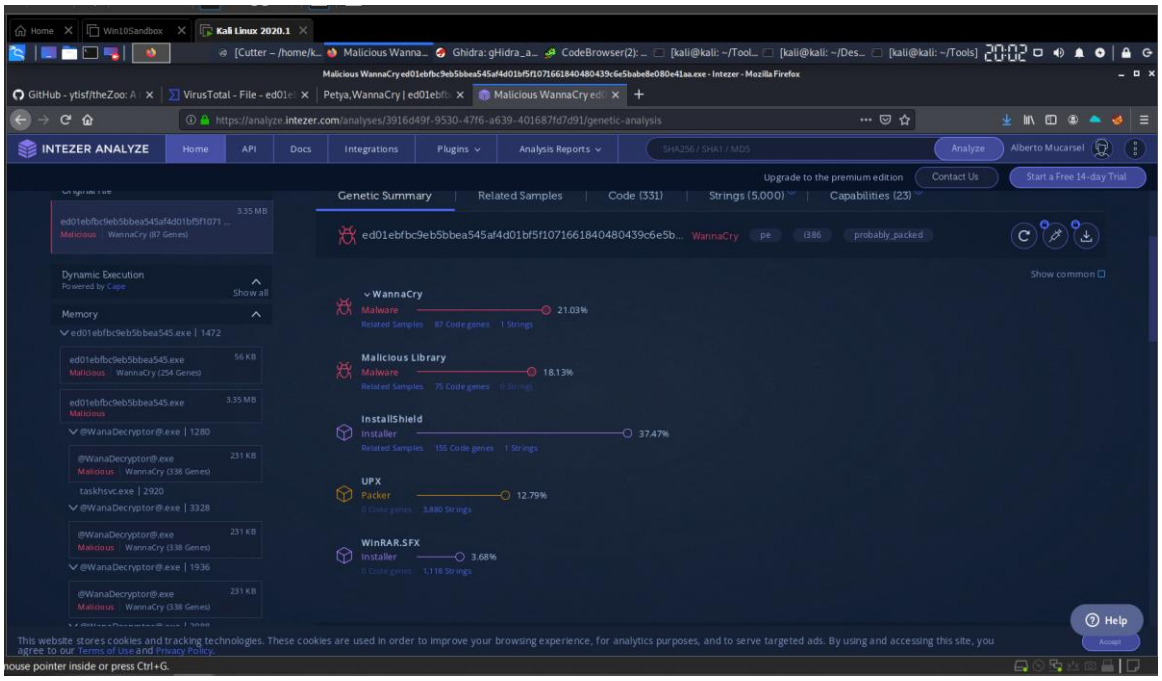


Ilustración 102: Genes detectados en WannaCry en Intezer

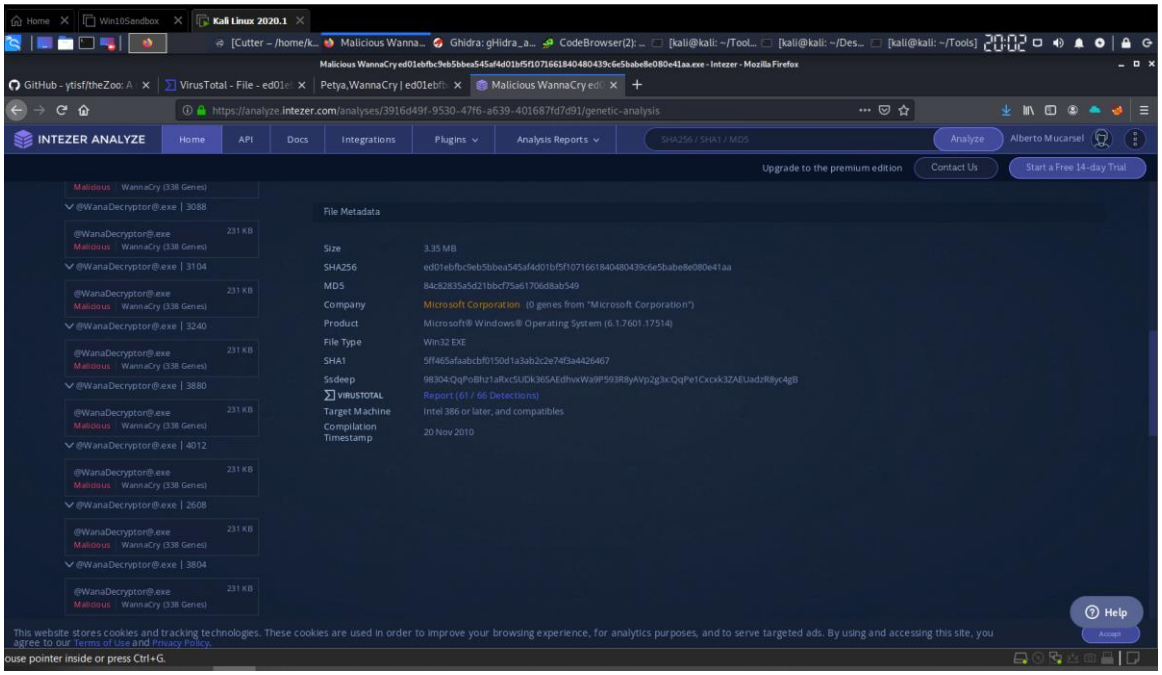


Ilustración 103: Metadata del archivo WannaCry

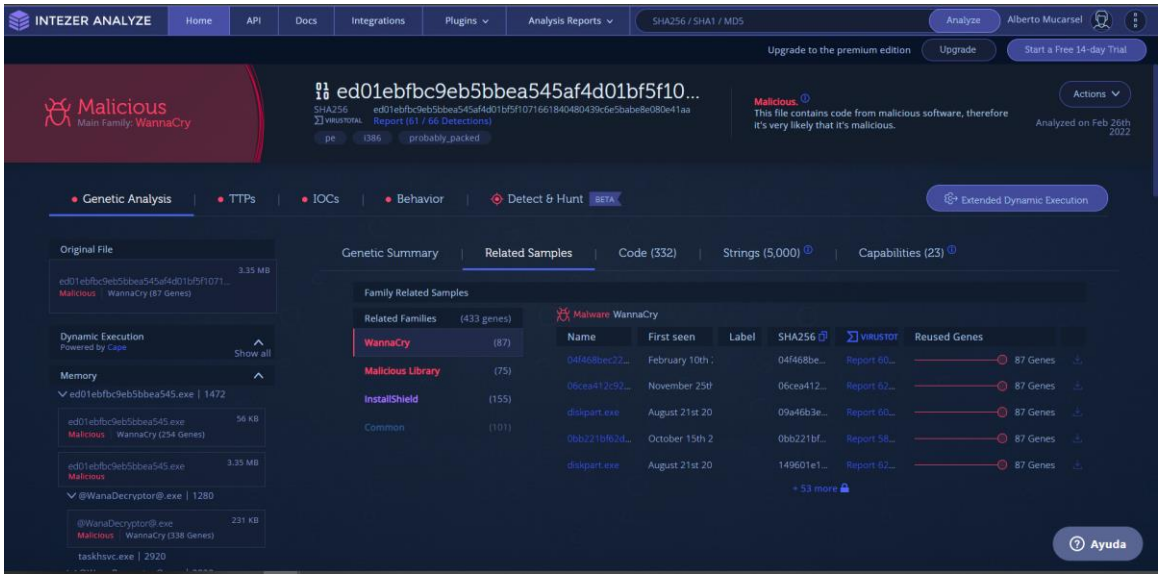


Ilustración 104: Muestras relacionadas a WannaCry en Intezer

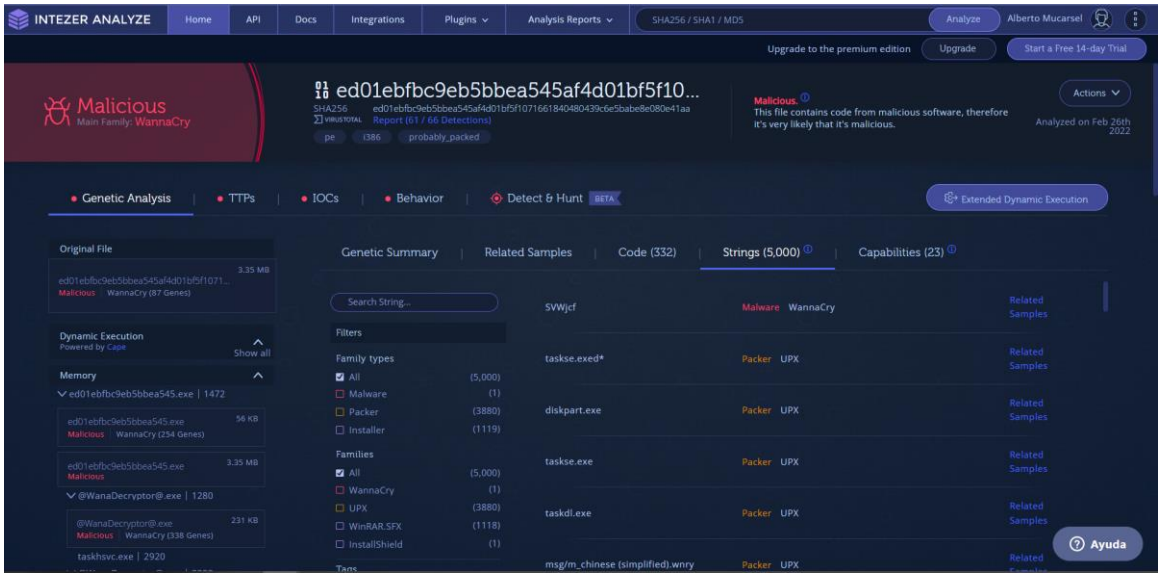


Ilustración 105: Strings encontradas dentro de WannaCry en Intezer

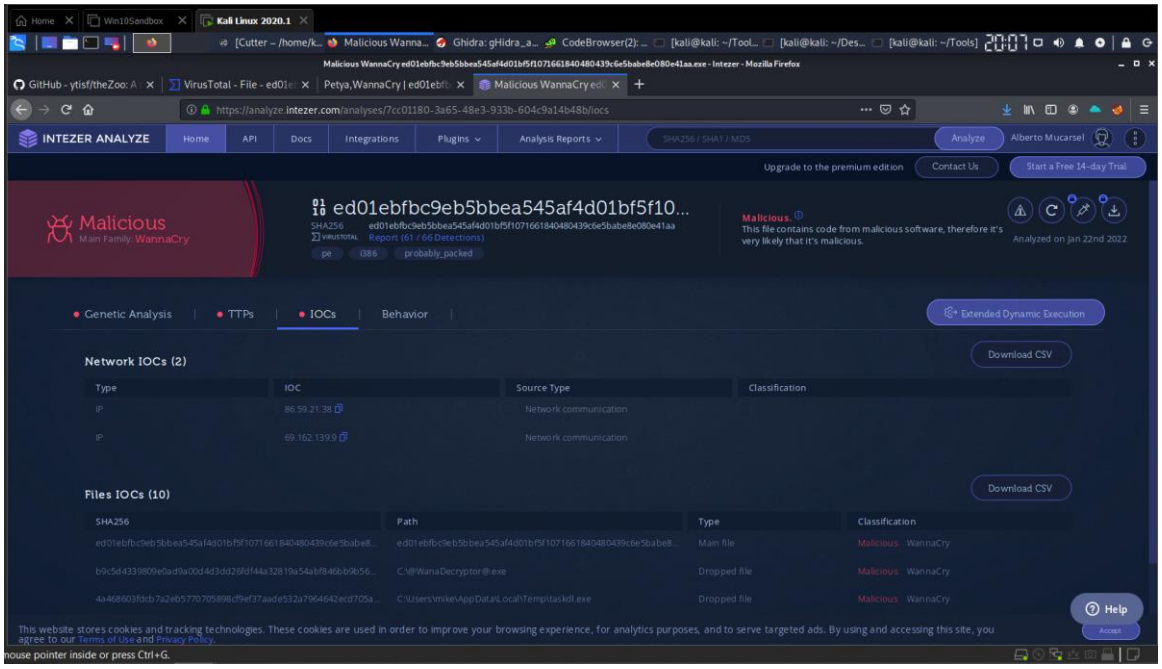


Ilustración 106: Detección de IoC dentro de WannaCry en Intezer

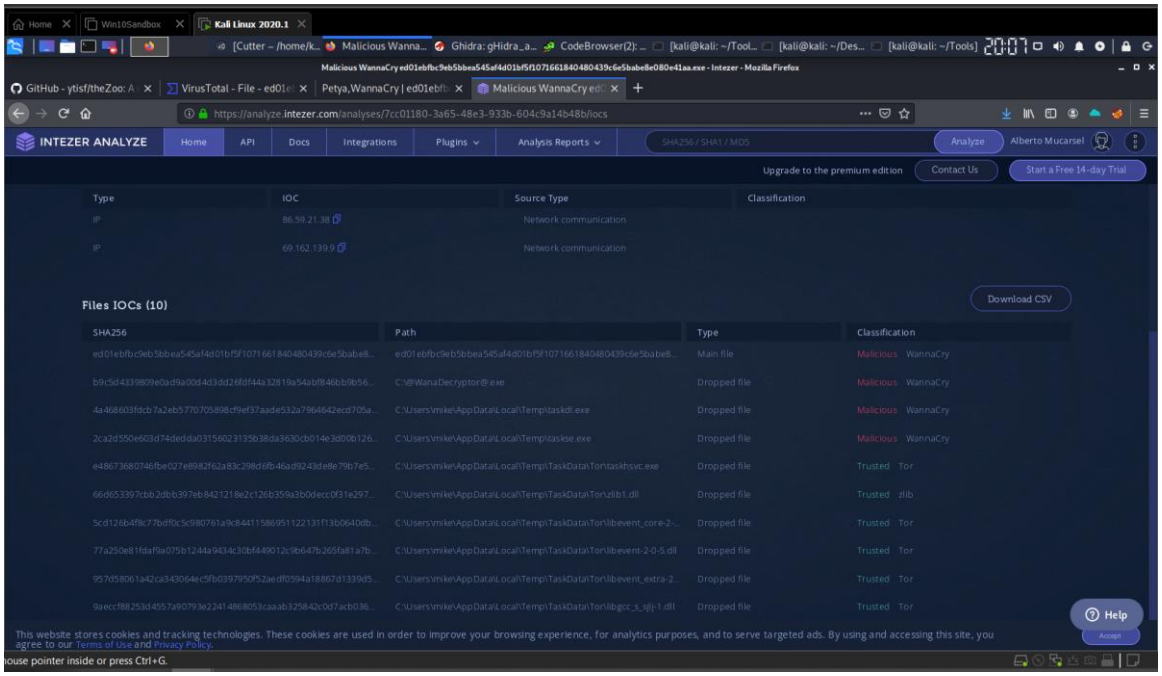


Ilustración 107: IoC de archivos encontrados dentro de WannaCry en Intezer

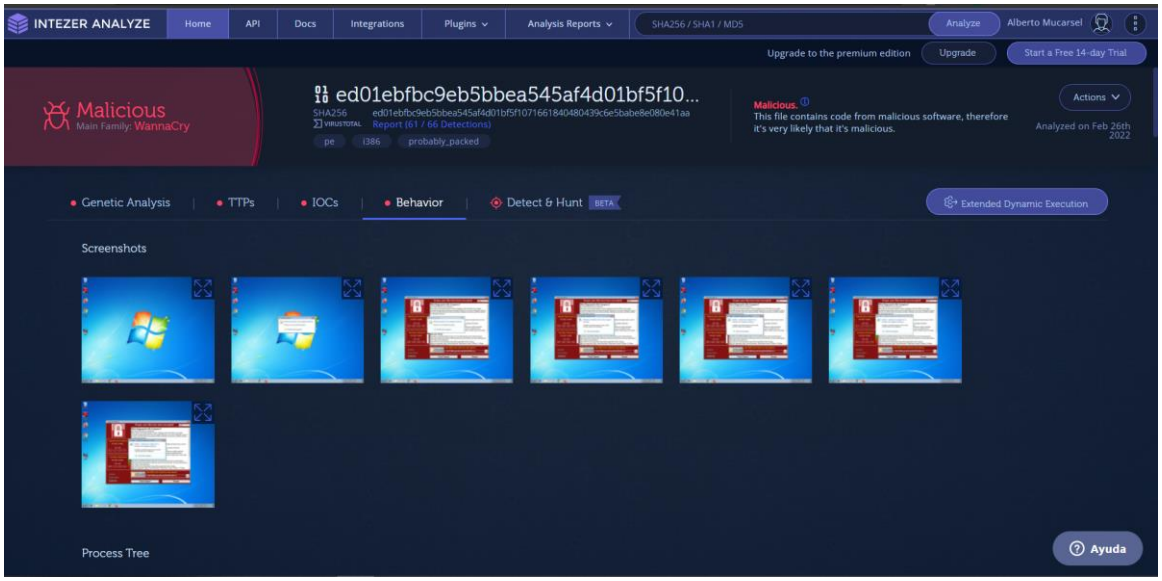


Ilustración 108: Comportamiento de WannaCry durante ejecución en entorno Sandbox en Intezer

Anexo C: Análisis realizado de la muestra de PoisonFrog

El archivo ejecutable de la muestra PoisonFrog es sometido a análisis estático con *Ghidra*

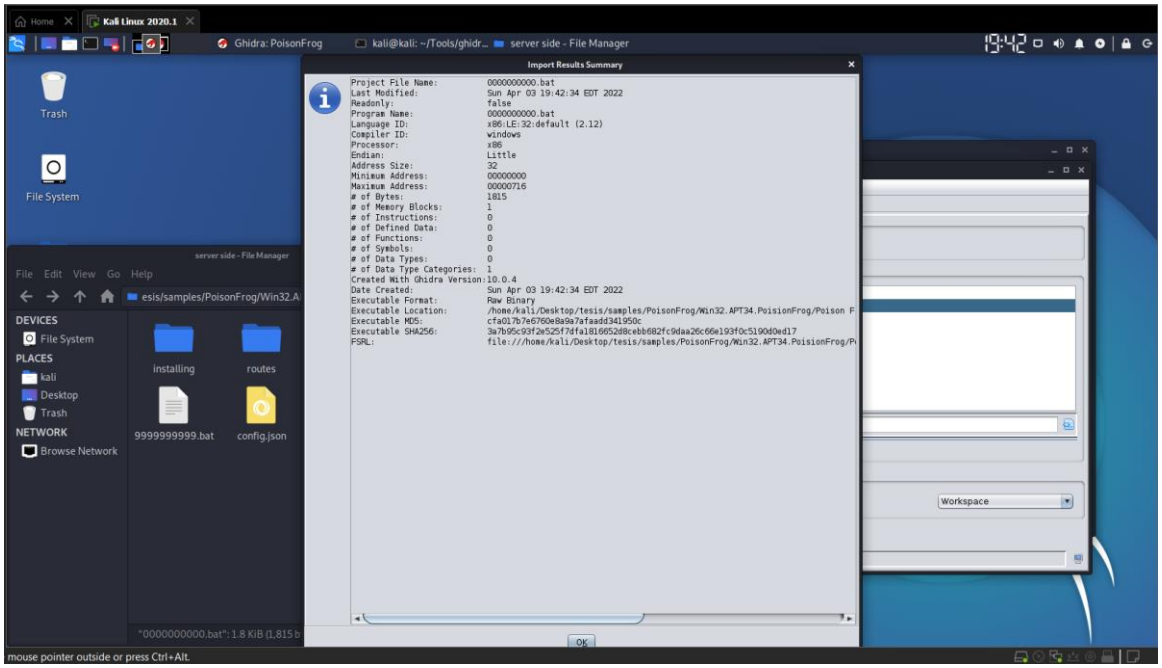


Ilustración 109: Resumen de resultados de análisis de PoisonFrog en Ghidra

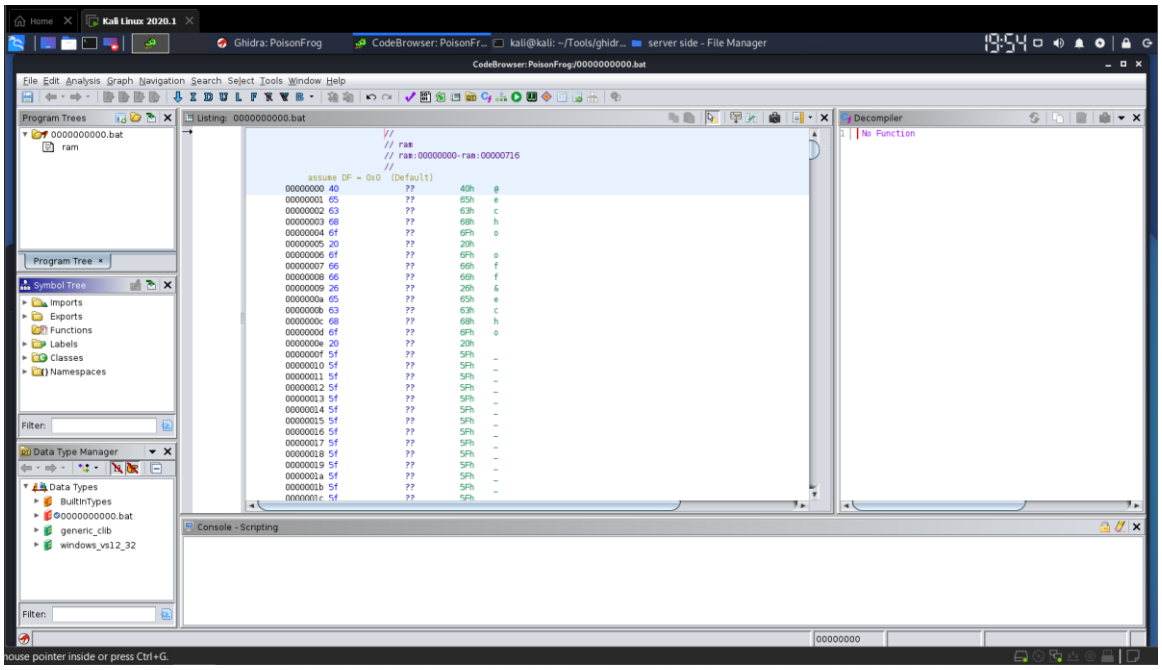


Ilustración 110: análisis de 000000000.bat (PoisonFrog) en Ghidra

Tras *Ghidra*, se realiza otro análisis estático de PoisonFrog en *Cutter*

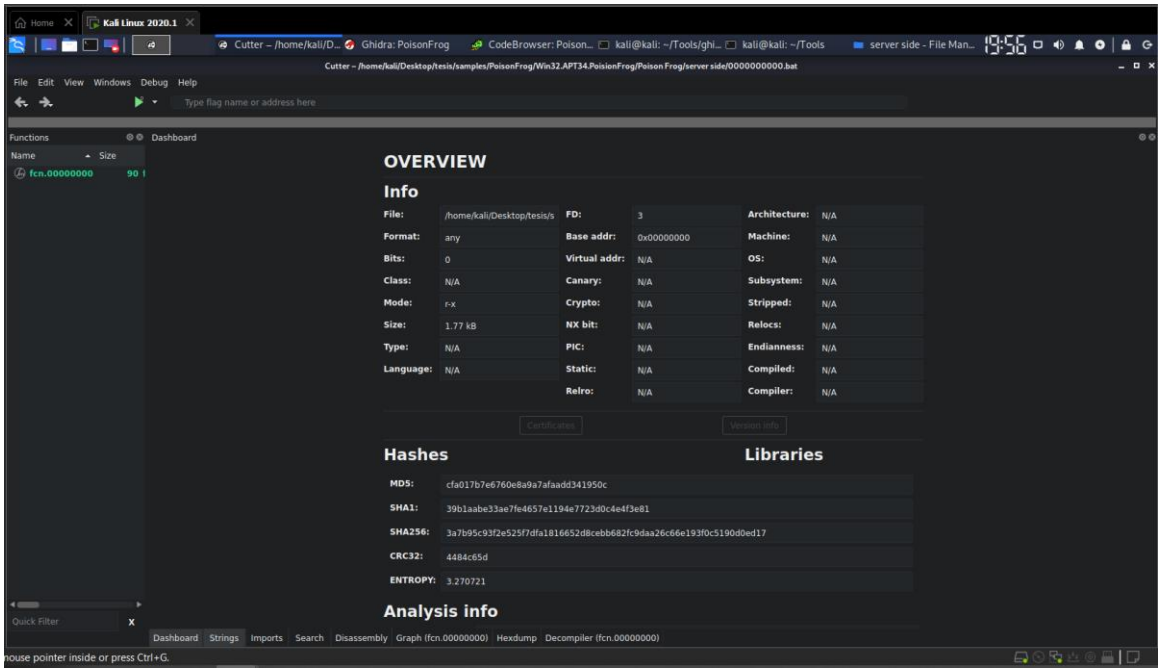


Ilustración 111: Resumen de información sobre muestra PoisonFrog

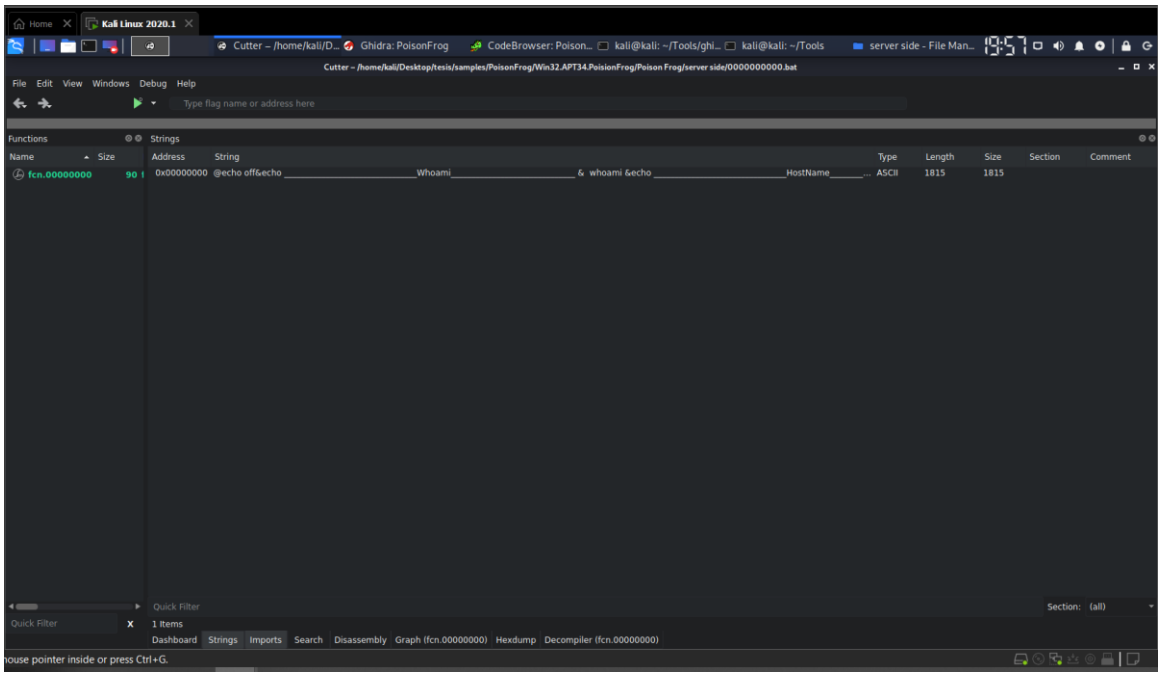


Ilustración 112: Extracción de strings de PoisonFrog en Cutter

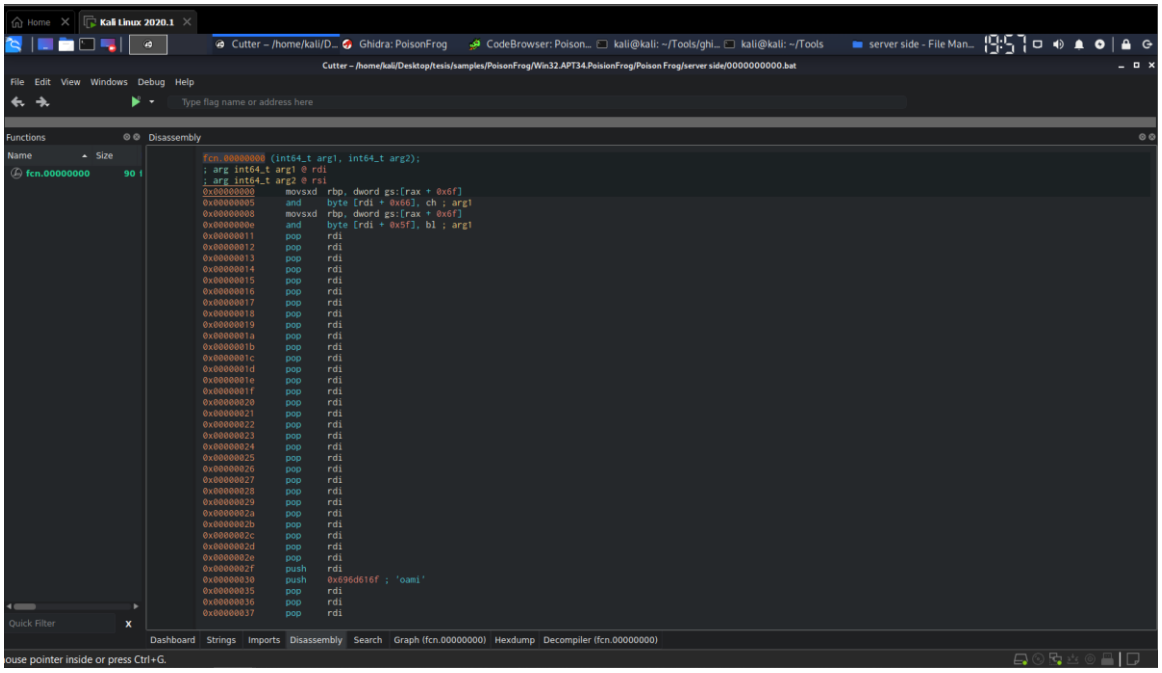


Ilustración 113: Disassembly code de PoisonFrog en Cutter

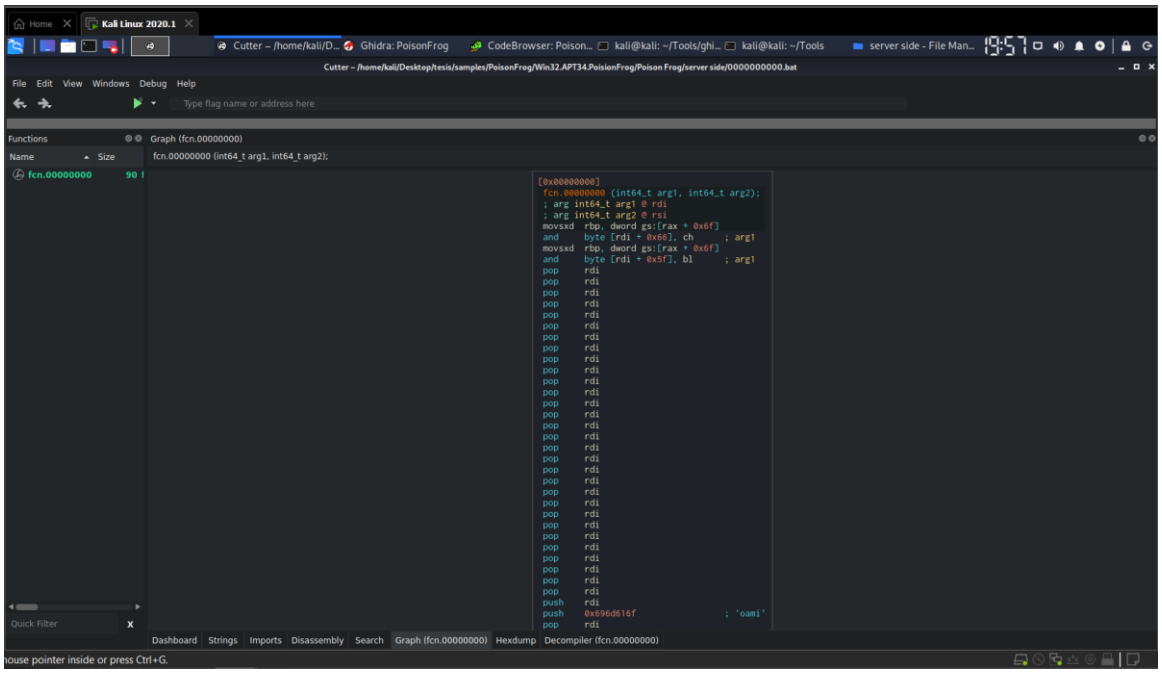


Ilustración 114: Grafo de ejecución de PoisonFrog en Cutter

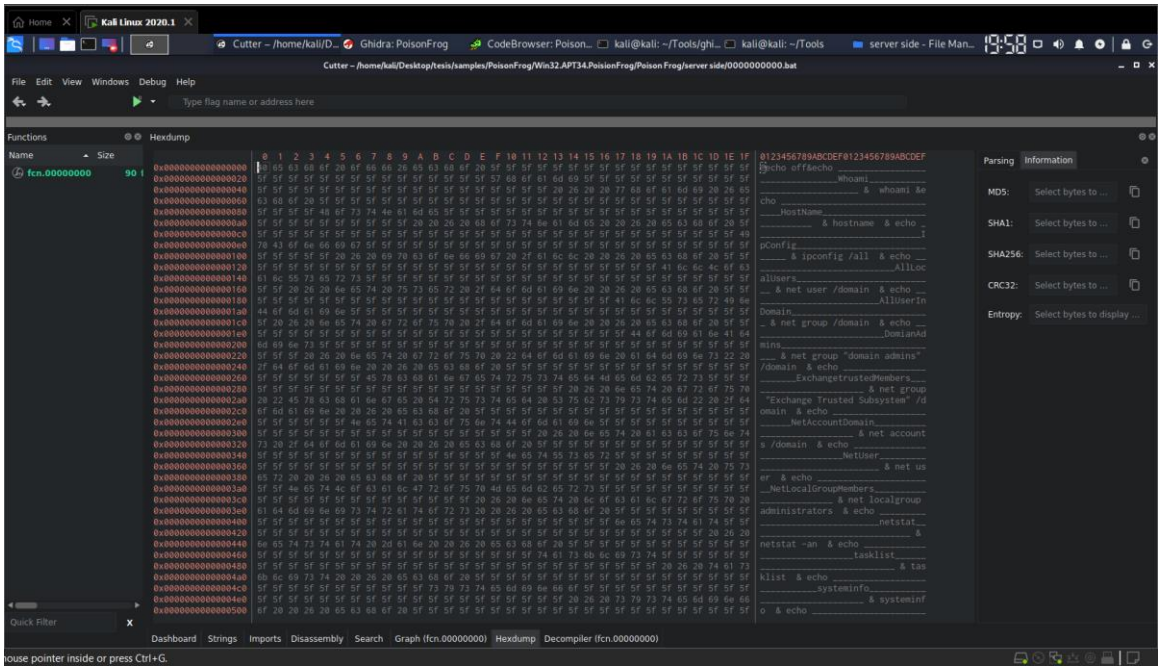


Ilustración 115: Vista hexadecimal de código de PoisonFrog en Cutter

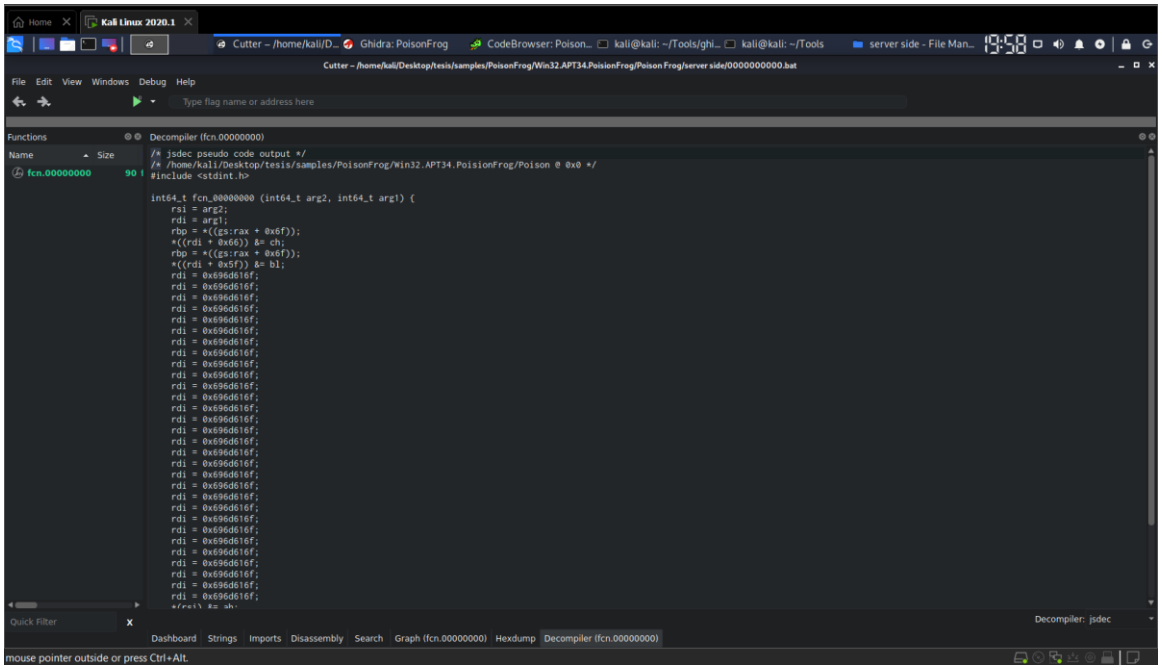


Ilustración 116: Código descompilado de PoisonFrog en Cutter

El archivo ejecutable con la muestra de PoisonFrog es subido a la plataforma *VirusTotal* para un análisis dinámico y la URL pública de su resultado es: https://www.virustotal.com/gui/file/3a7b95c93f2e525f7dfa1816652d8cebb682fc9daa26c66e193f0c5190d0ed17

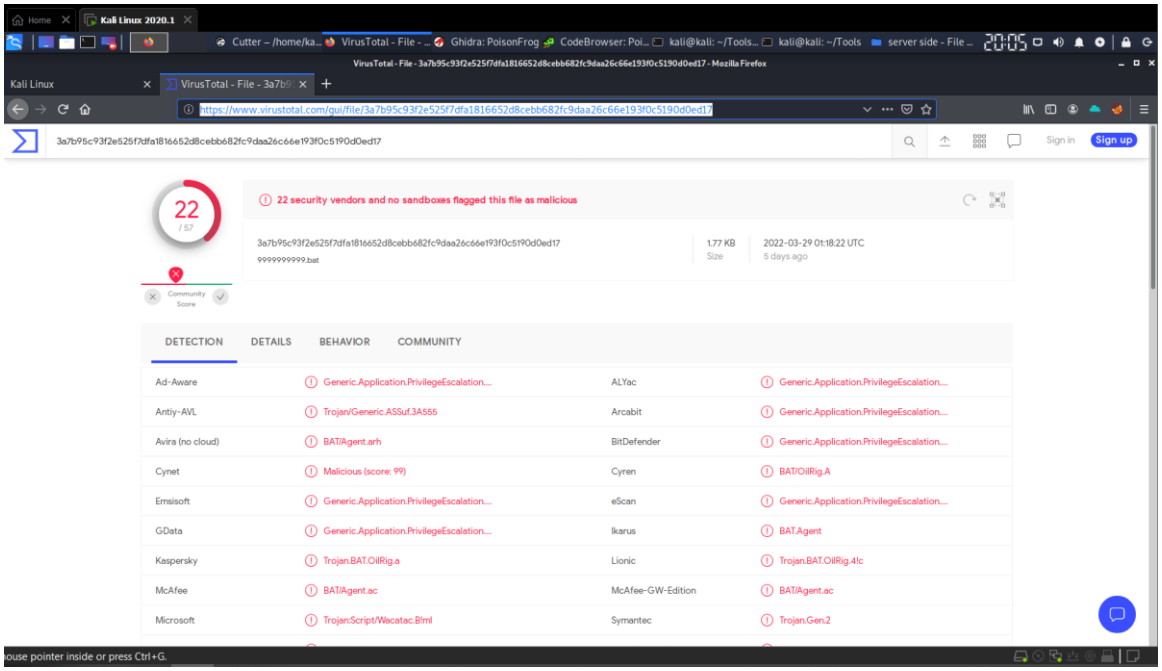


Ilustración 117: Listado de antivirus que detectaron a PoisonFrog

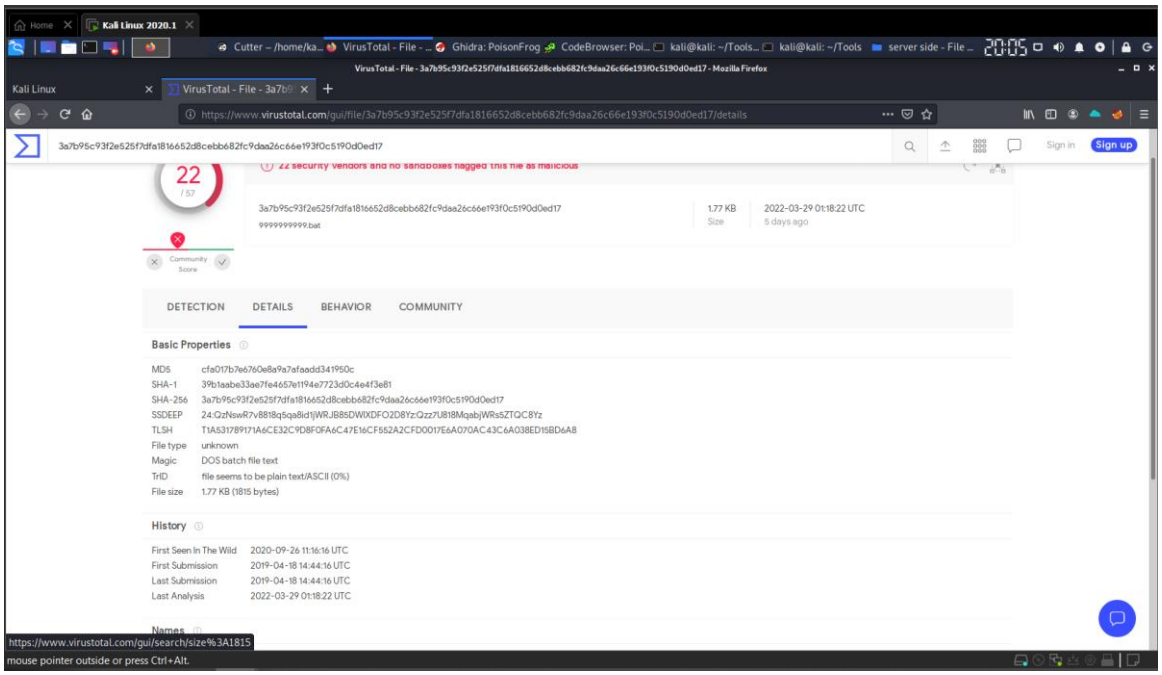


Ilustración 118: Detalles sobre la muestra de PoisonFrog en VirusTotal

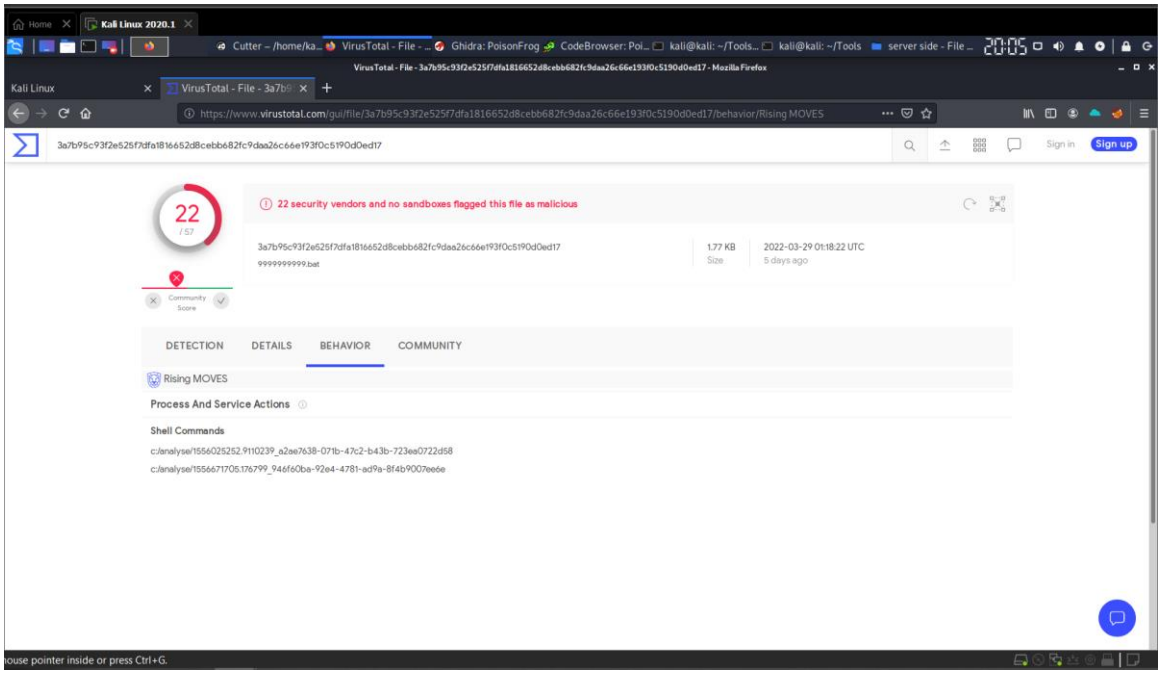


Ilustración 119: Comportamiento observado de PoisonFrog en ejecución en VirusTotal

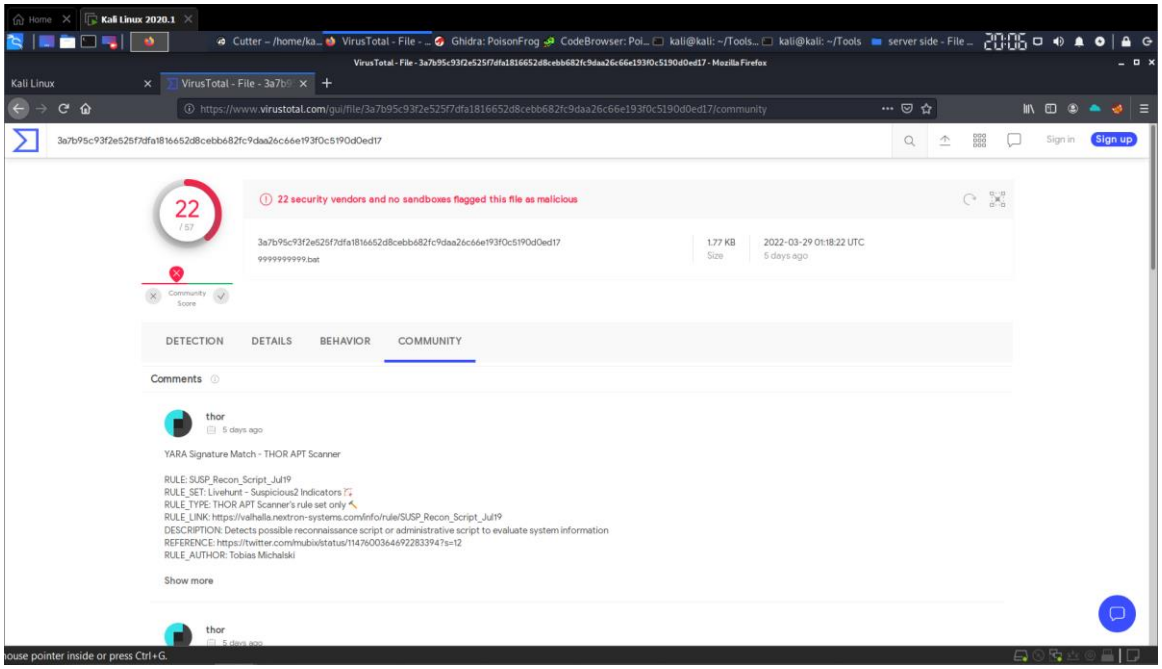


Ilustración 120: Aportes de la comunidad sobre PoisonFrog en VirusTotal

El archivo ejecutable de PoisonFrog es subido a la plataforma *Intezer Analyze* para realizar un análisis dinámico y el análisis genético, proceso propio de la plataforma

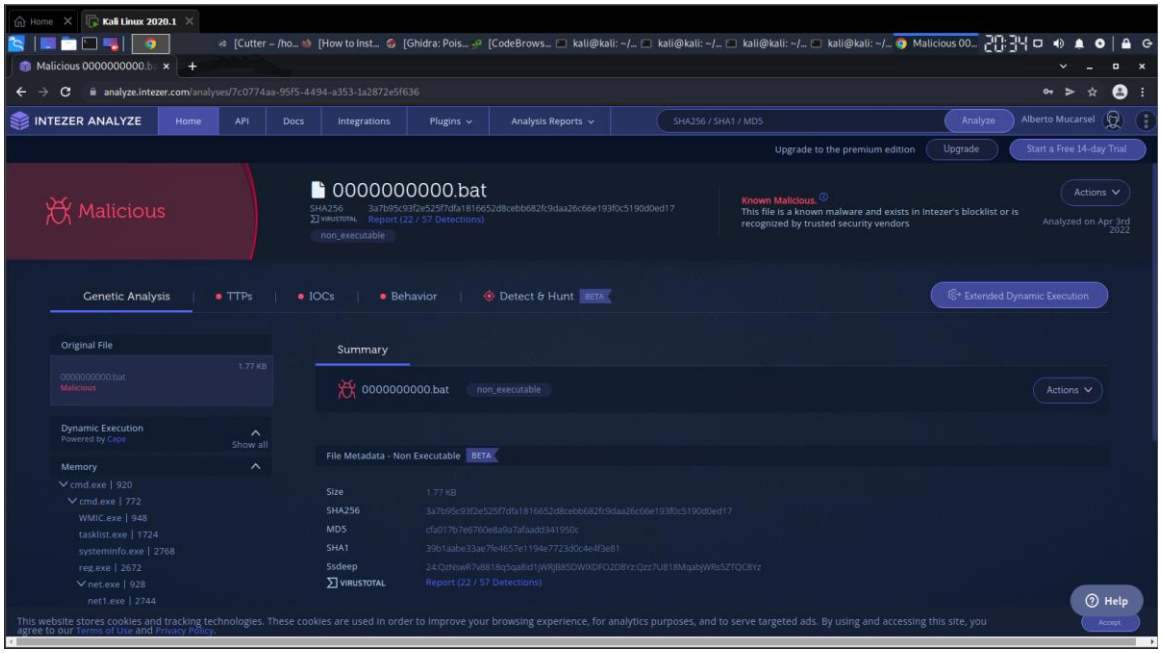


Ilustración 121: Resumen de análisis genético de PoisonFrog en Intezer

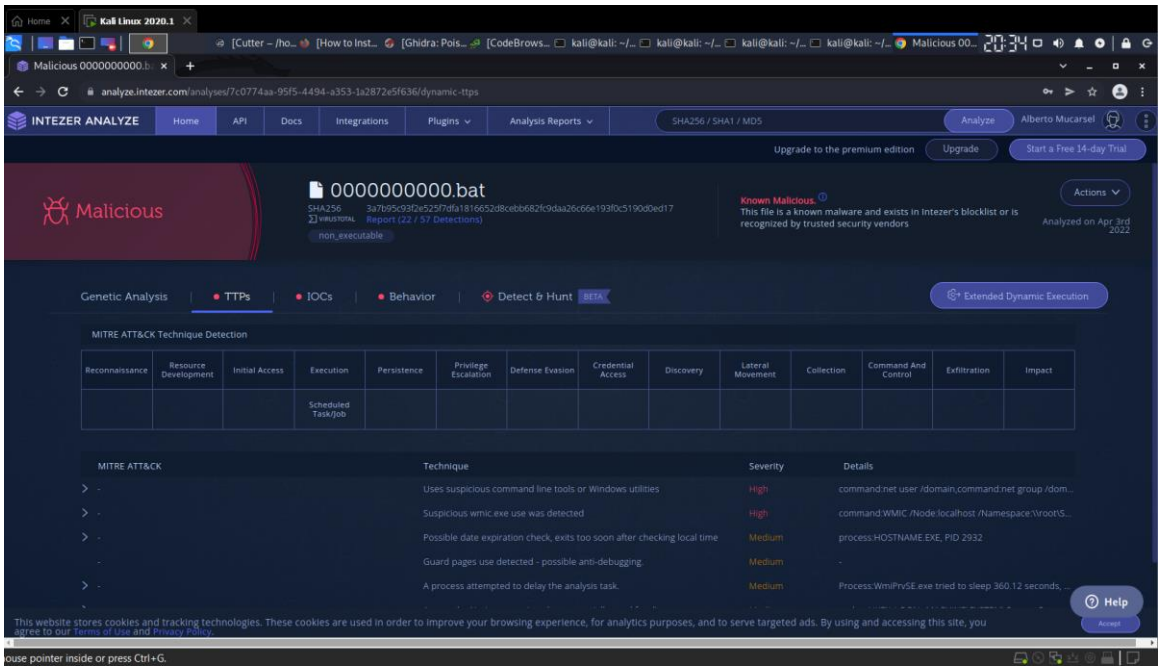


Ilustración 122: Detección de técnicas usadas por PoisonFrog en Intezer

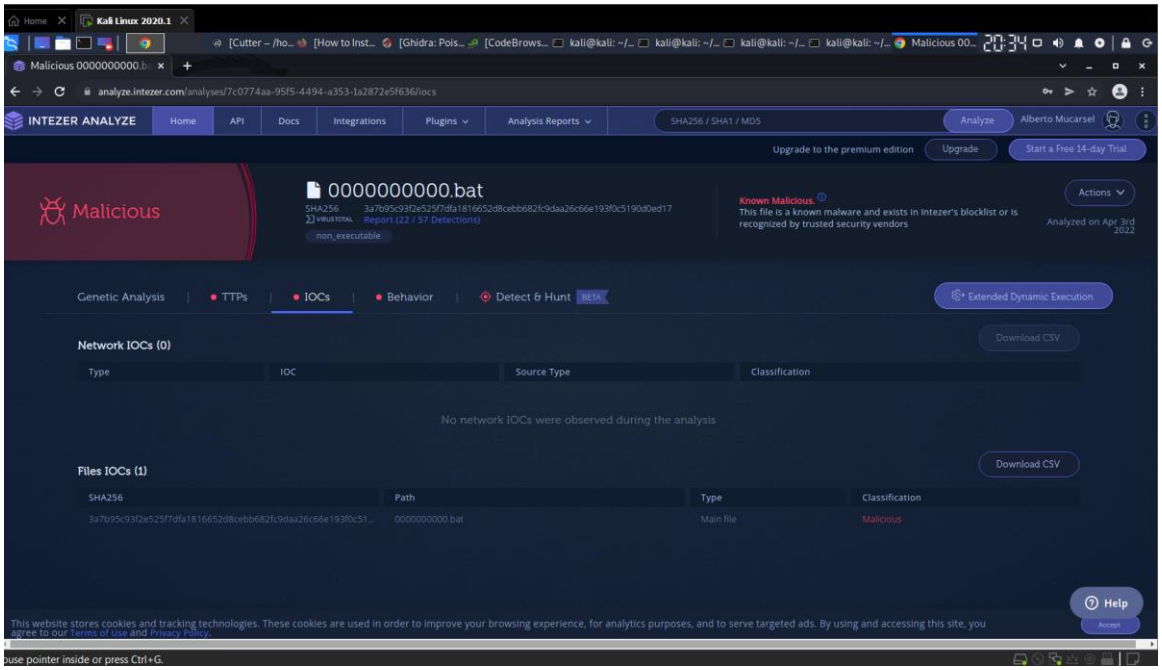


Ilustración 123: IoC detectados dentro de PoisonFrog en Intezer

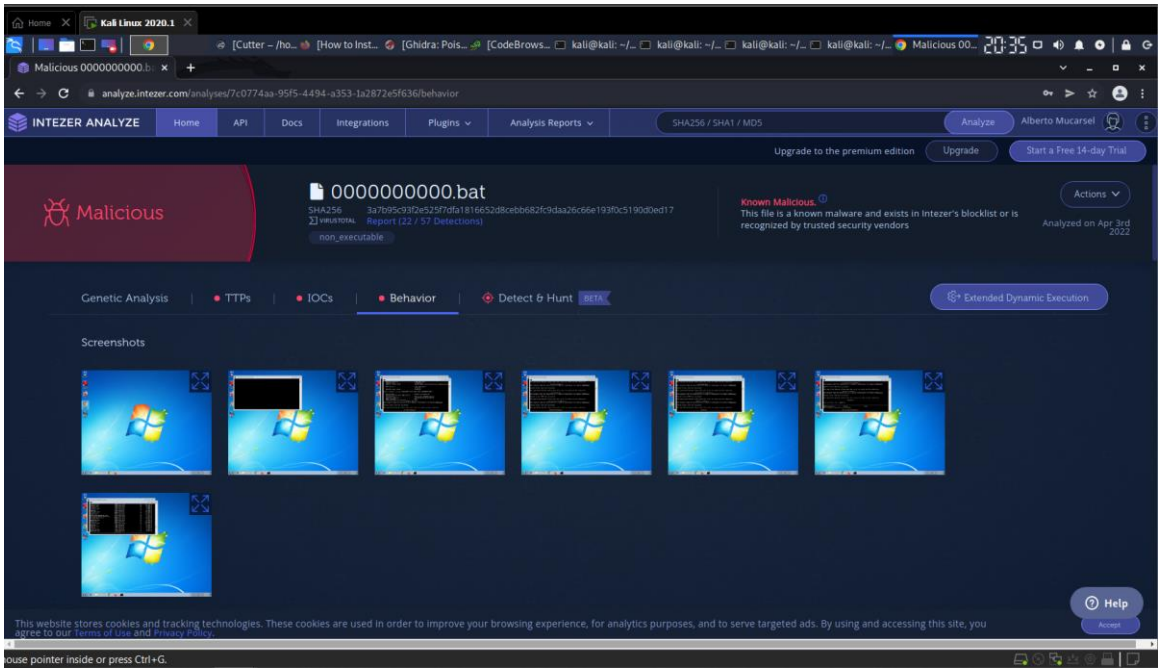


Ilustración 124: Comportamiento de PoisonFrog durante ejecución en entorno Sandbox en Intezer

Una vez la máquina víctima se encuentra infectada con PoisonFrog, se procede a la extracción del volcado de memoria y su envió a la máquina Kali para su análisis con Volatility.

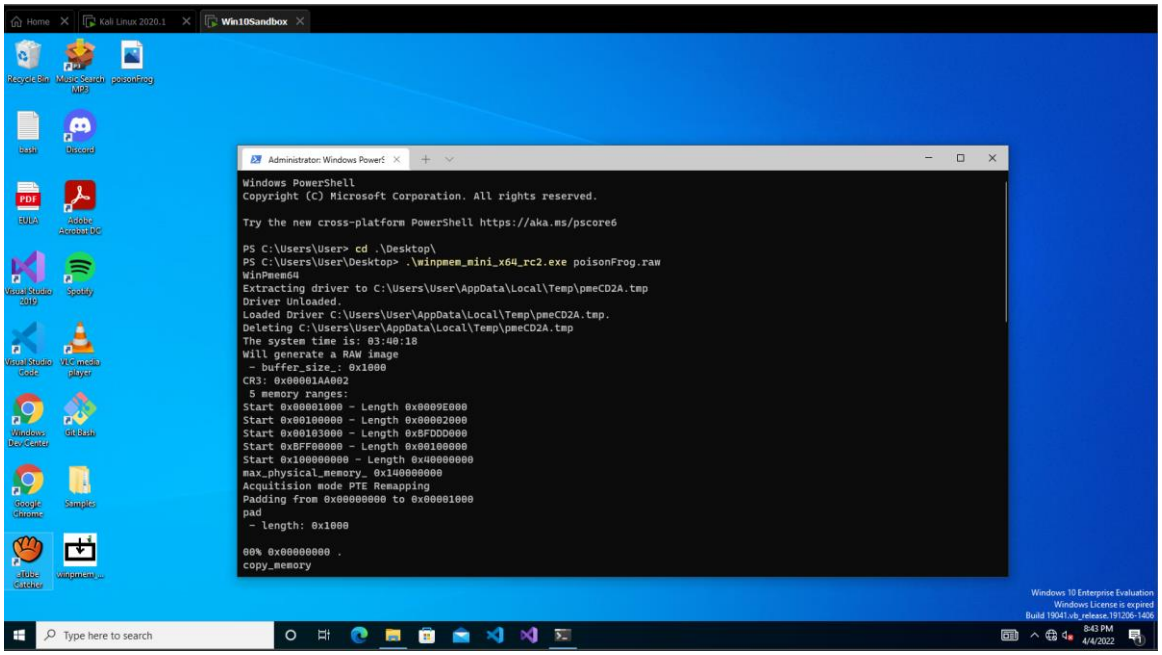


Ilustración 125: Obtención de volcado de memoria tras infección con PoisonFrog

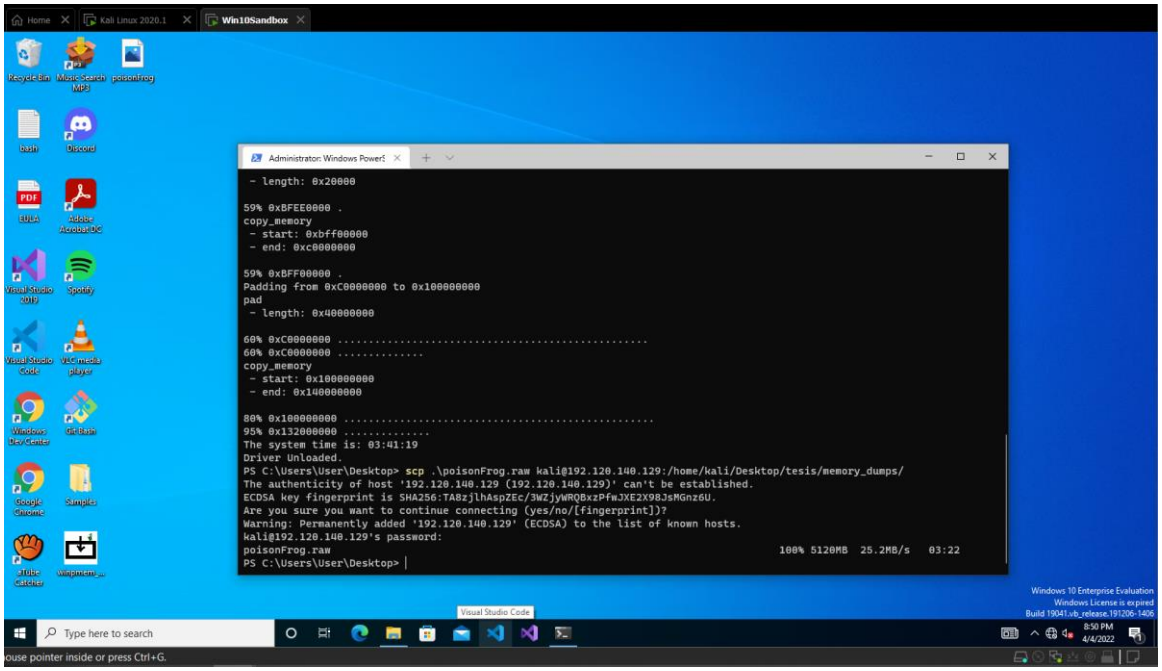


Ilustración 126: Envío de volcado de memoria de PoisonFrog hacia la máquina principal

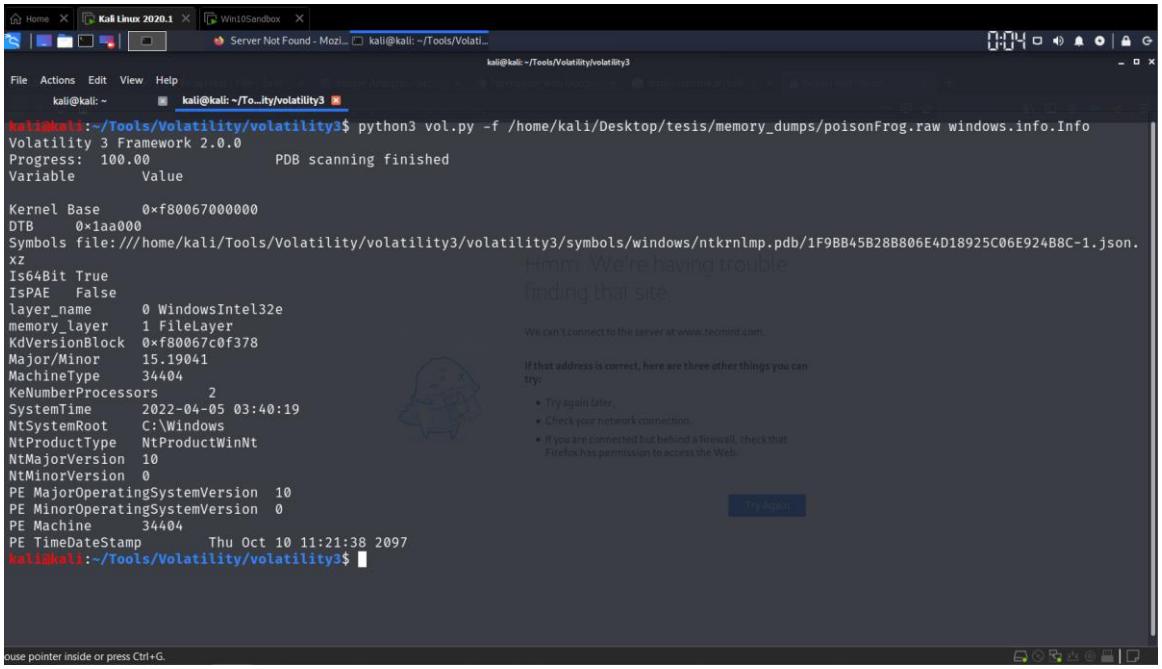


Ilustración 127: información general sobre el volcado de memoria de máquina infectada con PoisonFrog

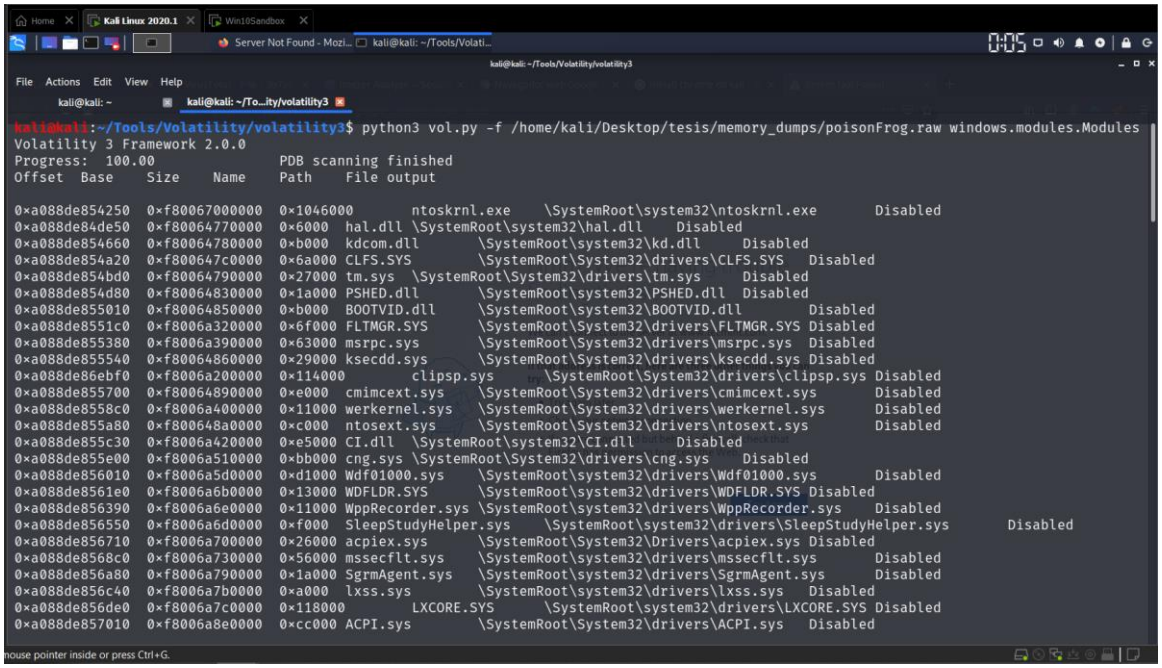


Ilustración 128: Resultados de extracción de información de plugin windows.modules.Modules con volcado de memoria de PoisonFrog

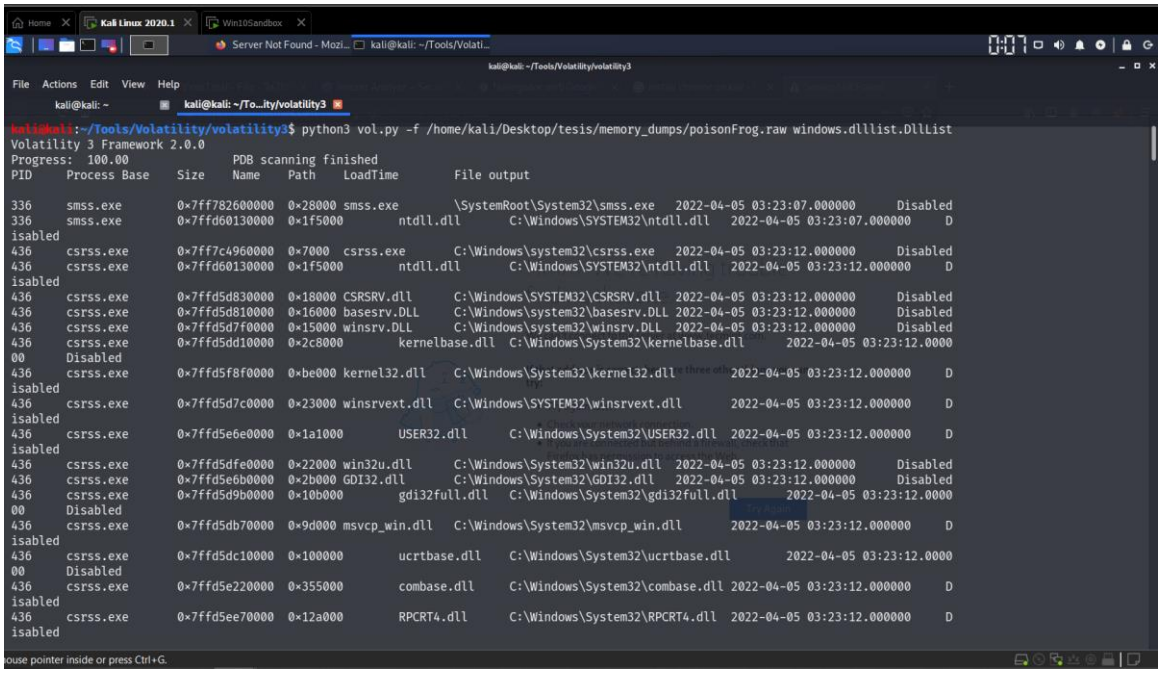


Ilustración 129: Resultados de extracción de información de plugin windows.dlllist.DllList con volcado de memoria de PoisonFrog

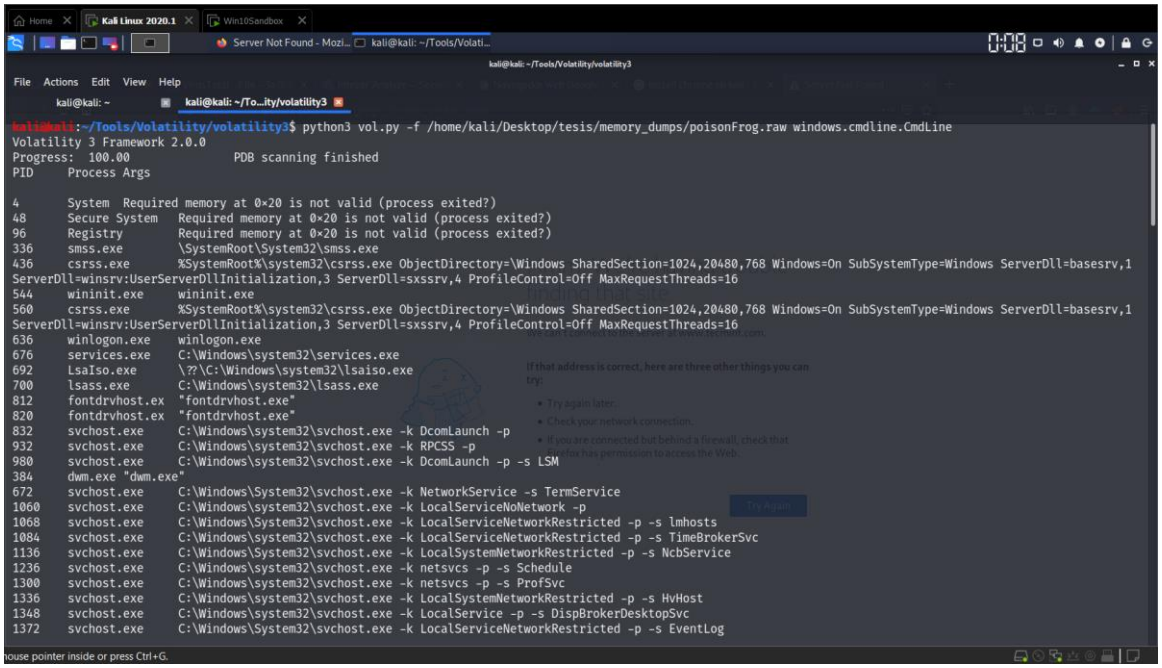


Ilustración 130: Resultados de extracción de información de plugin windows.cmdline.CmdLine con volcado de memoria de PoisonFrog

Anexo D: Análisis realizado de la muestra BadRabbit

El archivo de la muestra BadRabbit es sometido a análisis estático con *Ghidra*

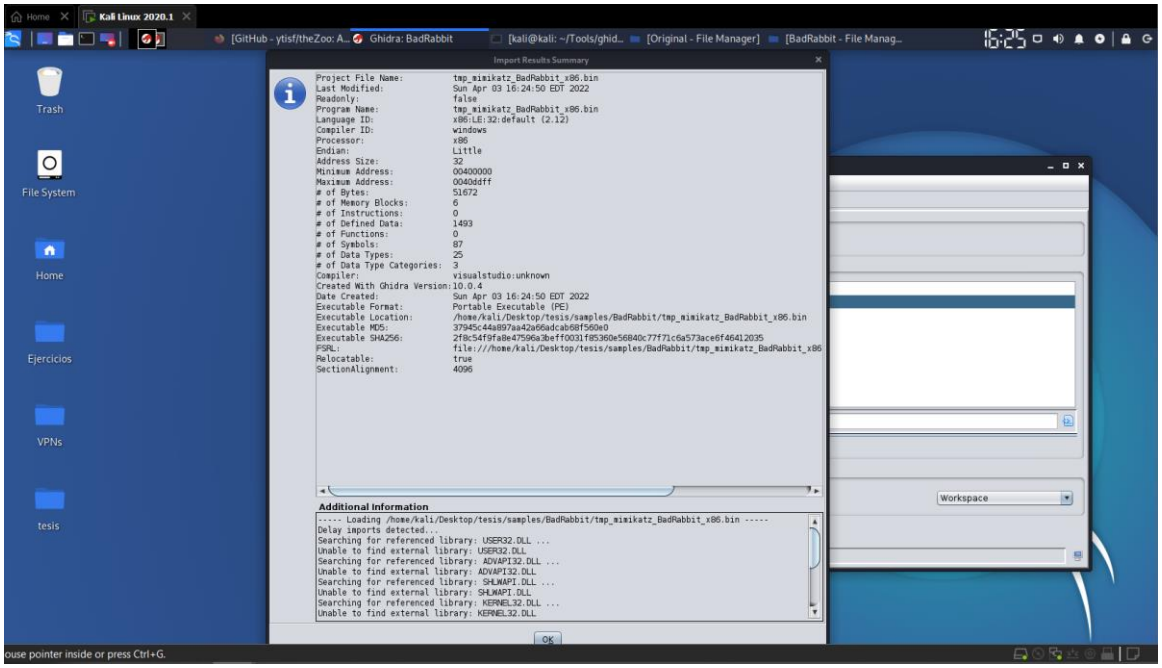


Ilustración 131: Resumen de resultados de análisis de BadRabbit en Ghidra

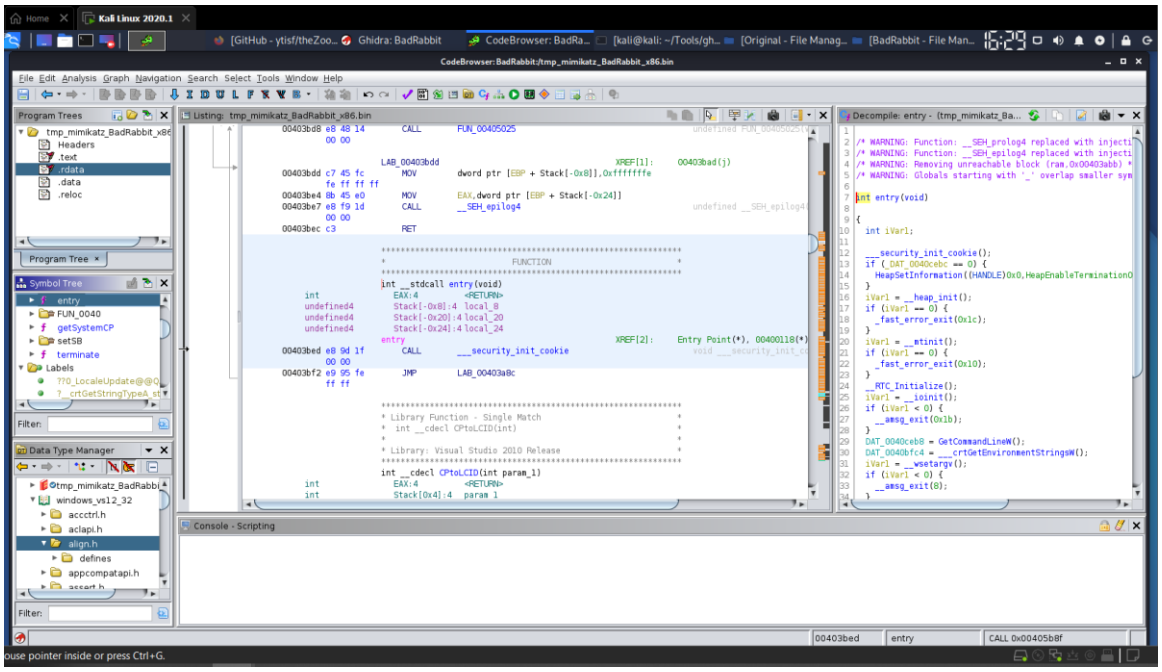


Ilustración 132: Revisión de entry point de BadRabbit

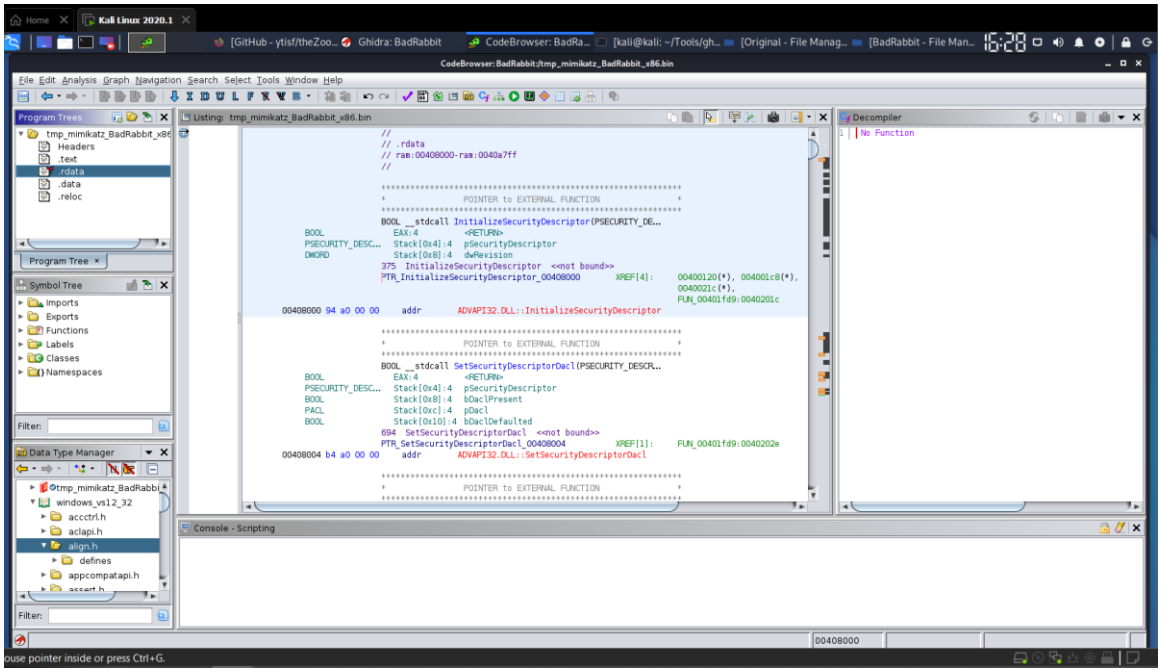


Ilustración 133: Revisión de secciones de BadRabbit

Tras *Ghidra*, se realiza otro análisis estático de BadRabbit en *Cutter*

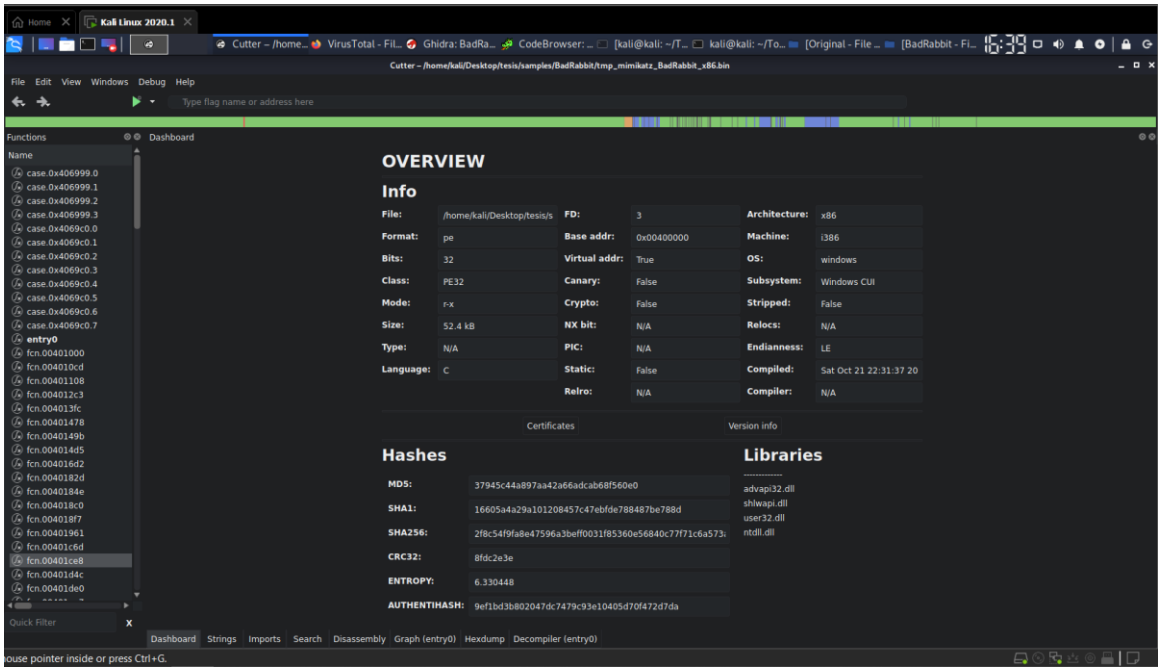


Ilustración 134: Resumen de información sobre muestra BadRabbit

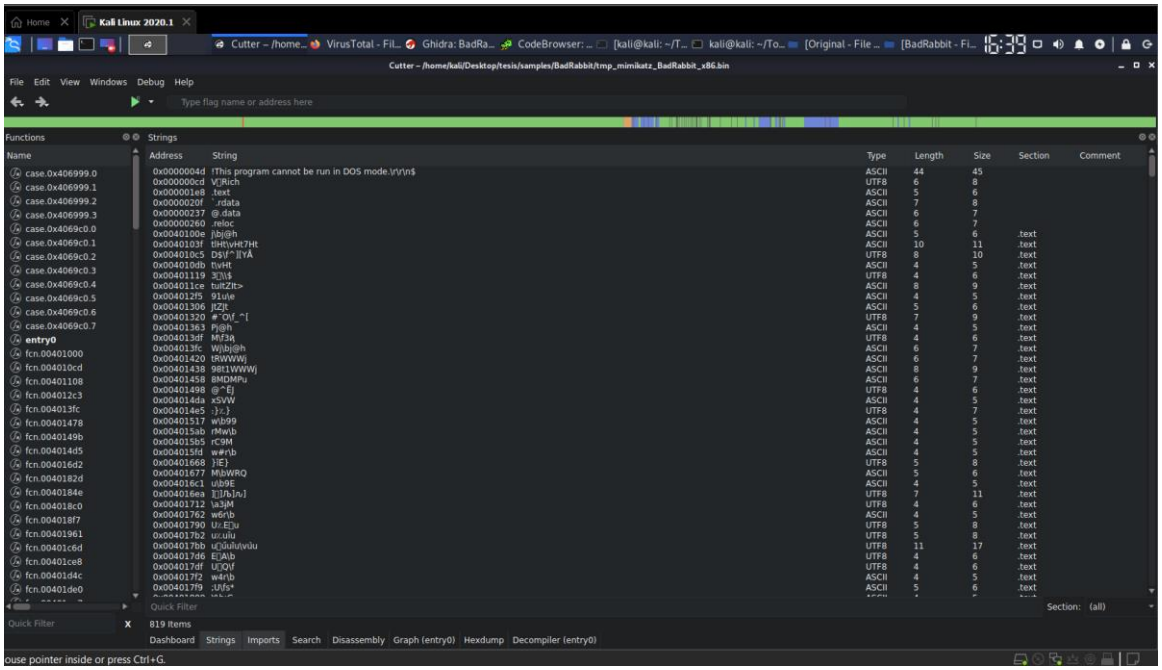


Ilustración 135: Extracción de strings de BadRabbit en Cutter

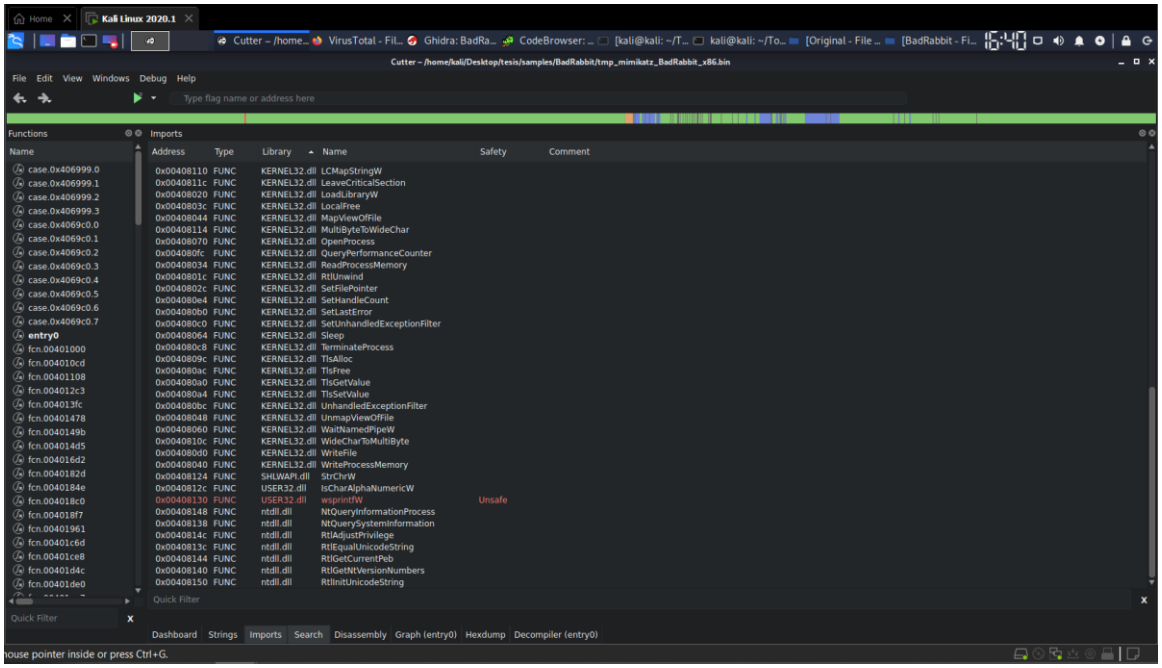


Ilustración 136: Detección de módulos inseguros importados en muestra BadRabbit

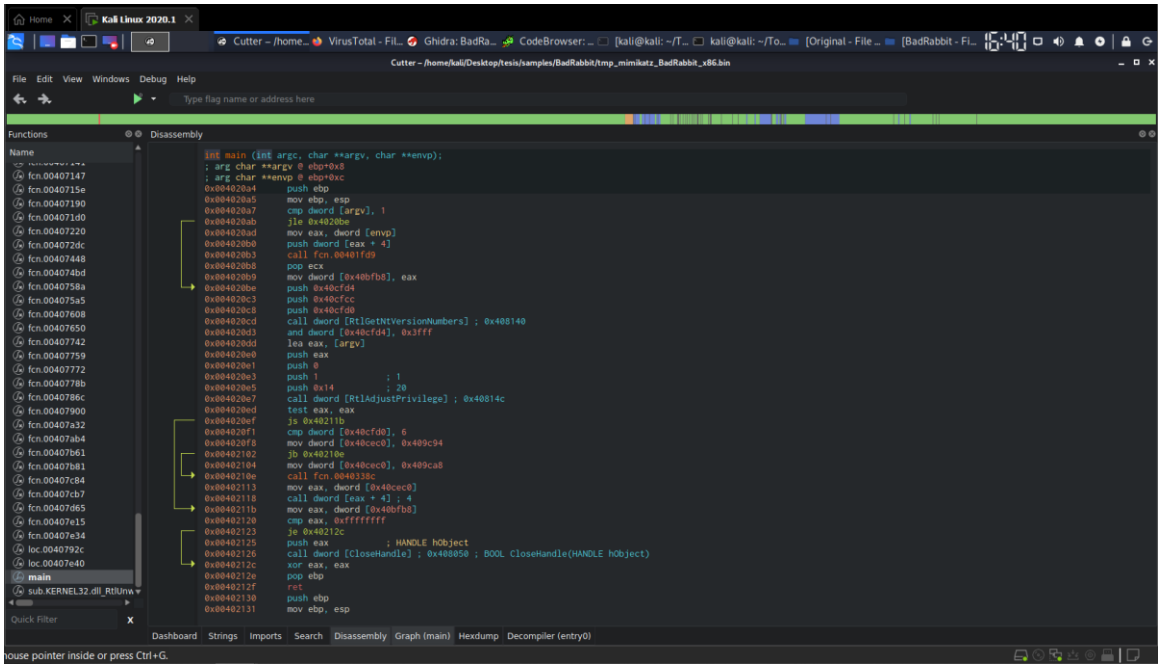


Ilustración 137: Dissasembly code de BadRabbit en Cutter

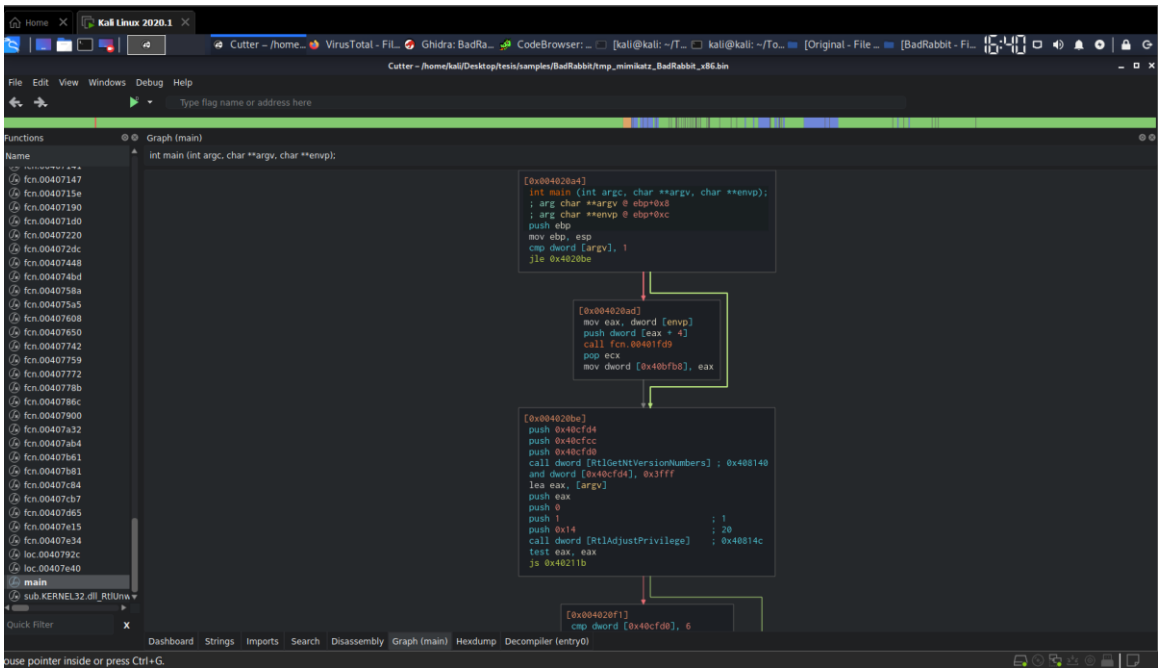


Ilustración 138: Grafo de ejecución de BadRabbit en Cutter

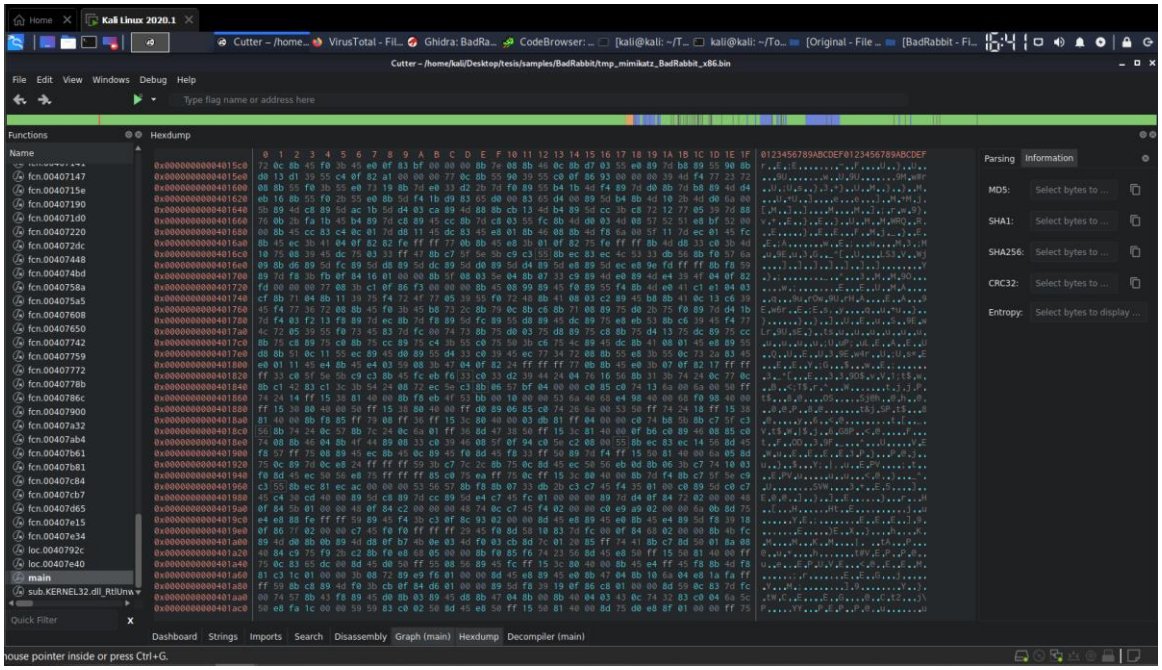


Ilustración 139: Vista hexadecimal de código de BadRabbit en Cutter

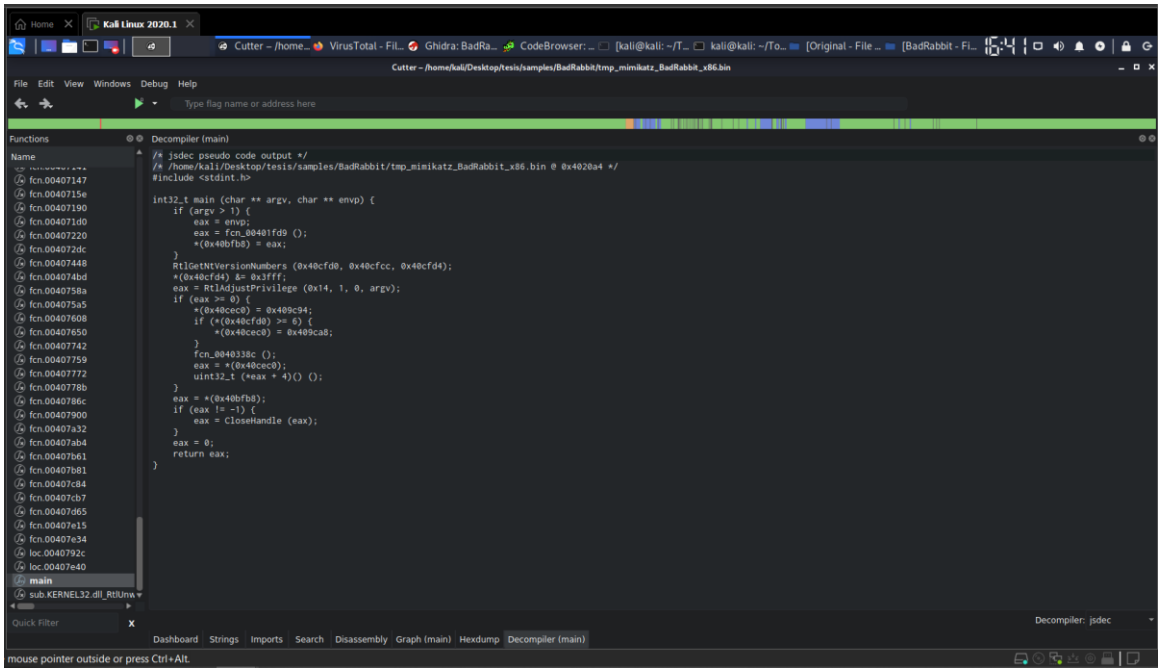


Ilustración 140: Código descompilado de BadRabbit en Cutter

El archivo de la muestra BadRabbit es subido a la plataforma *VirusTotal* para un análisis dinámico y la URL pública de su resultado es: https://www.virustotal.com/gui/file/2f8c54f9fa8e47596a3beff0031f85360e56840c77f71c6a573ace6f46412035/detection

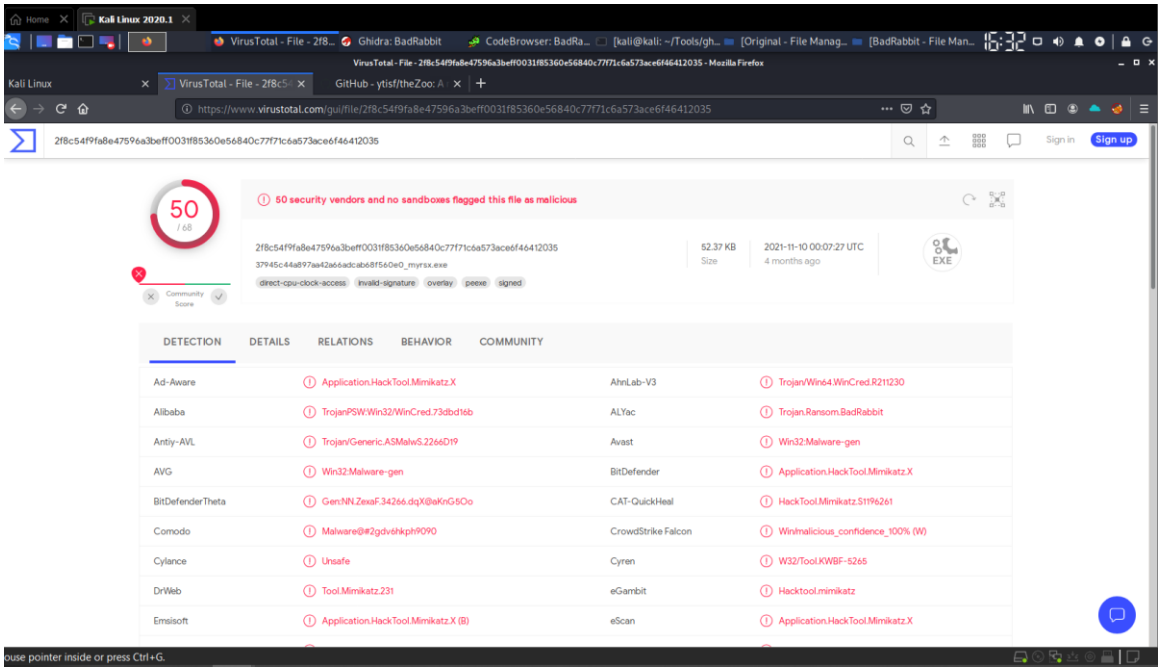


Ilustración 141: Listado de antivirus que detectaron la muestra de BadRabbit

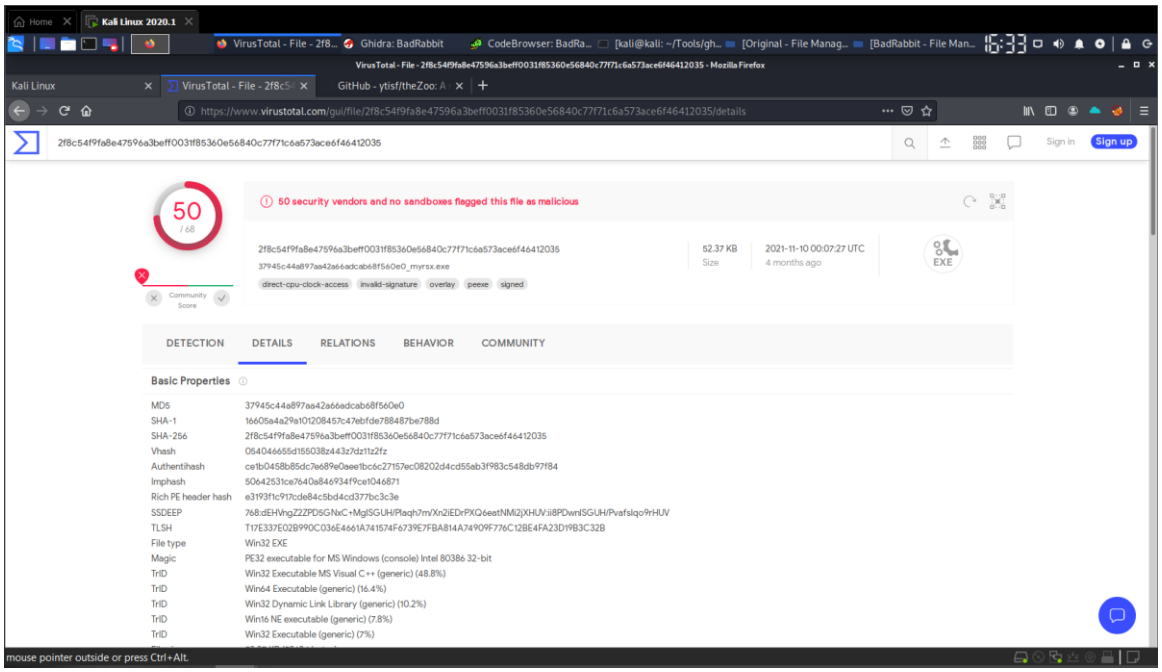


Ilustración 142: Detalles sobre la muestra BadRabbit en VirusTotal

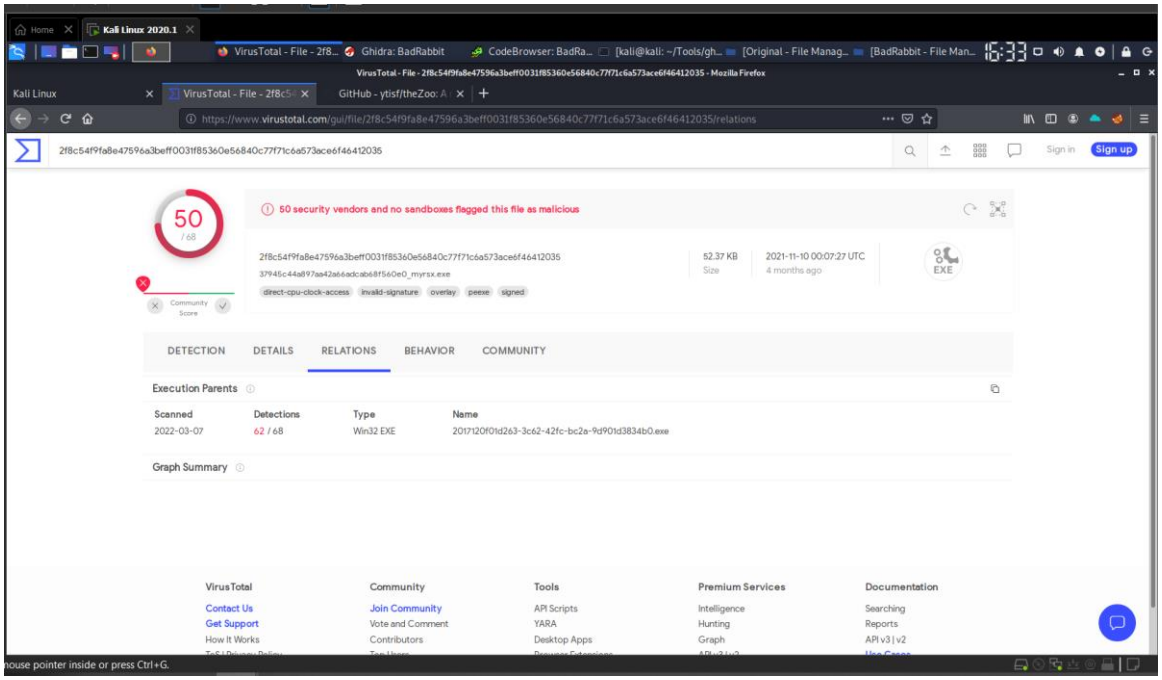


Ilustración 143: Relaciones detectadas de BadRabbit en VirusTotal

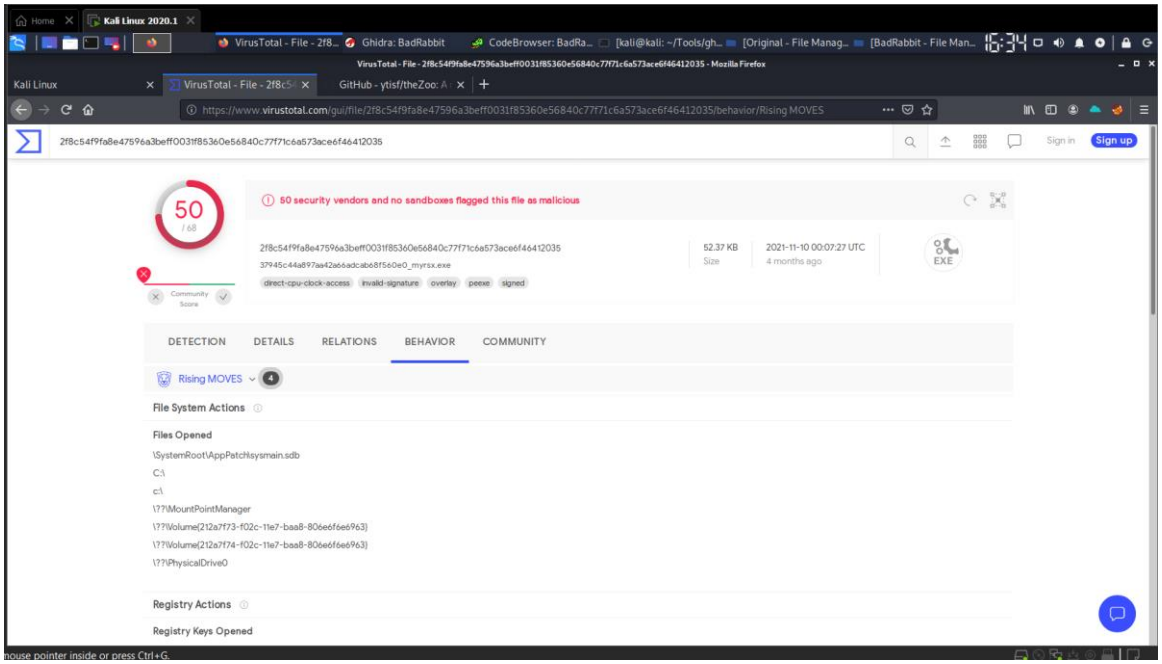


Ilustración 144: Comportamiento observado de BadRabbit en ejecución en VirusTotal

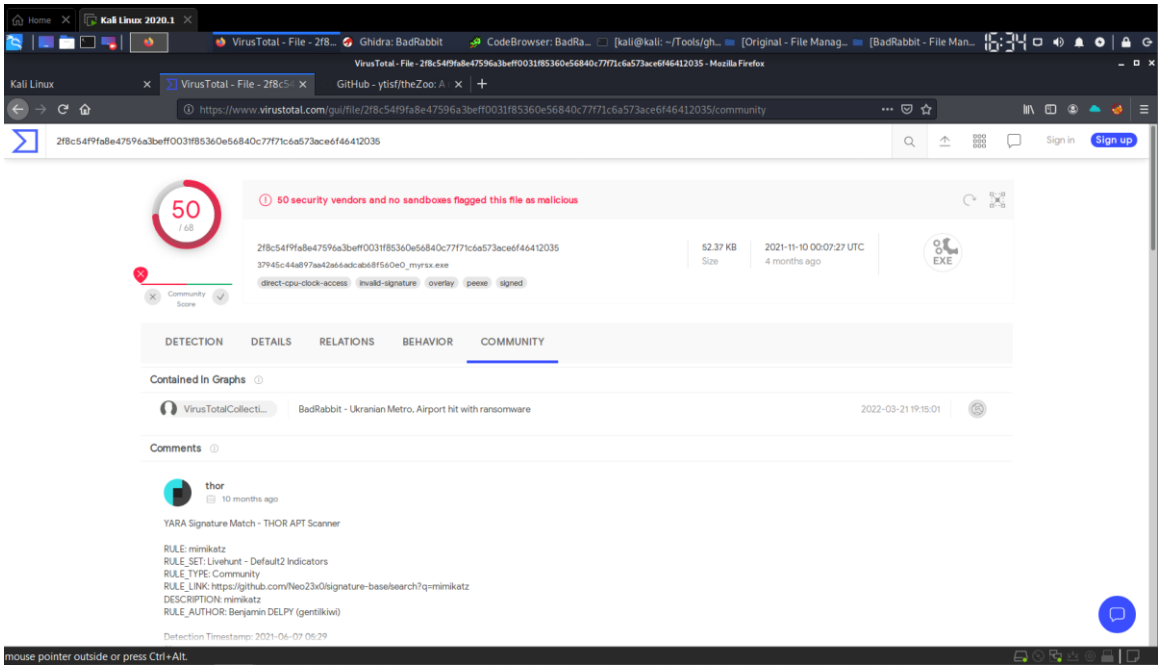


Ilustración 145: Aportes de la comunidad de VirusTotal sobre WannaCry

El archivo de BadRabbit es subido a la plataforma *Intezer Analyze* para realizar un análisis dinámico y el análisis genético, proceso propio de la plataforma

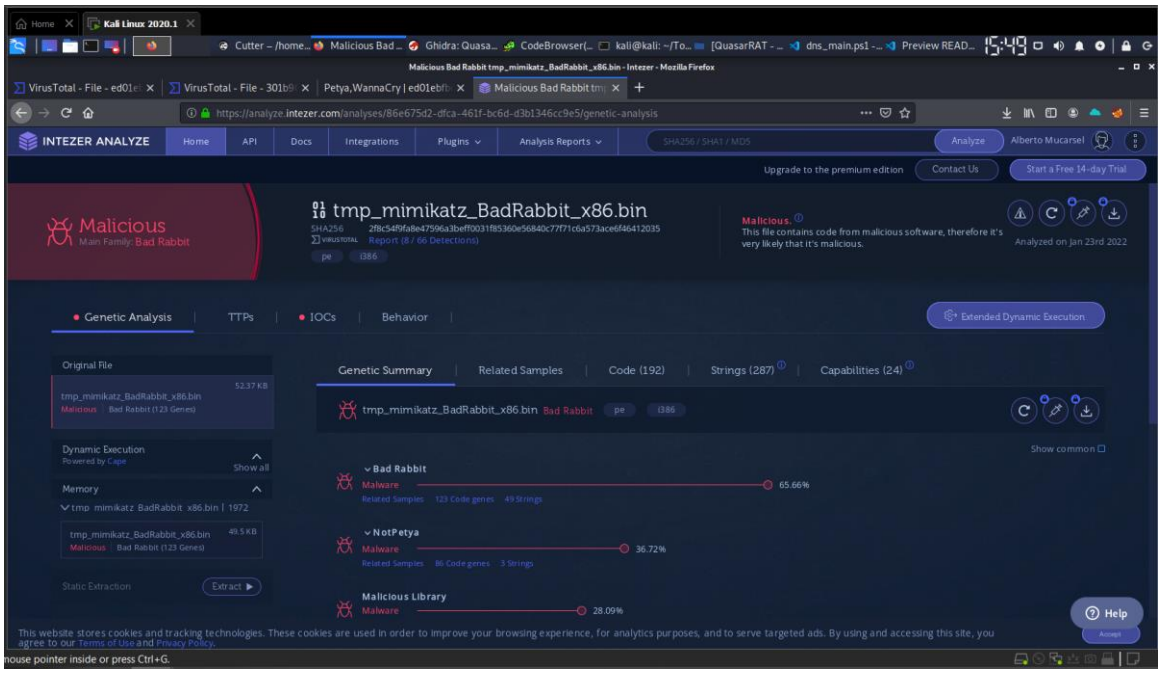


Ilustración 146: Resumen de análisis genético de BadRabbit en Intezer

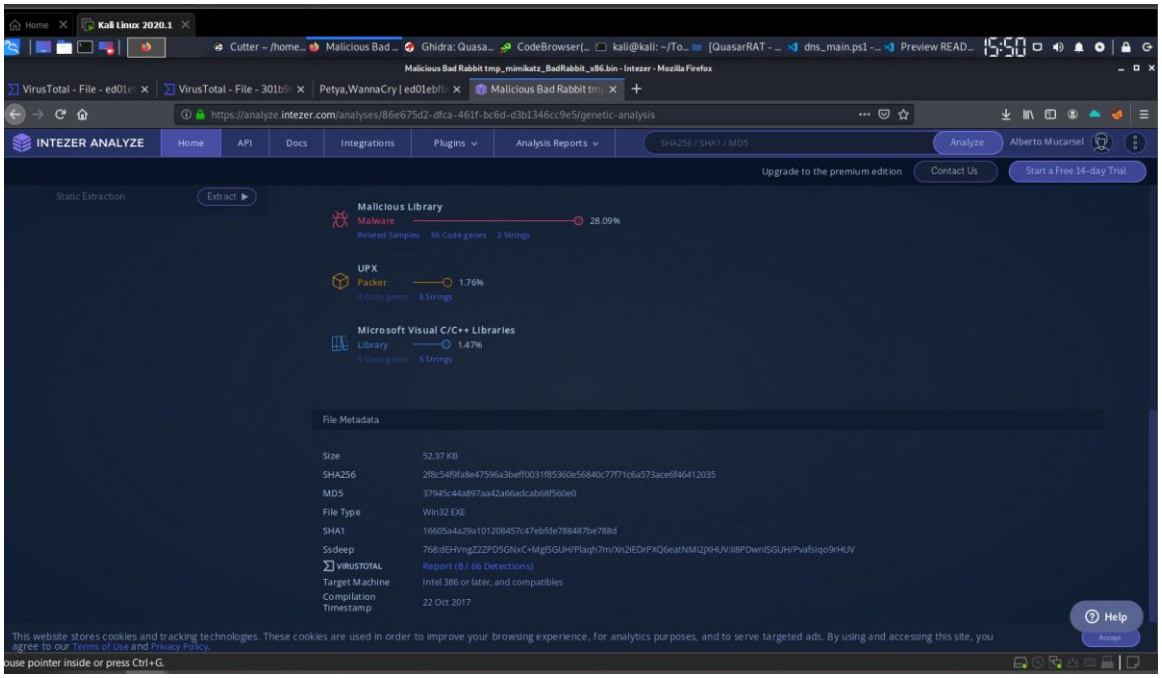


Ilustración 147: Metadata del archivo BadRabbit

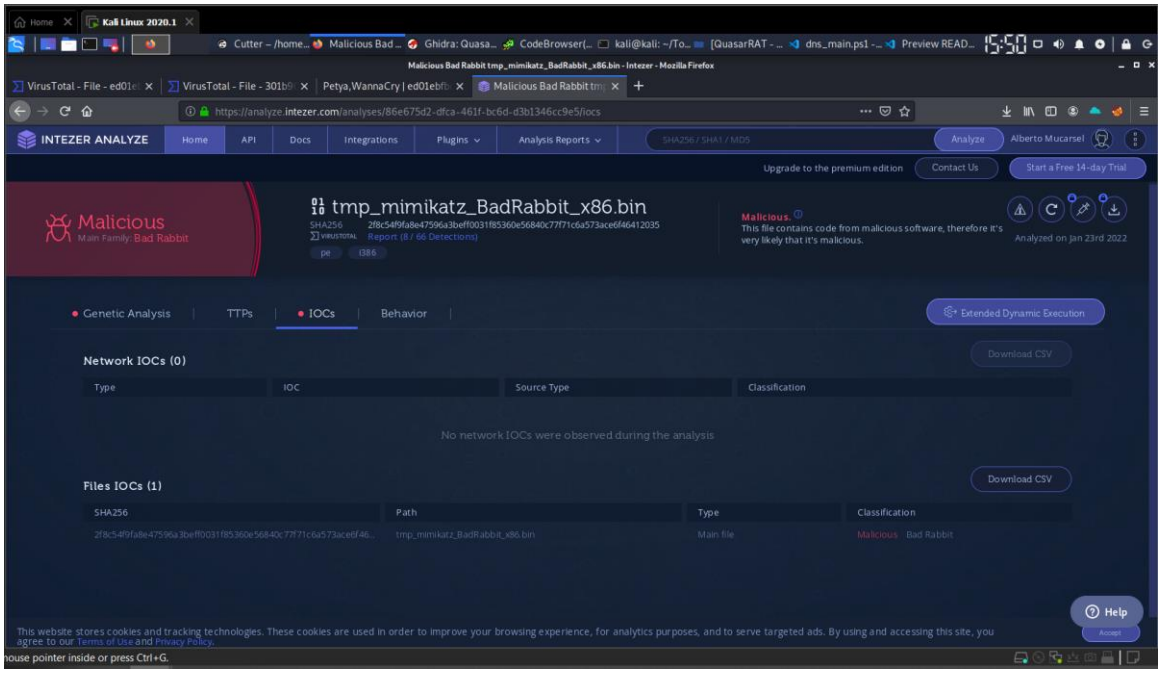


Ilustración 148: Detección de IoC dentro de BadRabbit en Intezer

Anexo E: Análisis realizado de la muestra Glimpse

El archivo de la muestra Glimpse es sometido a análisis estático con *Ghidra*

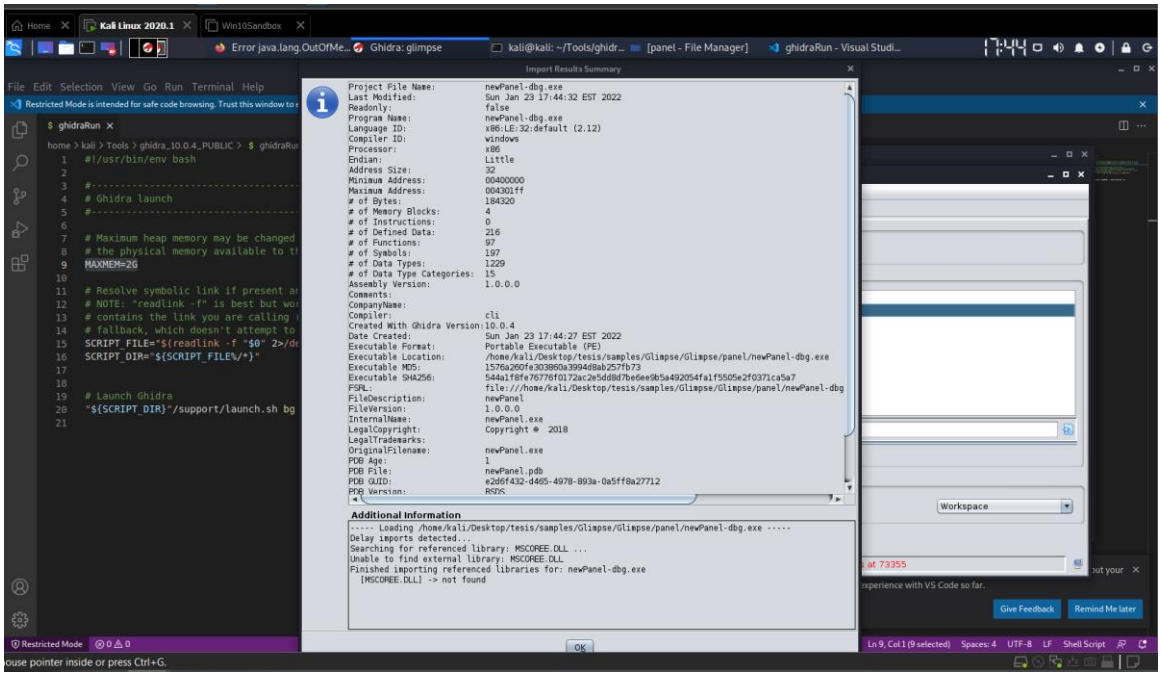


Ilustración 149: Resumen de resultados de análisis de Glimpse en Ghidra

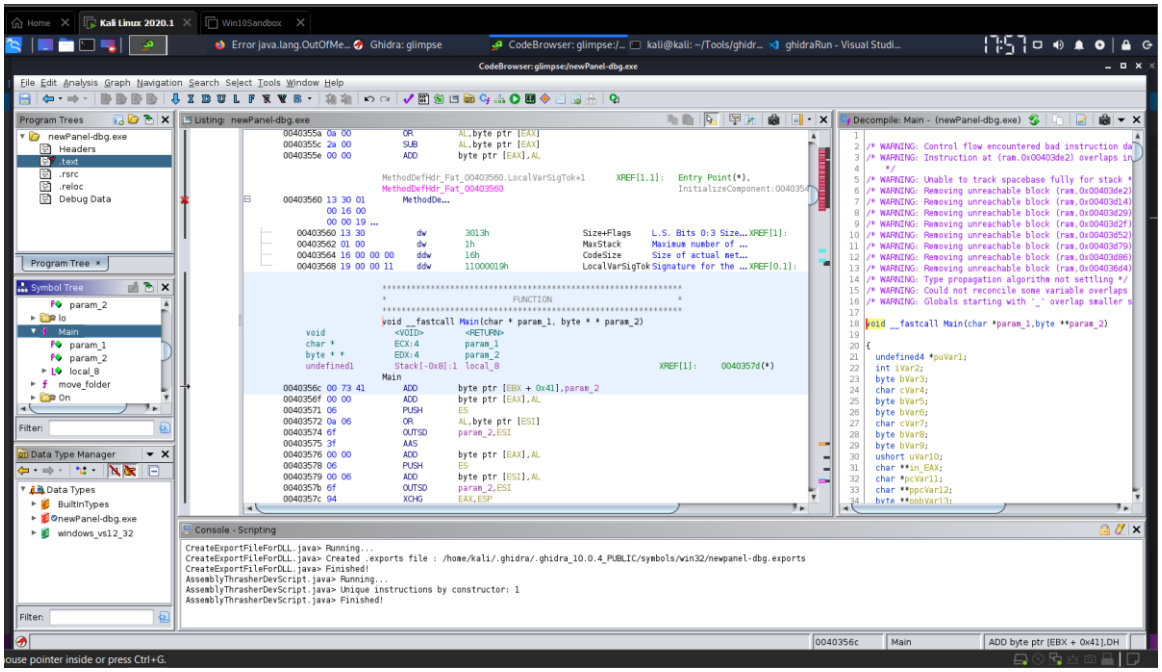


Ilustración 150: revisión de función main dentro de Glimpse en Ghidra

Tras *Ghidra*, se realiza otro análisis estático de Glimpse en *Cutter*

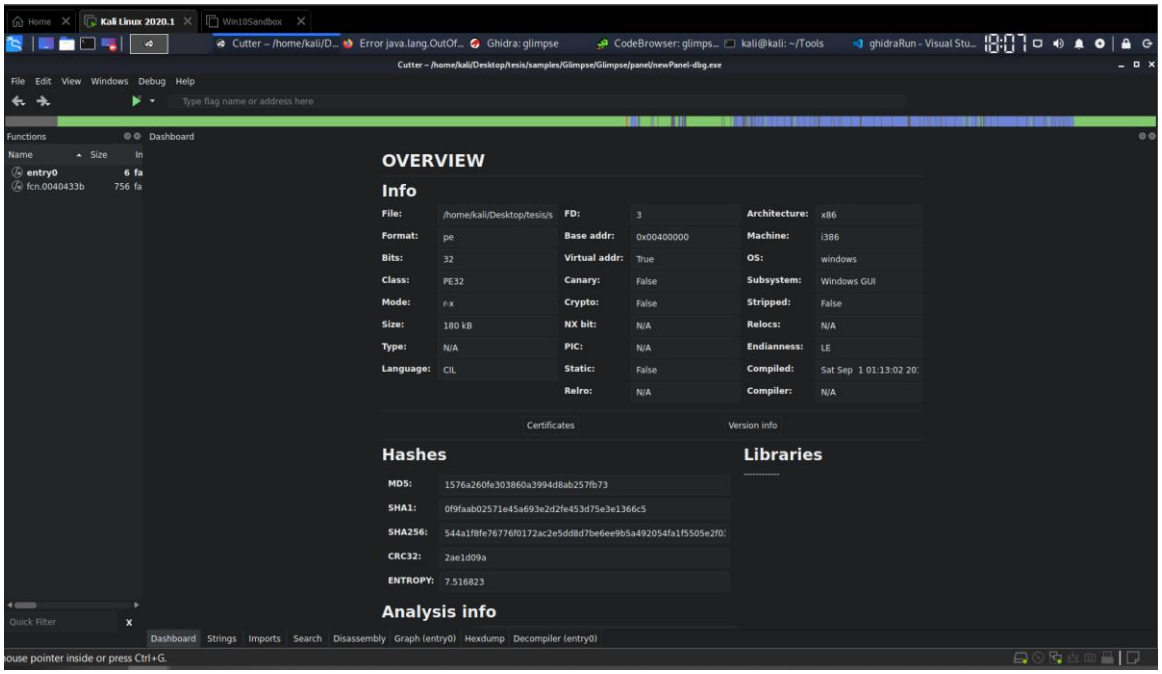


Ilustración 151: Resumen de información sobre muestra Glimpse

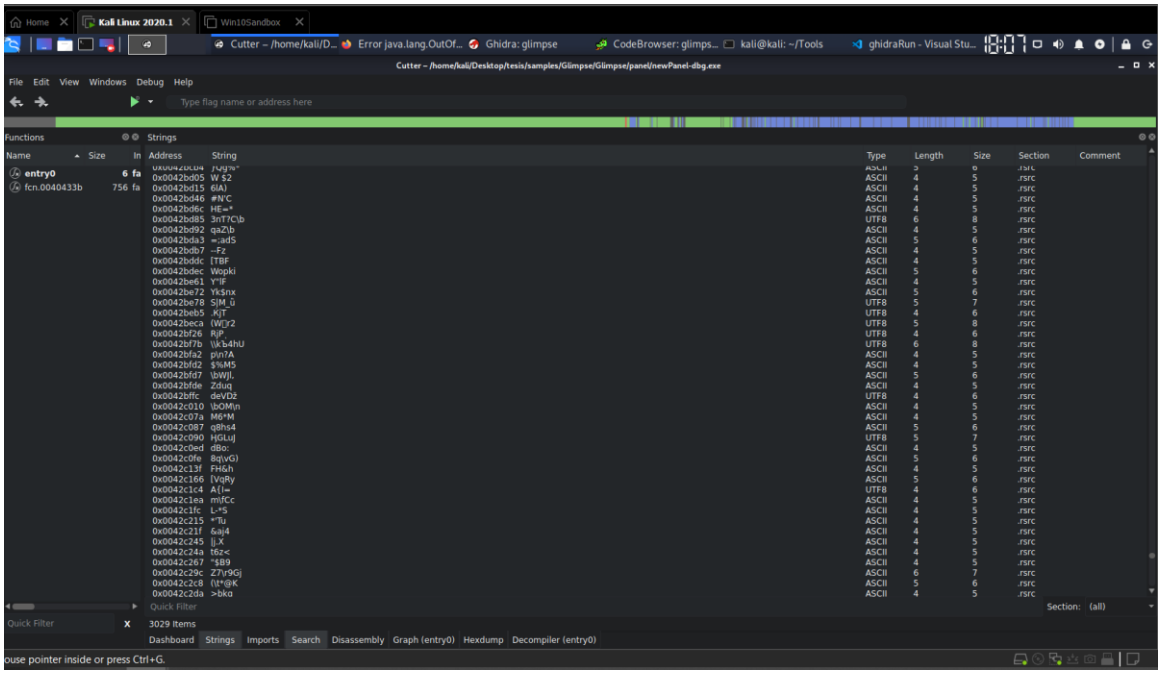


Ilustración 152: Extracción de strings de Glimpse en Cutter

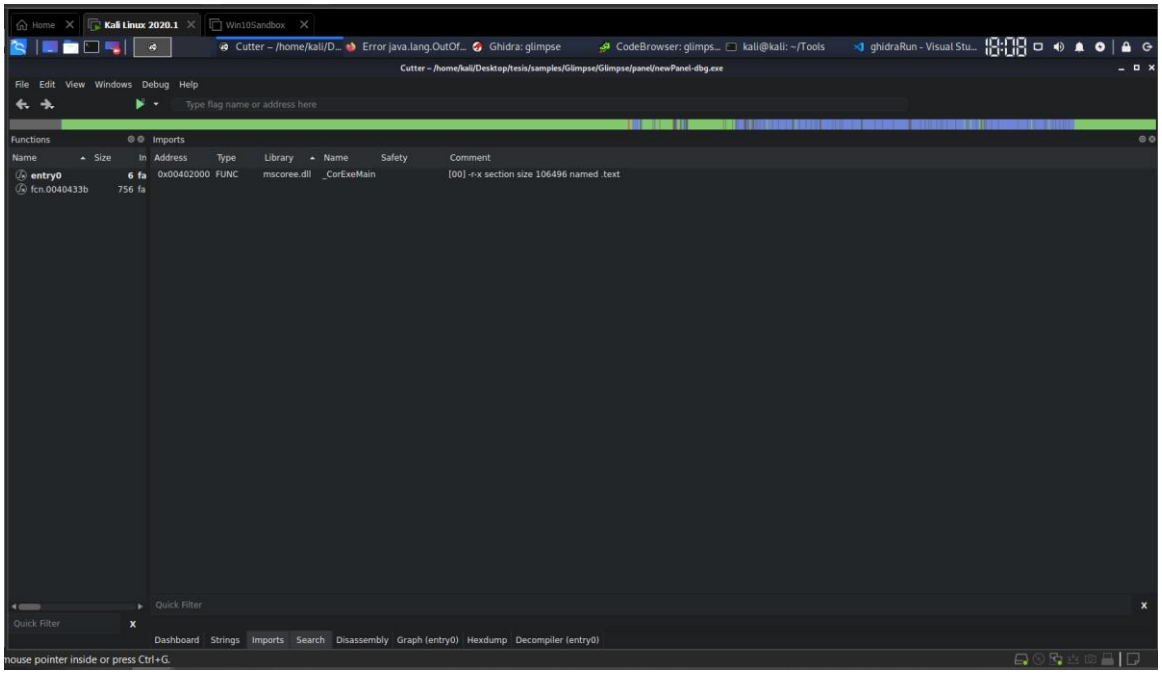


Ilustración 153: Detección de módulos importados en muestra Glimpse

El archivo de la muestra Glimpse es subido a la plataforma VirusTotal para un análisis dinámico y la URL pública de su resultado es: https://www.virustotal.com/gui/file/544a1f8fe76776f0172ac2e5dd8d7be6ee9b5a492054fa1f5505e2f0371ca5a7/details

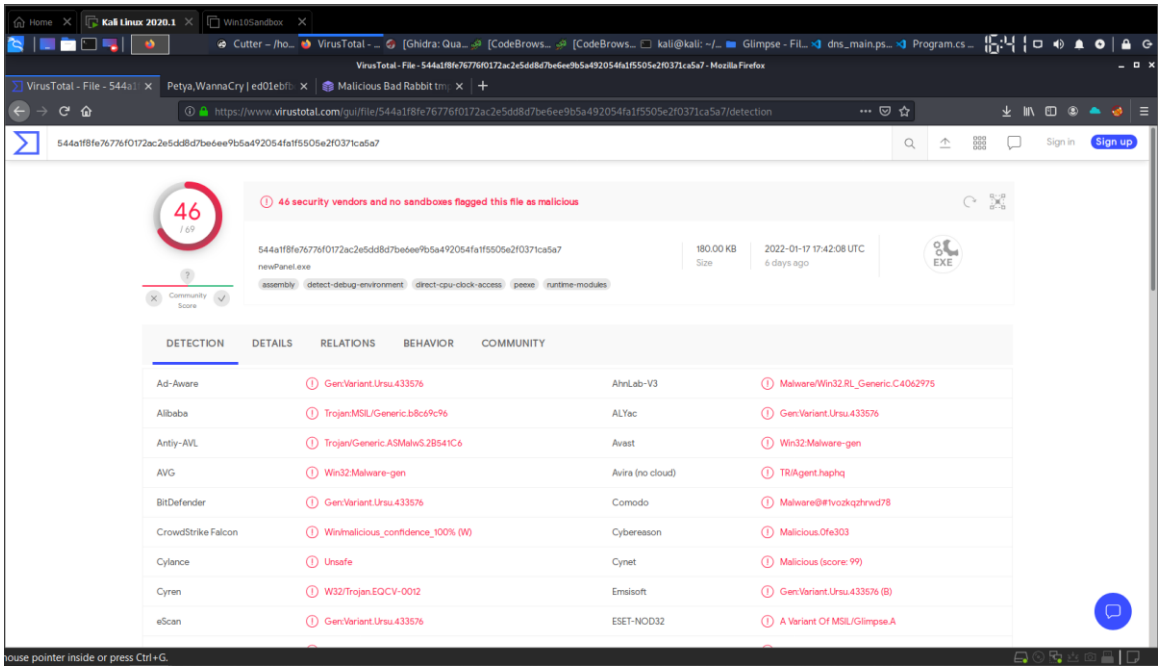


Ilustración 154: Listado de antivirus que detectaron la muestra Glimpse

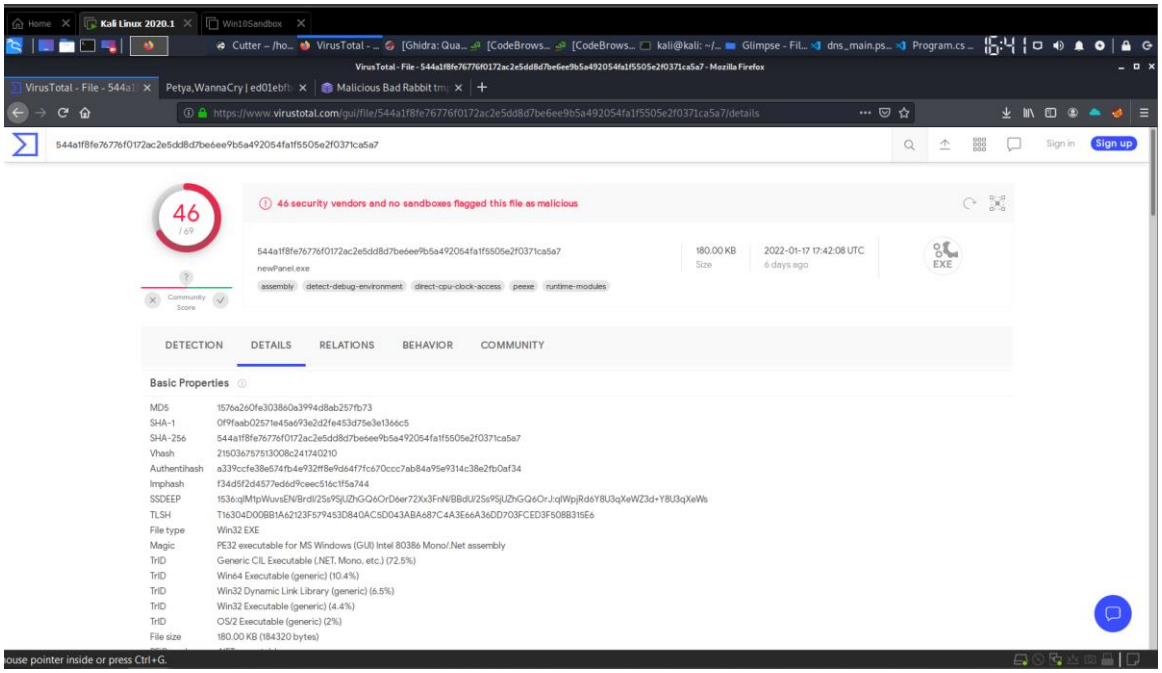


Ilustración 155: Detalles sobre la muestra Glimpse en VirusTotal

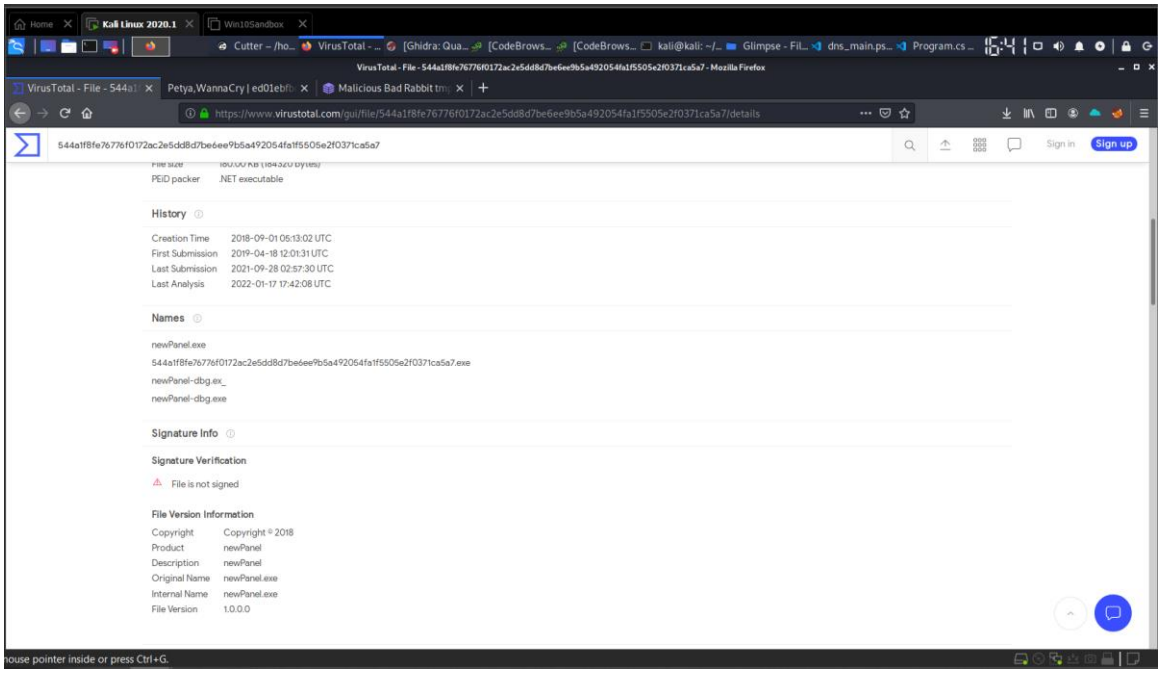


Ilustración 156: Historia y nombres de Glimpse en VirusTotal

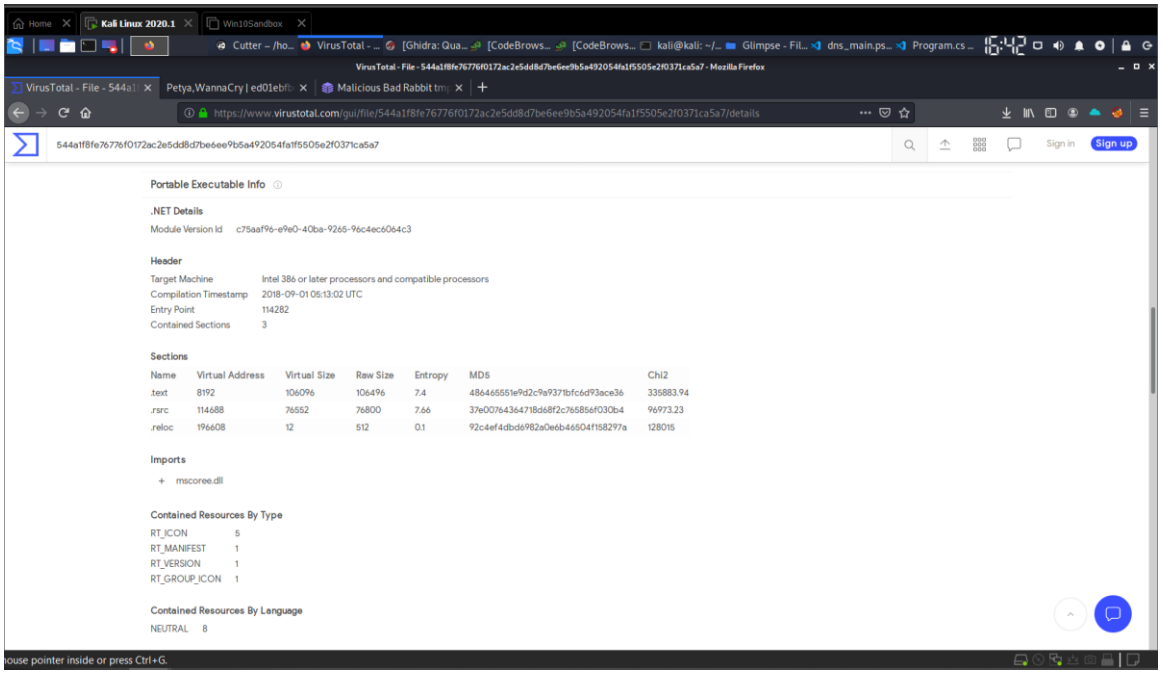


Ilustración 157: Headers, secciones y librerías importadas de Glimpse en VirusTotal

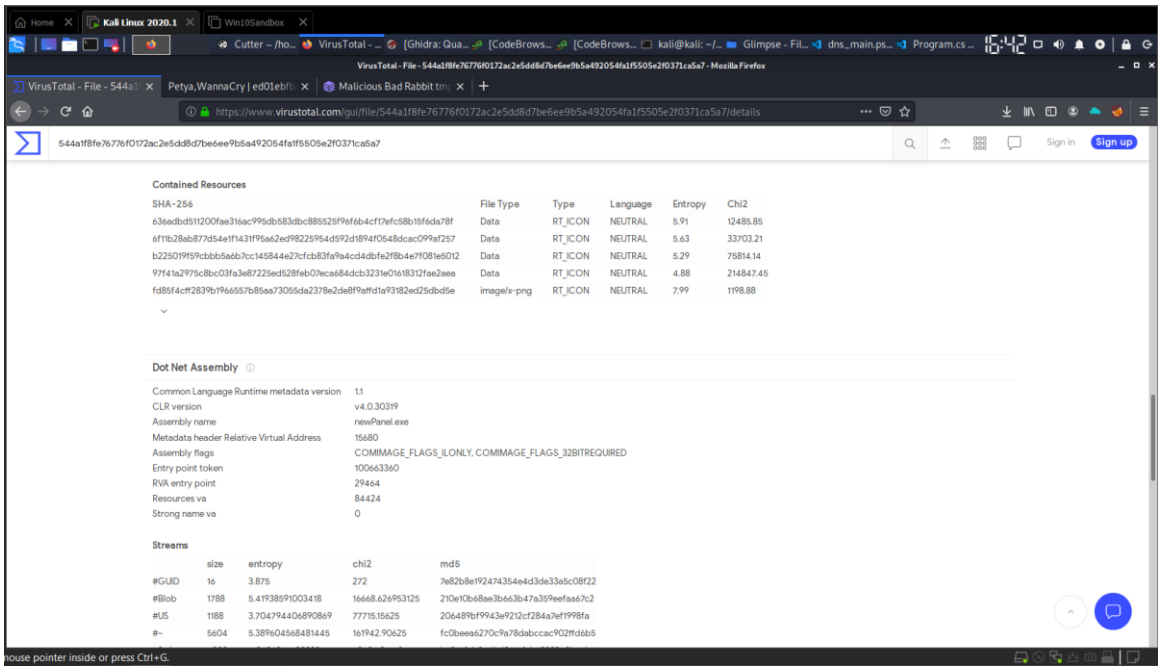


Ilustración 158: Recursos detectados dentro de Glimpse en VirusTotal

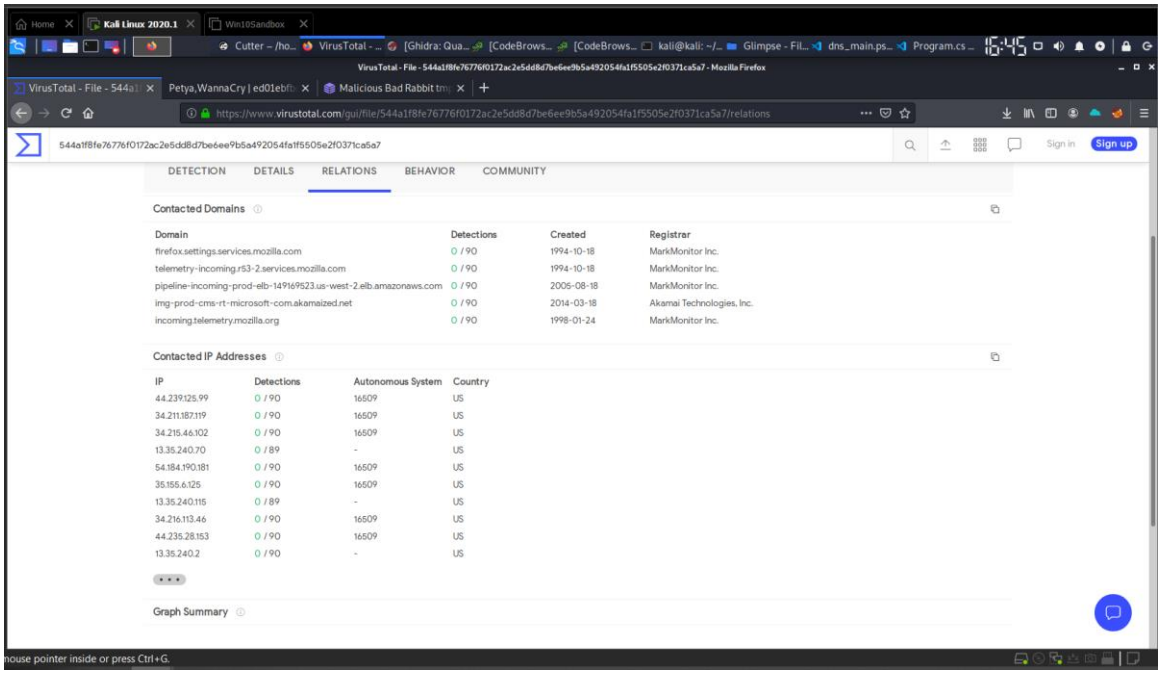


Ilustración 159: Relaciones detectadas de Glimpse en VirusTotal

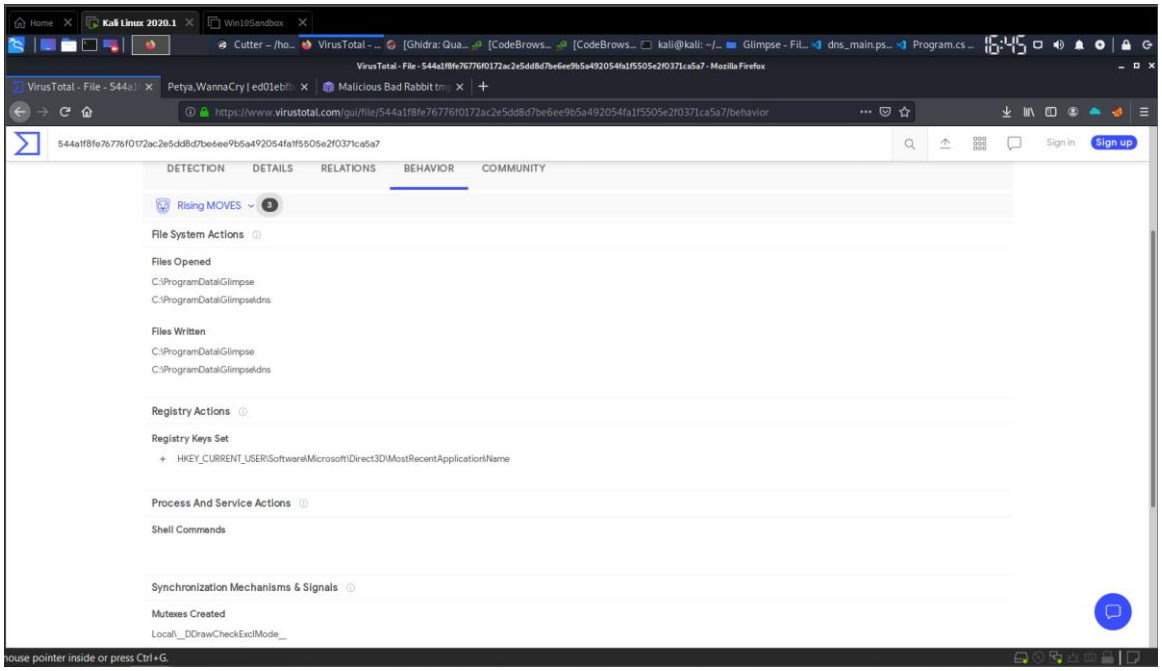


Ilustración 160: Comportamiento observado de Glimpse en ejecución en VirusTotal

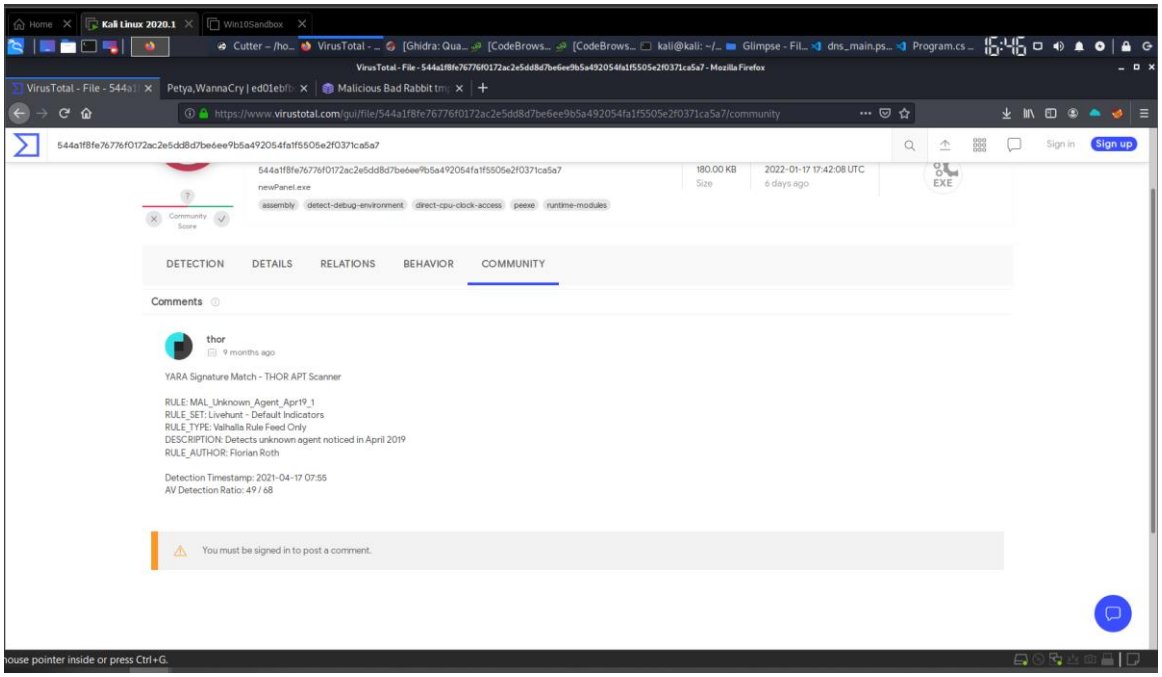


Ilustración 161: Aportes de la comunidad de VirusTotal sobre Glimpse

El archivo de Glimpse es subido a la plataforma *Intezer Analyze* para realizar un análisis dinámico y el análisis genético, proceso propio de la plataforma

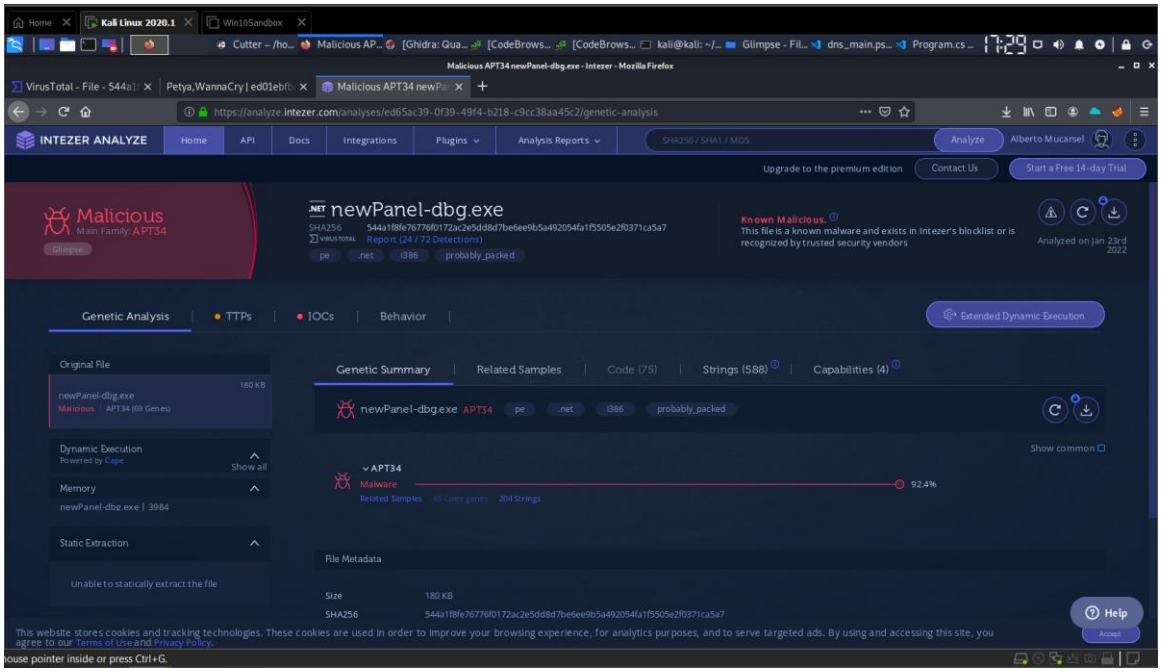


Ilustración 162: Resumen de análisis genético de Glimpse en Intezer

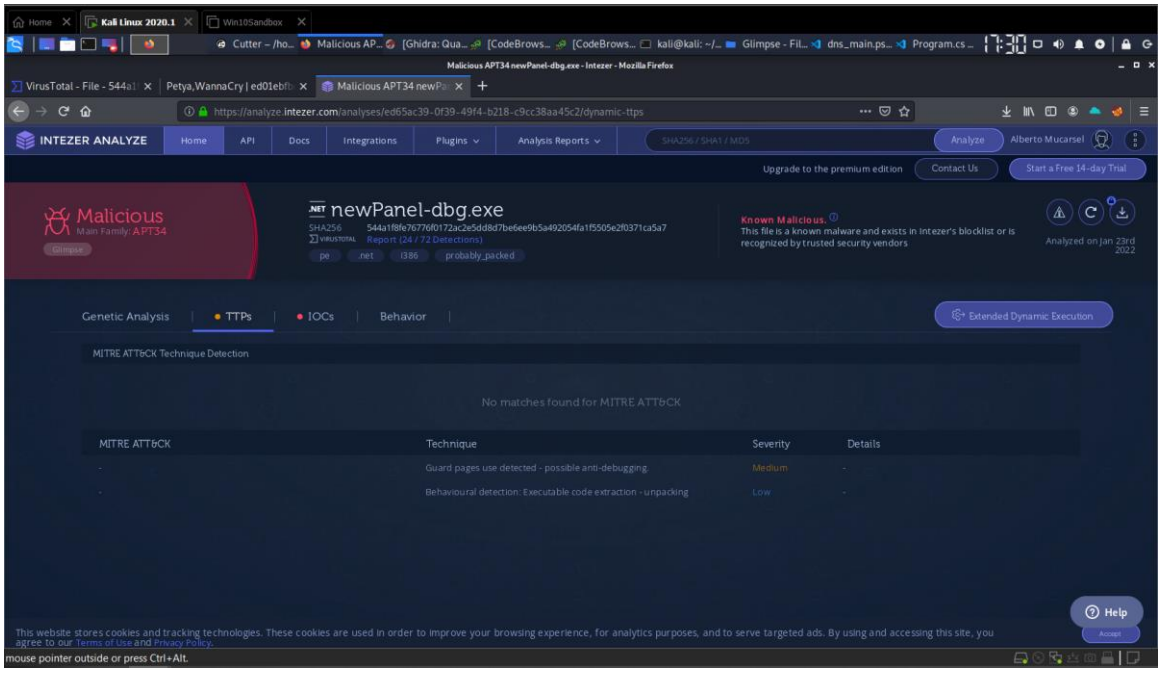


Ilustración 163: Detección de técnicas usadas por Glimpse en Intezer

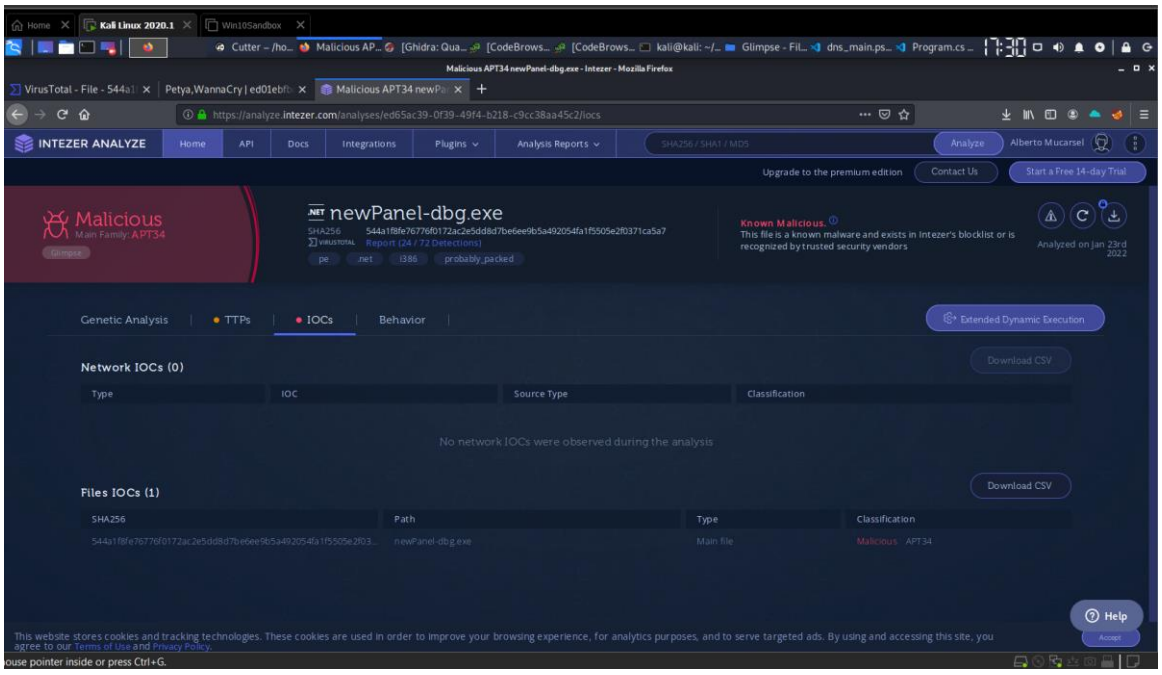


Ilustración 164: Detección de IoC dentro de Glimpse en Intezer

Anexo F: Análisis realizado de la muestra QuasarRAT

El archivo de la muestra QuasarRAT es sometido a análisis estático con *Ghidra*

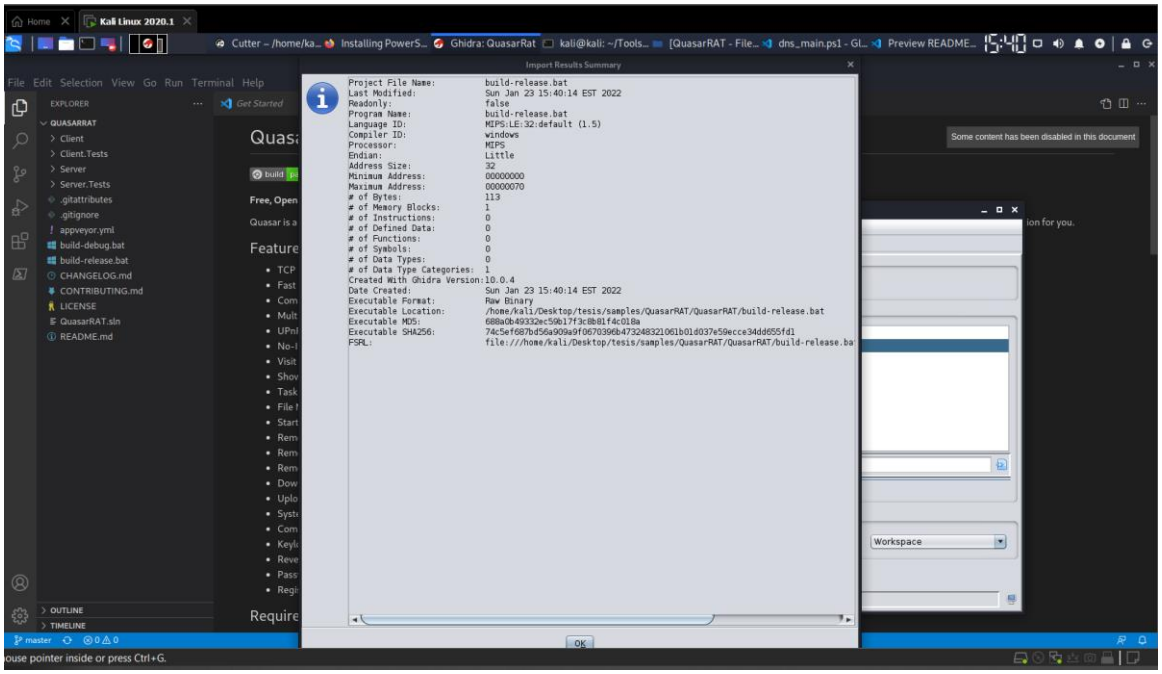
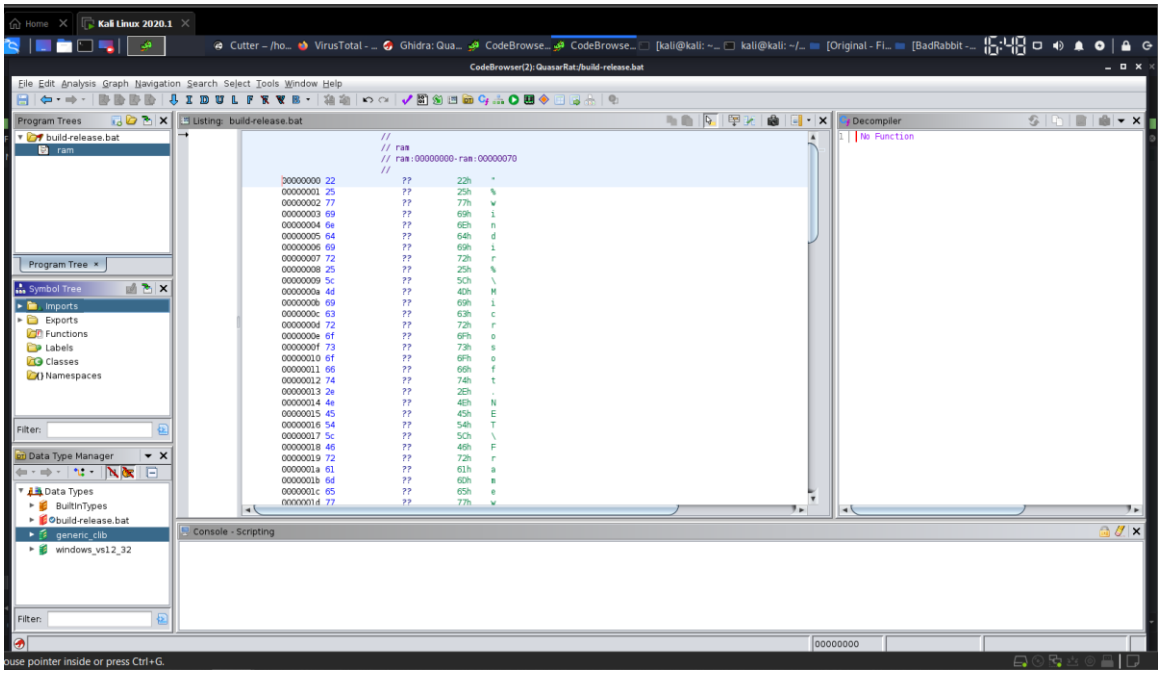


Ilustración 165: Resumen de resultados de análisis de QuasarRAT en Ghidra



Tras *Ghidra*, se realiza otro análisis estático de QuasarRAT en *Cutter*

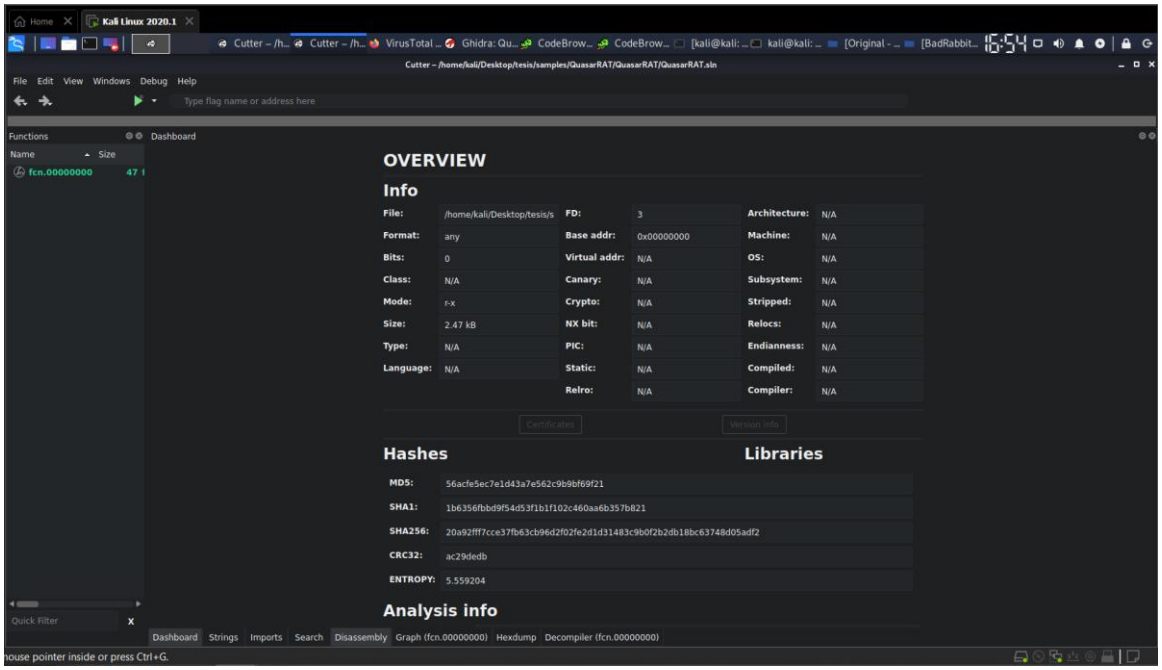


Ilustración 166: Resumen de información sobre muestra QuasarRAT

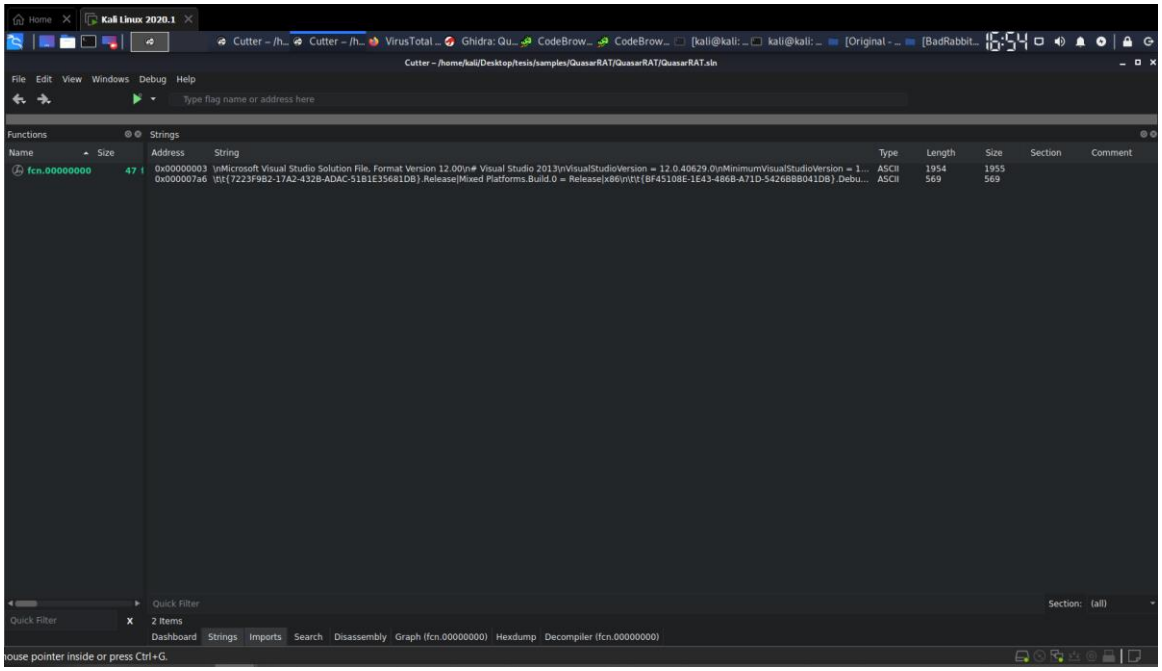


Ilustración 167: Extracción de strings de QuasarRAT en Cutter

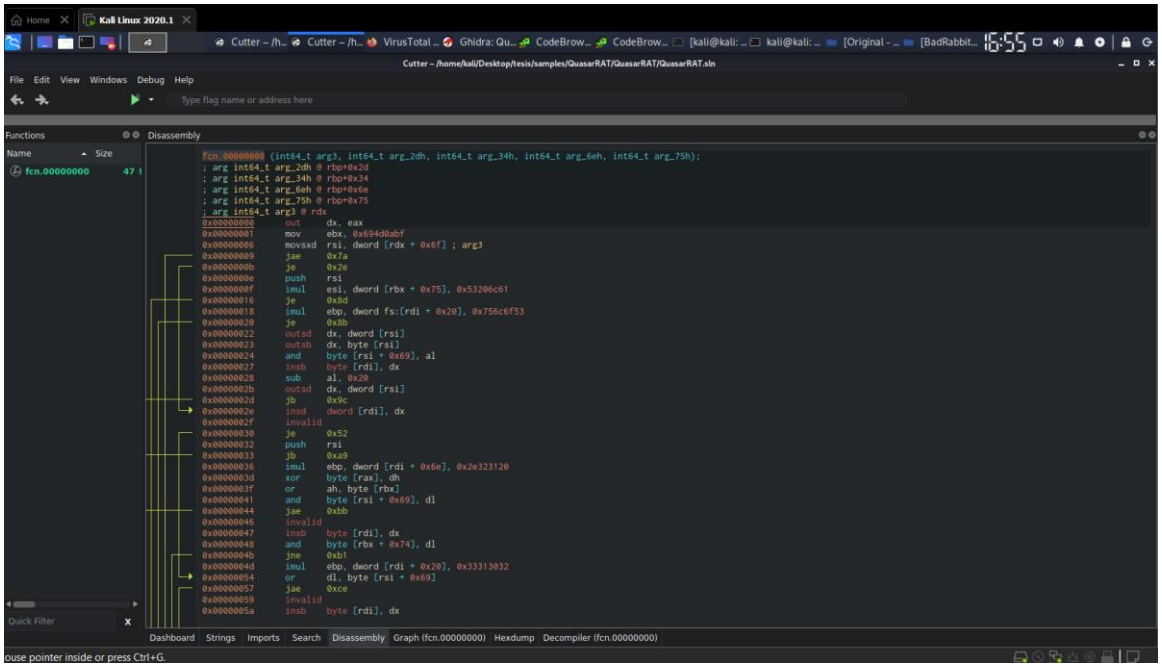


Ilustración 168: Dissasembly code de QuasarRAT en Cutter

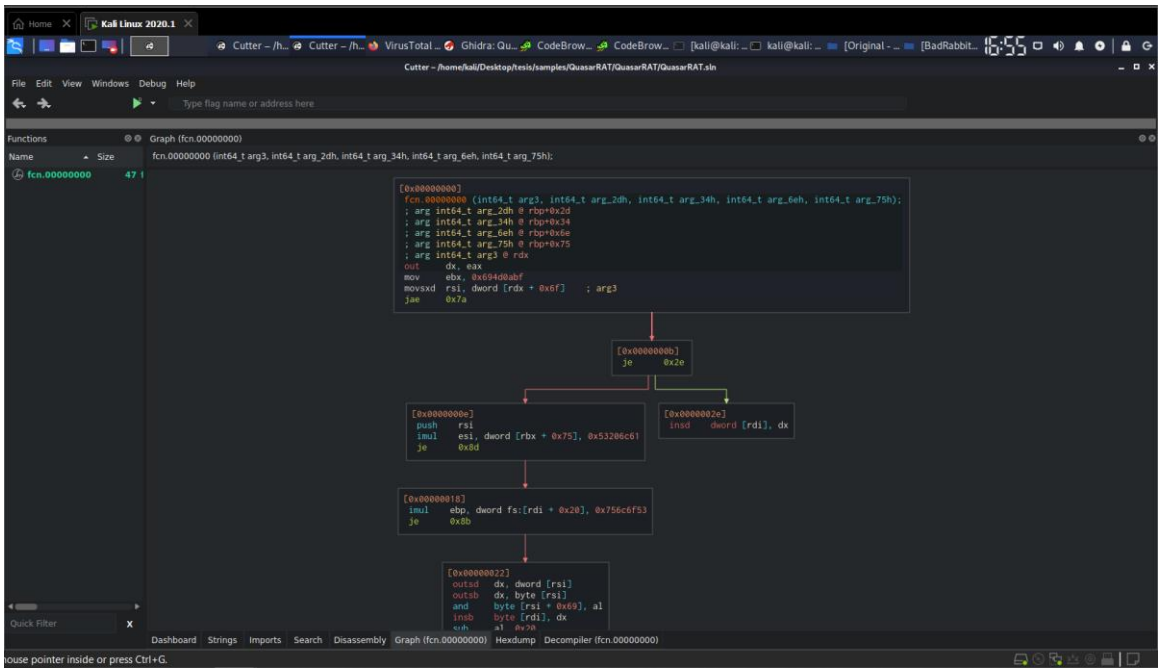


Ilustración 169: Flujo de ejecución de QuasarRAT en Cutter

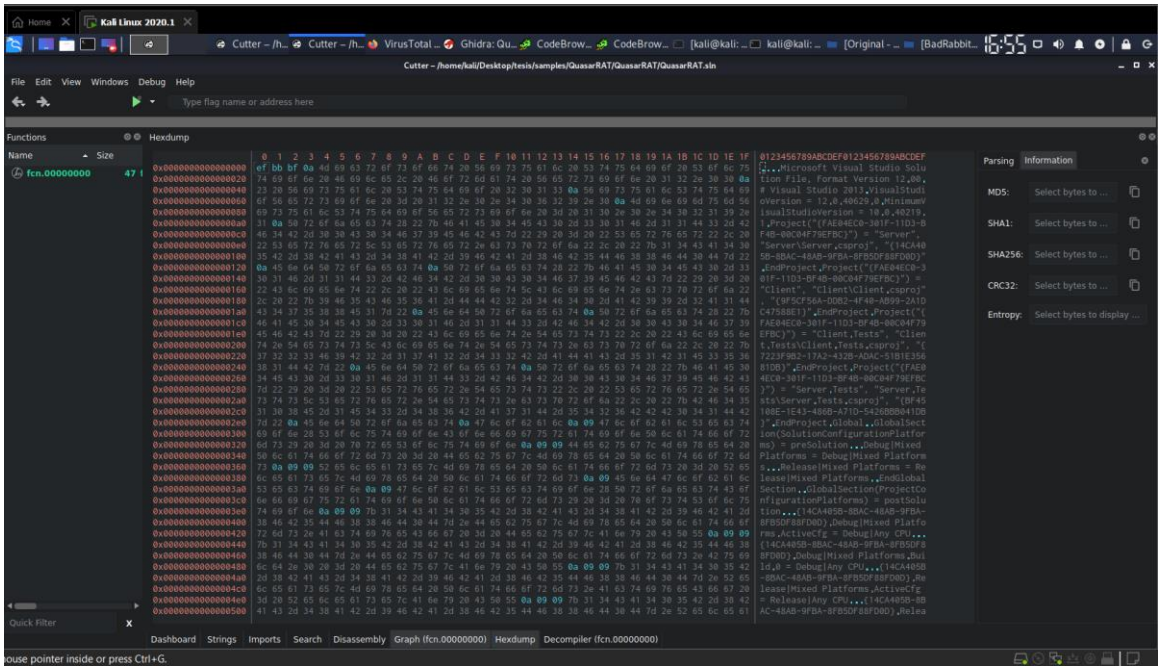


Ilustración 170: Vista hexadecimal de código de QuasarRAT en Cutter

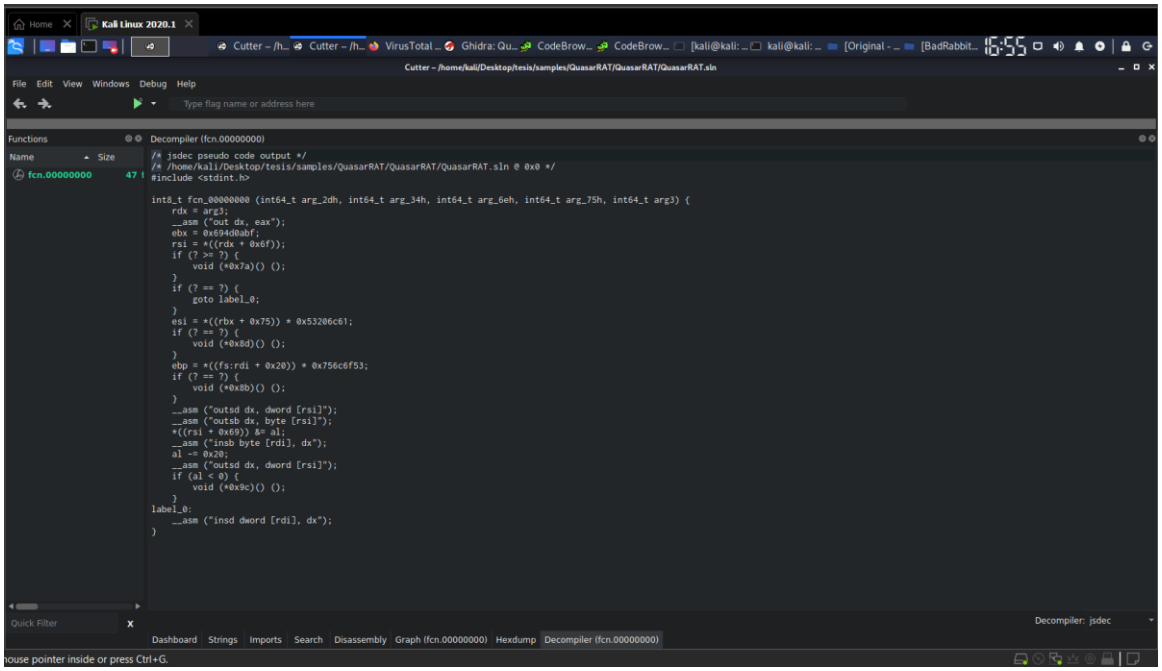


Ilustración 171: Código descompilado de QuasarRAT en Cutter

El archivo ejecutable con la muestra de WannaCry es subido a la plataforma *VirusTotal* para un análisis dinámico y la URL pública de su resultado es: https://www.virustotal.com/gui/file/4596df354fe078d9e43f9cb6cc28232a7d8e8bf3c5cf966dd1fbe2cf9656b076/detection

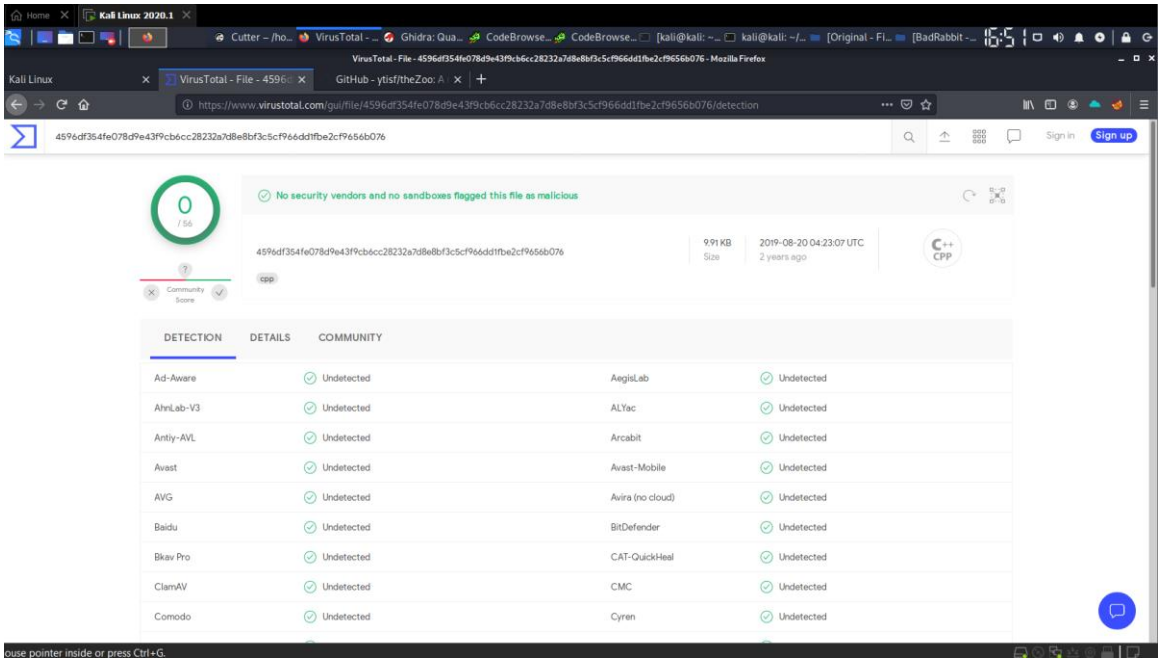


Ilustración 172: Listado de antivirus que detectaron la muestra de QuasarRAT

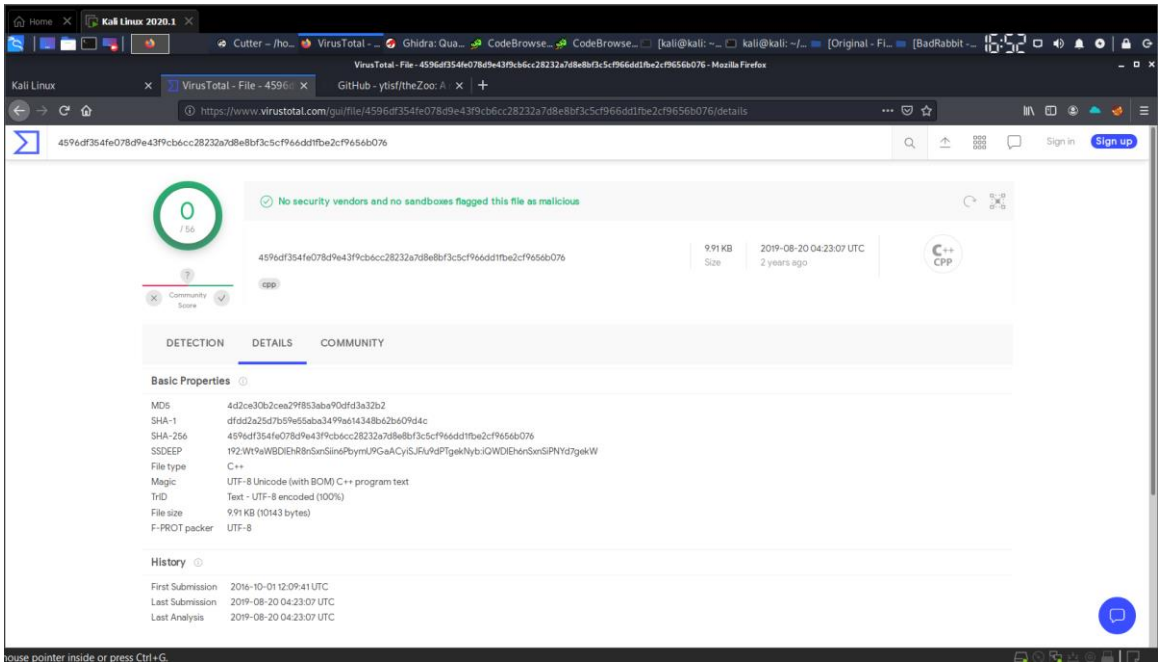


Ilustración 173: Detalles sobre la muestra QuasarRAT en VirusTotal

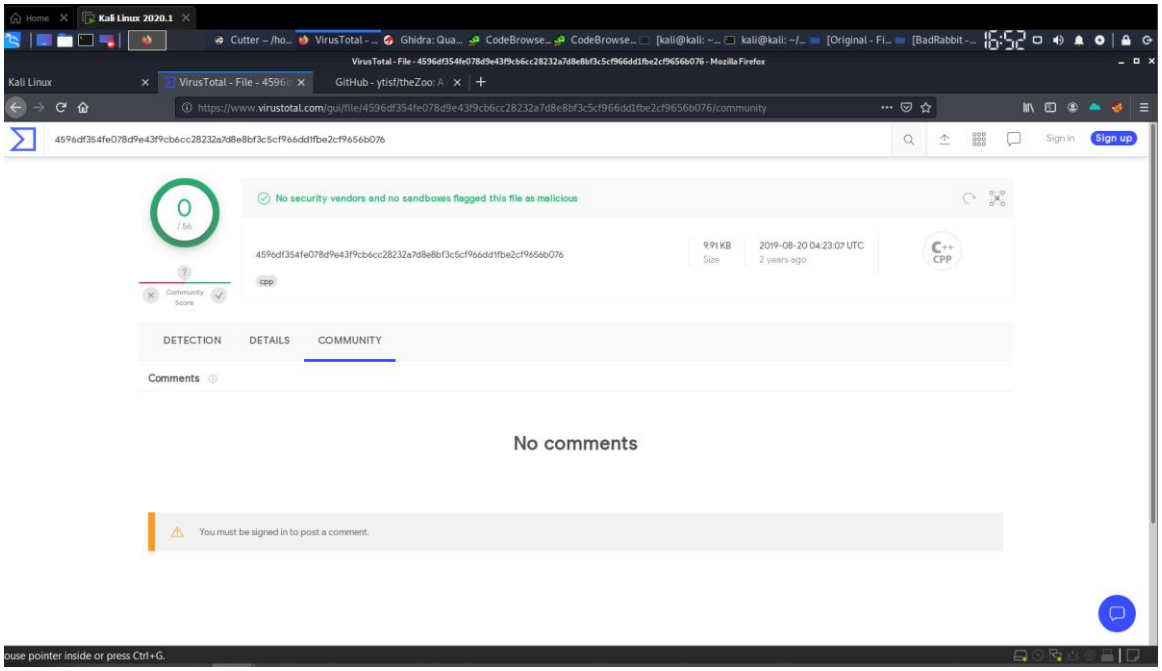


Ilustración 174: Aportes de la comunidad de VirusTotal sobre QuasarRAT

El archivo de QuasarRAT es subido a la plataforma *Intezer Analyze* para realizar un análisis dinámico y el análisis genético, proceso propio de la plataforma

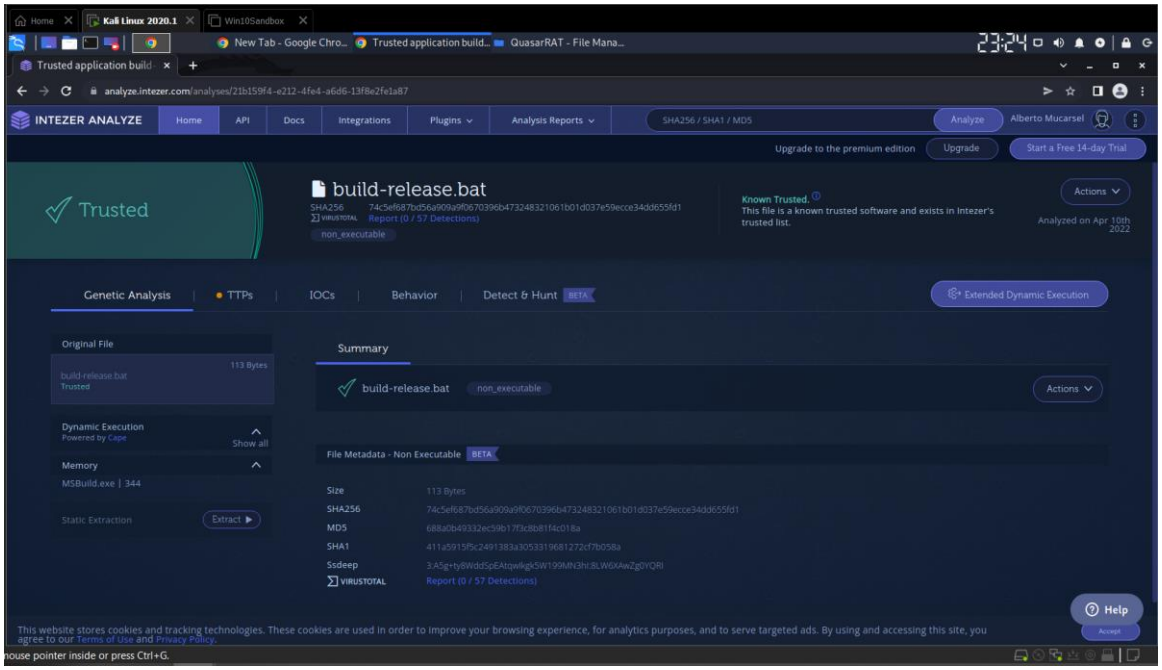


Ilustración 175: Resumen de análisis genético de QuasarRAT en Intezer

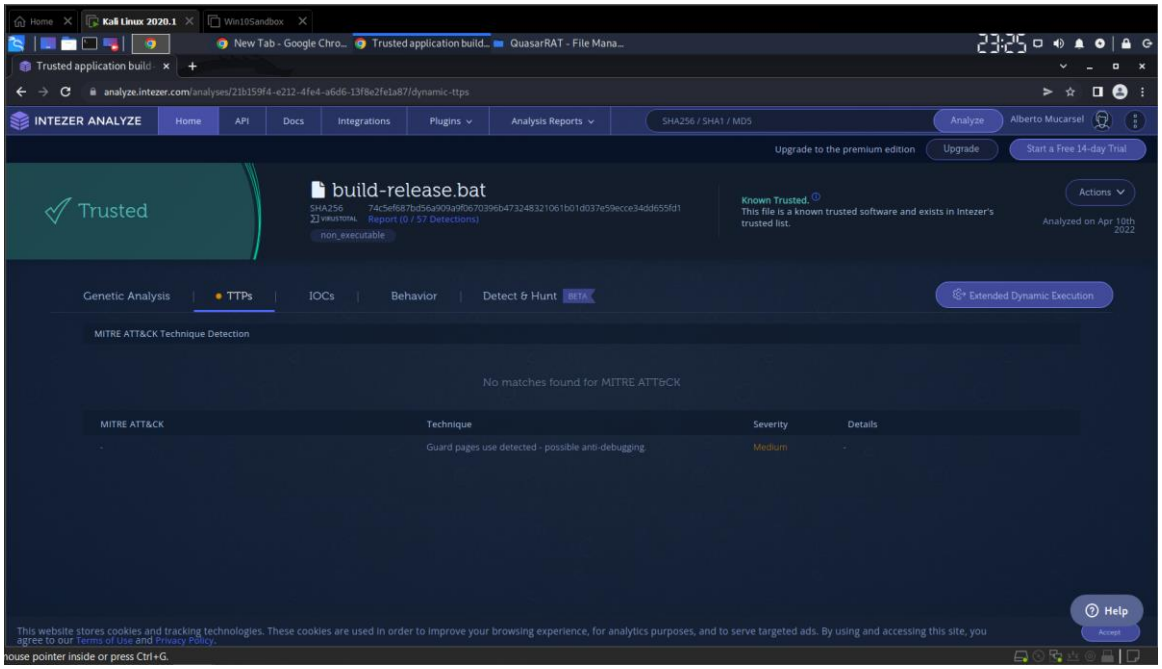


Ilustración 176: Detección de técnicas usadas por QuasarRAT

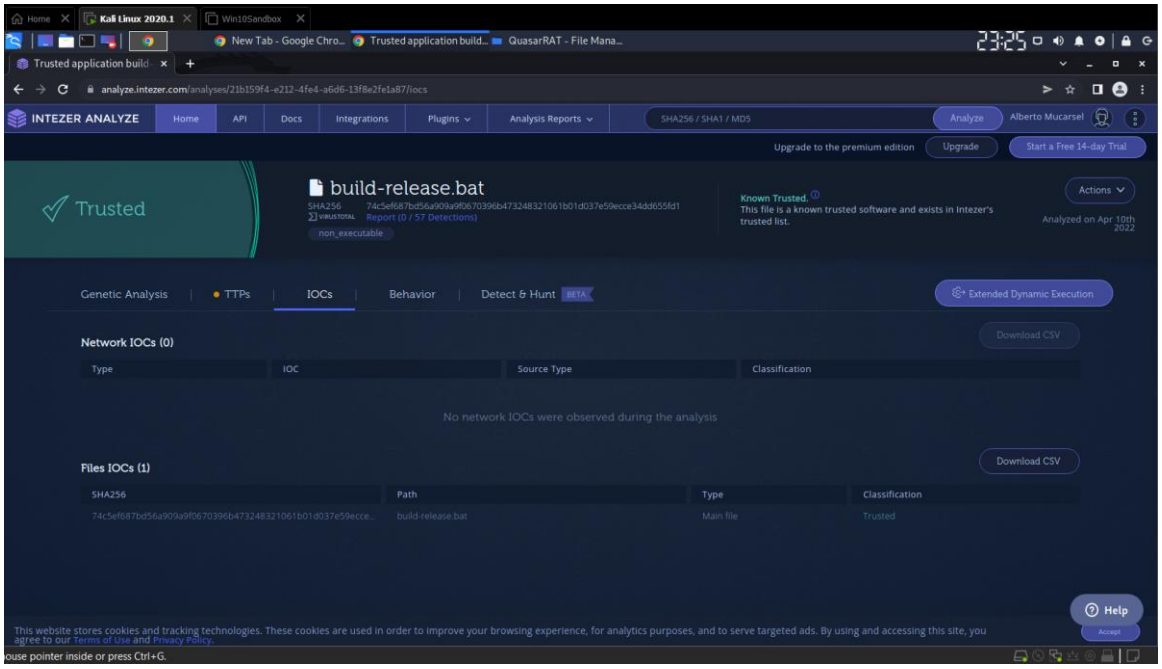


Ilustración 177: Detección de IoC dentro de QuasarRAT en Intezer

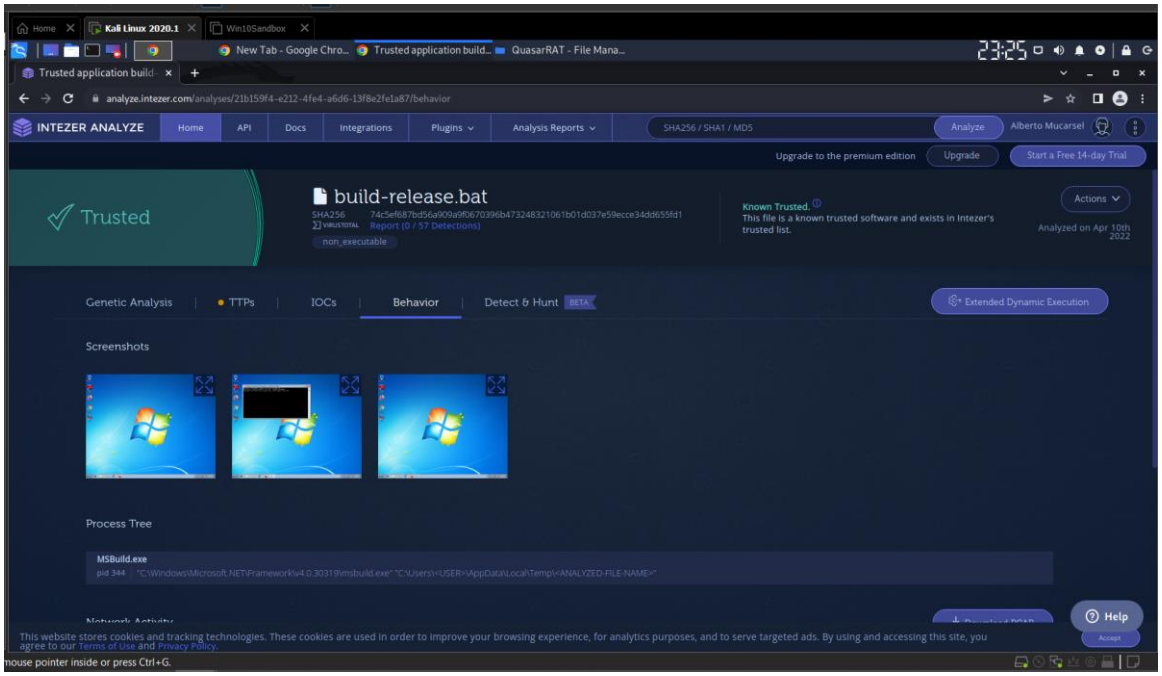


Ilustración 178: Comportamiento de QuasarRAT durante ejecución en entorno sandbox en Intezer

Una vez la máquina víctima se encuentra infectada con QuasarRAT, se procede a la extracción del volcado de memoria y su envió a la máquina Kali para su análisis con Volatility.

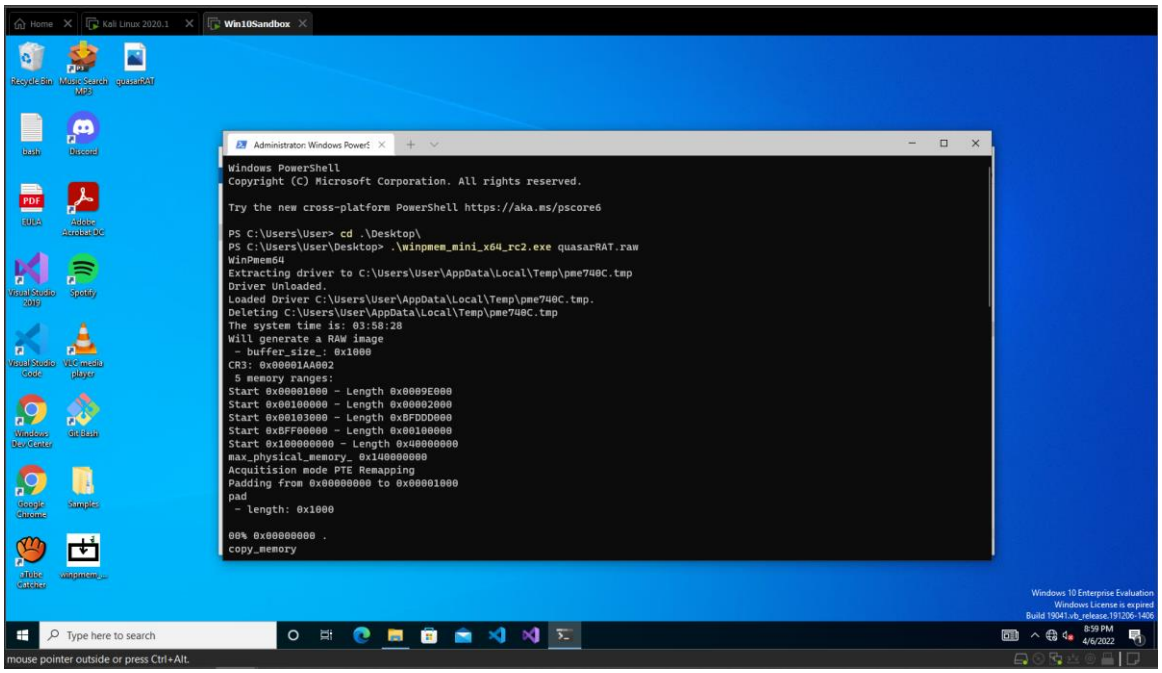


Ilustración 179: Obtención de volcado de memoria tras infección con QuasarRAT

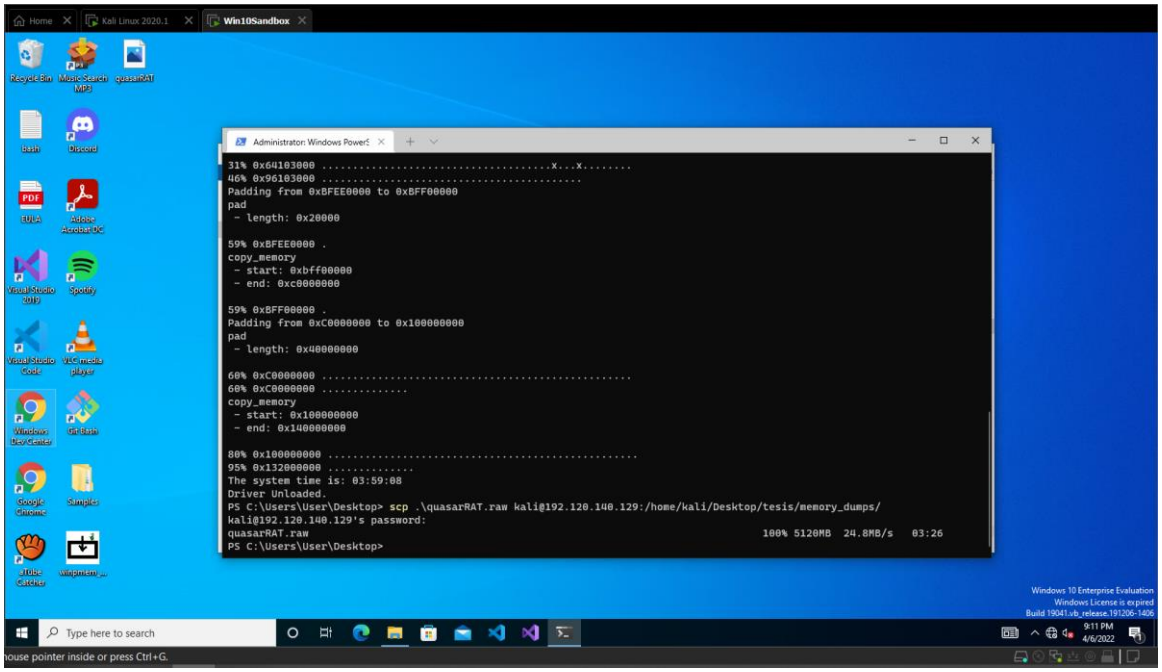


Ilustración 180: Envío de volcado de memoria de QuasarRAT hacia la máquina principal

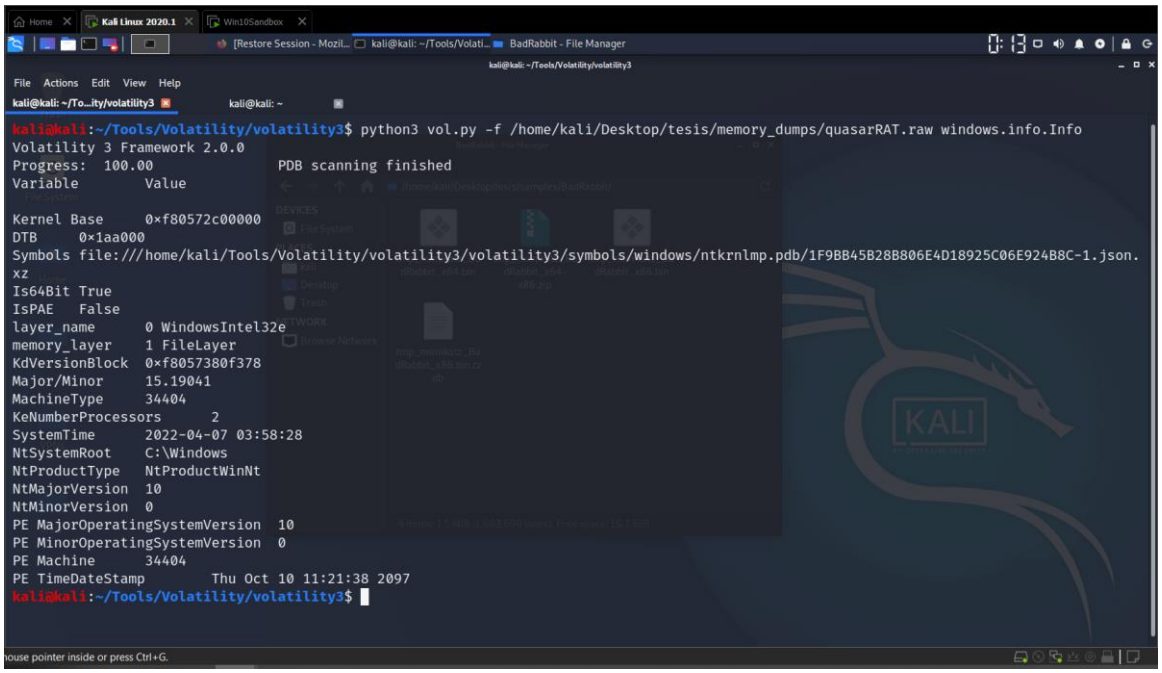


Ilustración 181: información general sobre el volcado de memoria de máquina infectada con QuasarRAT

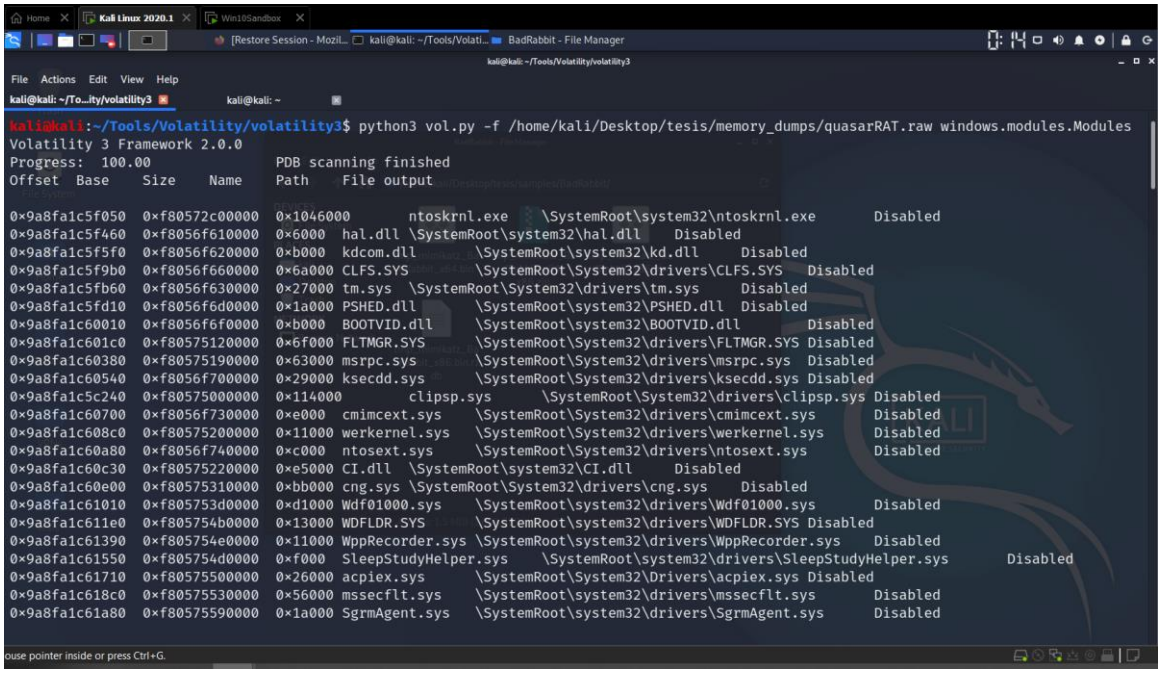


Ilustración 182: Resultados de extracción de información de plugin windows.modules.Modules con volcado de memoria de QuasarRAT

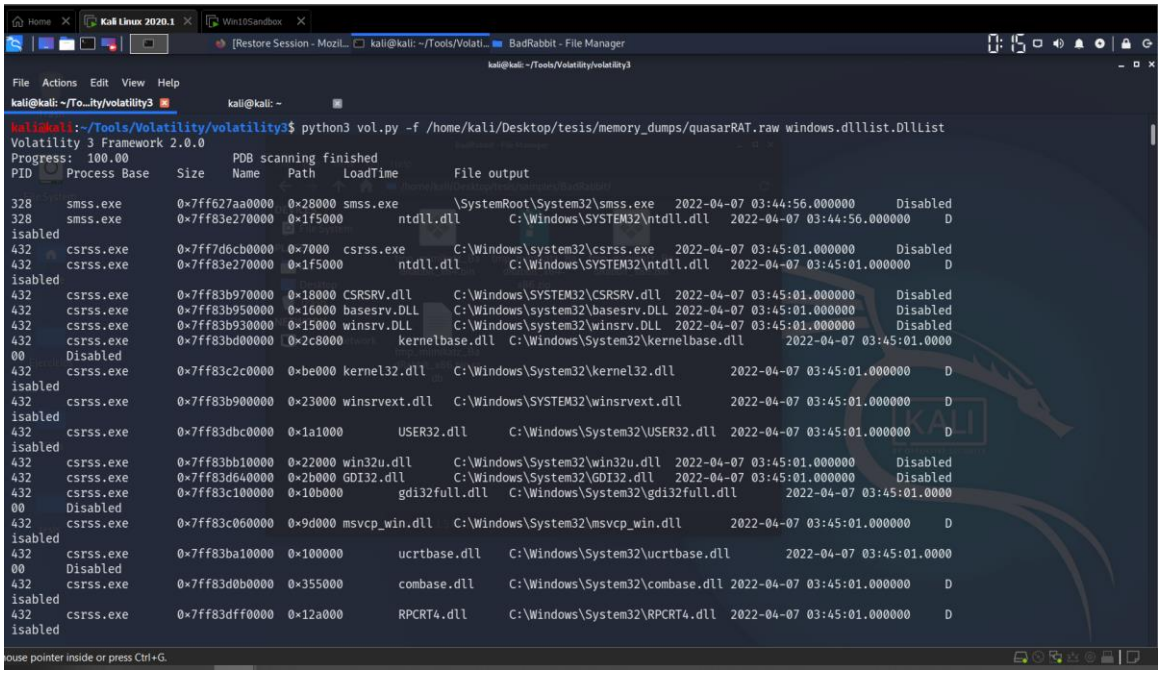


Ilustración 183: Resultados de extracción de información de plugin windows.dlllist.DllList con volcado de memoria de QuasarRAT

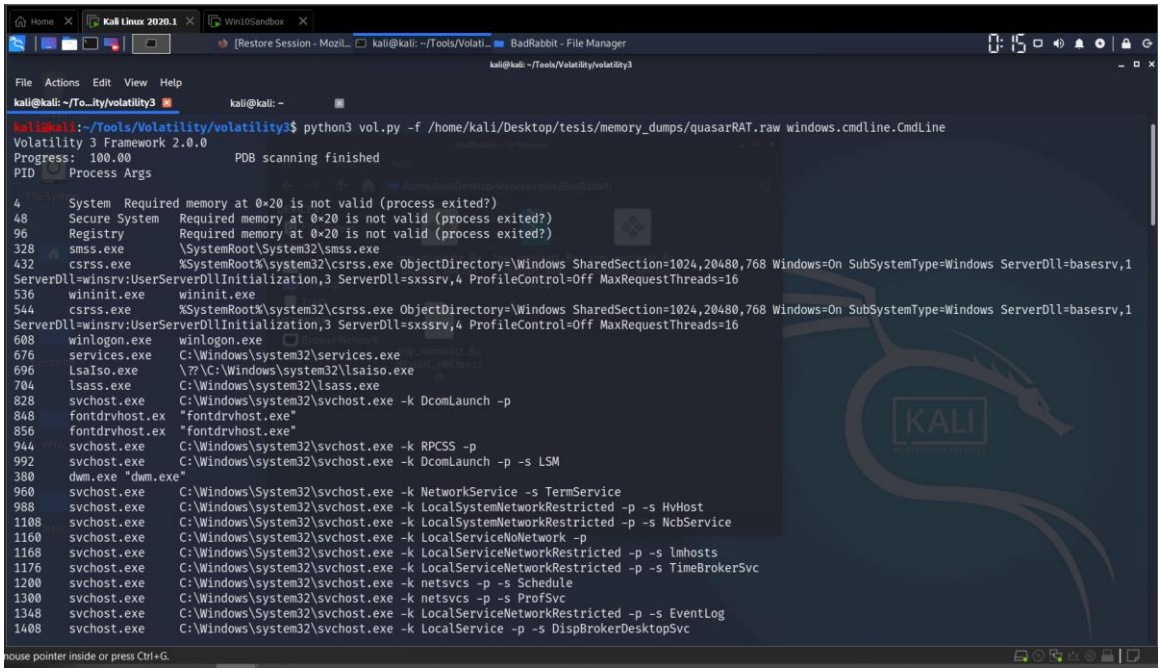


Ilustración 184: Resultados de extracción de información de plugin windows.cmdline.CmdLine con volcado de memoria de QuasarRAT

BIBLIOGRAFIA

Abuse.ch. (2020). *Introducing MalwareBazaar*. Abuse.Ch Blog. https://abuse.ch/blog/introducing-malwarebazaar/

ACSC. (2020). *Person-in-the-middle | Cyber.gov.au*. Australian Cyber Security Centre. https://www.cyber.gov.au/acsc/view-all-content/glossary/person-middle

Alvarez, V. (2020). *Welcome to YARA's documentation! — yara 4.1.0 documentation*. https://yara.readthedocs.io/en/stable/

Avira Protection Labs. (2021). *Q4 and 2020 Malware Threat Report*. https://www.avira.com/en/blog/q4-and-2020-malware-threat-report

Baker, K. (2020). *Malware Analysis Explained | Steps & Examples | CrowdStrike*. CrowdStrike. https://www.crowdstrike.com/cybersecurity-101/malware/malware-analysis/

Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*. CrowdStrike. https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/

Barnum, S. (2014). *STIX Whitepaper | STIX Project Documentation*. https://stixproject.github.io/getting-started/whitepaper/#stix-structure

Cambridge University Press. (n.d.). *Spam | meaning in the Cambridge English Dictionary*. Cambridge English Dictionary. Retrieved April 15, 2021, from https://dictionary.cambridge.org/dictionary/english/spam

Chauhan, P., & Bansal, P. (2021). Assessment of Forensics Investigation Methods. *Smart Innovation, Systems and Technologies, 213 SIST*, 317–324. https://doi.org/10.1007/978-981-33-4443-3_30

Chivers, K. (2020). *What is a man-in-the-middle attack?* NortonLifeLock Inc. https://us.norton.com/internetsecurity-wifi-what-is-a-man-in-the-middle-attack.html

CIS. (2018). What is Cyber Threat Intelligence? *Centre for Internet Security*. https://www.cisecurity.org/blog/what-is-cyber-threat-intelligence/

Cutter Dev Team. (2020). *User Documentation — Cutter 2.0.2 documentation*. Cutter User Documentation. https://cutter.re/docs/user-docs.html

Cyber and Infrastructure Security Agency. (2018). *Understanding Denial-of-Service*

Attacks | CISA. US-CERT. https://us-cert.cisa.gov/ncas/tips/ST04-015

Cyware Hacker News. (2019). *Quasar RAT: A sneak peek into the Remote Access Trojan's capabilities*. Cyware Social. https://cyware.com/news/quasar-rat-a-sneak-peek-into-the-remote-access-trojans-capabilities-18afa9a3

Defining Malware: FAQ | Microsoft Docs. (n.d.). Retrieved April 5, 2021, from https://docs.microsoft.com/en-us/previous-versions/tn-archive/dd632948(v=technet.10)?redirectedfrom=MSDN

Distler, D. (2020). *SANS Institute Information Security Reading Room Malware Analysis: An Introduction*.

Ducklin, P. (2019). *7 types of virus – a short glossary of contemporary cyberbadness*. Naked Security by Sophos. https://nakedsecurity.sophos.com/2019/12/28/7-types-of-virus-a-short-glossary-of-contemporary-cyberbadness/

ESET. (2017). *8 Things You Should Know About Spyware*. WeLiveSecurity by ESET. https://www.welivesecurity.com/2017/02/17/8-things-know-spyware/

F-secure. (2021). *Email-Worm:VBS/LoveLetter Description | F-Secure Labs*. https://www.f-secure.com/v-descs/love.shtml

Figueroa-Suárez, J. A., Rodríguez-Andrade, R. F., Bone-Obando, C. C., & Saltos-Gómez, J. A. (2018). La seguridad informática y la seguridad de la información. *Polo Del Conocimiento*, *2*(12), 145. https://doi.org/10.23857/pc.v2i12.420

Fox, N. (2021). *YARA Rules Guide: Learning this Malware Research Tool | Varonis*. Varonis. https://www.varonis.com/blog/yara-rules/

Ganacharya, T. (2018). *A worthy upgrade: Next-gen security on Windows 10 proves resilient against ransomware outbreaks in 2017 - Microsoft Security*. Microsoft Security Blog. https://www.microsoft.com/security/blog/2018/01/10/a-worthy-upgrade-next-gen-security-on-windows-10-proves-resilient-against-ransomware-outbreaks-in-2017/

GitHub - quasar/Quasar: Remote Administration Tool for Windows. (n.d.). Retrieved October 10, 2021, from https://github.com/quasar/Quasar

GitHub - rizinorg/cutter: Free and Open Source Reverse Engineering Platform powered by rizin. (n.d.). Retrieved October 10, 2021, from https://github.com/rizinorg/cutter

Gragido, W. (2013, October 3). *Understanding Indicators of Compromise (IOC)*. RSA Research -Speaking of Security - Security Operations; RSA. http://blogs.rsa.com/understanding-indicators-of-compromise-ioc-part-i/

GReAT [Global Research & Analysis Team]. (2017). WannaCry ransomware used in widespread attacks all over the world. *SecureList by Kaspersky*, 1. https://securelist.com/wannacry-ransomware-used-in-widespread-attacks-all-over-the-world/78351/

GReAT [Global Research & Analysis Team]. (2019). *OilRig's Poison Frog – old samples, same trick*. SecureList by Kaspersky. https://securelist.com/oilrigs-poison-frog/95490/

Hewlett Packard. (2019). *What Are the Most Common Types of Cyber Attacks?* Hewlelt Packard. https://www.hp.com/us-en/shop/tech-takes/most-common-types-of-cyber-attacks

Imperva. (2019). What is phishing | Attack techniques & scam examples | Imperva. In *Imperva* (pp. 1–2). https://www.imperva.com/learn/application-security/phishing-attack-scam/

INCIBE. (2017). *Amenaza vs Vulnerabilidad, ¿sabes en qué se diferencian? | INCIBE*. Web. https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian

Intezer. (2022). *Intezer- About*. Intezer. https://www.intezer.com/about/

Intezer Analyze – All-In-One Malware Analysis Platform. (n.d.). Retrieved January 8, 2022, from https://analyze.intezer.com/

Johansen, A. G. (2020). *What is a Trojan? Is It Virus or Malware? How It Works | Norton*. Norton - Security Centre. https://us.norton.com/internetsecurity-malware-what-is-a-trojan.html

Kabakus, A. T., & Dogru, I. A. (2018). An in-depth analysis of Android malware using hybrid techniques. *Digital Investigation*, *24*, 25–33. https://doi.org/10.1016/j.diin.2018.01.001

Kaspersky. (2016). *Malware: Difference Between Computer Viruses, Worms and Trojans*. Youtube. https://www.youtube.com/watch?v=n8mbzU0X2nQ&t=154s

Kaspersky. (2018). *RAT (Remote access tools)*. Kaspersky IT Encyclopedia.

https://encyclopedia.kaspersky.com/glossary/rat-remote-access-tools/

Kaspersky. (2019). *What is Cyber Security? | Definition, Types, and User Protection | Kaspersky*. Www.Kaspersky.Co.Uk. https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security

Kaspersky. (2020). *What is a Trojan Virus | Trojan Virus Definition | Kaspersky*. Kaspersky. https://www.kaspersky.com/resource-center/threats/trojans

Kaspersky. (2021a). *¿Qué es un virus troyano? Amenazas de seguridad en Internet Kaspersky*. https://www.kaspersky.es/resource-center/threats/trojans

Kaspersky. (2021b). *Todo sobre el ransomware WannaCry*. Centro de Recursos de Kaspersky. https://www.kaspersky.es/resource-center/threats/ransomware-wannacry

Khandelwal, S. (2018). *FBI seizes control of a massive botnet that infected over 500,000 routers*. The Hacker News. https://thehackernews.com/2018/05/vpnfilter-botnet-malware.html

Kulikova, T. (Kaspersky), Shcherbakova, T. (Kaspersky), & Sidorina, T. (Kaspersky). (2021). *Spam and phishing in 2020*. SecureList by Kaspersky. https://securelist.com/spam-and-phishing-in-2020/100512/

Kupreev, O., Badovskaya, E., & Gutnikov, A. (2021). *DDoS attacks in Q4 2020 | Securelist*. SecureList by Kaspersky. https://securelist.com/ddos-attacks-in-q4-2020/100650/

Lee, H.-K. (2020). A Study on Hacking E-Mail Detection using Indicators of Compromise★. *Journal of Information and Security*, *20*(3), 21–28. https://doi.org/10.33778/kcsa.2020.20.3.021

Lejonqvist, G., & Larsson, O. (2018). *Improving the precision of an Intrusion Detection System using Indicators of Compromise-a proof of concept* [Luleå University of Technology]. https://www.diva-portal.org/smash/get/diva2:1229171/FULLTEXT02.pdf

Léveillé, M.-E. (2017). *Bad Rabbit: el ransomware Not-Petya está de vuelta y con algunas mejoras*. WeLiveSecurity by ESET. https://www.welivesecurity.com/la-es/2017/10/25/bad-rabbit-not-petya-de-vuelta/

López, A. (2016). *Unidos contra las ciberamenazas: Information Sharing*. INCIBE-CERT. https://www.incibe-cert.es/blog/unidos-las-ciberamenazas-information-sharing

Lord, N. (2017). *What are Indicators of Compromise? | Digital Guardian*. https://digitalguardian.com/blog/what-are-indicators-compromise

Maltego Technologies. (2021). *Intezer Analyze : Maltego Support*. Maltego Docs. https://docs.maltego.com/support/solutions/articles/15000042307-intezer-analyze#overview-0-0

Malwarebytes. (2020a). *Ransomware - What Is It & How To Remove It | Malwarebytes*. Malwarebytes. https://www.malwarebytes.com/ransomware/

Malwarebytes. (2020b). *Spyware - What is it & how to remove it? | Malwarebytes*. Malwarebytes. https://www.malwarebytes.com/spyware/

Mandiant. (2022). *mandiant/capa: The FLARE team's open-source tool to identify capabilities in executable files.* GitHub. https://github.com/mandiant/capa

Matthews, B. (2017). Sophisticated Indicators for the Modern Threat Landscape: An Introduction to OpenIOC. *OpenIOC.Org*, 1–10. https://silo.tips/download/sophisticated-indicators-for-the-modern-threat-landscape-an-introduction-to-open

MaxXor. (2020). *Quasar: Remote Administration Tool for Windows*. GitHub. https://github.com/quasar/Quasar

Mcafee. (2020). *What Is Ransomware? | McAfee*. What Is Ransomware? https://www.mcafee.com/enterprise/en-us/security-awareness/ransomware.html

McAfee Labs. (2021). *McAfee Labs Threats Report 04.2021* (Issue March).

NortonLifeLock. (2018). *What Are Bots?* Norton. https://us.norton.com/internetsecurity-malware-what-are-bots.html

NortonLifeLock. (2019). *What is a computer worm and how does it work?* Norton. https://us.norton.com/internetsecurity-malware-what-is-a-computer-worm.html

NSA. (2019). *Ghidra is a software reverse engineering (SRE) framework: NationalSecurityAgency/ghidra*. https://github.com/NationalSecurityAgency/ghidra

OASIS. (2019). *STIX™ Version 2.1*. https://docs.oasis-open.org/cti/stix/v2.1/cs01/stix-v2.1-cs01.html

Omar, K., Sihwail, R., & Ariffin, K. A. Z. (2018). A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis. *Article in International*

Journal on Advanced Science Engineering and Information Technology · September, *8*(2). https://doi.org/10.18517/ijaseit.8.4-2.6827

Or-Meir, O., Nissim, N., Elovici, Y., & Rokach, L. (2019). Dynamic malware analysis in the modern era—A state of the art survey. *ACM Computing Surveys*, *52*(5), 1–48. https://doi.org/10.1145/3329786

Perekalin, A. J. (2017). *Bad Rabbit: A new ransomware epidemic is on the rise*. Kaspersky Daily. https://www.kaspersky.com/blog/bad-rabbit-ransomware/19887/

Ramsdale, A., Shiaeles, S., & Kolokotronis, N. (2020). A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages. *Electronics 2020, Vol. 9, Page 824*, *9*(5), 824. https://doi.org/10.3390/ELECTRONICS9050824

Regan, J. (2020). *A Brief History of Viruses and Malware | AVG*. Signal by AVG. https://www.avg.com/en/signal/history-of-viruses

Robert Moir. (2009). *Defining Malware: FAQ | Microsoft Docs*. Microsoft. https://docs.microsoft.com/en-us/previous-versions/tn-archive/dd632948(v=technet.10)?redirectedfrom=MSDN

Schatz, D., Bashroush, R., & Wall, J. (2017). Towards a More Representative Definition of Cyber Security. *The Journal of Digital Forensics, Security and Law*, *12*(2), 8. https://doi.org/10.15394/jdfsl.2017.1476

Seguin, P. (2020). *¿Qué es el spyware? | Definición de spyware | Avast*. Avast Academy. https://www.avast.com/es-es/c-spyware

Smith, B. (2017). Microsoft and Facebook disrupt ZINC malware attack to protect customers and the internet from ongoing cyberthreats. *Microsoft on the Issues*, 2019–2020. https://blogs.microsoft.com/on-the-issues/2017/12/19/microsoft-facebook-disrupt-zinc-malware-attack-protect-customers-internet-ongoing-cyberthreats/

Sophos. (2019). *What are Botnets and How Can I Block Bots on My Home Computers?* Sophos Home. https://home.sophos.com/en-us/security-news/2019/what-is-a-bot.aspx

US-CERT. (2005). *An Undirected Attack Against Critical Infrastructure A Case Study for Improving Your Control System Security An Undirected Attack Against Critical Infrastructure A Case Study for Improving Your Control System Security Sponsor*. https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/CaseStudy-

002.pdf

VirusTotal. (2017). *I am experiencing a false positive, my file or site should not be detected. – VirusTotal*. VirusTotal FAQ. https://support.virustotal.com/hc/en-us/articles/115002121185-I-am-experiencing-a-false-positive-my-file-or-site-should-not-be-detected-

VirusTotal. (2021). *How it works – VirusTotal*. https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works

Woodham, W. (2019). *Trojan:Win32/Generic — Virus Removal Guide*. How to Fix.Guide. https://howtofix.guide/trojan_win32_generic/

Young, A., & Yung, M. (1996). Cryptovirology: extortion-based security threats and countermeasures. *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, 129–140. https://doi.org/10.1109/secpri.1996.502676

Zeltser, L. (2021). *How You Can Start Learning Malware Analysis*. SANS Institute. https://www.sans.org/blog/how-you-can-start-learning-malware-analysis/

Zhou, S., Long, Z., Tan, L., & Guo, H. (2018). Automatic identification of indicators of compromise using neural-based sequence labelling. *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, PACLIC 2018*, 849–857.