Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Seguridad Informática

Implementación de un laboratorio para el análisis de malware

Trabajo fin de estudio presentado por:	Carolina Lituma Briones
Tipo de trabajo:	Piloto experimental
Director/a:	Dr. Javier Bermejo
Fecha:	2 de marzo del 2022

Agradecimientos

Agradezco a Dios por las bendiciones recibidas. A mi esposo Kléber por el apoyo incondicional en este proceso, a mi mamá y mis hermanas por inspirarme a seguir creciendo tanto personal como profesionalmente y al Ing. Iván por ser mi mentor en el ámbito profesional y por los aportes brindados en esta Memoria. Expreso mis sentimientos de gratitud al Dr. Bermejo por su profesionalismo, paciencia y guía en todo el proceso de TFM. Así mismo, a mi tutora Leidy por la orientación concedida en la maestría.

Resumen

A nivel mundial las infecciones y ataques por código malicioso crecen exponencialmente, según Kaspersky informa que en el año 2020 se encontraron un promedio de 360.000 nuevos archivos maliciosos por día, que infectan a dispositivos como ordenadores, teléfonos móviles e infraestructura tecnológica clave para las organizaciones. El presente piloto experimental estableció como objetivo implementar un laboratorio virtual que permita realizar análisis de malware con la finalidad de lograr una mejor comprensión de los mecanismos internos de su funcionamiento, debido a que una de las principales amenazas en la red es la infección por código malicioso. La metodología seleccionada para alcanzar este objetivo se basó en Systematic Approach to Malware Analysis (SAMA), que proporcionó una guía eficaz en todo el ciclo del análisis del malware a pesar de la complejidad de este, lográndose la creación del laboratorio virtual en entornos de sistemas operativos Windows, Linux y Android que permitirá establecer prospectivamente técnicas de defensas para proteger los diversos sistemas operativos existentes.

Palabras claves: Malware, análisis de código, análisis de comportamiento, laboratorio virtual, metodología SAMA.

Abstract

Worldwide infections and attacks by malicious code are growing exponentially, according to Kaspersky reports that in 2020 an average of 360,000 new malicious files were found per day, which infect devices such as computers, mobile phones and key technological infrastructure for organizations. The objective of this experimental pilot was to implement a virtual laboratory that allows malware analysis in order to achieve a better understanding of the internal mechanisms of its operation, due to the fact that one of the main threats on the network is infection by malicious code. The methodology selected to achieve this objective was based on Systematic Approach to Malware Analysis (SAMA), which provided an effective guide throughout the malware analysis cycle despite its complexity, thus achieving the creation of the virtual laboratory in Windows, Linux and Android operating system environments that will allow the prospective establishment of defense techniques to protect the various existing operating systems.

Keywords: Malware, code analysis, behavior analysis, virtual laboratory, SAMA methodology.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

1.1. Motivación

El malware es un software que se inserta en un sistema sin autorización con fines maliciosos, su nominación nace de la combinación de las palabras “malicioso” y “software” y es utilizado en el campo de las Tecnologías de la Información (TI). Este tipo de ataque está creciendo en volumen y evolucionando en complejidad [1], es por eso por lo que en los últimos años se ha evidenciado que los grupos de cibercriminales se han volcado al uso de técnicas cada vez más complejas para lograr ataques cada vez más certeros [2].

En el reporte anual de seguridad de la empresa ESET 2021, se manifiesta que la primera causa de incidentes de seguridad es la infección de código malicioso, por lo tanto, ésta se convierte en la principal preocupación en el área de seguridad de las empresas latinoamericanas. Así mismo, obtuvieron como resultado de sus encuestas que las soluciones antimalware son los controles de seguridad técnicos más utilizados [3].

La red de la Facultad de Sistemas y Telecomunicaciones (FACSISTEL) – UPSE se inestable e insegura, atenta contra la integridad, disponibilidad y confidencialidad de los datos que fluyen a través de la misma, conllevando a que sus usuarios estén expuestos a la infección de software malicioso. El presente estudio pretende que la arquitectura del laboratorio virtual sea moderna e instalada en los servidores de FACSISTEL, utilizando herramientas de análisis de malware, para tener un amplio conocimiento de los posibles daños que puede ocasionar el software malicioso en la red, además del comportamiento del mismo.

1.2. Planteamiento del trabajo

En el momento actual el malware es una de las principales amenazas a las redes de comunicaciones de la Facultad de Sistemas y Telecomunicaciones de la UPSE, tomando en cuenta la cantidad de información que fluye por esta red, y por lo anteriormente expuesto, es de gran importancia crear un laboratorio virtual de malware con una arquitectura moderna que permita el análisis en todas las fases del ciclo de vida del software malicioso (acciones iniciales, clasificación, análisis estático y dinámico de su código, análisis del comportamiento) y presente un informe de los hallazgos encontrados, clasificación del malware, sus comportamientos y el nivel de daño que podría ocasionar.

El fin del presente trabajo de investigación será crear un laboratorio de análisis de malware para lograr una mejor comprensión de los mecanismos internos de funcionamiento del código malicioso y de las propias herramientas de análisis propuestas.

1.3. Estructura del trabajo

El presente trabajo de Fin de Máster está estructurado de la siguiente manera:

Capítulo 1: Introducción. En este apartado se describe de manera general la sinopsis del trabajo fin de máster.

Capítulo 2: Estado del arte. En este apartado se realiza una recopilación de las investigaciones que contienen soluciones propuestas referente al tema de investigación.

Capítulo 3: Objetivos y metodología de trabajo. En este apartado se describen los objetivos, tanto general como específicos, y la metodología que se empleará durante todo el trabajo de piloto experimental.

Capítulo 4: Propuesta. En este capítulo se desarrollará la propuesta del trabajo de investigación, en el que se describirán los componentes y diseño de la propuesta, arquitectura del laboratorio y las herramientas seleccionadas para el análisis.

Capítulo 5: Evaluación y resultados. En este apartado se desarrollarán las pruebas en el laboratorio junto a las herramientas seleccionadas para el análisis y siguiendo la metodología propuesta. Se elegirá una muestra de los malwares encontrados en la red para su estudio.

Capítulo 6: Conclusiones y futuras líneas de desarrollo. En este capítulo se definen las conclusiones y recomendaciones principales del trabajo investigativo, además se enlista una serie de posibles líneas de trabajo futuro que permitan tener una continuación del presente piloto experimental.

2. Contexto y Estado del Arte

2.1. Introducción

Antes de realizar un estudio en profundidad sobre el análisis de malware, es necesario entender la definición del mismo.

Existen varias definiciones para el malware, las más breves son: software malicioso o programa malicioso [4]. Sikorski lo define como “cualquier software que realice una acción que pueda causar daño a un usuario, computadora o red” [5].

El programa maligno es una amenaza cada vez mayor para la informática moderna. Los autores de malware incorporan continuamente varias funciones sofisticadas, como la ocultación de códigos, para crear variantes de malware y eludir la detección por parte de los sistemas de detección de malware existentes [6]. El presente apartado está enfocado en las diversas investigaciones previas relacionadas con el análisis del software malicioso que permitan concatenar y contextualizar el objeto de estudio de la presente investigación.

2.2. Investigaciones relacionadas

En la investigación “Desarrollo de un sistema de análisis e ingeniería inversa de código malicioso”, realizada por el Ph.D Javier Bermejo Higuera, expresa que “el ciberespacio ha pasado a ser el quinto dominio de la guerra, uniéndose a los tradicionales tierra, mar, aire y espacio. Un arma clave en este dominio es el código malicioso, en adelante malware, que se convierte así en un nuevo tipo de arma o instrumento de proyección de poder e influencia”, por esta razón el autor considera relevante conocer la estructura, funcionamiento e interacción del malware, no sólo para el diseño y desarrollo de contramedidas eficaces, sino también para ayudar a conocer el origen de un ataque [7].

Para llevar a cabo esta investigación se consideró escoger un tipo sofisticado de ciberataque organizado denominado “Advanced Persistent Threads (APT)” y a su vez, desarrollar una metodología de análisis e ingeniería inversa de código malicioso, con la arquitectura del laboratorio donde se llevará a cabo el análisis de malware, así como un estudio de las herramientas de análisis de malware a utilizar en cada paso de la metodología de análisis propuesta [7].

Por otro lado, el artículo científico “Enfoque sistemático para el análisis de malware (SAMA)” desarrollado por varios autores, entre ellos el Ph.D Javier Bermejo y el Ph.D Carlos Abad, describen un método de análisis de malware sistemático y metodológico con el objetivo de comprender el comportamiento de un malware en particular [8].

La metodología propuesta tiene las siguientes características: no depende de la complejidad del malware a analizar, es independiente de las herramientas utilizadas facilitando su adopción, en realza la importancia de la clasificación del malware como uno de los primeros pasos a realizar, considera la integración de cualquier información existente del malware como entrada para el análisis, define una secuencia específica para el análisis basada en realizar el análisis de código, antes del análisis dinámico o conductual, e incluye un circuito de retroalimentación sistemático para retomar el proceso cuando la complejidad del malware lo requiera, así como para mejorar el análisis actual alimentando etapas anteriores con los resultados obtenidos de las posteriores [8].

Para la ejecución de esta metodología los autores eligieron dos softwares maliciosos a los que ellos consideran como “más complejos y conocidos” denominados “Flame” y el “Red October”, a su vez proporcionaron un informe de aplicación de estos [8].

Así mismo, el proyecto “MMALE: Una metodología para el análisis de malware en entornos Linux” fue realizado porque los autores en su investigación expresan que “los sistemas basados en Linux se han vuelto más atractivos para los ciberdelincuentes debido al uso cada vez mayor del sistema operativo Linux en servidores web y dispositivos de Internet de las cosas (IoT)” por tal motivo en la presente investigación los autores encontraron la necesidad de desarrollar una metodología para el análisis de malware en el sistema operativo Linux. Esta metodología fue evaluada mediante un caso de estudio e incluyó un análisis completo del malware de Linux. El software malicioso elegido para las pruebas fue Linux.Encoder.1 perteneciente a la familia de Ransomware [9].

Aslam y Samet [10] realizaron una investigación comparativa titulada “Una revisión completa sobre los enfoques de detección de malware” referente a los enfoques y métodos de detección de malware, en donde según su análisis los enfoques basados en firmas y heurísticas son rápidos y eficientes para detectar malware, sin embargo, no son capaces de detectar software malicioso desconocidos o nuevos. Así mismo indican que los enfoques basados en el

comportamiento, en la comprobación de modelos y en la nube funcionan de manera eficiente en los malwares desconocidos o nuevos, pero no detectan todo el malware en su naturaleza.

Con base en lo anteriormente expuesto, los autores presentan una revisión detallada de los métodos y enfoques de detección de malware con el fin de ayudar a los investigadores a tener una idea general de los mismos, los pros y los contras, los métodos que se utilizan, además de la gran brecha para nuevos estudios [10].

La investigación científica denominada "Clasificadores de machine learning para análisis de malware en Android" [11] desarrollada por Christian Camilo Urcuqui López y Andrés Navarro Cadavid aplican que Android emplea varios tipos de mecanismos de seguridad para restringir la difusión de malware, entre los más destacados son la API de permisos en los privilegios de accesos en las aplicaciones, el entorno sandbox al nivel Kernel, los mecanismos de seguridad en el desarrollo de las aplicaciones y su plataforma de descarga Google Play, sin embargo, estos mecanismos no eliminan la propagación de software malicioso, tan solo la limita.

Por esta razón los autores desarrollaron un módulo de Inteligencia Artificial que permita detectar el malware en Android, a partir de los algoritmos Naive Bayes, Bagging, Kneighbors, Super Vector Machines, Stochastic Gradient Descent y Decision Tree [11].

2.3. Laboratorio de análisis de MALWARE

El análisis de malware es una actividad compleja que tiene como misión entender las acciones que realiza el malware y el fin de las mismas [4].

Realizar las acciones de comprender, analizar e investigar las amenazas de un malware conlleva el uso de herramientas, metodologías y técnicas, integradas en un entorno, por lo que, es necesario crear un laboratorio para el análisis del código malicioso [12].

Existen dos tipos de enfoques principales para la creación de un laboratorio de análisis de malware [13], el primero es mediante dispositivos físicos y el segundo a través de virtualización [14].

2.3.1. Enfoque 1: Laboratorio físico de análisis de malware

La implementación de un laboratorio físico es un conjunto de dispositivos (computadoras, teléfonos móviles, tabletas, entre otros) [4], en donde se lleva a cabo las fases de metodología

para el análisis de malware [15]. A continuación, se enlistará las ventajas y desventajas de este enfoque:

Tabla 1: Ventajas y desventajas de un laboratorio físico

Ventajas
•El entorno de análisis es más realista •Mayor exactitud en el análisis
Desventajas
•Los dispositivos seleccionados deberán ejecutar los sistemas operativos y sus versiones seleccionados para la prueba •Mayor costos, espacio y tiempo •Posible infección de los dispositivos conectados a la red en donde se están realizando las pruebas

2.3.2. Enfoque 2: Laboratorio virtual de análisis de malware

La implementación de un laboratorio virtual consiste en la implantación de un hipervisor que contiene máquinas virtuales que simulan un entorno real [16]. Existen una gran gama de software de virtualización, los más destacados son VirtualBox, VMWare, SandBoxie, etc. [4]. A continuación, se enlistará las ventajas y desventajas de este enfoque:

Tabla 2: Ventajas y desventajas de un laboratorio virtual

Ventajas
•Captura de instantaneas que permiten el respaldo de cada fase en el análisis y la restauración del mismo. •Costos reducidos y temporalidad •Proporciona un contexto genuino, sin descuidar la protección que el usuario debe de tener al enfrentarse a riesgos que podrían aparecer mientras realiza ciertas pruebas y ensayos en los laboratorios de la investigación. •Permiten usar cualquier tipo de tecnología
Desventajas
•Fáciles de detectar •Posible infección de la máquina host

2.4. Clasificación y comportamiento del malware

Según Kaspersky en su blog de noticias destacadas informa que, en el año 2020 se encontraron un promedio de 360.000 nuevos archivos maliciosos por día [17]. Lo cual hace que en esta

investigación sea necesario identificar los principales tipos de malwares existentes y una breve descripción del comportamiento de cada uno de ellos.

Tabla 3: Clasificación del malware

Nombre	Comportamiento	Métodos de infección
Virus	Son un tipo de código auto-replicante, es decir, se reproducen por sí mismos [18]. Infectan el sistema operativo [14]. Se instalan sin el consentimiento del usuario [18].	Generalmente se alojan en los correos electrónicos como archivos ejecutables o documentos de Word [14].
Gusanos	Al igual que los virus, el worms o gusanos también son programas auto-replicantes [18]. Se instalan directamente en las computadoras, pero no infectan a los archivos existentes [18]. Realizan múltiples copias de sí mismo [19]. Esperan a que aparezca la oportunidad de infectar en otros sistemas a través de la red [14].	Realizan la infección a través de correos electrónicos, mensajes instantáneos o compartiendo archivos [18]. Las técnicas más populares son: Exploit de buffer overflow: un gusano envía más datos al programa que los que esperaba originalmente, desbordando el buffer y corrompiendo varias estructuras críticas de memoria. Ataque de archivos compartidos: este tipo de ataque se realiza usando archivos compartidos en Windows o NFS de Unix, los usuarios pueden leer o escribir archivos transparentemente a través de la red. Gusanos de correo electrónico: Suelen propagarse a través de los mensajes valiéndose de la utilización de ciertos programas clientes, reenviándose automáticamente a los contactos de la libreta de direcciones. Gusanos de IRC: Estos se propagan a través de canales de IRC (Chat), empleando habitualmente para ello al mIRC y al Pirch. Gusanos de VBS (Visual Basic Script): Son gusanos escritos o creados en Visual Basic Script Gusanos de Windows 32: se propagan a través de las API de Windows
Troyanos	Toman la apariencia de software legal, pero internamente están programados para atacar [18]. No infectan archivos [14]. No son auto-replicantes, por lo que necesitan del usuario para ser replicados [19]. Se clasifican de acuerdo con la acción que realizan, como los Troyanos Backdoor (que quieren tomar el control remoto de los ordenadores de las víctimas) y los Troyanos Downloader (que instalan códigos maliciosos) [18].	Engañan al usuario disfrazándose de software de gran utilidad, para así ser descargados y cumplir con sus funciones destructivas como el espionaje, robo de información, control de accesos, etc. [18].
Ransomwore	Es un software creado para extorsionar dinero a sus víctimas [18].	Generalmente aparece en forma de pop up, enlace de phishing o web malicioso [18].

	Crea una vulnerabilidad al sistema del usuario para secuestrar el ordenador de la víctima utilizando cifrado de archivos [14].	Al cliquear en el enlace impulsa una vulnerabilidad en el sistema del usuario, que hace que la máquina sea secuestrada a través del cifrado de archivos, y finalmente los cibercriminales proceden a realizar la extorsión [14].
Rootkit	Es un malware que actúa de forma silenciosa, ocultando las actividades que realiza el atacante [18], ya que modifican ficheros y librerias, para que ni el usuario ni el software de protección puedan detectarlo [7]. Uno de los tipos de rootkit es el Bootkit, el cual se activa antes de que arranque el sistema operativo [18].	Se trata de técnicas de enmascaramiento para engañar a los antivirus para que no puedan encontrar malwares en el sistema y considerarlos como aplicaciones normales [20].
Backdoor (RAT)	Acceden al sistema sin dejar rastros ignorando los procedimientos de autenticación [21] y toman el control absoluto a distancia de la computadora afectada [14].	Tienen varias formas de ser instaladas, puede ser por medio de una explotación de vulnerabilidad del sistema, por configuraciones no seguras en el sistema o por medio de worms o virus [21].
Dowloader	Son pequeñas fragmentaciones de código diseñadas para que al descargarse en la máquina de la víctima pueda ejecutar las órdenes desde el servidor del delincuente informático [18].	Se adjuntan en correos electrónicos o imágenes [18]. Al descargarlos los cibercriminales envían instrucciones para descargar otros programas maliciosos en el equipo [18].
Spyware	Es un software malicio que actúa como espía [22]. Recopila información sin la autorización del usuario, o vigila las actividades del mismo [22].	Se auto-instala en el sistema víctima. Se ejecuta cada vez que se inicia el ordenador (utilizando CPU y memoria RAM, reduciendo la estabilidad del ordenador) [7].
Adware	Reproduce anuncios en el computador del usuario sin su permiso, interrumpiendo las actividades generadas [22]. El objetivo de este malware es beneficiarse económicamente a través de los anuncios [22].	Generalmente son instalados mediante software de distribución gratuita [14].
Keylogger	Es un software malicioso espía que se utiliza para registrar pulsaciones de teclas para robar contraseñas, datos de tarjetas de crédito y otras cifras importantes y sensibles [7].	Se transfiere a una computadora cuando se instala algún otro programa malicioso o el usuario visita cualquier sitio infectado [14].
Botnet	Permite al atacante controlar una computadora infectada. Una red de computadoras infectadas controladas por piratas informáticos / atacantes para realizar actividades maliciosas sin el conocimiento del propietario [22]. Pueden realizar ataques de denegación de servicio, enviar mensajes de spam, robar información [22].	Se instala en el sistema objetivo a través de una vulnerabilidad del equipo o usando técnicas de ingeniería social y consiste básicamente en la creación de una red de ordenadores zombis o robots de software autónomos, que son controlados remotamente mediante un sistema de mando y control «C2» por el ciberatacante normalmente a través de un canal de Chat IRC [23].
Exploits	Aprovecha una vulnerabilidad y se ejecutan dependiendo de los sistemas operativos y sus configuraciones, de las configuraciones de los programas que se están ejecutando en una computadora y de la LAN donde están [23].	Se beneficia de algún error que tenga el sistema operativo [23].

2.5. Análisis del malware

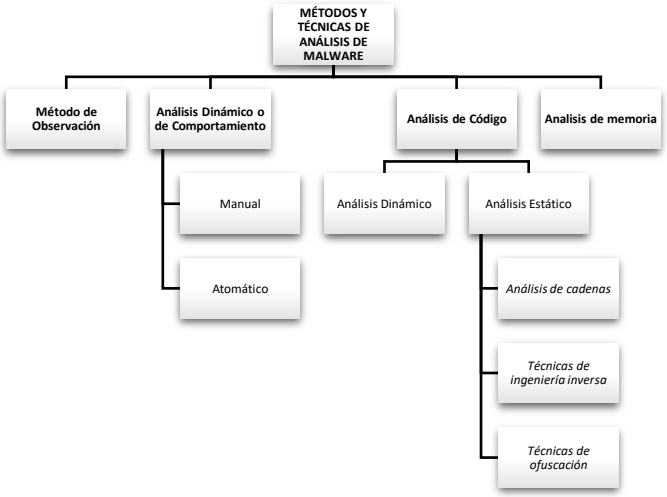
El análisis de malware tiene el objetivo de identificar la estructura, funcionamiento e interacción de un programa malicioso, para a través de sus resultados crear técnicas de defensa, evaluar las capacidades de detección y conocer el origen de un ataque [23].

Los principales beneficios del análisis de malware son los siguientes [8]:

- Evaluar la capacidad de detección de malware de los sistemas de protección de las organizaciones.
- Evaluar el daño causado por las acciones de intrusión y malware.
- Descubrir otras máquinas que han sido afectadas por el mismo malware.
- Identificar la vulnerabilidad que fue explotada por el malware, obtener la actualización de software que la mitigue, si está disponible.
- Obtener suficientes datos e información para implementar los mecanismos de defensa adecuados para mitigar y neutralizar el daño causado por el malware, incluidas las reglas de firewall, sistemas de detección de intrusiones en el host y la red, firewall y antivirus.
- Determinar el nivel de sofisticación y complejidad del malware.
- Si es posible, para determinar el origen de un ataque e identificar al intruso y al desarrollador de malware.

Existen varios métodos y técnicas de análisis de malware. A continuación, se explicará cada una de ellas:

Figura 1: Métodos de análisis de malware



Fuente: Elaboración Propia

2.5.1. Método de observación

Fue el primer método en ser usado para el análisis de malware. Consistía en observar el estado de un sistema antes y después de la infección, para luego realizar la comparación y búsqueda de archivos alterados o modificados por el programa malicioso [8].

Este método quedó obsoleto y actualmente no es usado por la poca profundidad y eficiencia en su análisis [23].

2.5.2. Análisis dinámico o de comportamiento

La técnica de análisis dinámico o de comportamiento analiza el código durante el tiempo de ejecución [23]. Si bien esta técnica no es exhaustiva tienen la ventaja significativa de que solo analiza las instrucciones que el código realmente ejecuta [24].

Esta técnica analiza el binario o espécimen del malware en ejecución, supervisa y monitoriza su interacción y comportamiento con su entorno tanto a nivel de los cambios que realiza en la máquina donde se ejecuta, como de tráfico de red y comunicaciones con sus servidores de mando y control para recibir órdenes [24]. Se tienen dos tipos de análisis de comportamiento:

- Manual.
- Automático.

Análisis Comportamiento Manual

El análisis dinámico permite visualizar y comprender la actividad que realiza el código malicioso más no la forma en como está programado [1]. Este análisis inicia en un entorno controlado para posteriormente observar su comportamiento en el mismo [10]. Esto se lleva a cabo con snapshots para tener estados del sistema que permitan comparar el antes y después de la infección, así mismo, se hace uso del monitoreo en tiempo real para verificar las acciones que realiza el programa maligno [4].

La información que se obtiene en este análisis es suficiente para evaluar la peligrosidad del código malicioso y proporcionar una visión básica de su funcionalidad [1].

Las actividades que se pueden analizar del malware en este tipo de análisis son, por ejemplo, abrir una conexión de red y transferir paquetes, crear, modificar o borrar un archivo, modificar registros, acceder a la memoria virtual, crear procesos etc. [1].

Para alcanzar los objetivos del análisis dinámico, se necesitan herramientas que permitan conocer su comportamiento, estas herramientas deben de ser capaces de detectar comportamientos de seguridad significativos de los códigos maliciosos [15].

Las herramientas generalmente usadas para este análisis son: Disk Pulse, Process Explorer,Process Monitor, AutoRuns

Análisis Comportamiento Automático

Se realiza con los sistemas Sandbox que consisten básicamente en un entorno supervisado y controlado de ejecución del código malicioso de forma que no pueda realizar ningún daño al sistema real que lo soporta [25]. La salida de estos sistemas consiste en informes con información completa de la trazabilidad de todas las acciones que realiza el malware, incluidas las llamadas a la API nativa del sistema operativo [26].

Es posible que esto no proporcione información sobre la lógica del software, pero es extremadamente útil para comprender su clasificación más amplia y a qué familia de malware podría pertenecer [27].

Las herramientas generalmente usadas para este análisis son: Cuckoo, Anubis, Malwr, Valkyrie.

2.5.3. Análisis de código

El análisis de código también se conoce como inversión de código, ya que esencialmente inicia desde el software final, retrocediendo al código y luego llegando a la lógica original [26].

Este análisis hace que un analista pueda optar por desglosar el código manualmente, utilizando herramientas como depuradores, descompiladores y descifradores. Así mismo, revela la intención estratégica detrás del software malintencionado; porque examina la lógica central del algoritmo [26].

Las herramientas generalmente usadas para este análisis son: ProcessMonitor, Sysinternals, WinDBG, Wireshark.

El análisis de código pretende investigar el funcionamiento del malware. Este método se compone de dos tipos de análisis (estático y dinámico) para obtener un conocimiento completo del objeto a analizar [12]. Se tienen dos tipos de análisis de código:

- Dinámico.

- Estático.

Análisis dinámico de código

El examen único de códigos malignos permite conocer de manera rápida y acertada qué actividades realiza una amenaza en el sistema. De esta manera puede adquirir datos sobre los registros realizados, conexiones de redes, cambios en los logs, etc.

Las herramientas generalmente usadas para este análisis son: PExplorer, IDA Pro, OllyDBG, etc.

Análisis estático de código

El análisis estático permite inspeccionar y comprender la funcionalidad e intención del código malicioso a través de la revisión de su programación [28]. Este análisis no ejecuta el binario, pero si recopila la información necesaria de su código [10].

Para la obtención de información en este análisis se requiere conocimiento de ingeniería inversa, y el uso de técnicas de desamblado, ya que los creadores de malware utilizan estrategias para ocultar el código (ofuscación de código) [22]. Se tienen tres tipos de análisis:

- Análisis de cadena.
- Técnicas de ingeniería inversa.
- Técnicas de ofuscación.

Análisis de cadena. Tiene el objetivo de realizar búsquedas en archivos ASCII, Unicode o ambos para encontrar información que permita deducir el comportamiento del programa malicioso [29]. Estos archivos pueden proporcionar datos de gran relevancia como: URL´s, datos de autenticación, protocolos, puertos, archivos, direcciones IP, nombres de procesos de máquinas, librerías DLL´s, información del malware, entre otros [23]. Herramientas: String, BinText, Dependency Walker

Técnicas de ingeniería inversa. La ingeniería inversa es una técnica que consiste generalmente en retroceder a través del ciclo de desarrollo y comprender la implementación y el funcionamiento del software. La ingeniería inversa es una fase que precede a la reingeniería, por lo tanto, no modifica el código fuente [30]. Esta técnica permite analizar y comprender el funcionamiento de un malware [31]. Herramientas: IDA Pro, WinDBG, PEstudio, ProcessExplorer, RegShot, TCPView, PEBrowse

Técnicas de ofuscación. La ofuscación es una técnica que se emplea al código fuente y al código binario [24], y se ha convertido en una gran estrategia para los creadores del software malicioso para camuflar la información relevante del programa maligno, socavar el software antimalware y frustrar el análisis de malware [32]. Los creadores de malware utilizan empaquetadores, técnicas polimórficas y técnicas metamórficas para evadir los sistemas de detección de intrusos [32].

Estas técnicas se clasifican en ofuscación de los especímenes ejecutables y restricción de los entornos de ejecución. Para tratar de evadir esta técnica, es necesario utilizar programas desempaquetadores que permitan realizar el análisis de la estructura del software malicioso. Herramientas: PEiD, PE Explorer, OllyDbg

2.5.4. Análisis de memoria

Es una técnica que permite estudiar actividades y características del malware extrayendo funciones basadas en la memoria, tales como: procesos activos y terminados, bibliotecas en vínculos finámicos (DLL), servicios en ejecución, registros, conexiones de red activas, sistema operativo en ejecución y el estado general de la computadora [33].

Las herramientas generalmente usadas para este análisis son: String, Dumpit, RamCapturer, Winpmem, FTK imager, Volatility, Rekal, Memorize, Red Line, etc.

2.6. Conclusiones de las investigaciones relacionadas

Una vez revisada la literatura científica relacionada con esta temática, se puede determinar que, para la creación del laboratorio virtual, todos los autores convergen en que es necesario conocer la forma en cómo está estructurado el malware, las actividades que ejecuta, y la interacción del mismo con el entorno, para poder tener una idea más clara de su comportamiento, las medidas de detección y prevención y a su vez determinar la génesis de su ataque.

El trabajo investigativo realizado por el Doctor Bermejo denominado SAMA, se considera como una propuesta metodológica esencial para desarrollar el análisis de malware, porque se adapta al entorno estudiado en esta tesis.

Con base a las investigaciones previas, el aporte del presente trabajo de fin de máster será crear un laboratorio virtual que integrará la metodología mencionada en el párrafo anterior

para realizar un análisis de malware en sistemas operativos Windows, Linux y Android, utilizando los métodos y técnicas existentes para el malware que fluye en las redes de telecomunicaciones de los laboratorios de FACSISTEL.

3. Objetivos concretos y metodología del trabajo

3.1. Introducción

3.2. Objetivo General

Implementar un laboratorio virtual mediante una metodología de análisis estático y dinámico de malware para lograr una mejor comprensión de los mecanismos internos de funcionamiento del malware.

3.3. Objetivos Específicos

- Estudiar las técnicas y métodos existentes en el análisis del malware.
- Examinar y determinar las herramientas de análisis de código y de comportamiento del malware
- Diseñar e implementar una arquitectura moderna del laboratorio de malware.

3.4. Metodología del trabajo

Para llevar a cabo de manera sistemática la presente investigación se desarrollarán las siguientes fases:

3.4.1. Fase 1: Recolección de datos

En esta fase se obtendrá información relevante para comprender el estado del contexto en el que se realizará el análisis tales como: estructura de la red, inventario de las máquinas, capacidad de los servidores para implementar el laboratorio de malware, obtención directa del malware.

Esta fase se llevará a cabo con las siguientes técnicas y herramientas:

- Cuestionario: se realizará una encuesta a los usuarios de los laboratorios de FACSISTEL referente al uso y los diversos casos de infección que han experimentado en el mismo.
- Entrevista: se realizará un guion de entrevista a los encargados de la red de FACSISTEL para obtener información referente a la estructura de la red, servidores, firewall, etc.
- Inventario: se realizará una recopilación de información con respecto a las PC, sistemas operativos, programas instalados en las máquinas conectadas a la red alámbrica de los laboratorios de FACSISTEL.

- Testeo de red: mediante las herramientas de testeo se monitorizará la red para verificar los datos que fluyen por la misma y la velocidad del internet.
- Dispositivos de almacenamiento: para la obtención del malware en las máquinas de los laboratorios de FACSISTEL

3.4.2. Fase 2: Laboratorio virtual

Una vez obtenida la información necesaria se procederá a realizar la virtualización de las máquinas necesarias para la creación del laboratorio virtual de malware. Esta virtualización se llevó a cabo en los servidores de FACSISTEL.

Para realizar la virtualización se utilizó el software VMWare, porque permite realizar instantáneas de las máquinas.

3.4.3. Fase 4: Análisis del malware

En esta fase se procedió aaplicar la metodología de análisis de malware SAMA, junto a las herramientas propuestas en cada fase.

3.4.4. Fase 5: Reportes

Al finalizar la etapa de análisis de malware, el laboratorio facilitará los informes necesarios para la eliminación o mitigación de los riesgos encontrados.

3.4.5. Fase 6: Conclusiones y trabajos futuros

Los reportes mencionados, permitirán tomar decisiones, por lo que, es necesario tener una fase de conclusiones y trabajo a futuros, para finalmente tener de manera clara el estado actual del entorno y las futuras mejoras que podrían implementarse.

4. Propuesta

4.1. Introducción

Analizar un software malicioso tiene como riesgo principal la propagación de este en la red, por lo tanto, se debe aplicar las medidas de seguridad pertinentes para evitar este suceso, desde el diseño del laboratorio hasta la metodología que se va a usar para el análisis del espécimen. En este apartado se describe de manera técnica lo que se pretende implementar en el presente trabajo investigativo.

4.2. Componentes de la Propuesta

4.2.1. Definición del entorno

Con base a la bibliografía revisada en este documento se puede determinar que el entorno más adecuado para llevar a cabo el análisis de malware es la creación de un laboratorio virtual, por las ventajas anteriormente explicadas.

El laboratorio será virtualizado mediante el software VirtualBox en su última versión, con los componentes necesarios para su correcto funcionamiento. Además, se instalará la herramienta Pafish en cada una de las máquinas virtuales con el fin de que el malware no detecte que se encuentra en un entorno de análisis, permitiendo simular un entorno real.

4.2.2. Escenarios

Para llevar a cabo el análisis de malware, es necesario crear los siguientes escenarios:

Víctima: se procederá a crear máquinas virtuales en donde se ejecutará el espécimen malicioso para proceder a su análisis y monitorización del comportamiento que realice en estas.

Para las máquinas víctimas tendremos tres tipos de ambientes de sistemas operativos que son:

- Sistema operativo Windows: Windows Flare
- Sistema operative Linux: Remnux
- Sistema operativo Android

Requisitos mínimos:

Para la creación de estas máquinas virtuales se establecieron estos requisitos mínimos de instalación:

Tabla 4: Requisitos de instalación de máquinas víctimas

Windows 7	REMnux	Android
Memoria RAM 4 GB	Memorian RAM 4GB	Versión 9.0 R2
DDR 80GB	DDR 60GB	
Procesador: 1	Procesador: 1	

Servicios: se procederá a crear máquinas virtuales para proporcionar al software malicioso servicios como DNS, DHCP, HTTP, FTP, SMTP, SMB, etc.

Para las máquinas de servicios tendremos dos tipos de ambientes de sistemas operativos que son:

- Sistema operativo Windows
- Sistema operative Linux: Remnux

Para la creación de esta máquina virtual se establecieron estos requisitos mínimos de instalación:

Tabla 5: Requisitos de instalación de la máquina de servicios

Windows 7	REMnux
Memoria RAM 4 GB	Memorian RAM 4GB
DDR 80GB	DDR 60GB
Procesador: 1	Procesador: 1

4.2.3. Selección de las herramientas

Las herramientas que se utilizarán serán las siguientes:

Tabla 6: Escenario de infección a Windows

WINDOWS VÍCTIMA						
Herramientas de análisis	BinText	Mobsf	WinMD5	Md5Summ	Dependency Walker	PEStudio
	PEiD	PE Explorer	PEBrowse	Process Explorer	IDA42 Pro	Disk Pulse
	Process Monitor	AutoRuns	OllyDBG	Systracer	Gmer	Antivirus
	SnapShot	Yara	WinHex	PE Viewer	Resource Hacker	VirusTotal
	Anubis	Norman SandBox	Cwsandbox	Fuzzy Hash	Command Prompt	Exeinfo PE
	ProcDump	LordPE	Reverse Engineering Compiler	Windbg	Regshot	TCPview
	Volatility	Netcat	Wireshark	VMMap	VirusTotal	Apktool
	Far Manager	Dex2jar				
Servicios	REMnux					

Tabla 7. Escenario de infección a Linux

LINUX VÍCTIMA						
Herramientas de análisis	Wxhexeditor	Trid	Exescan	DIE	Itrace	Readpe
	Pescanner	Pedump	Pestr	Bulkextractor	Strace	TCPDump
	Wireshark	YaraGenerator	IOCextractor	Autorule	Sysdig	Unhide
	VirusTotal	HashIdentifier	Objump	ELF Parser	ImageMahick	wxHexEditor
	Androwarn	AndroGuard				
Servicios	Windows Flare					

4.3. Diseño de la Propuesta

Tomando en cuenta el entorno y escenarios seleccionados para la creación del laboratorio, es necesario diseñar la arquitectura de red de este.

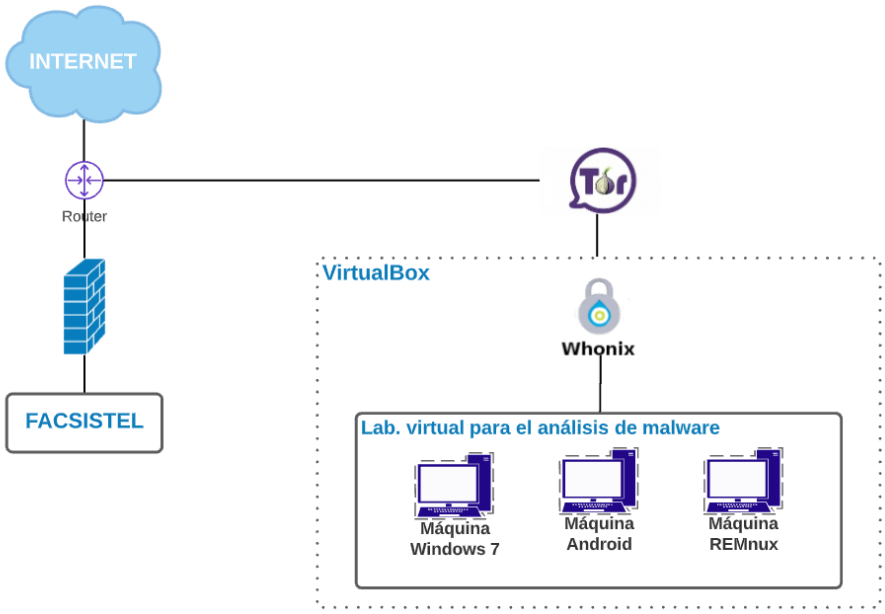
El elemento más crítico que se debe considerar en un laboratorio de malware es el peligro innato que existe en las tareas del análisis. Una gran cantidad de códigos maliciosos pueden diseminarse por la organización, intentar comunicarse con servicios externos y contaminar diferentes PC, explotando activos compartidos, ciclos remotos o debilidades, comprometiendo posteriormente la seguridad de diferentes marcos en caso de que los esfuerzos de seguridad no sean los suficientes. Por lo tanto, es insoslayable la implementación de entornos aislados.

4.3.1. Arquitectura del laboratorio virtual

El laboratorio virtual estará diseñado de tal manera que se encuentre aislado de la red de la organización, evitando así la propagación e infección de malware.

En la siguiente figura se puede visualizar que el entorno se encuentra virtualizado con el software VirtualBox, porque además de ser de libre acceso, permite tomar snapshots, es de fácil usabilidad y es compatible con todos los sistemas operativos.

Figura 2: Arquitectura del laboratorio de análisis de malware



Fuente: Elaboración propia

La arquitectura del laboratorio se encuentra dividida de la siguiente manera:

Red de FACSISTEL

La red de FACSISTEL se encuentra conectada a un router, y éste a su vez se conecta al proveedor de internet de la UPSE.

Como se puede observar en el diagrama, FACSISTEL utiliza el hipervisor PROXMOX en su servidor. Además, tiene un software como firewall para tener el control de los dispositivos conectados a la red.

Red del laboratorio virtual

El laboratorio virtual para el análisis de malware se implementará en una red aislada, detrás del firewall de la organización para evitar la infección en la misma y además estará desconectada del internet en los análisis.

En los casos en que se requiera conexión a internet (como por ejemplo el análisis de ransomware), se hará uso del sistema operativo Whonix, el mismo que permite la salida de internet a través de la red Tor y se configurará en las máquinas víctimas como gateway.

4.4. Implementación

Una vez definido las características, requerimientos, el entorno y escenarios para la creación del laboratorio virtual del análisis de malware, se procederá a llevar a cabo la implementación del mismo.

4.4.1. Laboratorio virtual

Preparación del entorno virtual

En esta fase se procedió a virtualizar los escenarios anteriormente descritos, para simular un entorno real.

Figura 3: Entorno virtual

Fuente: Elaboración propia

Configuraciones de las máquinas virtuales

La configuración general que se aplicó en las máquinas virtuales fue la obtención de los snapshots en su estado inicial.

Las máquinas en las que se requirió una configuración específica fueron las siguientes:

Sistema Operativo Windows

Una vez instalados los escenarios, se desactivaron las actualizaciones del sistema, restauración y el Windows Defender. Así mismo, se deshabilitaron las carpetas compartidas entre el host y las máquinas virtuales

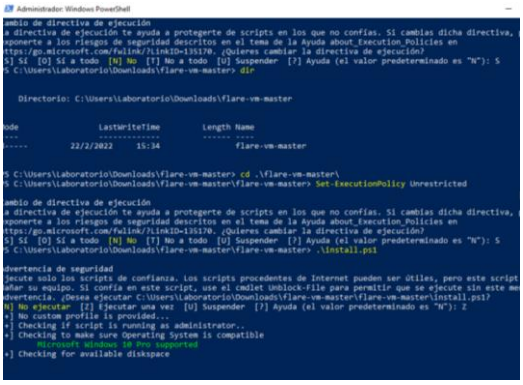
Para tener un estado inicial, se realizó una instantánea en vivo desde VirtualBox en las máquinas víctimas.

Los Snapshots se realizarán por cada evento que se haga en las máquinas virtuales con sus respectivas descricpiones.

En la máquina virtual de Windows se procedió a instalar la distribución Windows Flare. Esta distribución contiene las herramientas necesarias para crear el laboratorio de análisis de malware. A continuación, se detallan los pasos para instalar Windows Flare:

- En la máquina virtual de Windows, actualizar el sistema operativo
- Descargar FlareVM
- Abrir Power Shell como administrador
- En el Power Shell, ir a la carpeta en donde se descargó el flare-vm-master
- Escribir el comando Set-ExecutionPolicy Unrestricted para la ejecución del script
- Ejecutar el script insertado .\install.ps1

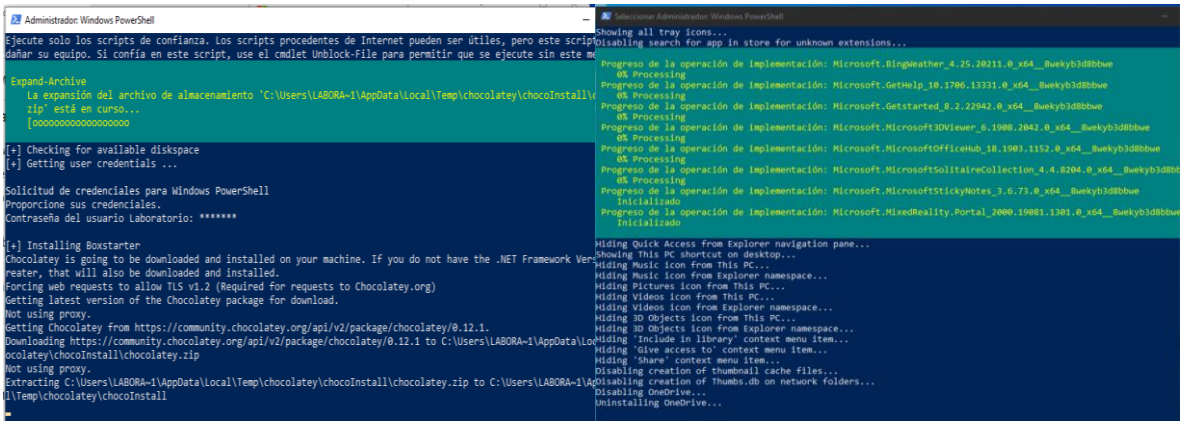
Figura 4: Ventana de Power Shell para instalar WF



Fuente: Elaboración propia

- A continuación, se empezarán a instalar todos los componentes

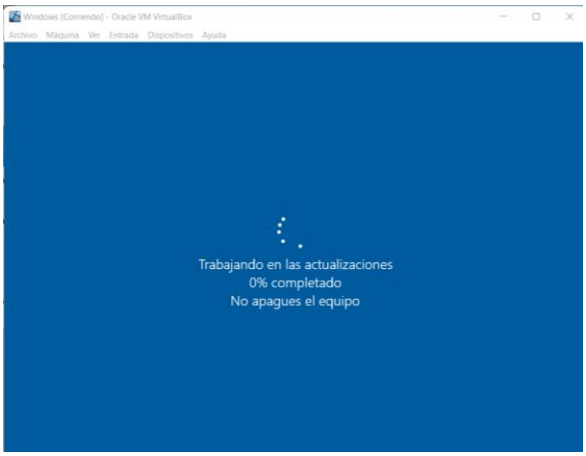
Figura 5: Componentes de WF



Fuente: Elaboración propia

- Después de realizar las instalaciones correspondientes, se reiniciará la máquina virtual

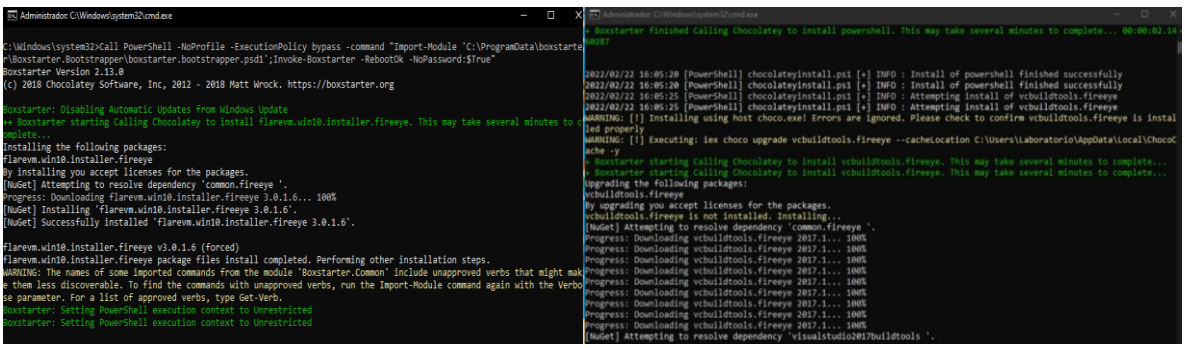
Figura 6: Instalación de las actualizaciones descargadas



Fuente: Elaboración propia

- A continuación, se abrirá la siguiente ventana para la descarga e instalación de las herramientas

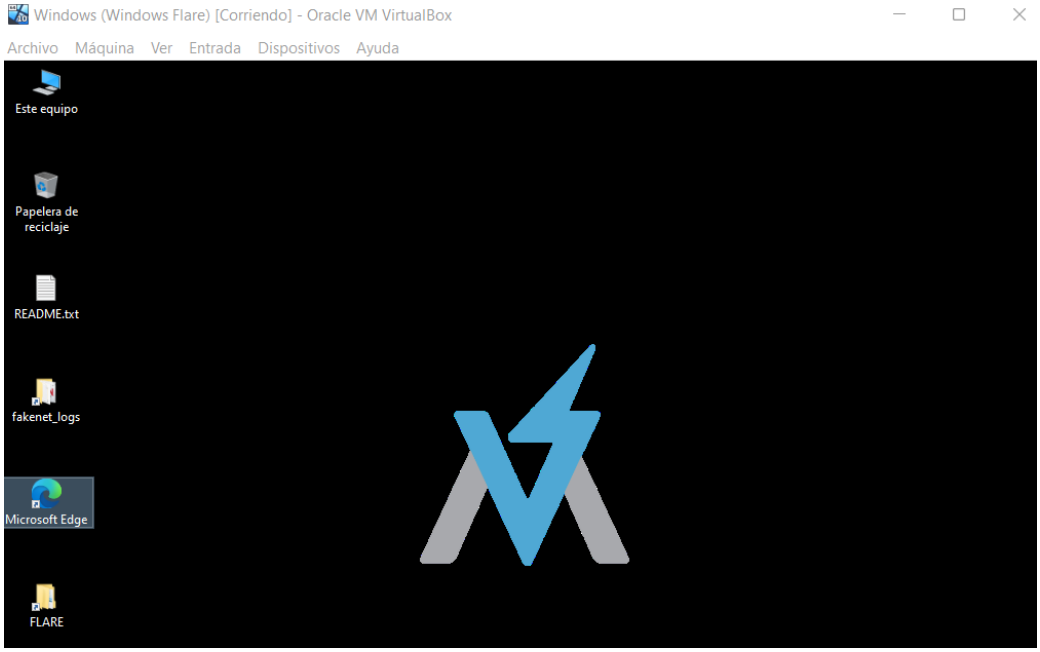
Figura 7: Descarga e instalación de ñas herramientas



Fuente: Elaboración propia

- Una vez finalizada la descarga e instalación, aparecerá un cambio de fondo de pantalla con el ícono de Flare y su carpeta con las herramientas

Figura 8: Ventana de WF



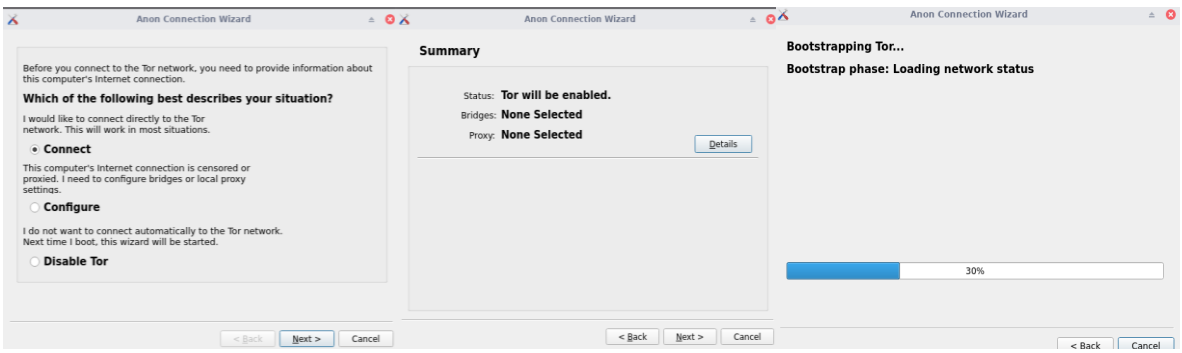
Fuente: Elaboración propia

Whonix-Gateway

Whonix es una distribución basada en Debian, que servirá como Gateway para que cuando las máquinas necesiten acceder a internet lo realicen a través de Tor.

Después de la instalación de Whonix, se procedió a realizar las configuraciones iniciales, tal como se muestra en las gráficas.

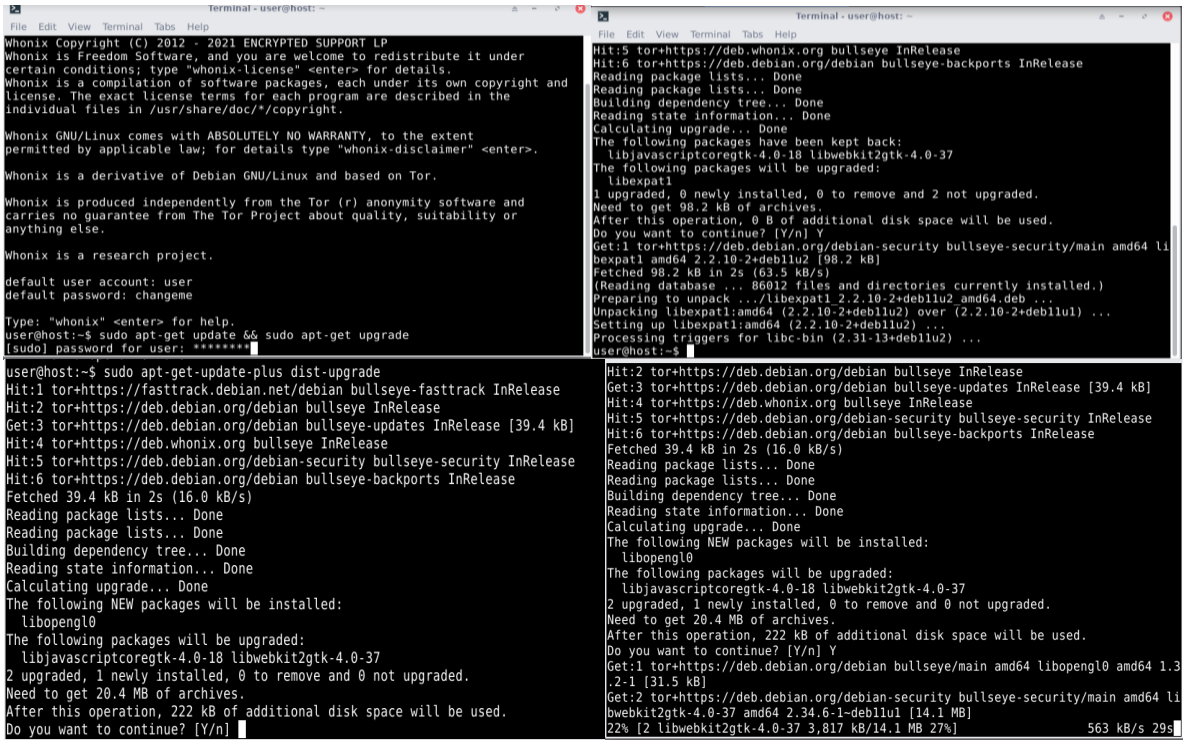
Figura 9: Configuraciones iniciales de Whonix



Fuente: Elaboración propia

Continuamente, se instalaron todas las actualizaciones correspondientes a través del terminal con el comando: sudo apt-get update && sudo apt-get upgrade y sudo apt-get-update plus dist upgrade

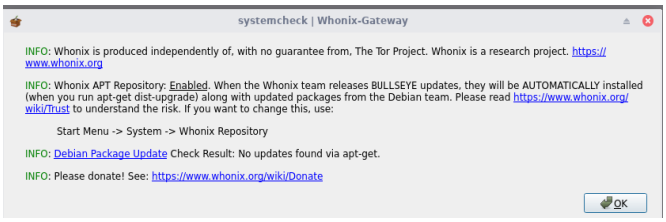
Figura 10: Instalación de las actualizaciones de Whonix



Fuente: Elaboración propia

Para comprobar que Whonix se encuentra actualizado se procede a verificar su estado en la opción System Check

Figura 11: Ventana del System Check Whonix



Fuente: Elaboración propia

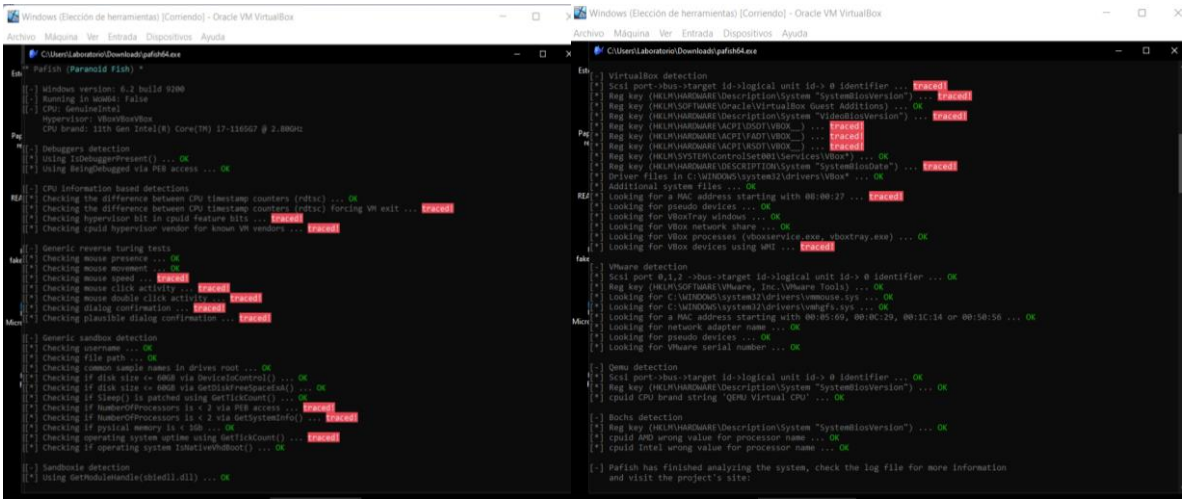
Herramienta Pafish

Es necesario realizar una configuración correcta, de tal manera que las máquinas virtuales sean casi indetectables por el malware, por esta razón se aplicó un testeo con la herramienta Pafish para verificar y corregir las configuraciones de los entornos.

A continuación, se describe los pasos para realizar el testeo:

- Descargar el archivo pafish.exe y ejecutar
- Se abrirá la consola, mostrando los resultados del testeo

Figura 12: Consola de resultados del análisis en Pafish



Fuente: Elaboración propia

Los resultados obtenidos con la herramienta indican que se rastreó siguiente:

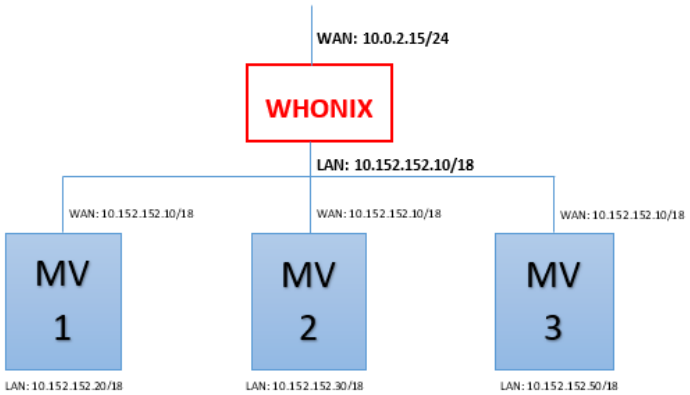
- Comprobación de la diferencia entre los contadores de marca de tiempo de la CPU de detección (rdtsc) que fuerzan la salida de la VM
- Comprobación de bit de hipervisor en bits de función de cpuid
- Comprobación del proveedor del hipervisor cpuid para conocer el proveedor de la VM
- Comprobación de la velocidad del ratón
- Comprobación de la actividad del clic del ratón
- Comprobación de la actividad de doble clic del ratón
- Comprobación de la confirmación del cuadro de diálogo
- Comprobación de la confirmación plausible del diálogo
- Comprobando si NumberOfProcessors es < 2 a través del acceso PEB
- Comprobando si NumberOfProcessors es < 2 a través de GetSystemInfo()
- Comprobación del tiempo de actividad del sistema operativo mediante GetTickCount()
- Puerto scsi->bus->id de destino->id de unidad lógica-> 0 identificador
- Clave de registro (HKLM\HARDWARE\Description\System "SystemBiosVersion")
- Clave de registro (HKLM\HARDWARE\Description\System "VideoBiosVersion")
- Clave de registro Clave de registro (HKLM\HARDWARE\ACPI\DSDT\VBOX)
- Clave de registro Clave de registro (HKLM\HARDWARE\ACPI\FADT\VBOX)

- Clave de registro Clave de registro (HKLM\HARDWARE\ACPI\RSDT\VBOX)
- Clave de registro (HKLM\HARDWARE\DESCRIPTION\System "SystemBiosDate")
- Buscando una dirección MAC que comience con 08:00:27
- Buscando dispositivos VBox usando WMI

Configuración de la red

La red del laboratorio estará estructurada de acuerdo con el siguiente diagrama:

Figura 13: Diagrama de red del laboratorio virtual

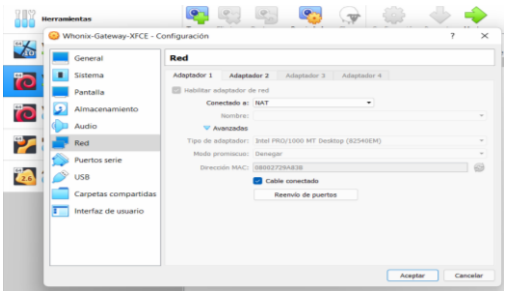


Fuente: Elaboración propia

Se procedió a configurar la red de la máquina virtual Whonix-Gateway con los siguientes pasos:

- Ir a la configuración de red de la máquina virtual
- El adaptador 1 es definido como WAN

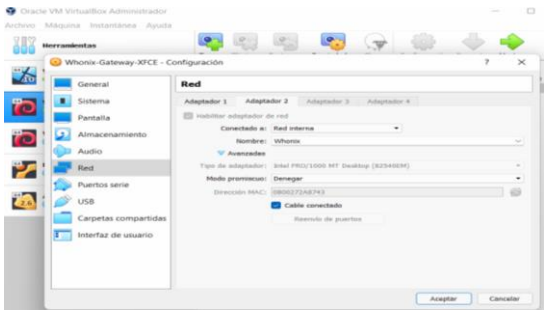
Figura 14: Configuración de red de la máquina virtual de Whonix



Fuente: Elaboración propia

- El adaptador 2 es definido como LAN. Configurar el adaptador 2 como red interna. El cual será el Gateway de las máquinas víctimas

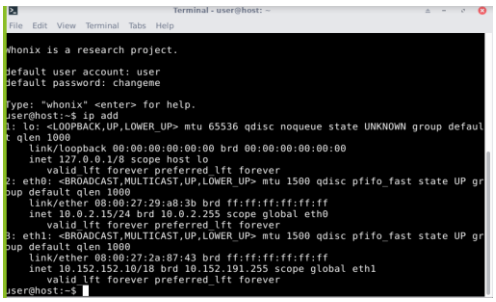
Figura 15: Configuración del adaptador de Whonix



Fuente: Elaboración propia

- Terminada la configuración de la red de la máquina, se procederá a inicar la Whonix
- Ir al terminal para verificar las IP haciendo uso del comando "ip add". Estas IPs definirán el rango en las máquinas víctimas y la LAN servirá como Gateway para las mismas.

Figura 16: Verificación de IP en Whonix

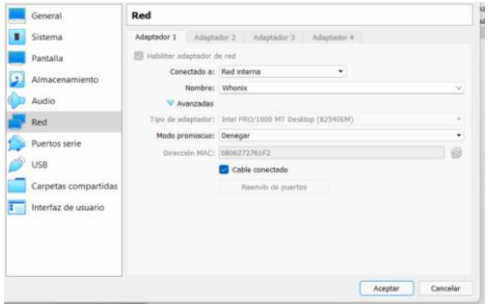


Fuente: Elaboración propia

Finalizado todos los pasos anteriormente descritos, se procedió a configurar las máquinas víctimas. A continuación, se muestra la configuración general de los escenarios:

- Configurar en todas las máquinas la red, estableciendo el Adaptador 1 como Red Interna

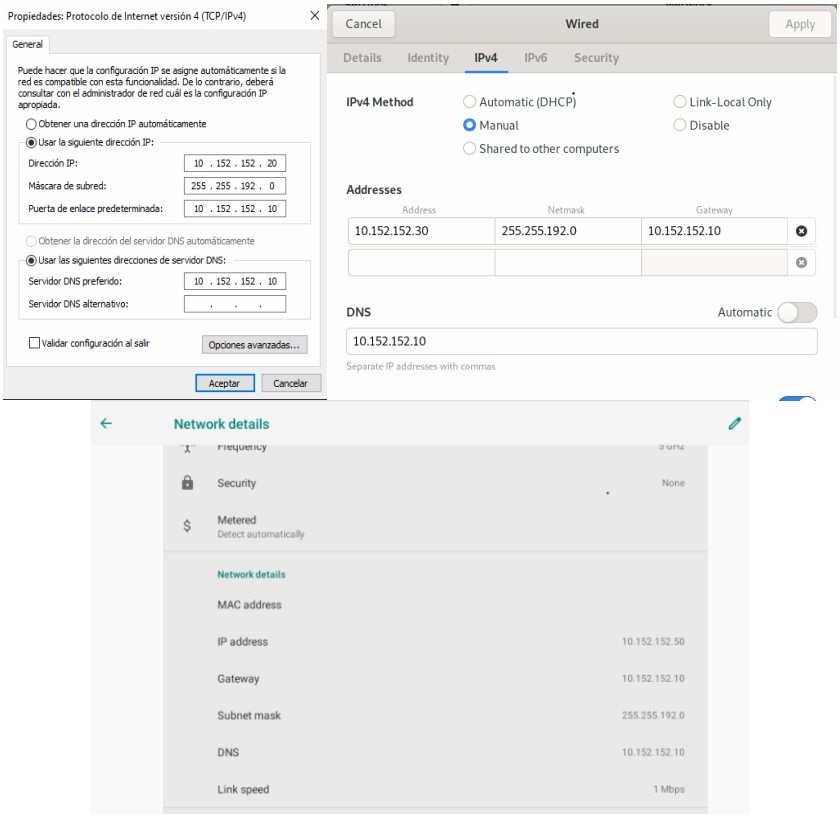
Figura 17: Configuración del adaptador en las máquinas virtuales



Fuente: Elaboración propia

- Terminada la configuración en la red, iniciar la máquina.
- Configurar una IP estática, siguiendo el rango establecido por Whonix. La puerta de enlace y el DNS serán la LAN de Whonix con el fin de que exista conexión entre las dos máquinas para que tenga acceso a internet a través de Tor.

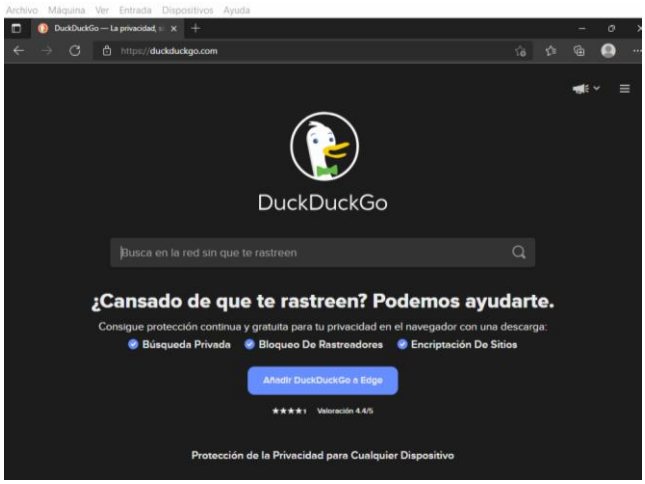
Figura 18: Configuración de IP, Gateway y DNS



Fuente: Elaboración propia

- Abrir el navegador y verificar la conexión a internet

Figura 19: Red conectada a Internet por medio de Tor



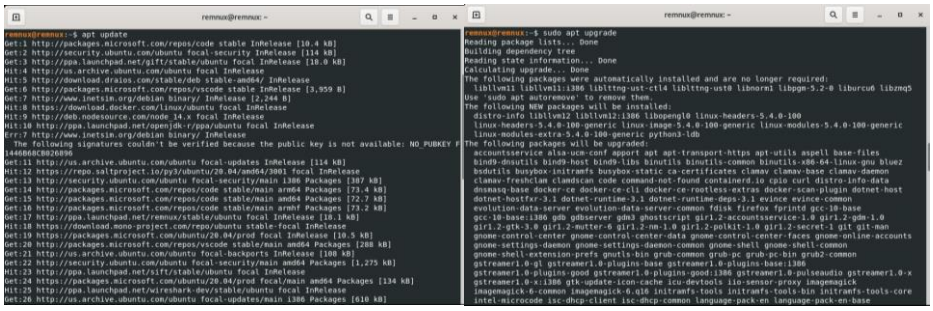
Fuente: Elaboración propia

Configuración de REMnux como servicio

La configuración de REMnux como servicio se realizó instalando la herramienta InetSim, la cual simula entornos con servicios DNS, HTTP, etc. A continuación, se muestran los pasos a seguir para instalar esta herramienta:

- Actualizar REMnux con los comandos “sudo apt update” y “sudo apt upgrade”

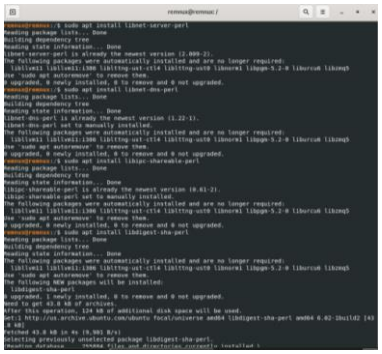
Figura 20: Actualización de REMnux



Fuente: Elaboración propia

- Instalar los paquetes libnet-servidor-perl, libnet-dns-perl, libipc-shareable-perl, libdigest-sha-perl

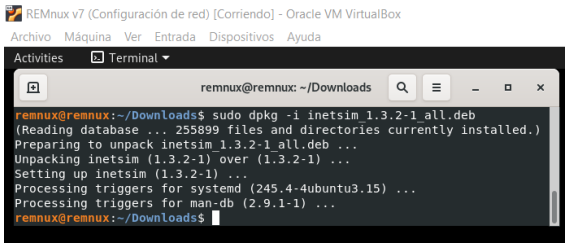
Figura 21: Instalación de paquetes



Fuente: Elaboración propia

- Ir a la página oficial de InetSim “https://www.inetsim.org/packages.html” y descargar el paquete InetSim
- Instalar el paquete descargado con el comando “sudo dpkg -i inetsim_1.3.2-1_all.deb”

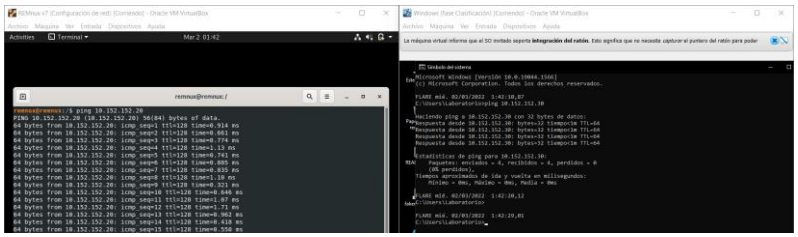
Figura 22: Instalación del paquete InetSim



Fuente: Elaboración propia

- Finalmente verificar la conexión entre la máquina Windows y la máquina Linux.

Figura 23: Comunicación entre las dos máquinas virtuales



Fuente: Elaboración propia

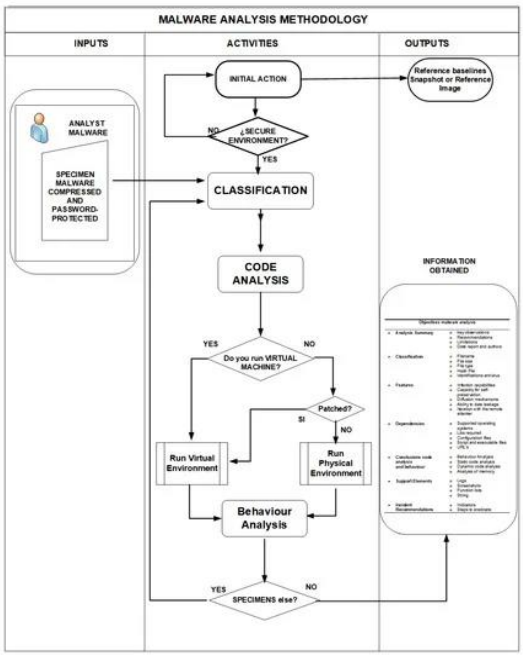
4.4.2. Análisis de malware aplicando la metodología SAMA

Para realizar un correcto análisis de los malwares encontrados en la red se deberá establecer la metodología a implantar en el laboratorio que se adapte al contexto (herramientas, complejidad del análisis, etc.).

La metodología seleccionada fue la establecida por el Doctor Javier Bermejo en su investigación denominada “Systematic Approach to Malware Analysis (SAMA)” [8].

Tal como lo muestra el diagrama en la Figura 1 esta metodología proporciona una guía eficaz en todo el ciclo del análisis del malware, sin importar la complejidad del mismo. El diagrama de dicha metodología se establece de la siguiente manera:

Figura 24: Diagrama de la metodología [8]



Fuente: Obtenido desde el artículo Systematic Approach to Malware Analysis [8]

Las fases aplicadas en la metodología para el desarrollo del análisis de malware en el laboratorio virtual se detallan a continuación:

Fase 1: Acciones iniciales.

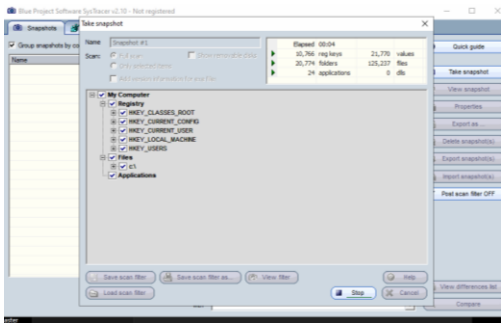
En esta fase se realiza una serie de acciones principales que nos faciliten un registro de las configuraciones del estado inicial de las máquinas involucradas en el análisis. Este registro nos permitirá hacer una comparación del antes y después de la infección [8].

Para llevar a cabo esta fase el doctor Bermejo describe los siguientes pasos [8]:

- Preservación de la integridad

En el laboratorio de análisis se utilizó la herramienta Systracer que permitió realizar un escaneo del sistema y a su vez obtener un snapshot mediante un archivo de imagen binaria. Esta instantánea del sistema servirá para comparar las modificaciones realizadas después de la infección y determinar las alteraciones en los archivos o entradas de registro.

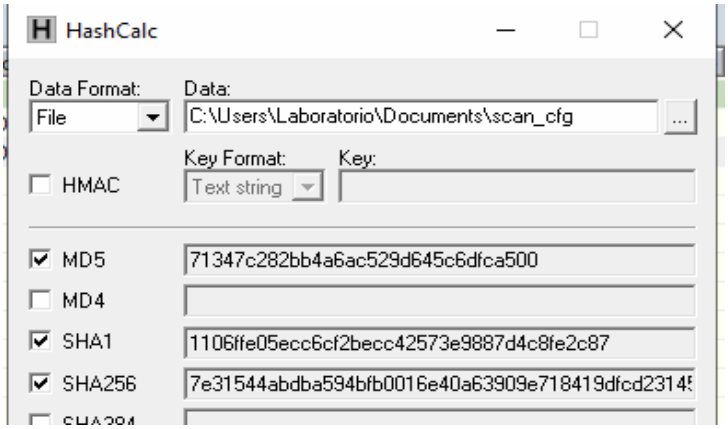
Figura 25: Snapshot en la herramienta Systracer



Fuente: Elaboración propia

Al finalizar el escaneo, se procedió a obtener el hash del sistema y del archivo generado por la herramienta Systracer utilizando MD5Summer y HashCalc.

Figura 26: Calcular hash del archivo generado por Systracer

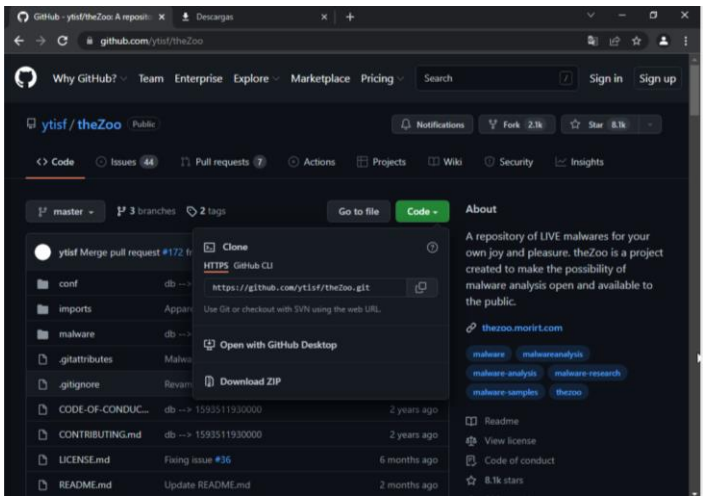


Fuente: Elaboración propia

- Garantía de la integridad de las muestras generadas.

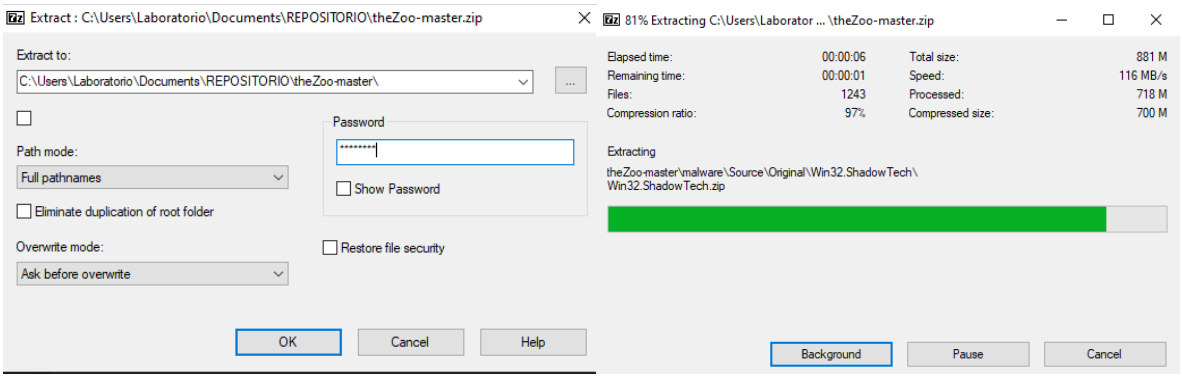
Las muestras de malware fueron obtenidas en el repositorio theZoo a través de github. Se procedió a comparar el hash de los malwares para verificar su integridad.

Figura 27: Repositorio theZoo



Fuente: Elaboración propia

Figura 28: Extracción del repositorio



Fuente: Elaboración propia

Fase 2: Clasificación

En esta fase se requiere examinar el ejecutable del programa malicioso, sin acceder al código. Al comprobar que se trata de un malware se procederá a obtener información de este a través de fuentes abiertas [8].

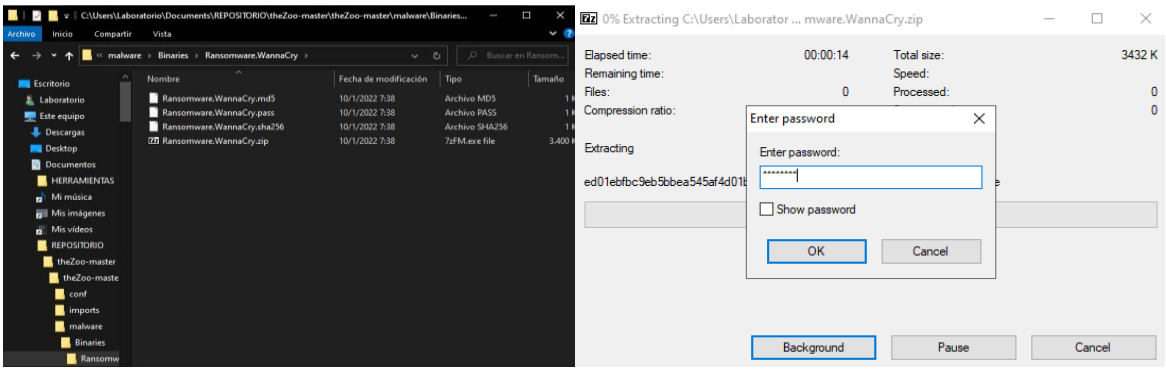
Para llevar a cabo esta fase el Doctor Bermejo enlista las tareas que se deben realizar [8]:

- Transferir el malware.

Se extrajeron de forma segura las muestras de malware, debido a que, al descomprimir el repositorio, éste solicitó una contraseña por defecto que era “infected”.

Para la presente investigación se eligió el malware “WannaCry”.

Figura 29: Extracción de la muestra WannaCry

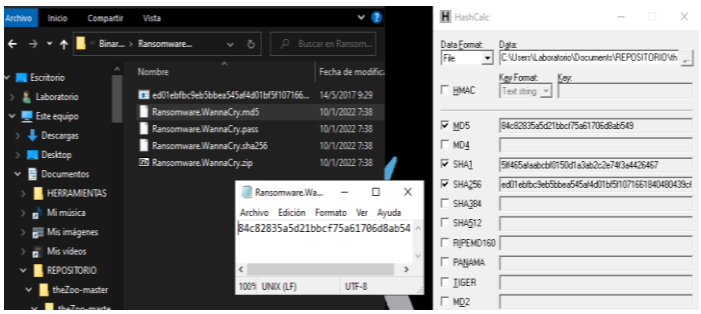


Fuente: Elaboración propia

- Identificar malware.

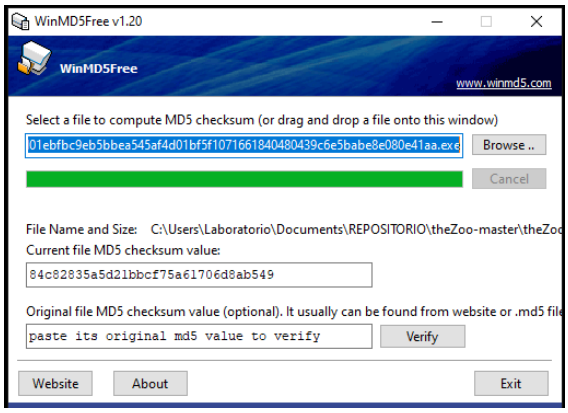
Para realizar este apartado, se eligieron las herramientas WinMD5 y HashCalc. Estas herramientas permitieron calcular el hash del malware seleccionado y el tamaño del archivo.

Figura 30: Calculo de la muestra con HashCalc



Fuente: Elaboración propia

Figura 31: Calculo de la muestra con WinMD5



Fuente: Elaboración propia

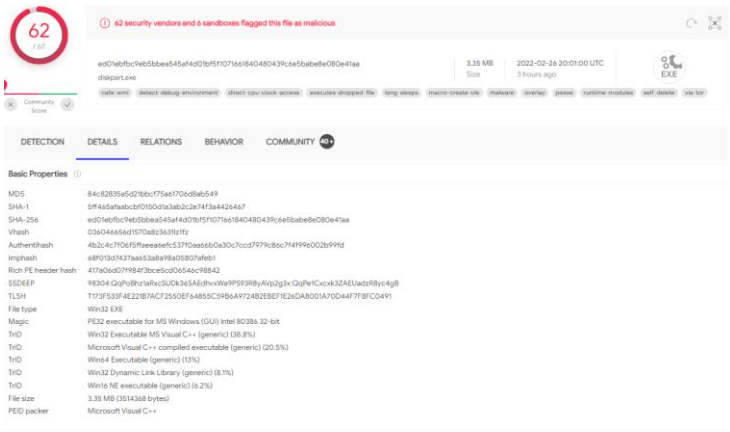
- Compruebe la familia de malware.

A través de antivirus ClamWin se pudo detectar la familia del malware estudiado.

- Encontrar información sobre el malware en código abierto.

Con la información obtenida en los apartados anteriores, se pudo realizar la búsqueda de información del malware en fuentes abiertas.

Figura 32: Información obtenida en VirusTotal

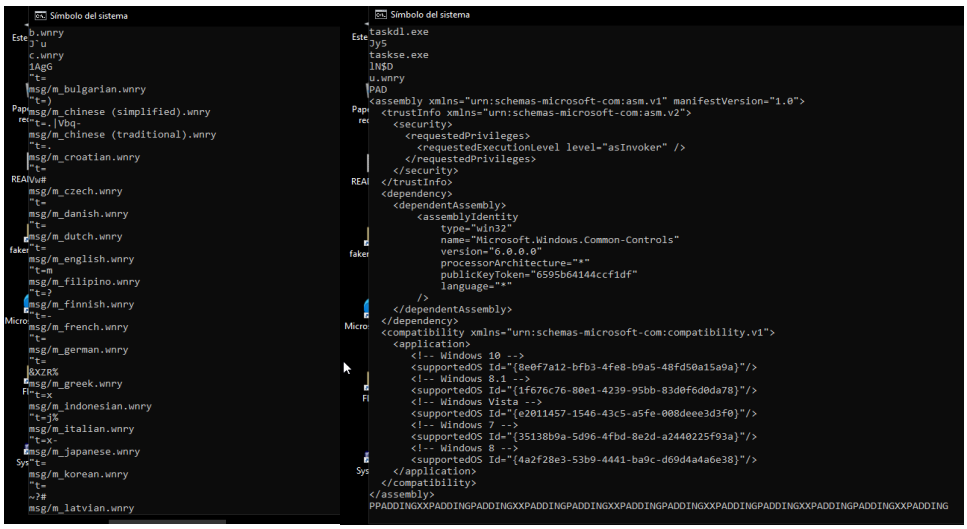


Fuente: Elaboración propia

- Análisis de cadenas de malware binario.

Para realizar el análisis de cadena se utilizó la herramienta String, con el comando “string -a nombreMalware”. Esto permitió realizar un escaneo total del espécimen y extraer la cadena de texto del mismo.

Figura 33: Resultados de la herramienta String



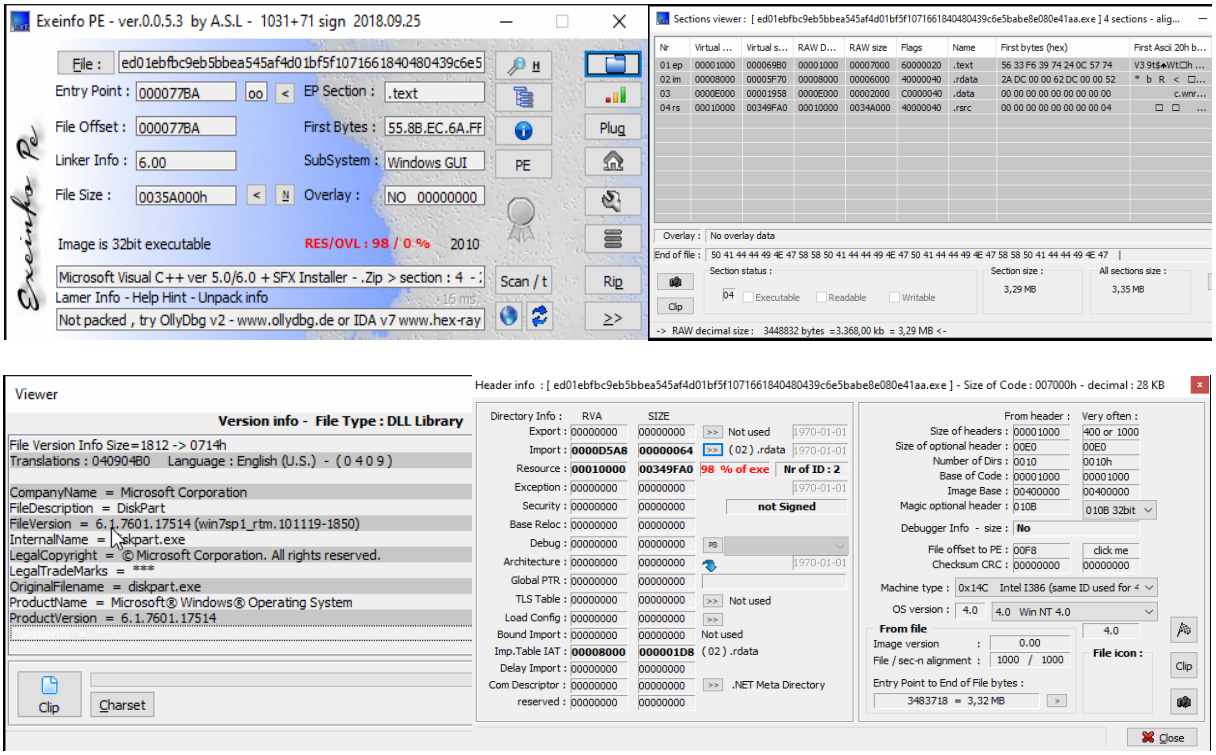
Fuente: Elaboración propia

- Analizar las técnicas utilizadas en la ofuscación de malware.

El análisis que se llevó a cabo en esta fase permitió identificar las herramientas que se utilizaron en las técnicas de ofuscación del malware. Se hizo uso de varias herramientas para comparar los datos, porque además de obtener la información antes mencionada, estas herramientas permiten conocer la estructura del binario.

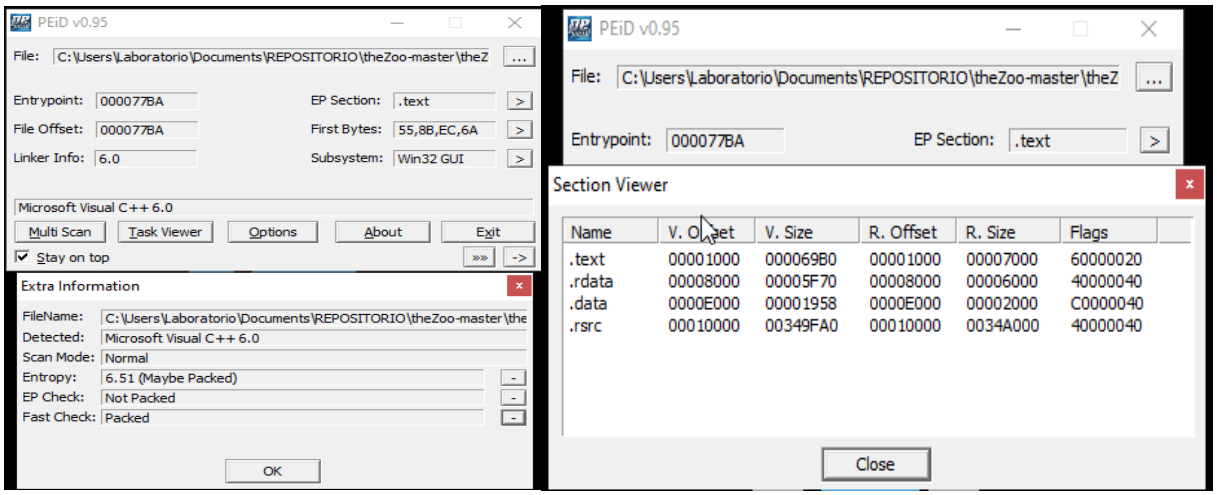
Las herramientas utilizadas fueron Exeinfo PE, PEiD y WinHex

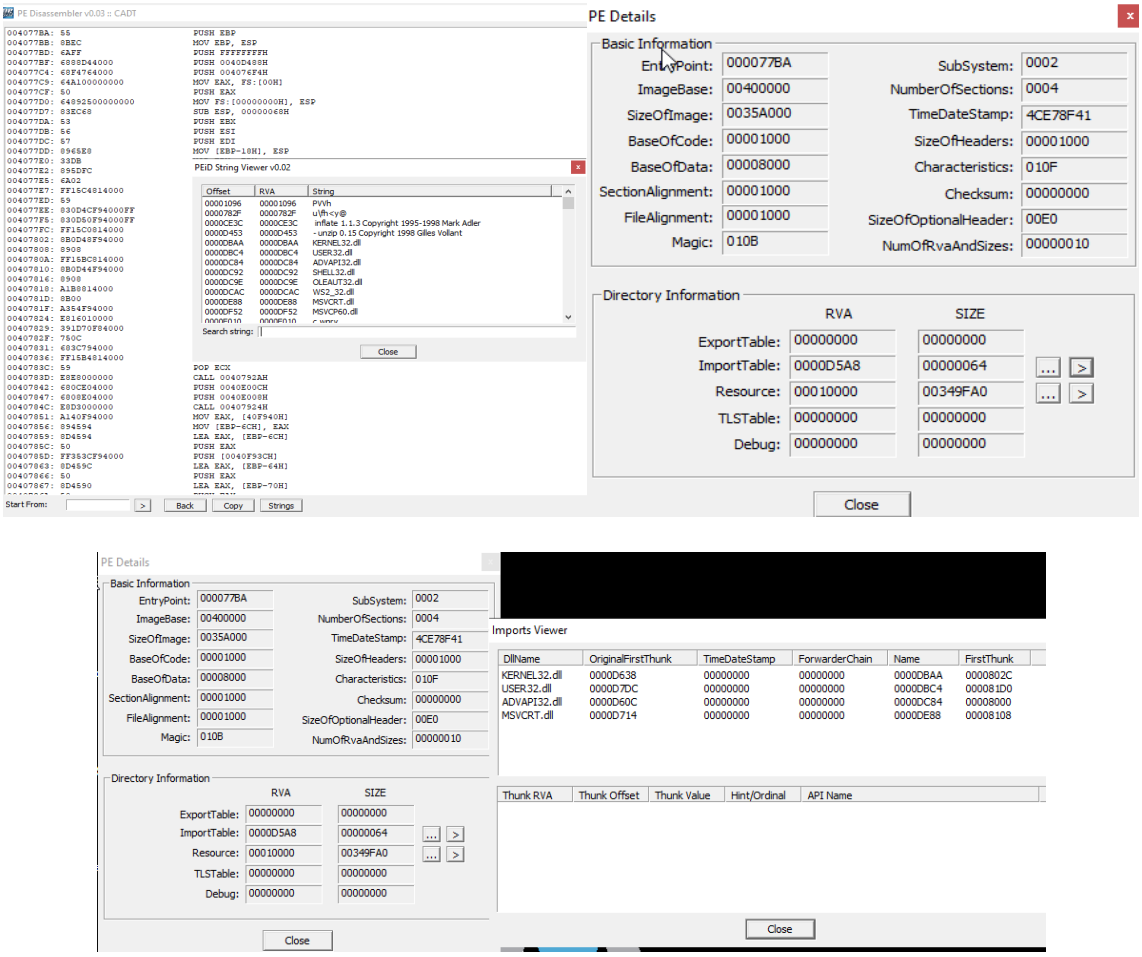
Figura 34: Resultados del análisis en la herramienta Exeinfo PE



Fuente: Elaboración propia

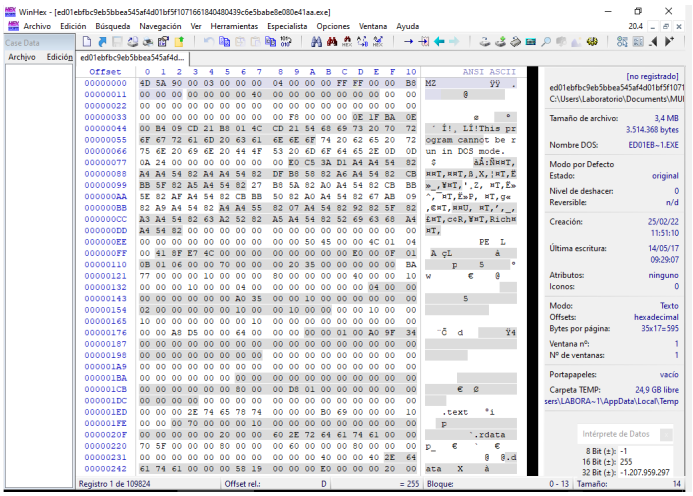
Figura 35: Resultados del análisis en la herramienta PEiD





Fuente: Elaboración propia

Figura 36: Resultados del análisis en la herramienta WinHex



Fuente: Elaboración propia

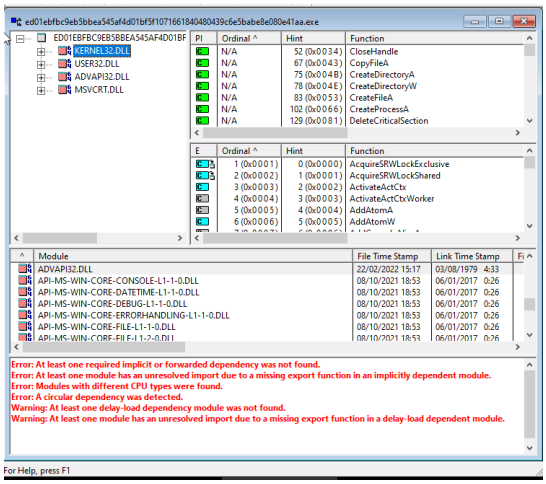
- Analizar formato de archivo.

En este apartado se obtuvo información sobre el formato y estructura del archivo.

Con el programa Dependency Walker se identificaron las librerías y módulos del espécimen.

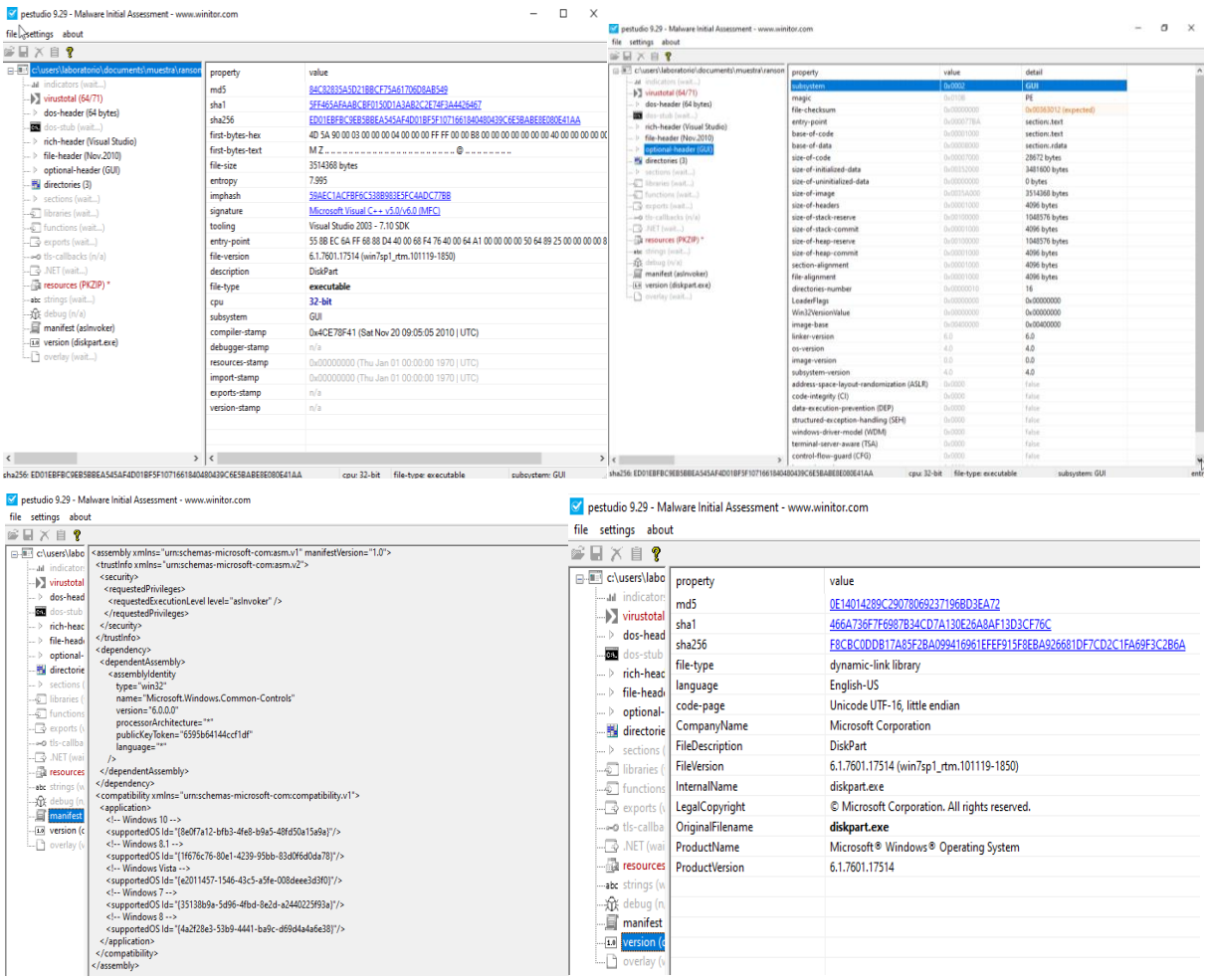
Finalmente, para tener una información global del análisis se utilizó el programa PEStudio.

Figura 37: Resultados del análisis en la herramienta Dependency Walker



Fuente: Elaboración propia

Figura 38: Resultados del análisis en la herramienta Pestudio



Fuente: Elaboración propia

Resultados

En esta fase se obtuvo información que permite clasificar el malware y conocer su estructura.

A continuación, se muestran los datos relevantes obtenidos en este apartado:

Tabla 8: Resultados de la fase de clasificación

Hash de archivo binario	84c82835a5d21bbcf75a61706d8ab549 5ff465afaabcbf0150d1a3ab2c2e74f3a4426467 ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa	
Hash MD5 PE secciones	db349b97c37d22f5ea1d1841e3c89eb4	
Firmas hexadecimales	FEEF04BDh	
Firmas de metadatos de archivos	Microsoft Visual C++ v5.0/v6.0 (MFC)	
Patrones de firma	Ransom_WANA.A Ransom_WCRY.B Ransom_WCRY.C Ransom_WCRY.H Ransom_WCRY.I Ransom_WCRY.J Ransom_WCRY.K Ransom_WCRY.L	Ransom_WCRY.DAM Ransom_WCRY.F117D7 Ransom_WCRY.F117DB Ransom_WCRY.F117E8 Ransom_WCRY.SM Ransom_WCRY.SM1 Ransom_WCRY.SMB WORM_WCRY.A
Cadenas de metadatos de archivos PE	msg b.wnry c.wnry r.wnry	s.wnry t.wnry taskdl.exe taskse.exe u.wnry
Familias de malware	Crypto ransomware worm	
Cadenas de texto	.text .rdata .data .rsrc	
Protocolo	SMB	
Direcciones IP	109.105.109.162 127.0.0.1 128.31.0.39 131.188.40.189 137.74.19.201	142.93.208.127 146.0.32.144 149.154.154.155 154.35.175.225 163.172.131.88
Clave	WNcry@2ol7 - WNcry@123	
Archivo	Ejecutable	
Dirección base del archivo	00008000h	
Dirección del punto de entrada	004077BAh	
Número de secciones en la tabla de secciones	0004h	
Tipo de aplicación	Intel I386	
Bibliotecas requeridas	kernel32.dll advapi32.dll user32.dll msvcrt.dll	
Requisitos de espacio	3,32 MB	

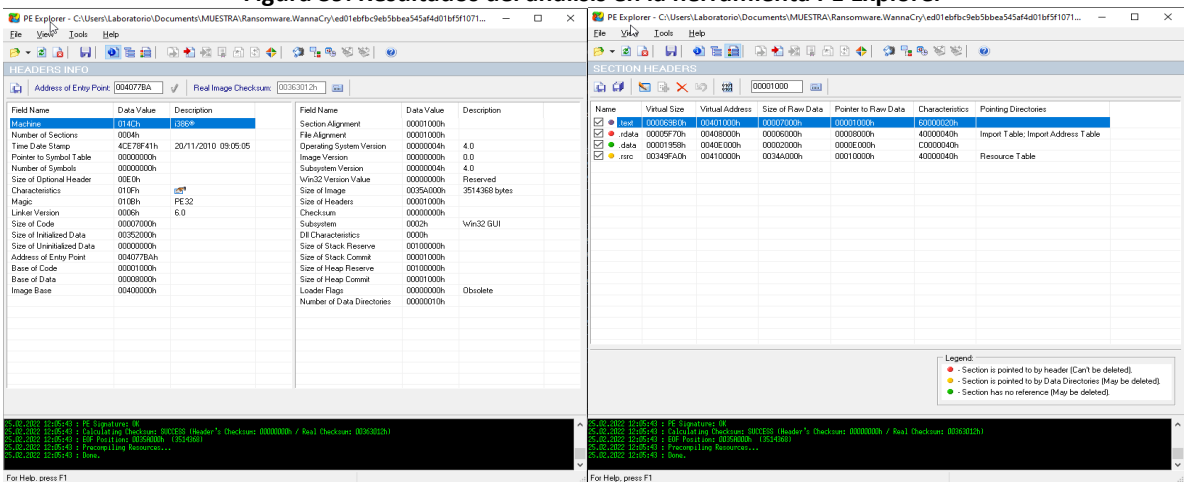
Fase 3: Análisis de código

En esta fase se lleva a cabo el análisis estático y dinámico en lenguaje ensamblador, el cual se realizará con mayor profundidad de navegación del malware para conocer de manera más clara su funcionalidad.

- Operación General

Para conocer los datos del encabezado del malware se utilizó el software PE Explorer, que además proporciona información de los recursos contenidos en el archivo PE como las importaciones y dependencias.

Figura 39: Resultados del análisis en la herramienta PE Explorer



Fuente: Elaboración propia

- Análisis estático

De acuerdo con la metodología estudiada, en este apartado es fundamental encontrar los siguientes datos del malware: información para ejecutar el malware, información de instalación, comandos de ejecución, contraseñas, comandos de mando y control, canal IRC y contraseña de conexión, información para evitar restricciones operativas en entornos virtuales, funcionalidades ocultas y compresión del funcionamiento del malware.

Para la ejecución de esta fase se utilizaron las herramientas IDA Pro, Resource Hacker, HxD, PEview.

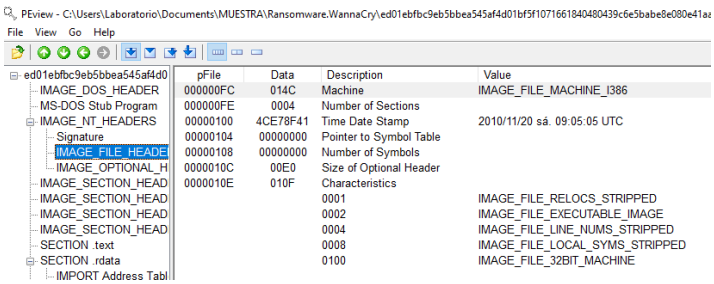
Resultados

La información obtenida en este análisis complementó los datos encontrados en la fase de clasificación como las secciones de la cabecera del malware Ver Tabla 9. Así mismo la creación de este.

Tabla 9: Secciones del Header

Nombre	Características	Directorios
.text	60000020h	
.rdata	40000040h	Tabla de Importaciones: Import Address Table
.data	C0000040h	
.rsrc	40000040h	Tabla de Recursos

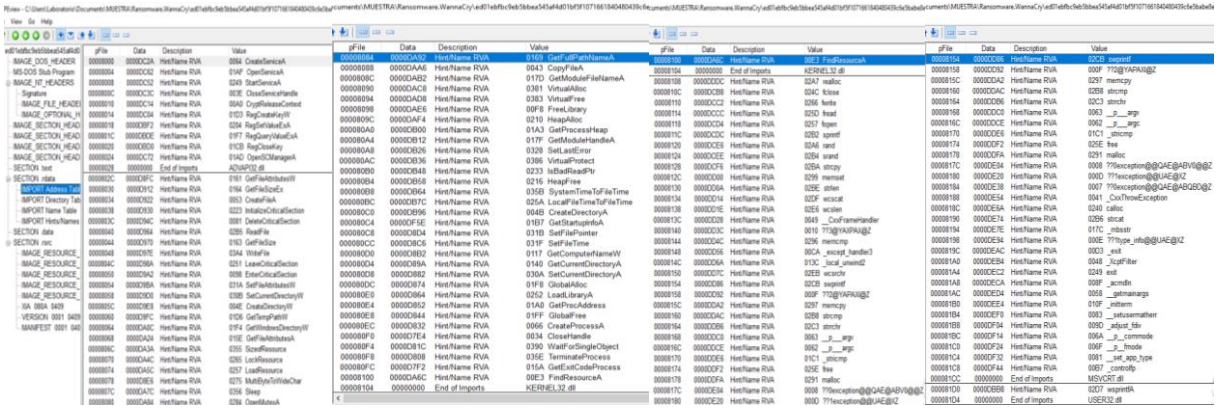
Figura 40: Fecha de creación del archivo. Herramienta PEview



Fuente: Elaboración propia

Además, se visualizó las importaciones realizadas por este malware Ver Figura 37.

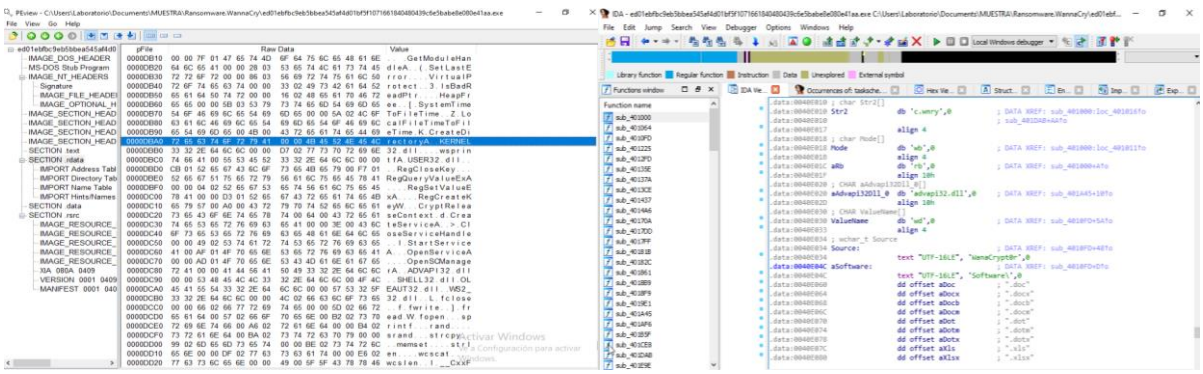
Figura 41: Tabla de direcciones de importaciones. Herramienta PEview



Fuente: Elaboración propia

Este espécimen crea entradas en el registro, a través de la librería ADVAPI32 y la función RegCreateKeyW.

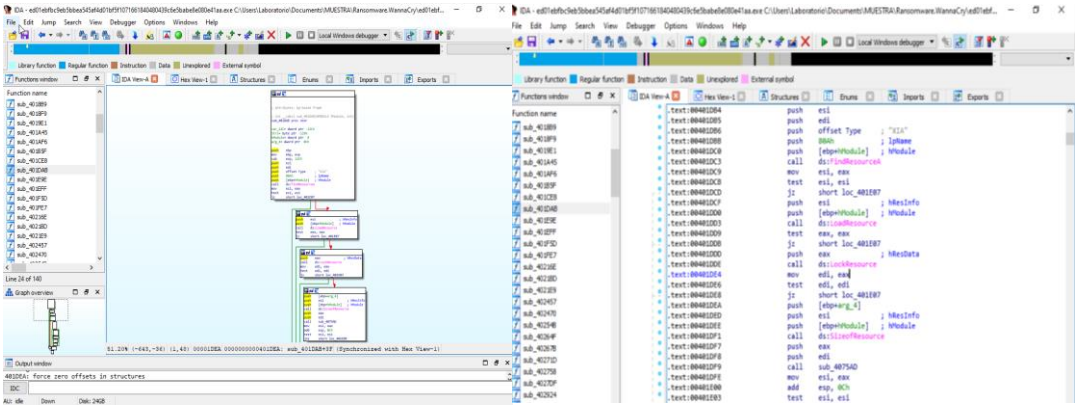
Figura 42: Librería ADVAPI32 y función RegCreateKeyW. Herramientas PEview e IDAPro



Fuente: Elaboración propia

En la búsqueda se evidenció el uso de FindResource, LoadResource, LockResource y SizeofResource, que implica cargar la sección de recursos y acceder a los datos que contiene.

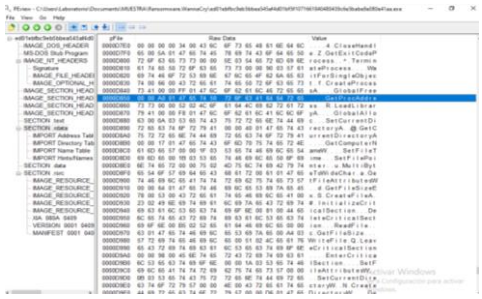
Figura 43: Funciones FindResource, LoadResource, LockResource y SizeofResource. Herramienta IDA



Fuente: Elaboración propia

Utiliza GetProcAddress para completar algunas variables con la dirección de API específicas, para que pueda llamarlas en las siguientes líneas de código.

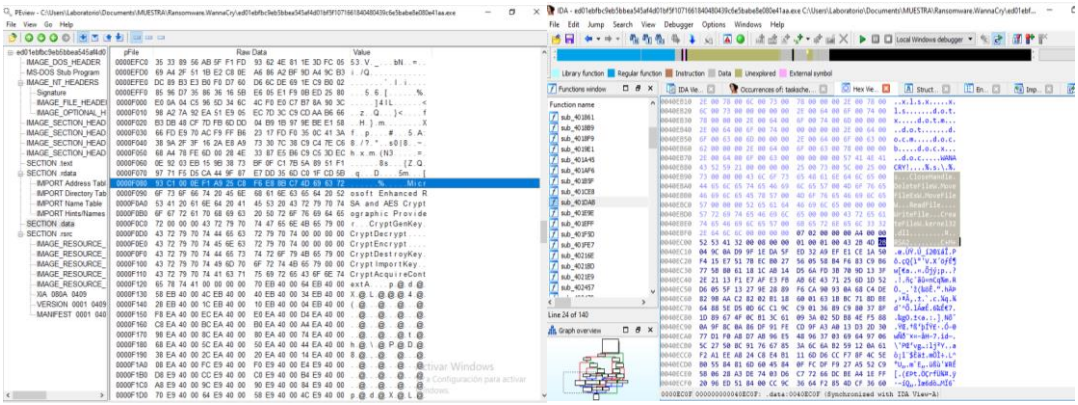
Figura 44: Función GetProcAddress. Herramienta PEview



Fuente: Elaboración propia

Utiliza cmd.exe para ejecutar procesos, así mismo crear o utilizar un proceso tasksche.exe para ganar persistencia o ejecutar alguna tarea posterior. Los algoritmos de encriptación que utiliza este malware son el RSA y el AES. Así mismo, enlista las extensiones de los archivos a encriptar.

Figura 45: Algoritmos de Encriptación: Herramientas PEview e IDA

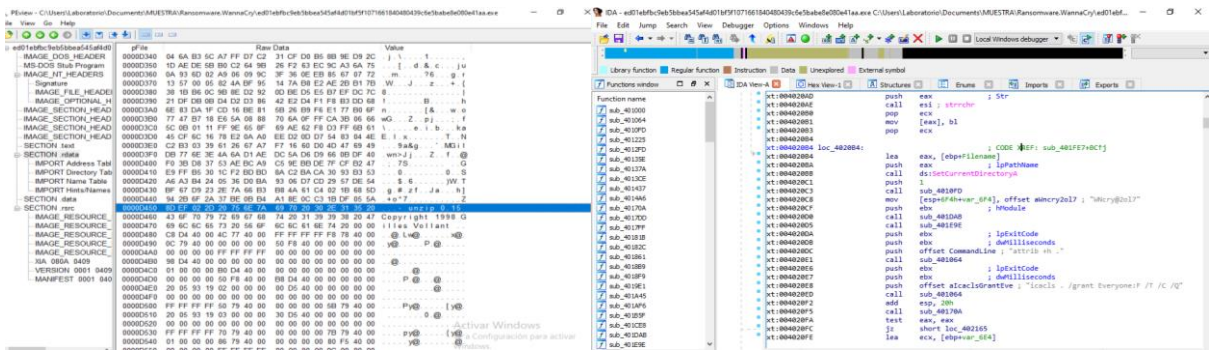


Fuente: Elaboración propia

Kernel32.dll y msvcrt.dll son las dos bibliotecas más utilizadas por el componente de cifrado. Esto puede indicar que estas dos bibliotecas implementaron la funcionalidad de cifrado principal de WannaCry.

La biblioteca Crypto API se utiliza para generar y administrar claves criptográficas simétricas y asimétricas aleatorias. Además, se encontró la contraseña “WNcry@2ol7” perteneciente a un archivo ZIP. El contenido se extrae en el directorio donde se encuentra tasksche.exe.

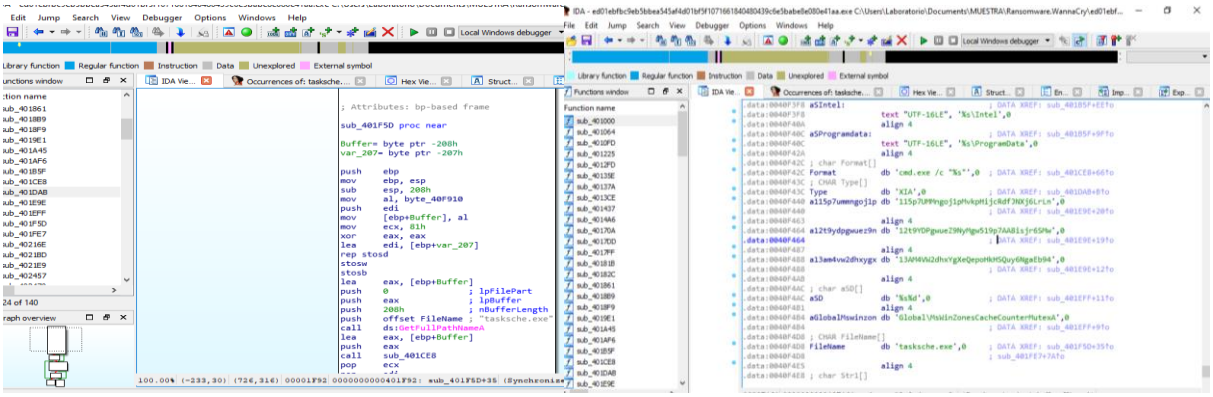
Figura 46: Contraseña del ZIP. Herramienta PEview e IDA



Fuente: Elaboración propia

En taskche.exe se cargan las direcciones del bitcoin wallet Ver Figura 43.

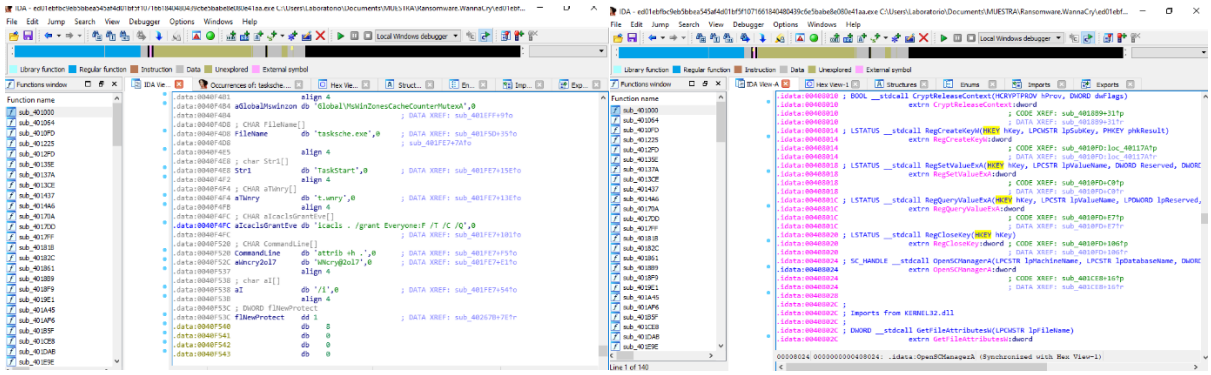
Figura 47: Direcciones del bitcoin. Herramienta IDAPro



Fuente: Elaboración propia

La función TaskStart realiza las siguientes acciones a través de hilos: recuperar la clave pública utilizada para cifrar archivos en el sistema, Crea un mutex global llamado "MsWinZonesCacheCounterMutex", cifra archivos en el sistema, cambia el nombre de las extensiones de archivo a .WNCRY e inicia varios procesos: taskdl.exe, taske.exe y @WanaDecryptor@.exe

Figura 48: Mutex. Herramienta IDAPro



Fuente: Elaboración propia

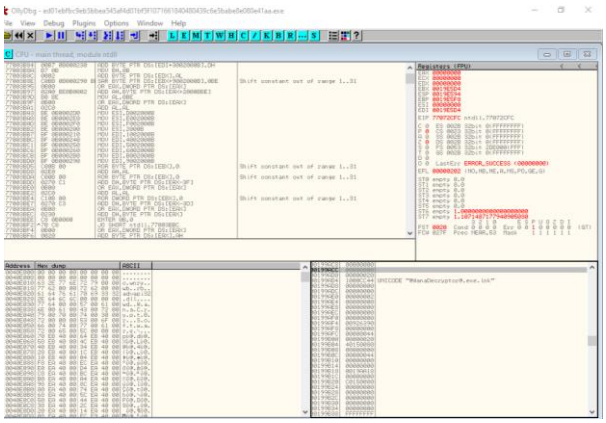
- Análisis dinámico

Para llevar a cabo este análisis, se usó la herramienta OllyDBG y se obtuvieron los siguientes resultados:

Al ejecutar el código, éste se instala como un servicio y extrae el recurso R codificado para copiarlo en C:\Windows\taskche.exe. El recurso R representa el binario del componente de cifrado WannaCry.

El cifrador WannaCry inicia el binario integrado del descifrador “@WanaDecryptor@.exe”.

Figura 49: Binario del descifrador



Fuente: Elaboración propia

Durante su ejecución, el componente de cifrado comprueba si existe el objeto de exclusión mutua (mutex) “GlobalnMsWinZonesCacheCounterMutexA”. La presencia de este mutex indica que una muestra de WannaCry ya está activa en el sistema.

Si el mutex está presente en el sistema, el componente de cifrado se detiene automáticamente sin realizar más acciones. De lo contrario, se inicia el proceso de cifrado.

Para cifrar cada archivo, se genera una clave AES simétrica. Luego, cada clave AES generada se cifra con la clave RSA pública y se almacena dentro del encabezado del archivo que comienza con "WANACRY!" valor de cadena. Los archivos cifrados se renombran y se agregan con la extensión de archivo ".WNCRY".

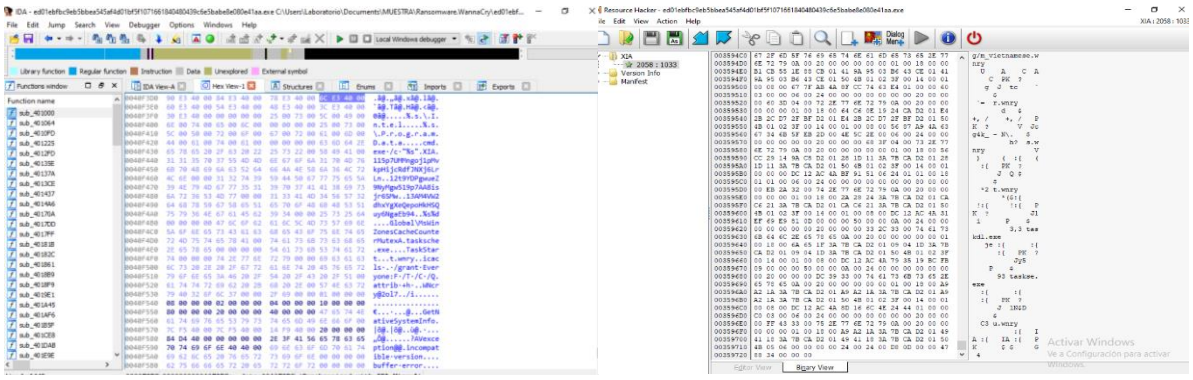
El componente de cifrado contiene un archivo ZIP protegido con contraseña. En el anterior análisis se pudo detectar que la contraseña de este archivo es "WNcry@2ol7". El contenido se extrae en el directorio donde se encuentra tasksche.exe .

El archivo ZIP contiene lo siguiente:

Tabla 10: Componentes del archivo ZIP

Componente	Descripción
.msg	Es una carpeta que contiene una lista de archivos de formato de texto enriquecido (RTF) con extensión wnry. Estos archivos son las instrucciones "Léame" utilizadas para mostrar el mensaje de extorsión a la víctima en diferentes idiomas, en función de la información obtenida del sistema por las funciones maliciosas de WannaCry.
b.wnry	Es un archivo de imagen utilizado para mostrar instrucciones para el descifrado de archivos de usuario.
c.wnry	Contiene una lista de direcciones Tor con extensión .onion y un enlace a un archivo de instalación comprimido del navegador Tor de Tor Project.
r.wnry	Es un archivo de texto en inglés con instrucciones de descifrado adicionales para ser utilizado por el componente de descifrado.
s.wnry	Es un archivo ZIP que contiene el ejecutable del software Tor.
t.wnry	Es un archivo encriptado con "WANACRY!" formato de cifrado.
Taskdl.exe	Es una herramienta de apoyo para la eliminación de archivos con extensión .WNCRY.
Taske.exe	Es una herramienta de apoyo para la ejecución de malware en sesiones de protocolo de escritorio remoto (RDP).
u.wnry	Es un archivo ejecutable con el nombre "@WanaDecryptor@.exe", que representa el componente de descifrado de WannaCry

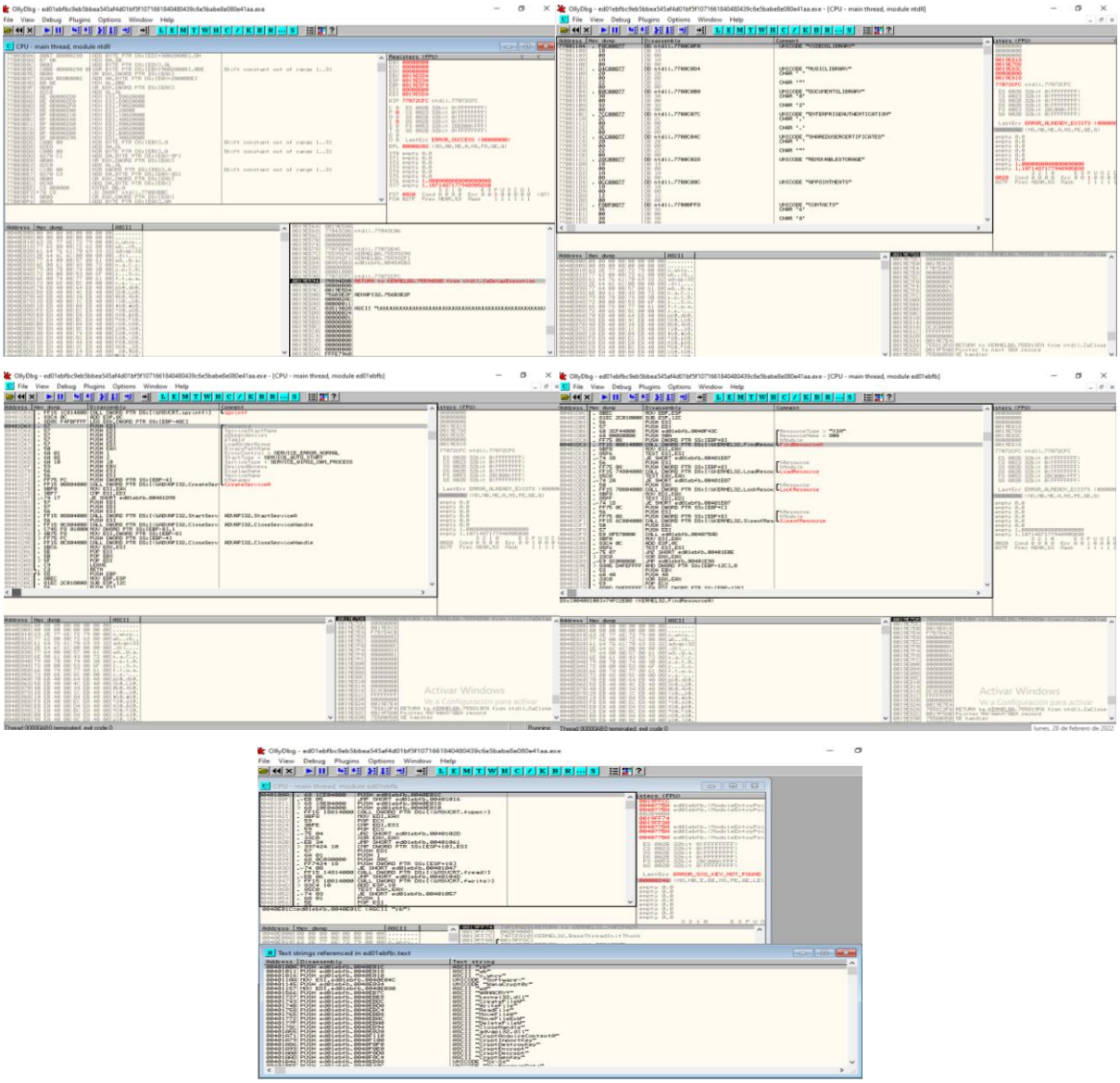
Figura 50: Componentes



Fuente: Elaboración propia

El malware encripta archivos en el sistema, buscando extensiones de archivo, que están codificadas en el binario, tales como docx, jpeg, mp4, avi, gif, etc.

Figura 51: Resultados del análisis en la herramienta IDAPro



Fuente: Elaboración propia

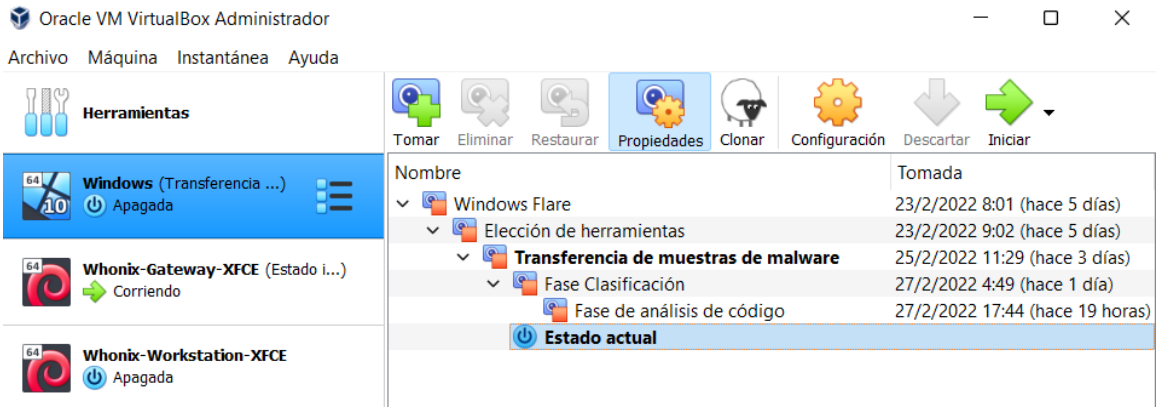
Fase 4: Análisis de comportamiento

En esta fase se pretende implantar los malwares en un laboratorio que simula un entorno real, para verificar su comportamiento en el mismo, el tráfico de red que genera y las acciones que realiza sobre el sistema objetivo [8].

- Tareas previas a la ejecución del malware

Para iniciar esta fase, se regresó al punto de partida “Transferencia de muestras de malware”, el cual es un estado previo a la ejecución de la muestra.

Figura 52: Retorno a la instantánea en un estado limpio



Fuente: Elaboración propia

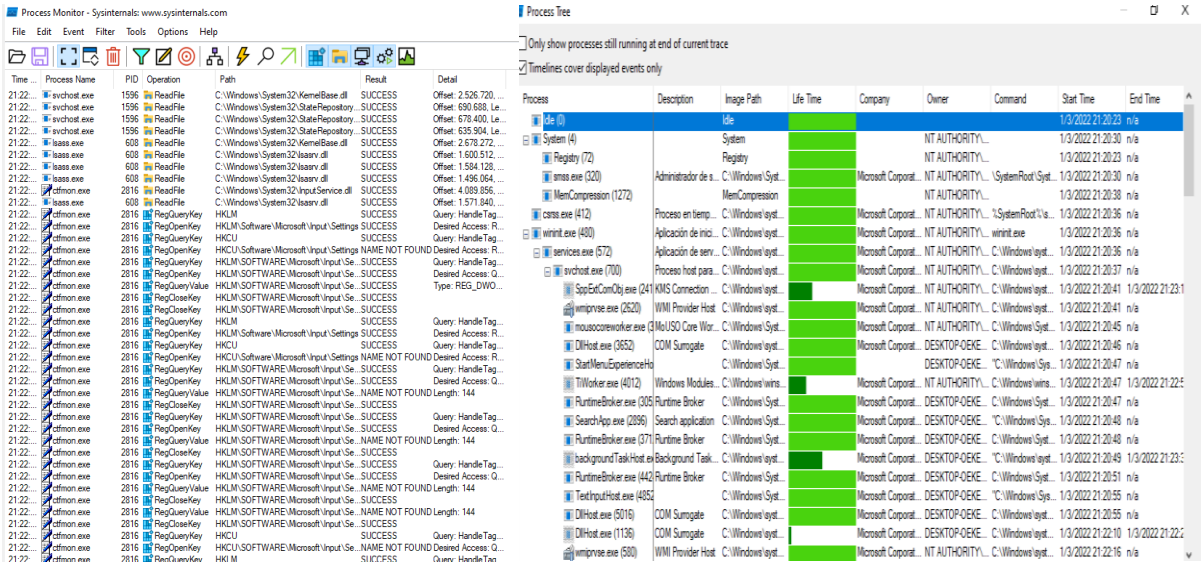
Es insoslayable tener una idea clara del sistema en general, por esta razón se debe realizar un monitoreo previo del sistema de archivos, procesos, el tráfico de la red, y los cambios en el registro. En este apartado se seleccionaron varias herramientas que cumplen cada una de estas funciones Ver Tabla 11.

Tabla 11: Herramientas para el monitoreo del sistema

	MONITOREO			
Nombre	**Sistema de archivos**	**Procesos**	**Registro**	**Tráfico de red**
Process Monitor	X	X	X	
Process Hacker		X		X
Autoruns	X	X	X	X
Wireshark				X

Antes de ejecutar el malware, se obtiene un estado inicial del sistema previo a la infección, por lo tanto, se utilizó la herramienta process monitor para conocer los procesos propios del computador en un funcionamiento normal como se aprecia en la Figura 53.

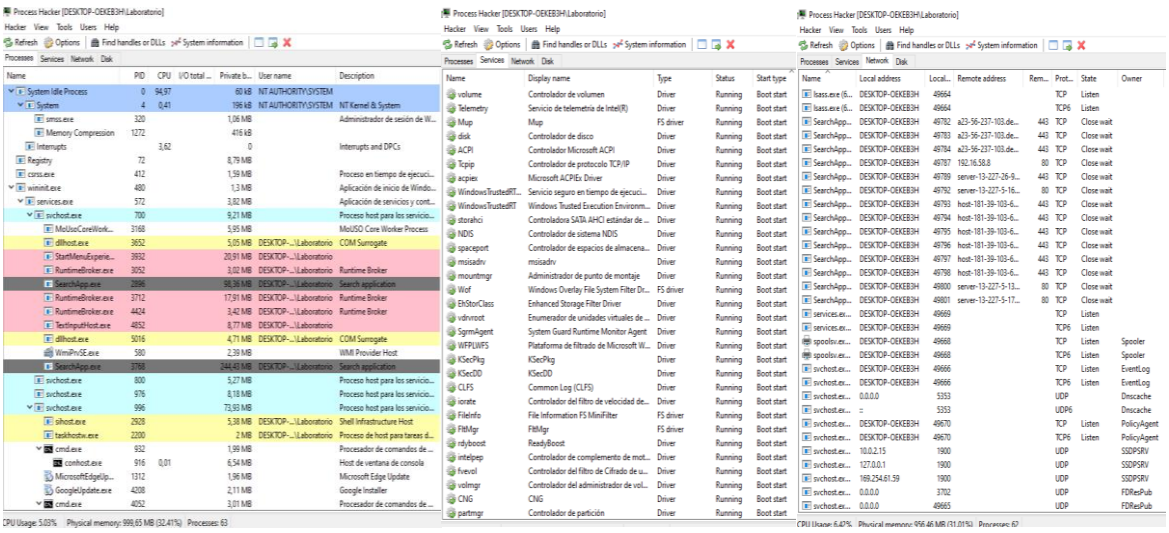
Figura 53: Resultados de la herramienta Process Monitor



Fuente: Elaboración propia

Para contrastar la información obtenida con la herramienta Process Monitor se utilizó Process Hacker que permitió obtener los procesos del sistema en esquema de árbol, donde se indican los procesos principales y secundarios. Así mismo, se obtuvo el comportamiento de la red, puertos y protocolos del computador.

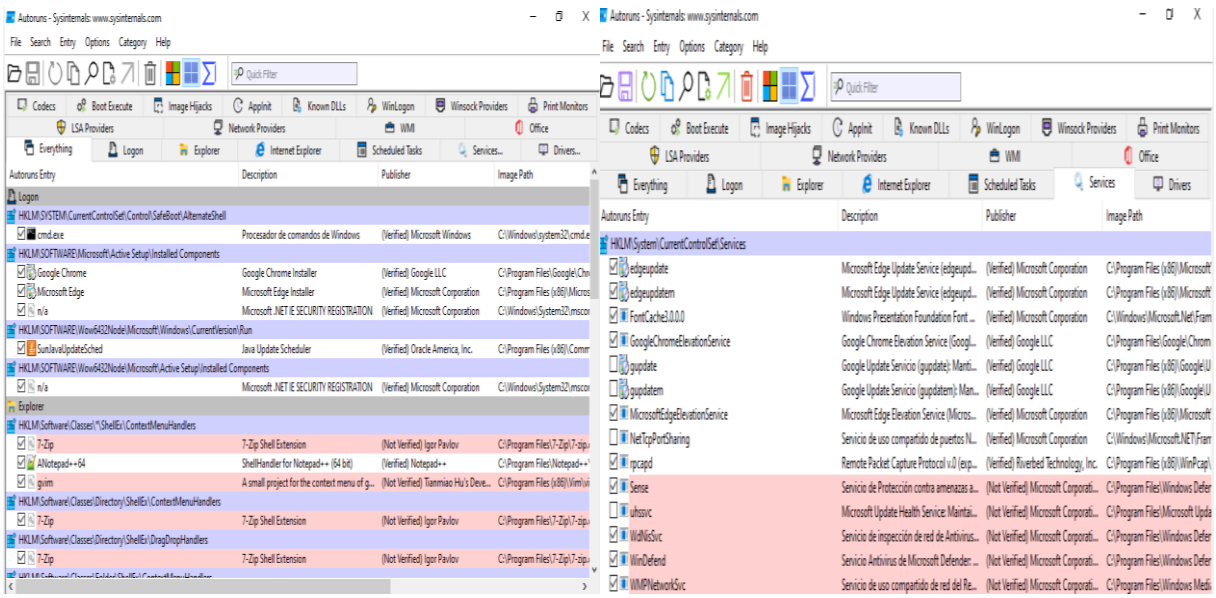
Figura 54: Resultados de la herramienta Process Hacker



Fuente: Elaboración propia

Una de las herramientas más completas utilizadas en análisis de malware de comportamiento es Autoruns. En el presente trabajo se la utilizó para verificar los resultados obtenidos anteriormente. Este software proporcionó información adicional en lo referente a los sistemas de archivos, procesos, tráfico de red y registros del computador.

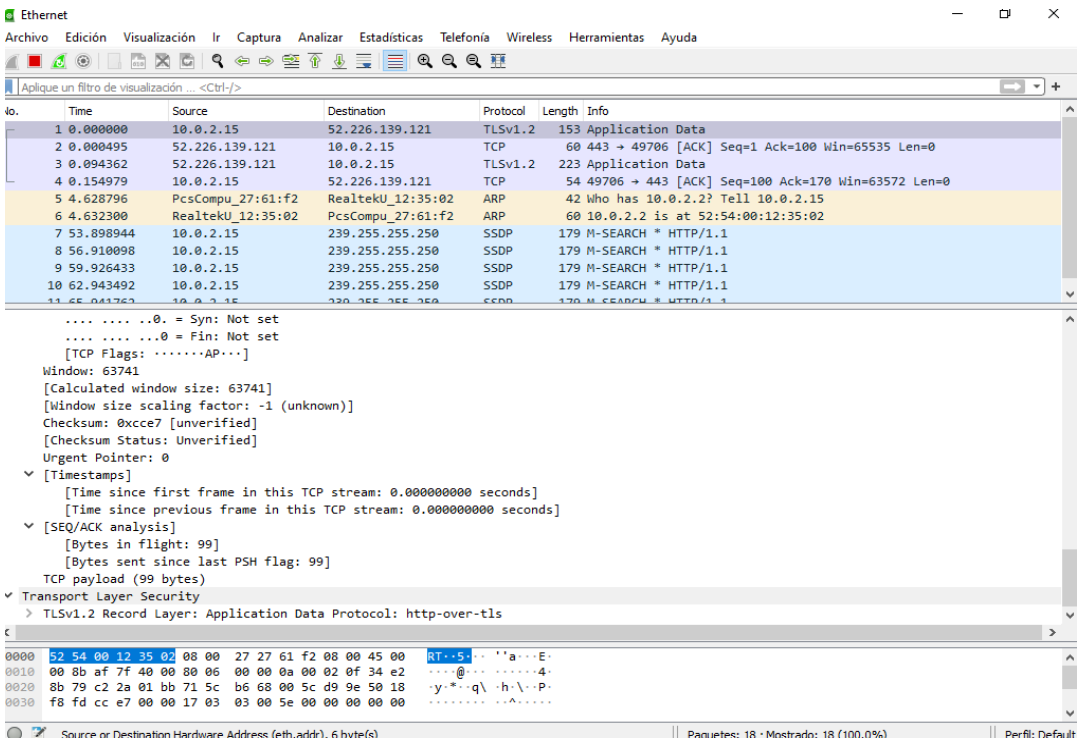
Figura 55: Resultados de la herramienta Autoruns



Fuente: Elaboración propia

Adicional a conocer los puertos y protocolos que nos ofrecen las tres herramientas aplicadas anteriormente, se ejecuta el Wireshark para verificar el comportamiento de la red en cuanto a paquetes, IPs (fuentes y destino) e incluso poder observar la información que se transmite en el medio.

Figura 56: Resultados de la herramienta Wireshark

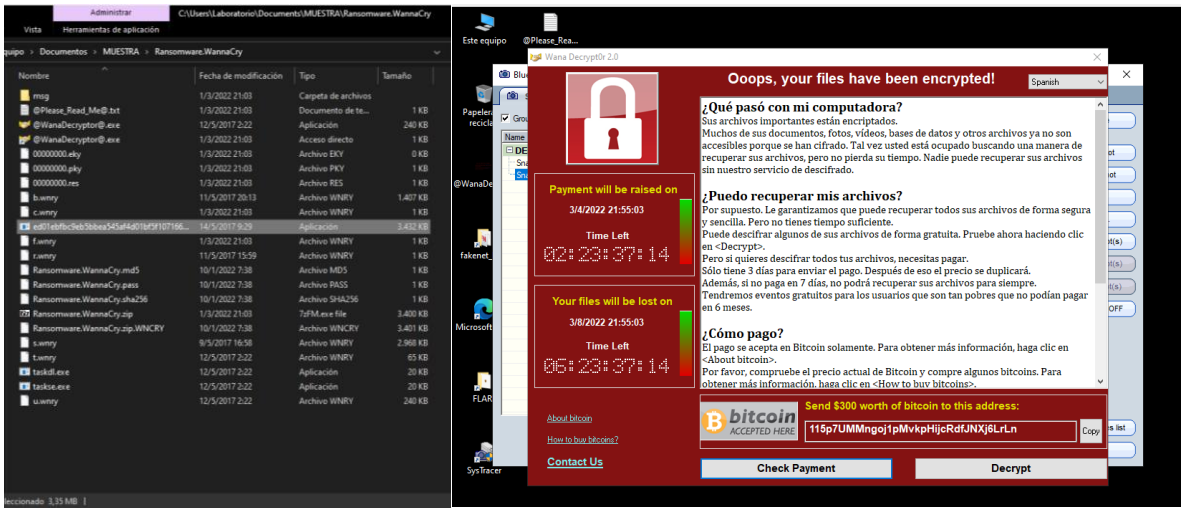


Fuente: Elaboración propia

- Ejecución del malware

Para realizar esta tarea se procede a ejecutar el malware, además según la metodología SAMA sugiere que antes de realizar el análisis se debe esperar veinte minutos para que el código malicioso realice las alteraciones correspondientes a su infección.

Figura 57: Ejecución del malware



Fuente: Elaboración propia

- Tareas de post ejecución del malware

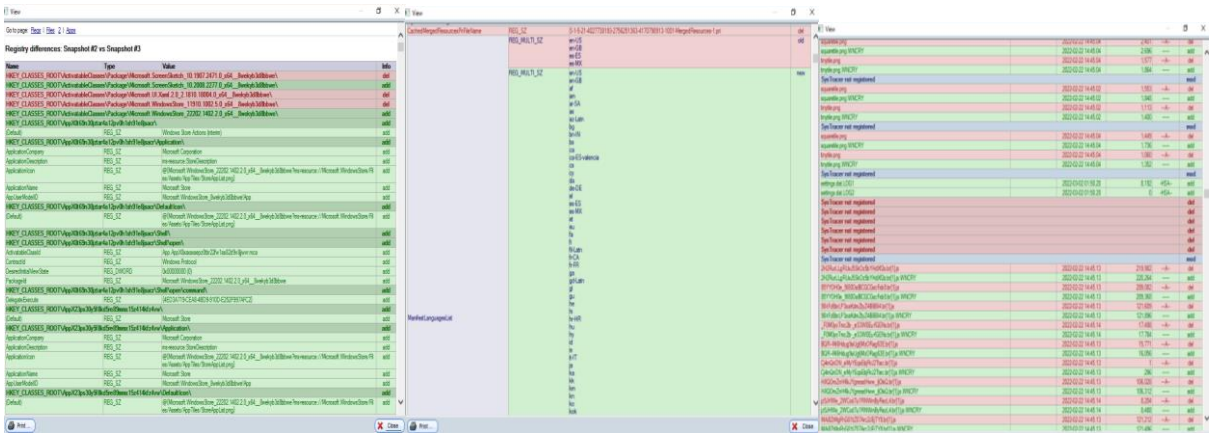
Para obtener los cambios realizados en el sistema después de la infección, se procedió a realizar otro snapshot en la herramienta Systracer, para comparar con la instantánea anterior. El reporte que emitió la herramienta evidencia los cambios realizados por el malware, tanto en el registro como en los archivos.

Con este reporte se puede constatar los resultados obtenidos en el análisis de código, tales como los lenguajes del mensaje, la encriptación de los archivos, la extensión .WNCRY que se agrega a los archivos, etc.

Al comparar las instantáneas tomadas antes y después de la ejecución del programa maligno, se obtuvo un reporte extenso en el que se evidenció la modificación, creación y eliminación de los registros y archivos como se muestra en la Figura 58.

Todas estas modificaciones las realiza primero a nivel de registros del sistema, analizando posteriormente las características de este en cuanto al lenguaje, hora, fecha, etc. y finalmente busca extensiones de imágenes, videos, documentos para encriptar una copia y proceder a eliminar el original.

Figura 58: Comparación de snapshots antes y después de la infección



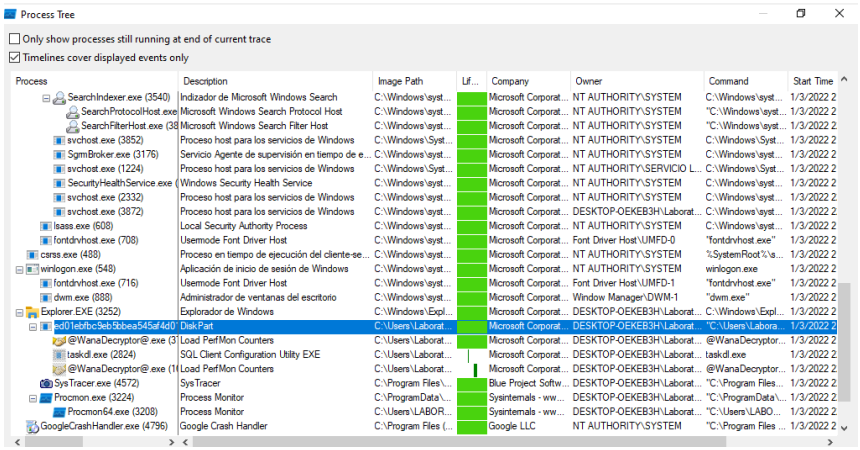
Fuente: Elaboración propia

Con base a lo realizado en la etapa de tareas previas a la ejecución del malware, se debe realizar un monitoreo del sistema para contrarrestar con la información obtenida anteriormente y poder identificar los cambios que ha realizado el malware en el sistema de archivos, procesos, el tráfico de la red y en los registros.

Se compararon los análisis de monitoreo realizados antes y después de la ejecución del malware y se obtuvieron los siguientes cambios:

Se visualizaron dos procesos nuevos mediante la herramienta procmon, que pertenecen al malware, ya que se crearon en el momento de la ejecución de la muestra. Estos PE tienen la función de mostrar el mensaje de recompensa.

Figura 59: Visualización de nuevos procesos después de la infección

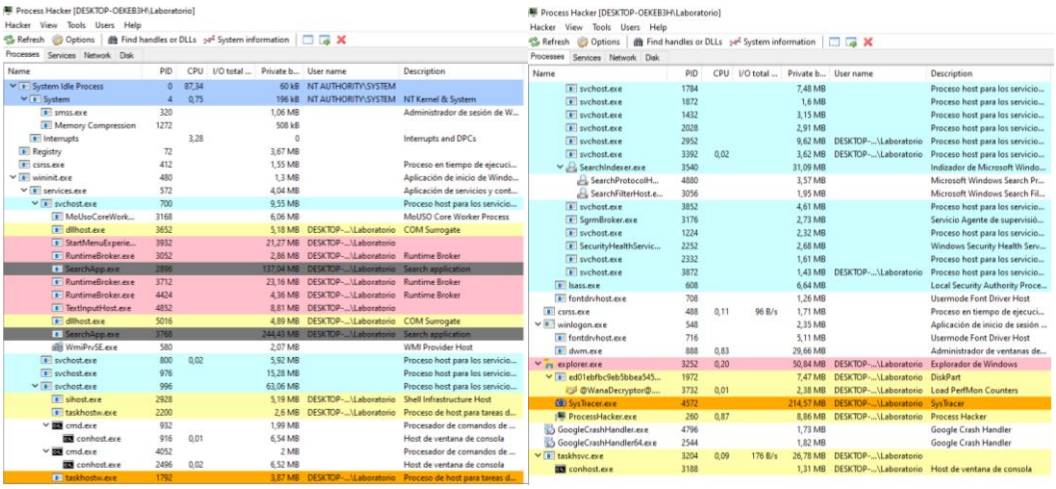


Fuente: Elaboración propia

Con la herramienta Process Hacker se confirmaron los resultados obtenidos anteriormente por medio del procmon, debido a que mostró los mismos archivos ejecutables creados por la

muestra analizada, estas dos herramientas tienen un funcionamiento similar en cuanto al monitoreo del sistema.

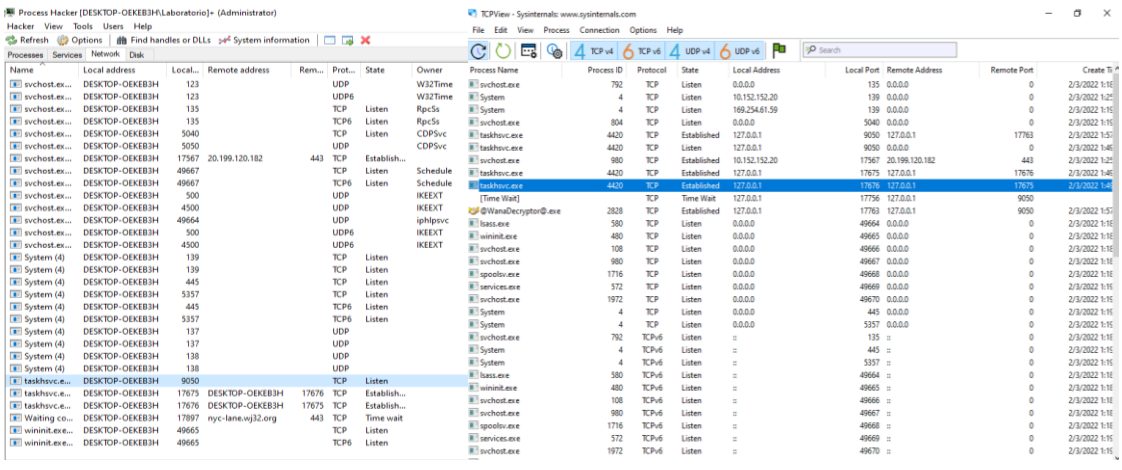
Figura 60: Visualización de los nuevos procesos después de la infección



Fuente: Elaboración propia

En la pestaña de red se mostraron las conexiones TCP que ha creado el malware en los puertos 9050, 1765, 17676, 17897. Además, en el último puerto, en la dirección remota aparece una url nyc-lane.wj32.org. Así mismo, se logró contrarrestar esta información con la herramienta TCPView. Figura 61.

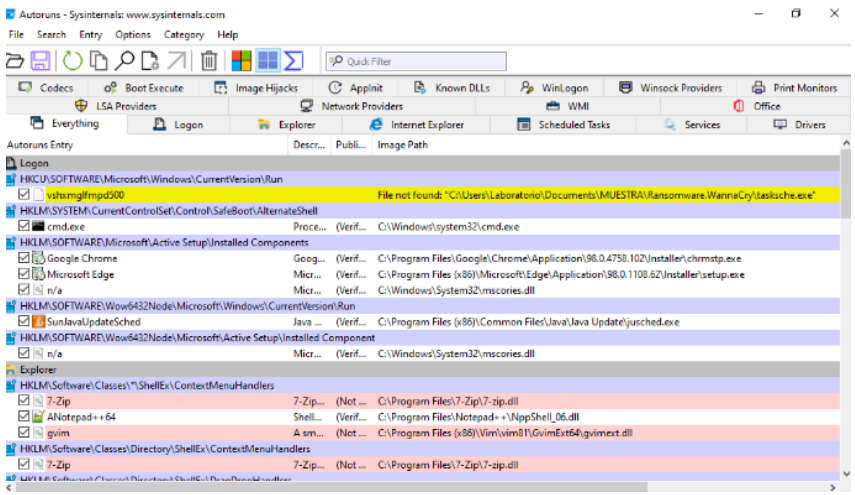
Figura 61: Visualización de las conexiones de red después de la infección



Fuente: Elaboración propia

En este monitoreo también se pudo visualizar que existió un cambio en el inicio de sesión del sistema, esta modificación se pudo detectar a través de la herramienta Autoruns, según este software, la alteración fue realizada en el registro HKCU\SOFTWARE\Microsoft\Windows\ CurrentVersion\Run, tal como se visualiza en la Figura 62.

Figura 62: Visualización de los cambios en el registro después de la infección

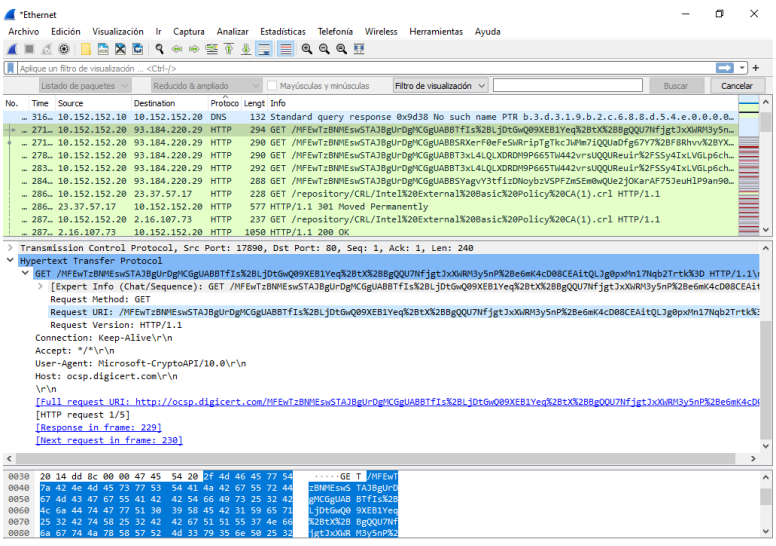


Fuente: Elaboración propia

Es importante recalcar que las máquinas Windows con REMnux deben establecer un canal de comunicación y que esta última conceda servicios a la máquina víctima, todas estas configuraciones se encuentra explicadas en el apartado **Configuraciones de las máquinas virtuales.**

Para comprender el comportamiento del malware en la red, se realizó un análisis de monitoreo de la misma con Wireshark, en los resultados de esta herramienta se puede observar que el malware realiza GET y POST constantemente a la url maliciosa ocsp.digicert.com e intenta recibir un archivo con un nombre extenso como se muestra en la imagen.

Figura 63: Visualización de las conexiones del malware con el protocolo HTTP



Fuente: Elaboración propia

Resultados

En los análisis de código y de comportamiento se analizó la información tanto de la estructura como de la ejecución del malware WannaCry.

Referente a su comportamiento el programa maligno realizó búsqueda de la información del sistema como la fecha, hora, lenguaje, etc. con el fin de utilizar esta información en el mensaje de recompensa. Seguido de esto, realiza modificaciones en los registros del sistema, crea archivos, asigna permisos e inicia procesos.

Los procesos ejecutados después de la infección buscan archivos de uso común, con extensiones de imágenes, documentos, audios, videos, etc. para proceder a encriptarlos y posteriormente eliminar los originales.

Finalizada la encriptación de los archivos, el malware inicia un proceso para mostrar el mensaje de recuperación junto a las direcciones de los wallet de bitcoin e intenta conectarse con direcciones externas maliciosas para descargar archivos

5. Conclusiones y trabajos futuros

5.1. Conclusiones

El malware es un software que se inserta en un sistema sin autorización con fines maliciosos [1]. Este tipo de ataque está creciendo en volumen y evolucionando en complejidad, en los últimos años se ha evidenciado que los grupos de cibercriminales se han volcado al uso de técnicas cada vez más complejas para lograr ataques cada vez más certeros [1]. La red de la Facultad de Sistemas y Telecomunicaciones (FACSISTEL) – UPSE no es ajena a esta realidad de ataques con malware al ser una red inestable e insegura, atenta contra la integridad, disponibilidad y confidencialidad de los datos, conllevando a que sus usuarios estén expuestos a la infección de software malicioso, esta vulnerabilidad motivó el presente trabajo investigativo de implementar un laboratorio virtual mediante una metodología Systematic Approach to Malware Analysis (SAMA) de análisis de código y de comportamiento de malware para lograr una mejor comprensión de los mecanismos internos de funcionamiento del código malicioso presentado en un informe de los hallazgos encontrados, sobre todo la clasificación del malware, su comportamientos y el nivel de daño que podría ocasionar. Este conocimiento de su estructura se podrá permitir prospectivamente cuáles son los mecanismos de defensa que se podrían aplicar.

Para el cumplimiento de los objetivos específicos se muestra en la Tabla 12 la conclusión de cada uno de ellos:

Tabla 12: Conclusiones de los objetivos específicos

Objetivo 1: Estudiar las técnicas y métodos existentes en el análisis del malware.
Conclusión: Las premisas más relevantes que se pueden evidenciar en este trabajo investigativo se concatenan en primer lugar con las técnicas y métodos existentes tanto en el análisis de código y comportamiento del malware, donde tiene preponderancia la revisión manual que posibilitó conocer la estructura del código malicioso en su accionar y ejecución, así como el análisis de comportamiento cuyo examen exhaustivo es de carácter automático, que permitió visualizar el comportamiento del malware y revela por ejemplo que problemas causa en el sistema, los cambios en las claves de registros, etc., por ende estas técnicas,

métodos y procedimientos se complementan porque la aplicación de las mismas posibilitaron que se realice un estudio más profundo y preciso en relación a la estructura del malware examinado.

Objetivo 2: Examinar y determinar las herramientas de análisis de código y de comportamiento del malware

Conclusión: Es de suma importancia destacar que las herramientas de análisis de código y de comportamiento del programa malicioso que se utilizaron son las establecidas principalmente en la metodología SAMA y de otras instancias como Flare. El uso de varias herramientas en la fase de análisis permitió contrastar la información analizada del código malicioso, emitiendo resultado contrastados, validados y más fidedignos. Existieron dos tipos de herramientas, las que se implantaban en las máquinas víctimas y las que proporcionaban servicios. Estas herramientas se situaban en máquinas virtuales que simulaban un entorno real en tres escenarios relacionados con los sistemas operativos Windows (Windows Flare), Linux (REMnux) y Android.

Objetivo 3: Diseñar e implementar una arquitectura moderna del laboratorio de malware.

Conclusión: Para el cumplimiento de este objetivo en la arquitectura del laboratorio se implantó la metodología SAMA la misma que puede adaptarse a cualquier tipo de organización y se relaciona sobre todo que no solo se utilizaron los sistemas operativos Window y Linux, sino que también se utilizó el sistema operativo Android que es de uso exclusivo de los dispositivos móviles como celulares y tablets, además se evidenció que las herramientas aplicadas son las últimas versiones. Por las razones expuestas queda claramente establecido que la modernidad y actualización de la arquitectura se dan por la metodología, herramientas y escenarios analizados en este TFM.

Se puede concluir que la metodología seleccionada para el análisis de malware en el laboratorio se fundamentó en Systematic Approach to Malware Analysis (SAMA) que cuenta con las fases de acciones iniciales, clasificación, análisis de código y de comportamiento, la misma que proporcionó una guía eficaz en todo el ciclo del análisis del código malicioso, sin importar la complejidad de este y de esta manera se logró comprender los mecanismos internos del malware estudiado.

5.2. Trabajos futuros

En cuanto a la prospectiva de este trabajo investigativo se podría crear un nuevo escenario relacionado con la aplicabilidad y factibilidad de este laboratorio en máquinas virtuales basadas en el sistema operativo IOS, complementando de esta manera el contexto y estado del arte de la problemática planteada. Además, los resultados obtenidos en relación con la estructura, clasificación y comportamiento de un malware especifico, proporcionarían los insumos necesarios para crear antivirus más efectivos en relación al código malicioso estudiado, y tipos de mecanismo de defensas que protejan de mejor manera los sistemas operativos utilizados en este proyecto.

6. Referencias

[1] Y. Yunus y S. Ngah, «Review of Hybrid Analysis Technique for Malware Detection,» *IOP Publishing,* pp. 1-7, 2020.

[2] ESET, «Tendencias en ciberseguridad para el 2021,» 2021.

[3] ESET, «ESET Security Report 2021,» 2021.

[4] J. Pepper, Creating a Malware Analysis Lab and Basic Malware Analysis, Iowa: Creative Components, 2018.

[5] M. Sikorski y A. Honig, Practical malware analysis: The hands-on guide to dissecting malicious software, San Francisco: No Starch Press, 2012.

[6] S. A. a. G. S. a. K. S. a. N. Y. Roseline, «Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm,» *IEEE Access,* pp. 206303-206324, 2020.

[7] J. Bermejo Higuera, Desarrollo de un análisis e ingeniería inversa de código malicioso, Escuela Politécnica Superior, 2015.

[8] J. Bermejo Higuera, C. Abad Aramburu, J. R. Bermejo Higuera, M. Á. Sicilia Urban y J. A. Sicilia, «Systematic Approach to Malware Analysis (SAMA),» *Applied Sciences,* pp. 1-31, 2020.

[9] J. J. Vicente Mohino, J. Bermejo Higuera, J. R. Bermejo Higuera, J. A. Sicilia, M. Sánchez Rubio y J. J. Martínez Herraiz, «MMALE a methodology for malware analysis in linux environments,» *Computers, materials and continua,* pp. 1-11, 2021.

[10] Ö. Aslan y R. Samet, «A Comprehensive Review on Malware Detection Approaches,» *IEEE Access,* pp. 6249-6271, 2020.

[11] C. C. Urcuqui López y A. Navarro Cadavid, «Clasificadores de machine learning para análisis de malware en Android,» *ResearchGate,* 2016.

[12] Academia ESET, «Introducción al Análisis de Malware,» ESET, 2021. [En línea]. Available: https://www.academiaeset.com/default/store/13599-analisis-de-malware. [Último acceso: 7 Noviembre 2021].

[13] G. Bell Bitjoka y A. Elang, «Network Malware Laboratory BasedOn Honeypots Technologies,» *Journal of Cybersecurity Research (JCR),* vol. 3, nº 1, pp. 1-12, 2018.

[14] R. Tenesaca Gómez, Definición de un laboratorio para obtención, gestión y análisis de malware, Cuenca: Universidad Internacional de La Rioja (UNIR), 2020.

[15] J. Aycock y K. Barker, «Creating a Secure Computer Virus Laboratory,» *13th Annual EICAR Conference,* pp. 1-13, 2004.

[16] X. Riofrío, F. Salinas y D. Galindo, «A Design for a Secure Malware Laboratory,» *Springer,* vol. 1099, pp. 273-286, 2019.

[17] Kaspersky, «En 2020, nuevos archivos maliciosos detectados cada día alcanzaron los 360,000,» Latam Kaspersky, 21 Enero 2021. [En línea]. Available: https://latam.kaspersky.com/blog/en-2020-nuevos-archivos-maliciosos-detectados-cada-dia-alcanzaron-los-360000/20901/. [Último acceso: 10 Noviembre 2021].

[18] Kaspersky, «Clasificación de Malwares,» Kaspersky, 29 Octubre 2013. [En línea]. Available: https://latam.kaspersky.com/blog/clasificacion-de-malwares/1608/. [Último acceso: 10 Noviembre 2021].

[19] J. Li y S. Shad, «Detecting smart, self-propagating Internet worms,» *IEEE Conference on Communications and Network Security,* pp. 193-201, 2014.

[20] M. Davis, S. Bodmer y A. Lemasters, Hacking Exposed Malware & Rootkits: Security Secrets & Solutions, New York: McGraw-Hill, 2010.

[21] R. Ramírez Gutiérrez y O. Reyes Fuentes, Implementación de un laboratorio de Análisis de malware, México: Universidad Nacional Auntónoma de México, 2009.

[22] R. Tahir, «A Study on Malware and Malware Detection Techniques,» *International Journal of Education and Management Engineering,* vol. 8, nº 2, pp. 20-30, 2018.

[23] Universidad de la Rioja, «Análisis de malware,» UNIR, Madrid, 2021.

[24] A. Valencia y S. Galicia, «Detección de malware con modelo de lenguaje y su clasificación mediante SVM,» *Res. Comput. Sci,* vol. 115, pp. 9-18, 2016.

[25] T. Jumbo Tene, Metodología para el análisis de malware en un ambiente controlado, Cuenca: Universidad Politécnica Salesiana, 2017.

[26] R. Rivera-Guevara, Detección y Clasificación de Malware con el Sistema de Análisis de Malware Cuckoo, Madrid: UNIR, 2018.

[27] V. Fernández Coderch, Sandbox environment for IOT malware, Barcelona: Universitat Oberta de Catalunya, 2019.

[28] J. R. Bermejo Higuera, J. Bermejo Higuera, J. A. Sicilia Montalvo, J. Cubo Villalba y J. J. Nombela Pérez, «Benchmarking Approach to Compare Web Applications Static Analysis Tools Detecting OWASP Top Ten Security Vulnerabilities,» *Computers, Materials & Continua,* vol. 64, nº 3, pp. 1555-1577, 2020.

[29] G. Quicasaque, Como realizar análisis dinámico y estático a una muestra malware, Bogotá: Universidad Piloto de Colombia, 2015.

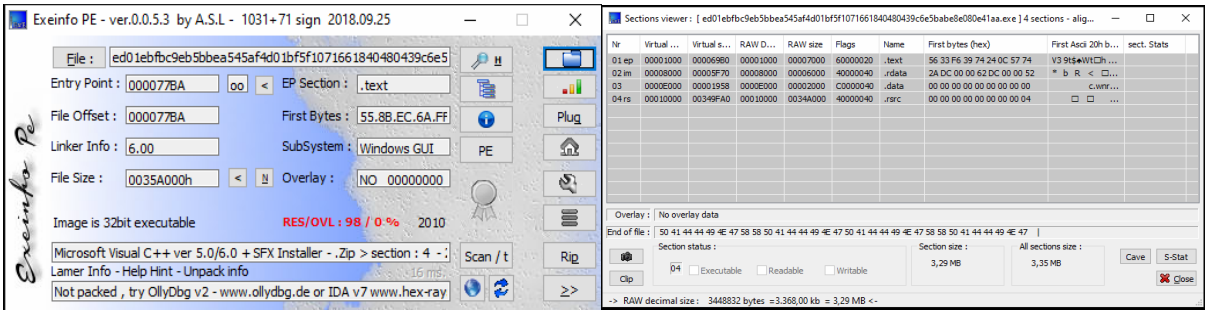
[30] A. Singhal y S. Gandhi, «Reverse Engineering,» *International Journal of Computer Applications,* vol. 108, nº 9, pp. 1-3, 2014.

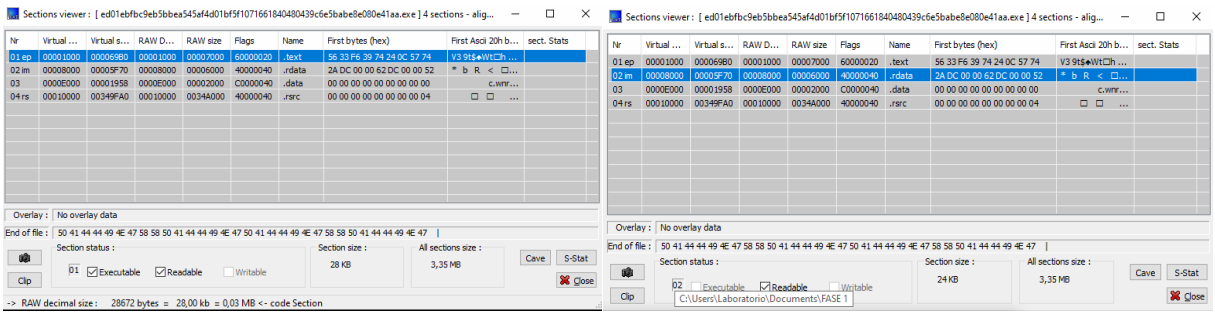
[31] Y. Prayudi y I. Riadi, «Implementation of malware analysis using static and dynamic analysis method,» *International Journal of Computer Applications,* vol. 117, nº 6, pp. 11-15, 2015.

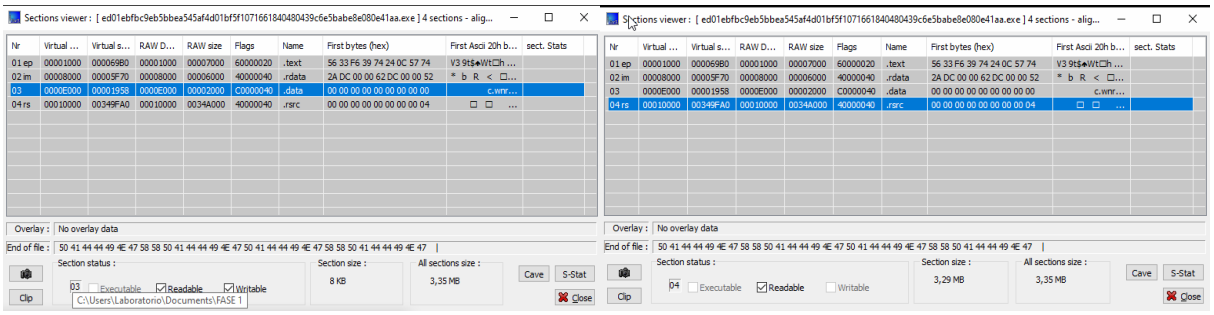
[32] P. O'Kane, S. Sezer y K. McLaughlin, «Obfuscation: The Hidden Malware,» *IEEE Security & Privacy,* vol. 9, nº 5, pp. 41-47, 2011.

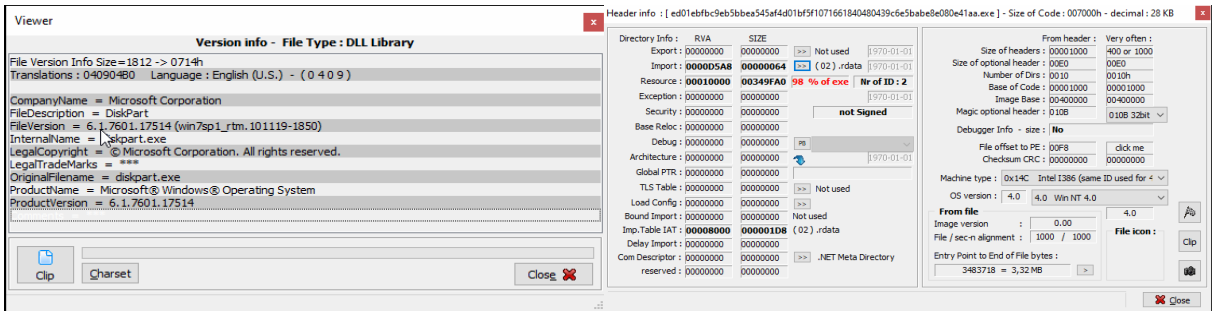
[33] A. H. e. a. Di Iorio, «Análisis forense de memoria: malware y evidencia oculta,» *XVI Simposio Argentino de Informática y Derecho,* pp. 83-95, 2016.

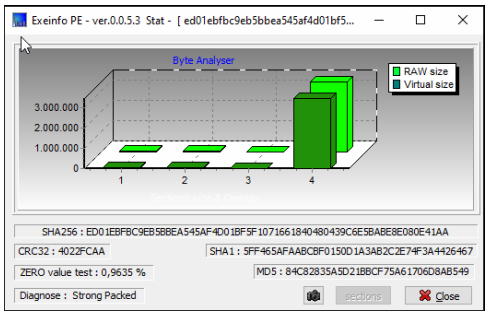
ANEXO A. Resultados del análisis de la herramienta Exeinfo PE



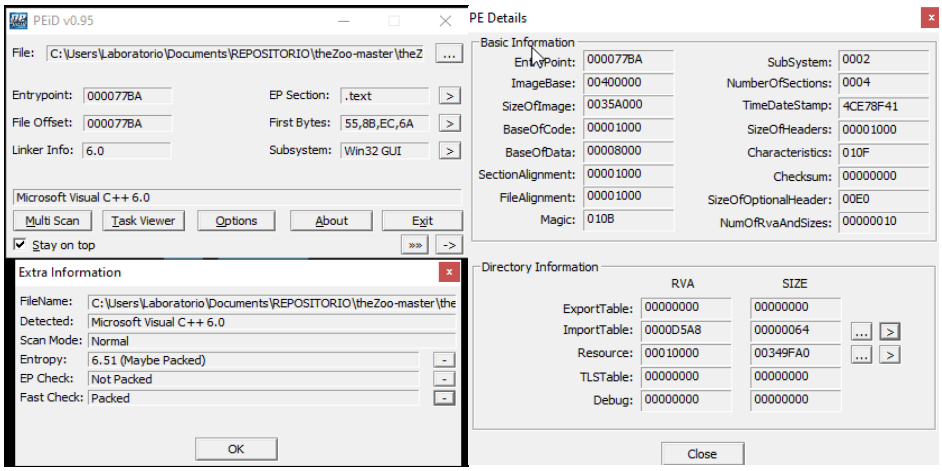


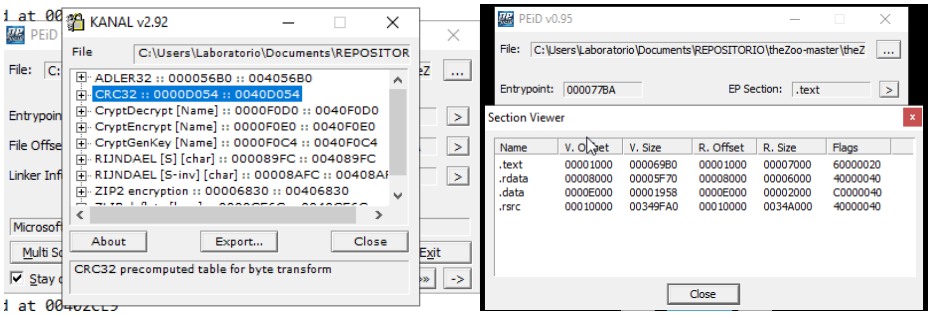


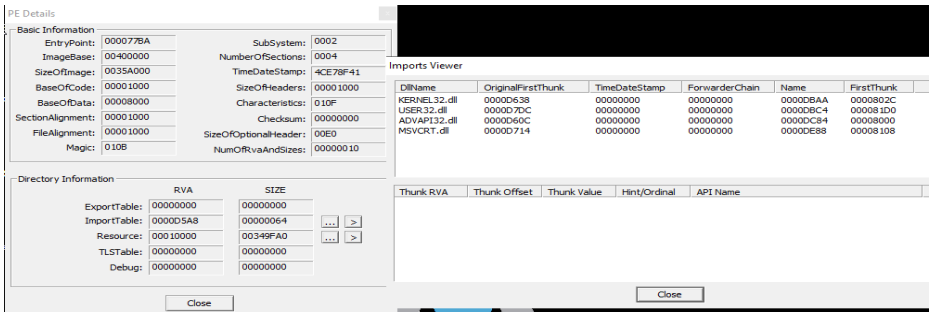


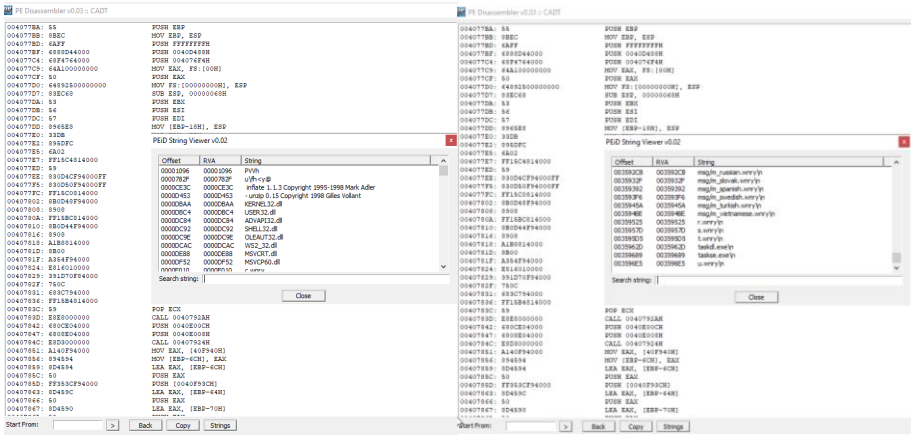


ANEXO B. Resultados del análisis de la herramienta PEiD

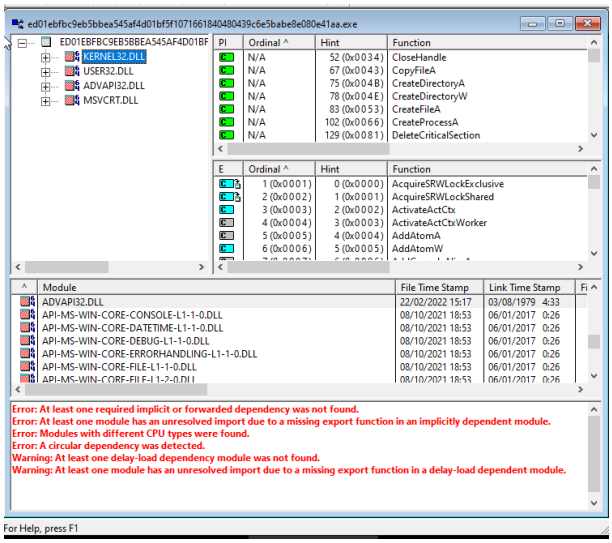




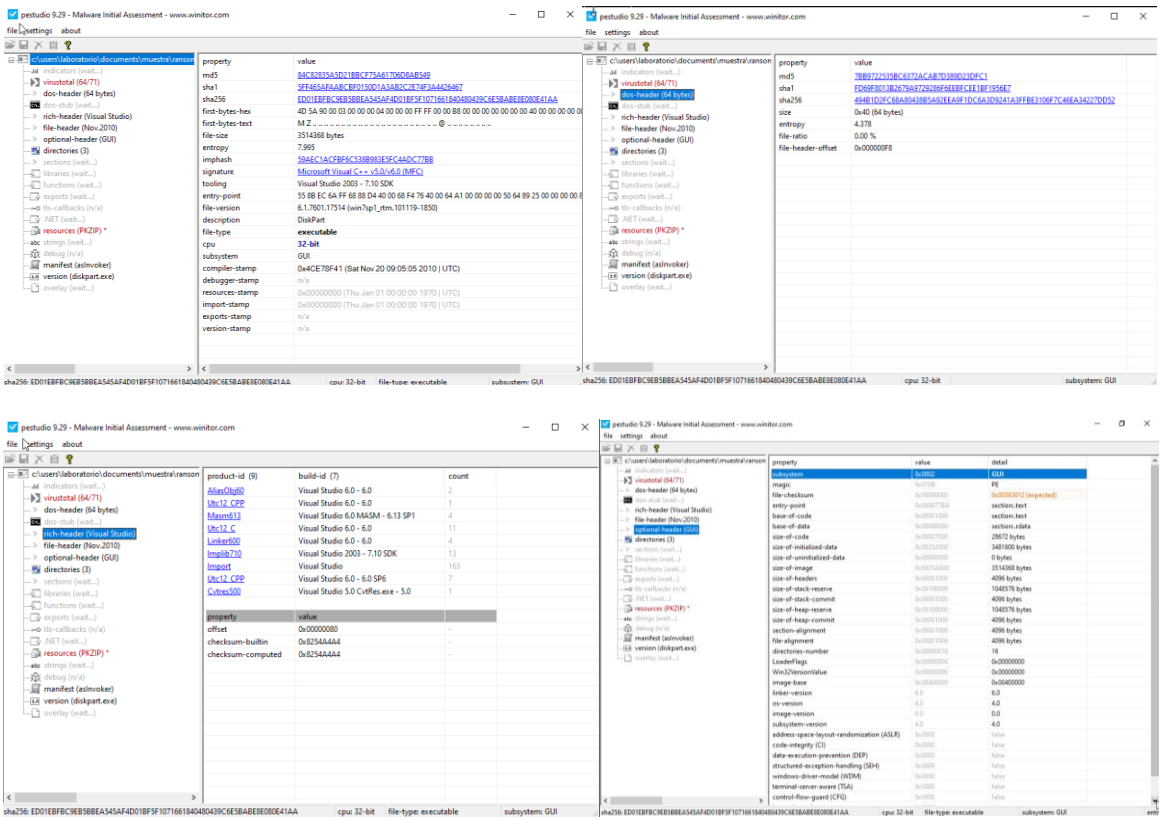


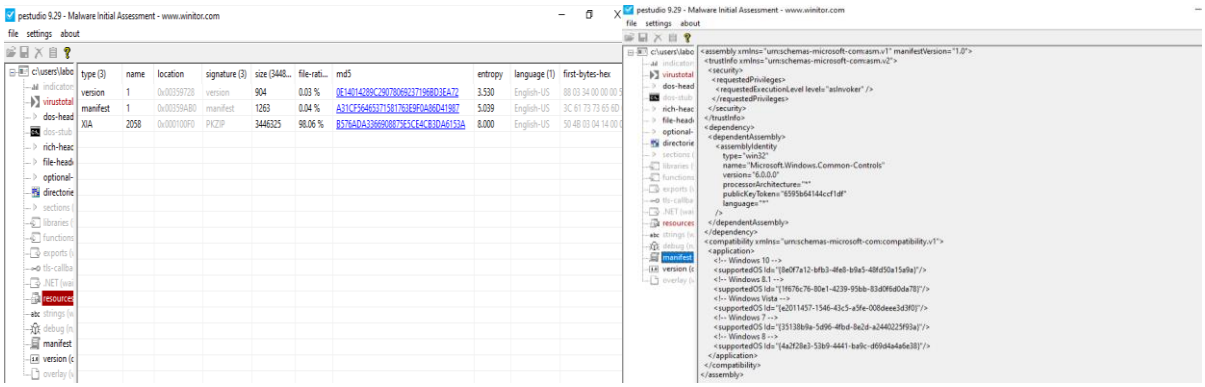


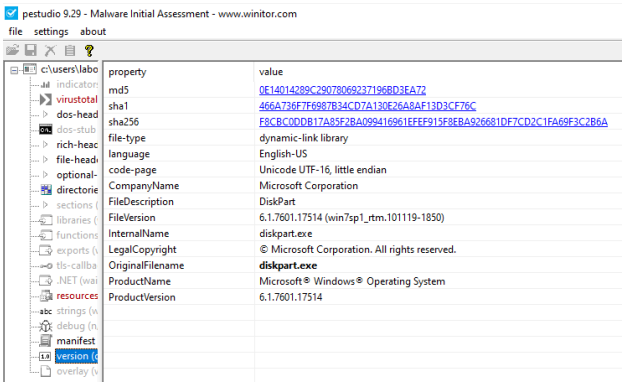
ANEXO C. Resultados del análisis de la herramienta Dependency Walker



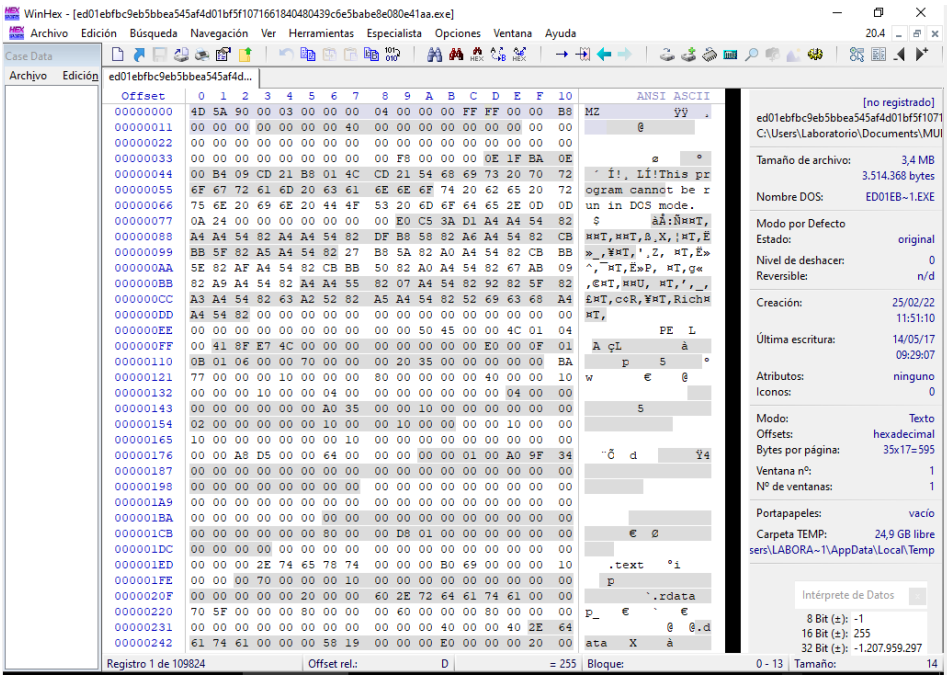
ANEXO D. Resultados del análisis de la herramienta Pestudio



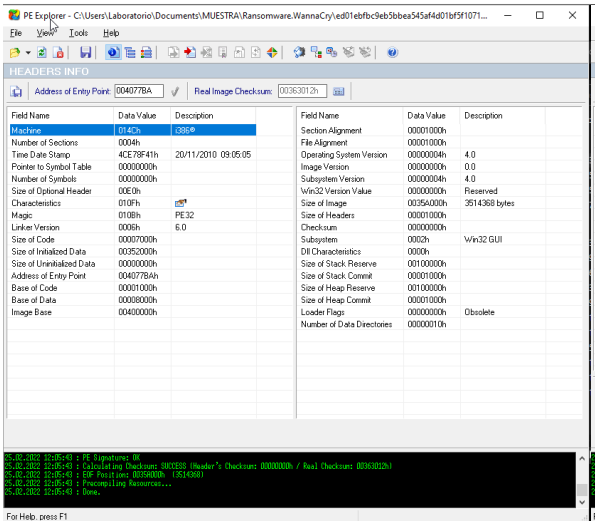


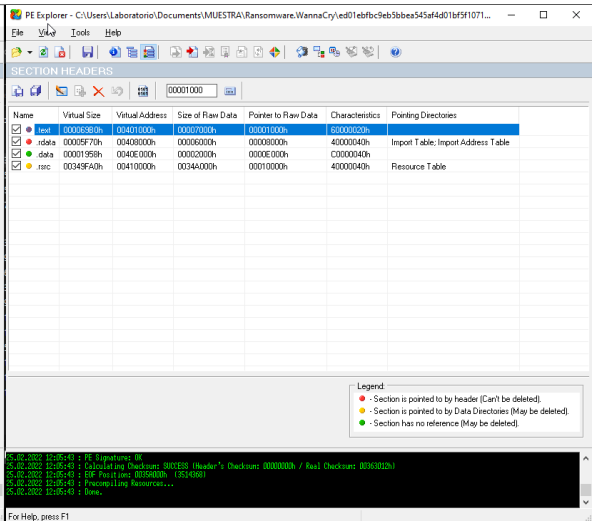


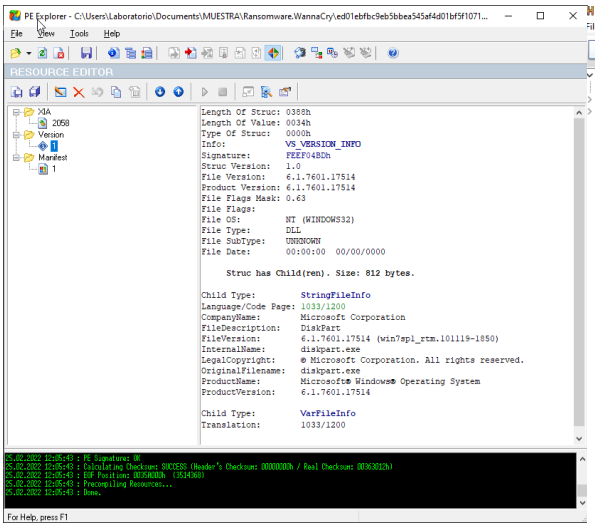
ANEXO E. Resultados del análisis de la herramienta WinHex

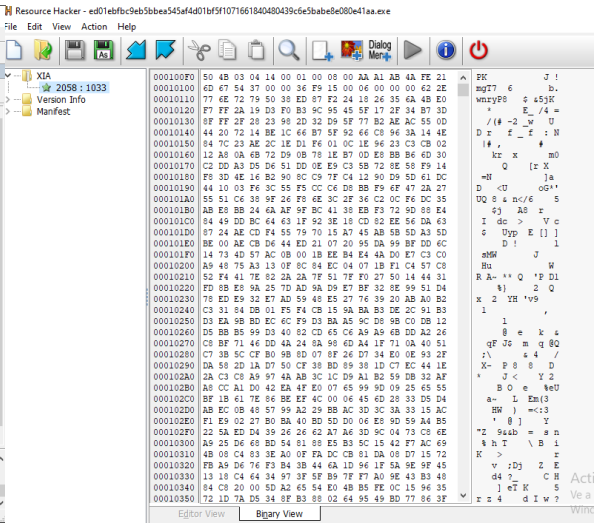


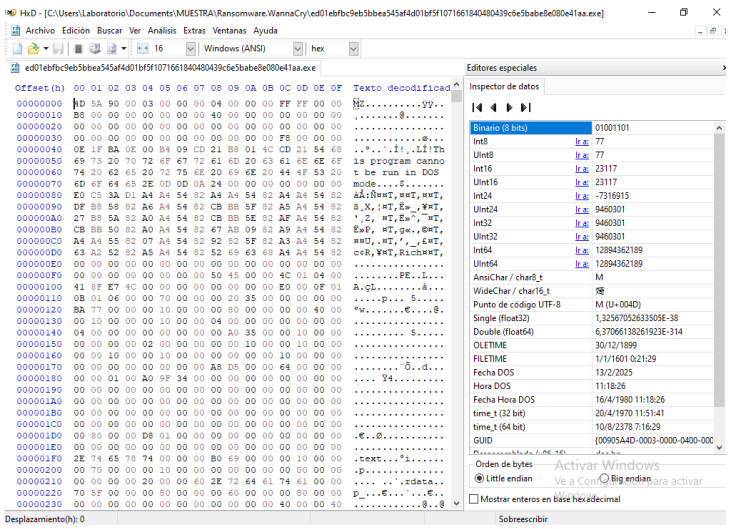
ANEXO F. Resultados del análisis de la herramienta PE Explorer



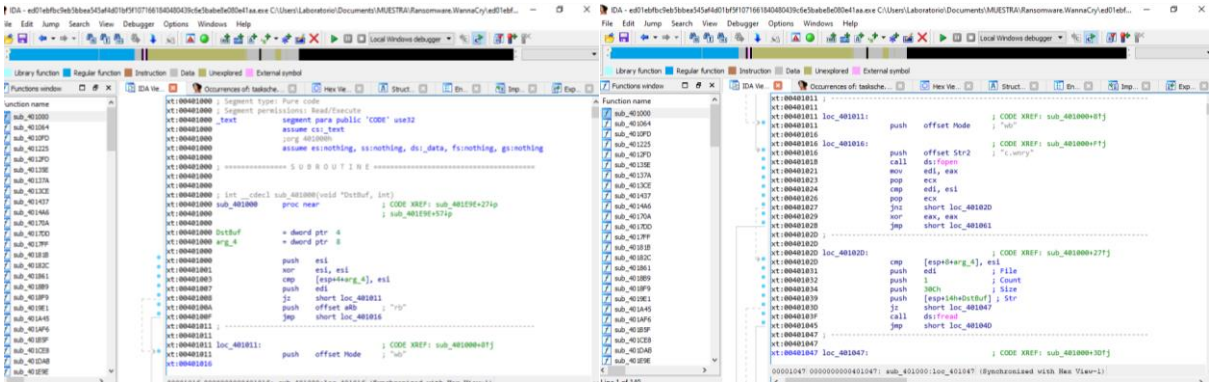


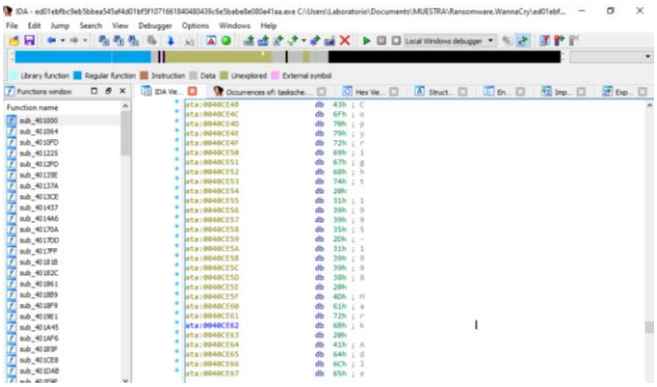




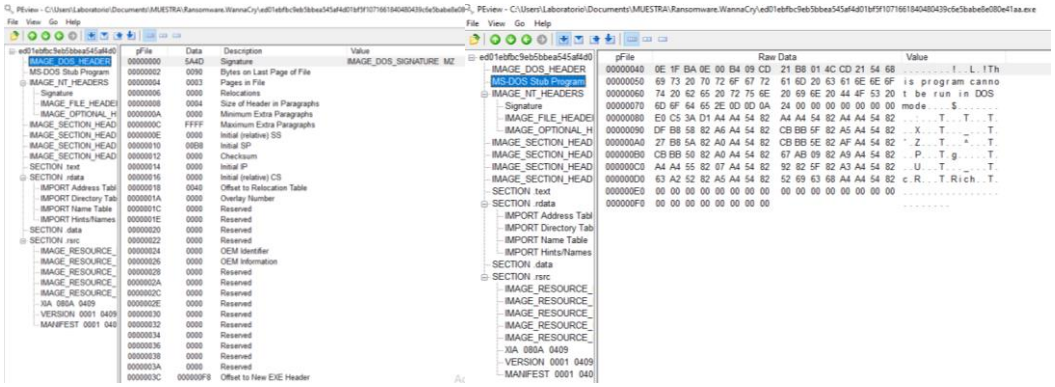


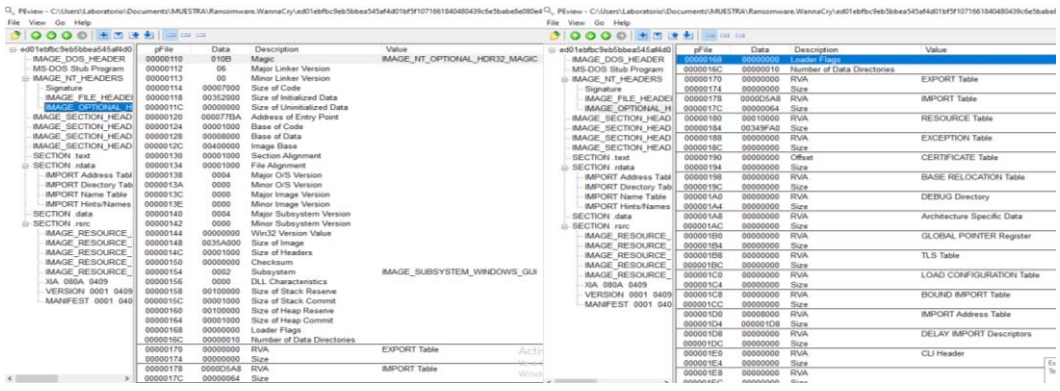
ANEXO G. Resultados del análisis de la herramienta IDA Pro

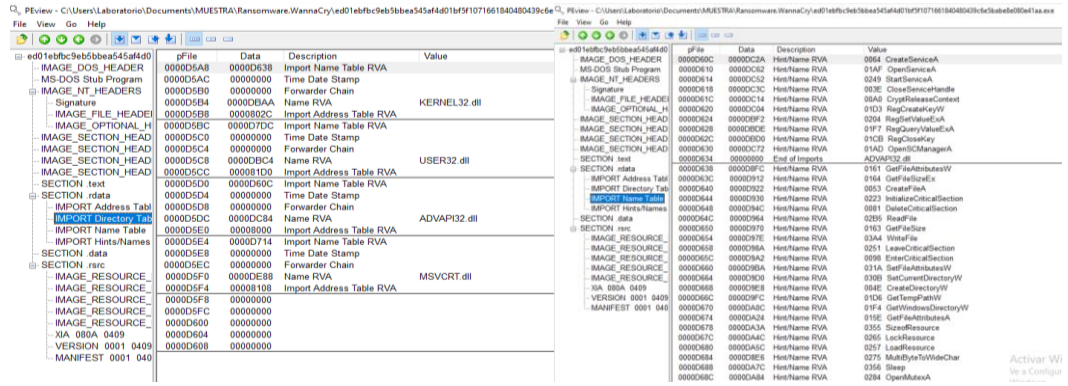


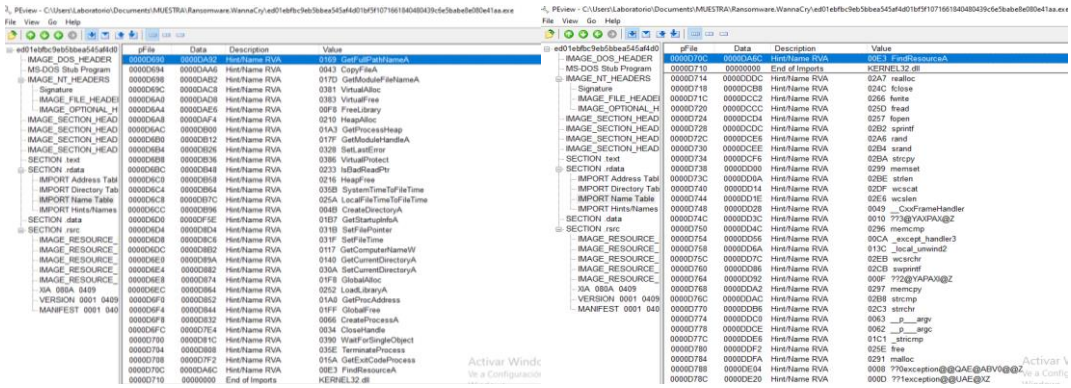


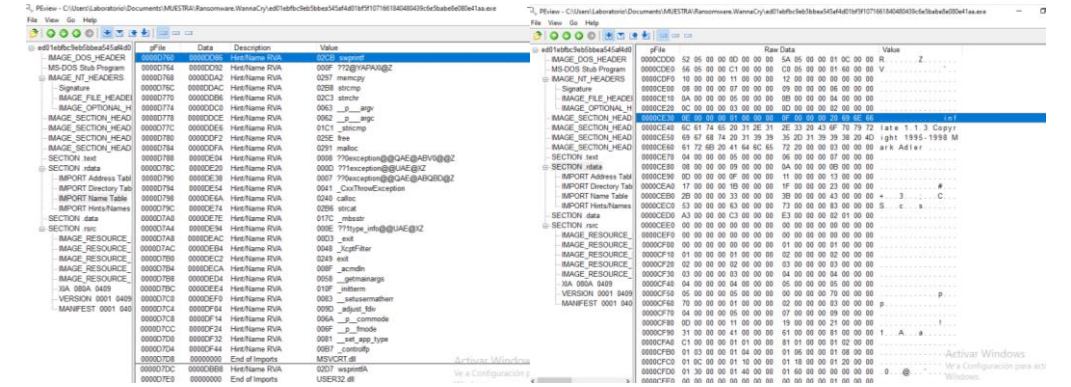
ANEXO H. Resultados del análisis de la herramienta PE View

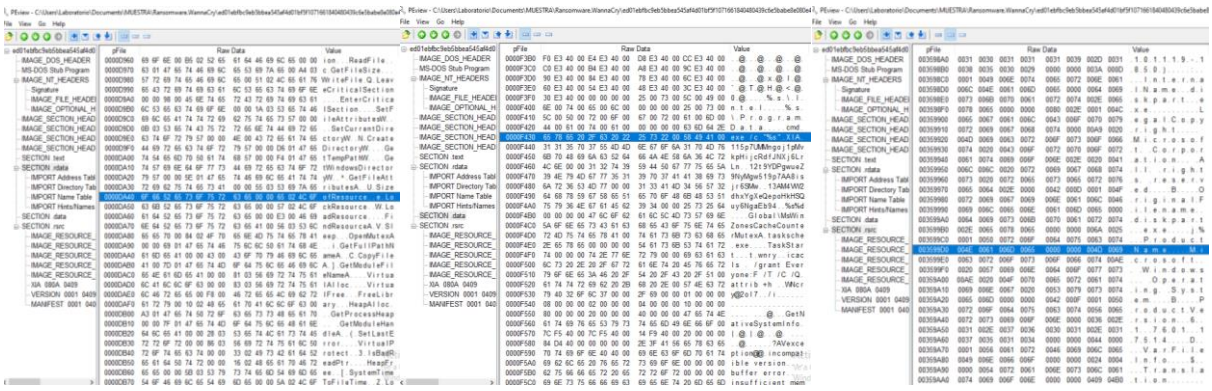




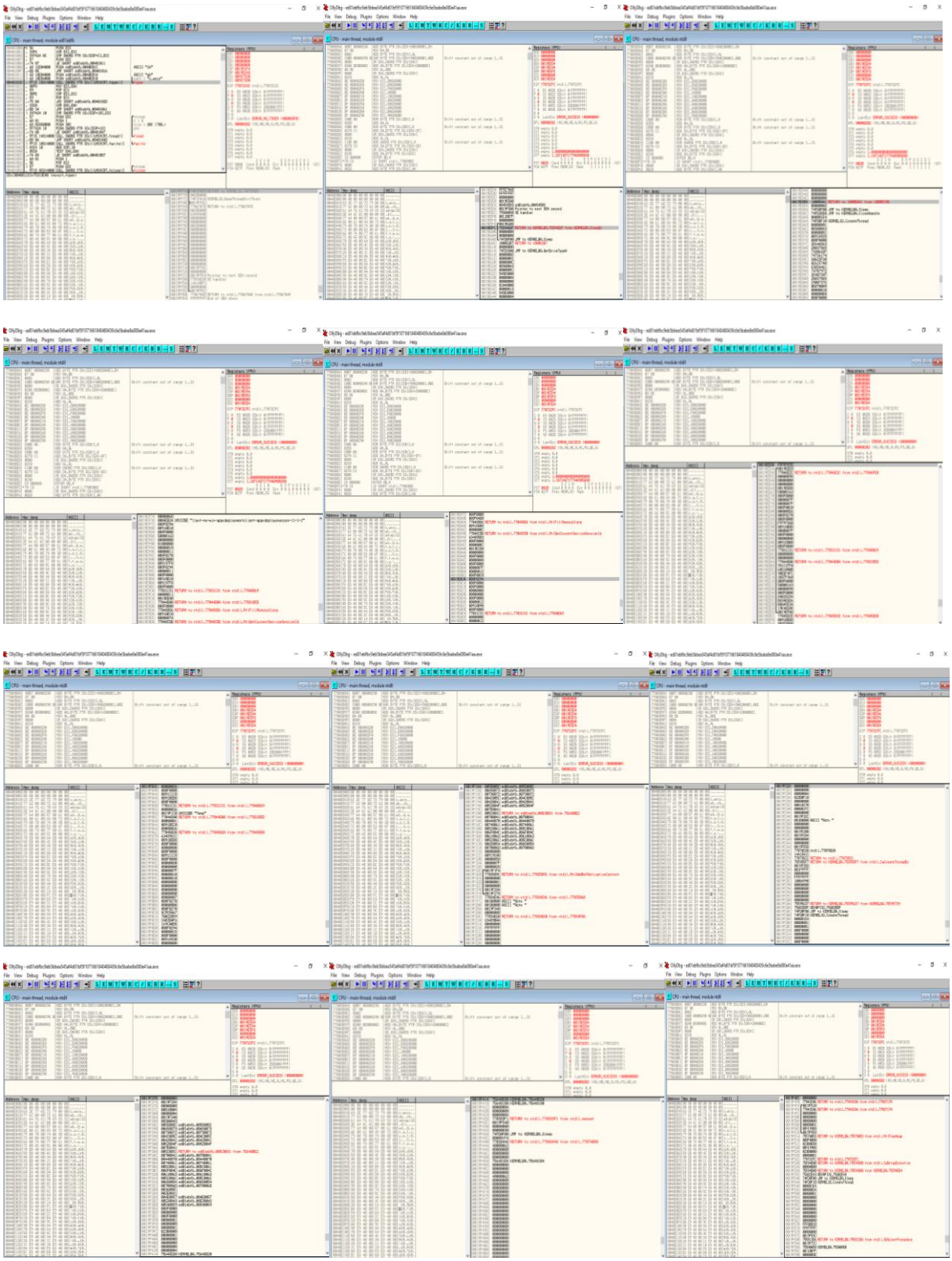


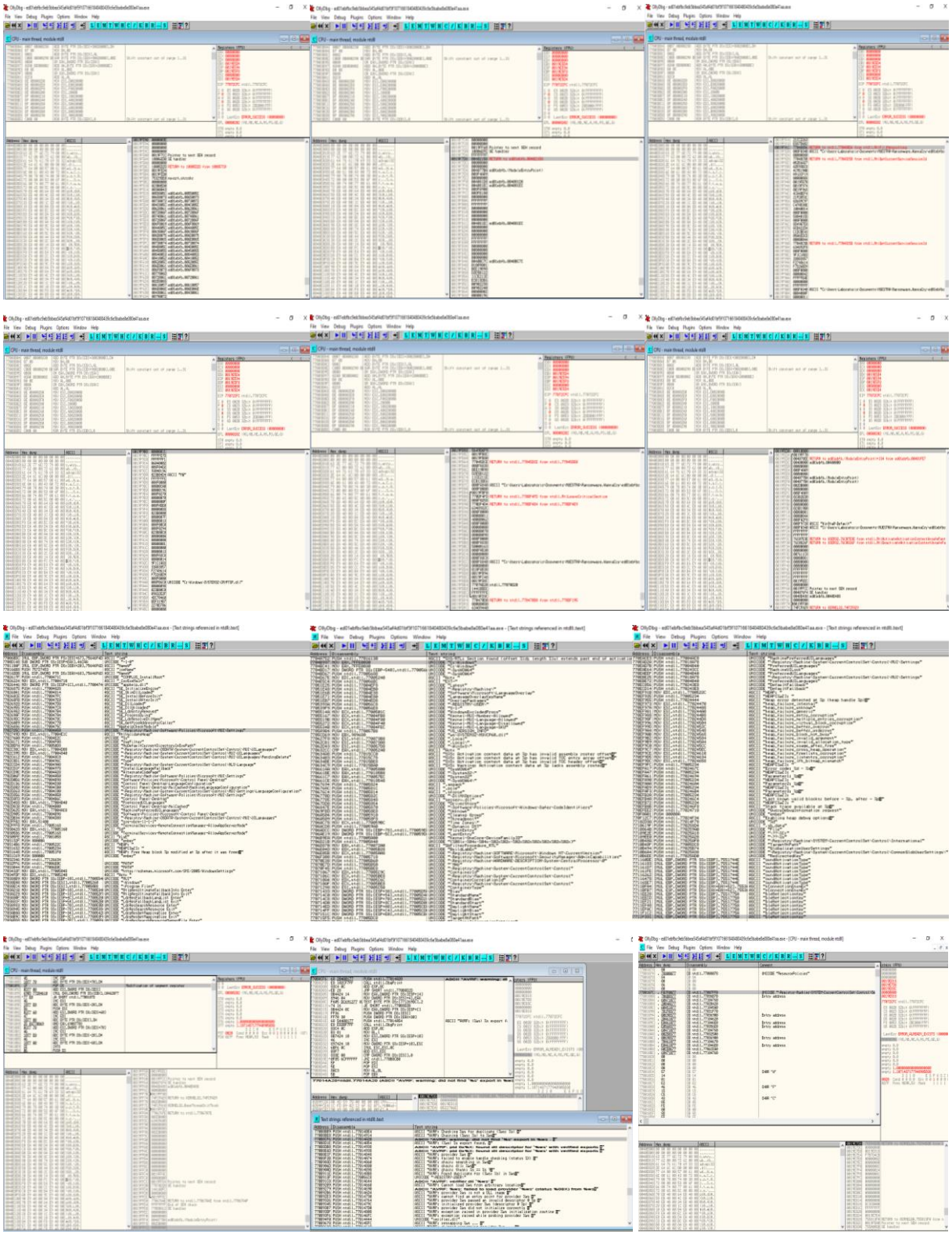




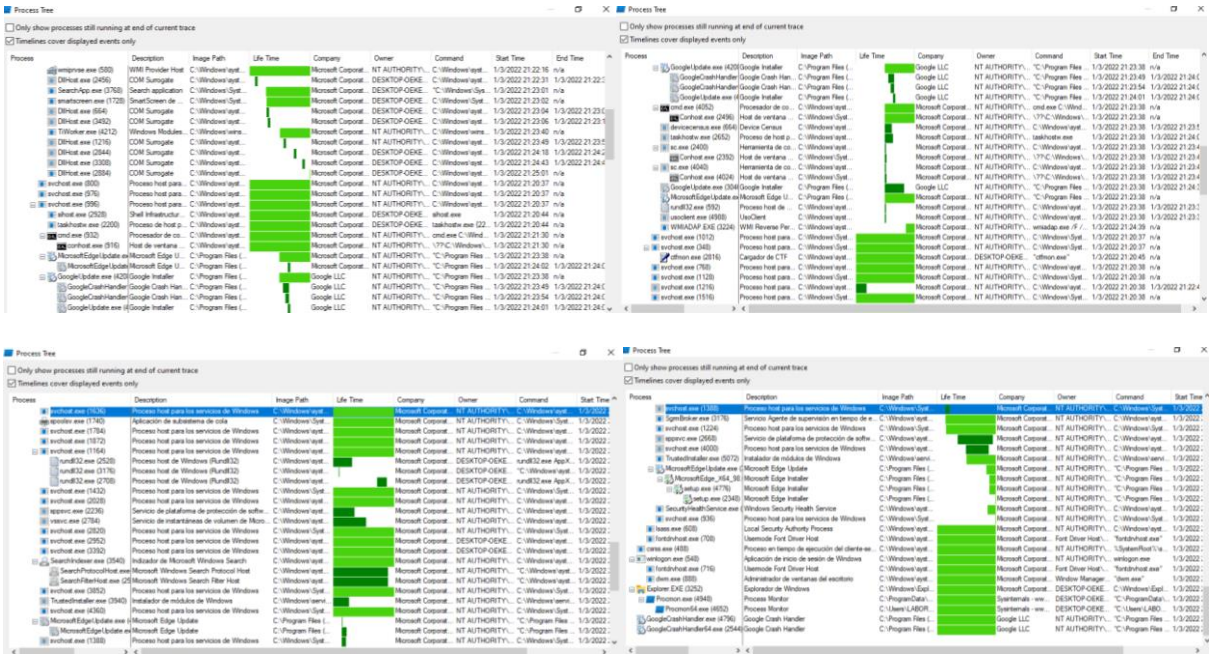


ANEXO I. Resultados del análisis de la herramienta OllyDbg

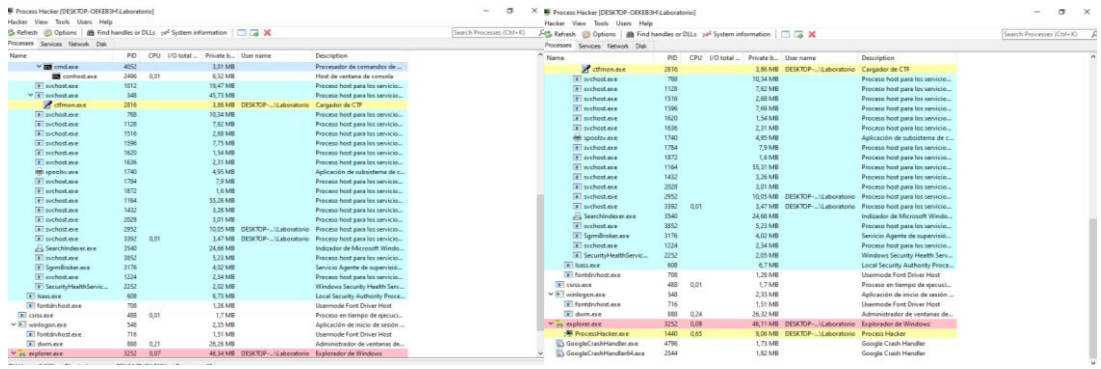


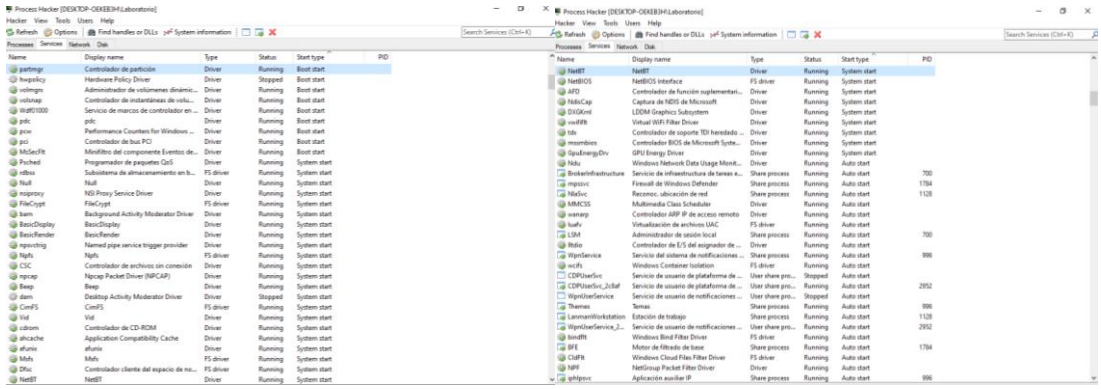


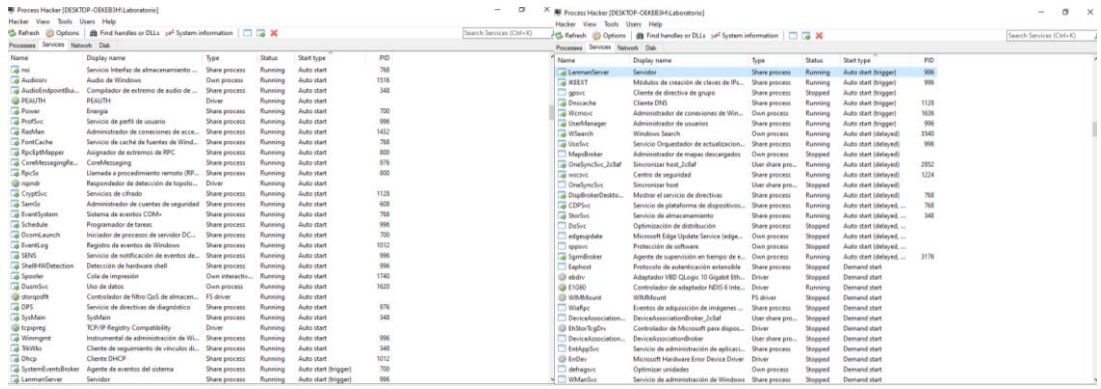
ANEXO J. Resultados del análisis de la herramienta Process Monitor. Antes de la ejecución del malware

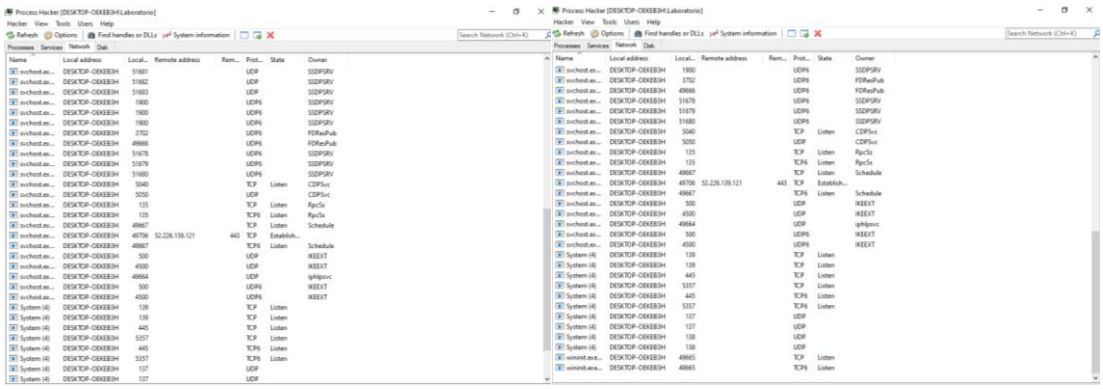


ANEXO K. Resultados del análisis de la herramienta Process Hacker. Antes de la ejecución del malware









ANEXO L. Resultados del análisis de la herramienta Autoruns.

Antes de la ejecución del malware

ANEXO M. Resultados del análisis de la herramienta Wireshark. Después de la ejecución del malware