TRABAJO FIN DE GRADO
GRADO DE INGENIERÍA INFORMÁTICA
CURSO 2017-2018

Herramienta de Extracción de Información de Malware

LENIN BENAVIDES QUINTANA

CARLOS ROA MEDINA

Directores:

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Departamento de Ingeniería del Software e Inteligencia Artificial

---

FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

Agradecimientos

Primero de todo, agradecer a mi madre, familia por su apoyo y comprensión, así como la oportunidad de poder realizar el Grado de Ingeniería en Computadores.

Segundo, agradecer a Luis Javier García Villalba y a Ana Lucila Sandoval Orozco, los directores de este Trabajo, todo el apoyo que nos han proporcionado para la realización de este trabajo.

Por último, agradecer a mis amigos y a Esteban Armas Vega por darnos la oportunidad de poder trabajar en este proyecto y por compartir con nosotros sus conocimientos acerca de seguridad.

Resumen

Los ataques de ransomware alrededor del mundo son cada vez más frecuentes debido al incremento en el uso de dispositivos con acceso a Internet. Este hecho provoca que el número de usuarios vulnerables a este tipo de software malicioso aumente vertiginosamente. La información recolectada en tiempo real acerca del comportamiento del ransomware es escasa. Esto se debe a que las herramientas forenses para la extracción de esta información se centran en entornos controlados y la toma de muestras o análisis de los ordenadores de las víctimas reales es muy difícil de realizar. En este trabajo se propone una herramienta que extrae información en tiempo real de un ataque de ransomware para facilitar las tareas de análisis forense, la clasificación del ransomware y la correlación de las cuentas bitcoin asociadas. La herramienta diseñada combina la captura de pantalla del ataque del ordenador victima con el reconocimiento de patrones para determinar si corresponden a una muestra de ransomware. Adicionalmente, se realiza el reconocimiento óptico de caracteres, el volcado de la memoria RAM y la extracción de archivos relevantes para el análisis. Con el fin de evaluar, la herramienta resultante, se realizaron experimentos con distintas muestras de ransomware sobre un ordenador real infectado con dichas muestras.

Palabras clave

Análisis Forense, Bitcoin, Internet, Ransomware, Reconocimiento de Patrones, Reconocimiento Óptico de Caracteres, Volcado de Memoria RAM.

Abstract

Ransomware attacks around the world are becoming more frequent due to the increase in the use of devices with Internet access. This fact causes that the number of vulnerable users to this type of malicious software increases vertiginously. The information collected in real time about the behavior of ransomware is scarce. This is because the forensic tools for extracting this information focus on the controlled environments and the sampling or analysis of the computers of the actual victims is very difficult to perform. In this work we offer a tool that extracts information in real time from a ransomware attack to facilitate the tasks of forensic analysis, the classification of ransomware and the correlation of the associated bitcoin accounts. The designed tool combines the capture of the computer attack screen with pattern recognition to determine if it corresponds to a sample of ransomware. Additionally, optical character recognition, RAM memory volume and extraction of relevant files for analysis are performed. For the purpose of the evaluation, the resulting tool, an experiment was registered with ransomware samples on a real computer infected with the samples.

Keywords

Forensic Analysis, Bitcoin, Internet, Ransomware, Pattern Recognition, Optical Character Recognition, Memory RAM Dump

Lista de Acrónimos

AES	Advanced Encryption Standard
API	Application Programming Interface
ARPARNET	Project Agency Network
C&C	Comando y Control
CMS	Content Management System
CPU	Unidad Central de Proceso
DES	Data Encryption Standard
DNS	Domain Name System
GUI	Interfaz Gráfica de Usuario
IP	Internet Protocol
LDT	Local Descriptor Table
MARE	Malware Analysis Reverse Engineering
MBR	Master Boot Record
OCR	Reconocimiento Óptico de Caracteres
PDF	Portable Document Format
PE	Portable Ejecutable
PIL	Python Imaging Library
PNG	Portable Network Graphics
RAM	Random Access Memory
RLO	Right to left Override
SIFT	Scale-Invariant Feature Transform
SVM	Máquinas de Soporte Vectorial

TLS	Thread Local Storage
USB	Universal Serial Bus

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

En sus inicios, el uso de Internet era mínimo siendo utilizado por sectores industriales, militares e investigación. Poco a poco se fue introduciendo en la sociedad aumentando rápidamente el número de usuarios de Internet. El continuo desarrollo de la tecnología y su facilidad de adquisición atrajo una gran cantidad de consumidores permitiendo que Internet sea accesible en todos lados.

Los dispositivos tecnológicos se han convertido en medios de almacenamiento y comunicación imprescindibles en la rutina diaria de la sociedad actual. Como consecuencia, la información privada almacenada en dispositivos con acceso a Internet. Se convierte en objeto de interés para un grupo minoritario de internautas, con la intención de extraerla y obtener un beneficio. Esto ha promovido la aparición de aplicaciones maliciosas para atacar los dispositivos conectados a Internet.

El ransomware es uno de los tipos de software malicioso más peligroso que se puede hallar en Internet. En los últimos años es utilizado por cibercriminales en campañas de ataque a diferentes entidades públicas o privadas.

En 2016 hubo una campaña de ransomware dirigida a hospitales como el Hollywood Presbyterian medical center en los Ángeles, donde un malware de tipo ransomware bloqueó el acceso al sistema, hasta que se efectuó el pago de $17000 dólares en concepto de "rescate". La portavoz del FBI Laura Miller comunicó que se hicieron cargo de las investigaciones, sin embargo, fuentes policiales explicaron al medio de comunicación The Time que el hospital había pagado el rescate antes de solicitar asistencia de la policía [1].

De igual forma, en Alemania distintos hospitales fueron blancos de ataque afectando el funcionamiento de sus sistemas informáticos. Por ejemplo, el sistema de rayos X no pudo acceder a los datos que necesitaba debido a que se

encontraban cifrados. Afortunadamente, los hospitales no llegaron a pagar el rescate ya que recuperaron los datos desde las copias de seguridad [2].

Otros objetivos destacables fueron importantes empresas de medios de comunicación como The New York Times y universidades de prestigio como Calgary en Canadá donde se pagó $16000 dólares para recuperar correos que fueron cifrados por una semana [3].

En general, los ataques pretenden interrumpir el funcionamiento normal del servicio informático, como sucedió con las máquinas expendedoras de billetes del tren de San Francisco, permitiendo a las personas viajar sin pagar dichos billetes [4]. La mayoría de software malicioso tiene como objetivo obtener información confidencial, tanto de los usuarios como de las grandes empresas. El almacenamiento de datos personales en la red es cada vez más común. Este hábito aumenta la importancia que tienen los servidores en las empresas tecnológicas, convirtiéndose en parte fundamental de su funcionamiento y blanco principal para los cibercriminales. Un ejemplo es el ataque masivo de ransomware a empresas como MongoDB, donde se "secuestraron" 32.000 servidores exigiendo el pago de un rescate mediante bitcoins para recuperar la información. Empresas como Telefónica, eBay, además de gobiernos que utilizan este servicio se vieron afectadas [5].

Hoy en día nadie está exento de sufrir una ataque informático, de hecho el medio de comunicación BBC dio a conocer, 17 bibliotecas estadounidenses comprometidas con este tipo de ataques [6].

1.1. Motivación

Actualmente en el mercado hay herramientas que se enfocan en distintos aspectos del análisis de ransomware; algunas se centran en el volcado de memoria, otras en la observación del comportamiento e incluso existen herramientas capaces de automatizar dichas funcionalidades, pero necesitan de personal cualificado para llevar a cabo estas tareas. La escasez de herramientas que unifiquen estas funcionalidades acerca de la obtención de muestras de ransomware, ha generado el interés de cuerpos de seguridad de adquirir una herramienta que automatice los procesos de obtención de información en muestras de ransomware para facilitar la labor del analista forense.

1.2. Contexto

El presente Trabajo de Fin de Grado se enmarca dentro de un proyecto de investigación titulado RAMSES aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (Convocatoria H2020-FCT-2015, Acción de Innovación, Número de Propuesta: 700326)y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Además de la Universidad Complutense de Madrid participan las siguientes entidades:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern

(Alemania)

- Trilateral Research & Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universitaet des Saarlandes (Alemania)
- Dirección General de Policía - Ministerio del Interior (España)

1.3. Objetivos

El presente trabajo tiene los siguientes objetivos:

- Elaborar un estado del arte sobre software malicioso, centrándose en el ransomware, su evolución, clases, familias, funcionamiento y su uso como negocio lucrativo.

- Extraer información de la captura de pantalla de un ordenador infectado con ransomware extrayendo el mensaje mostrado y anonimizando la información del usuario.

- Diseñar e implementar una herramienta que permita la adquisición de información a través del mensaje mostrado por el ransomware mediante el reconocimiento óptico de caracteres (OCR), así como la búsqueda de archivos relevantes y obtener la memoria volátil del ordenador.

1.4. Plan de Trabajo

El desarrollo del proyecto se dividió en 2 etapas que se presentan en la Tabla 1.1.

En la primera etapa se definieron los objetivos del trabajo, los métodos de investigación y documentación.

La segunda etapa tuvo como objetivo el diseño e implementación de la herramienta de análisis de la imagen y el volcado de memoria RAM. Las

principales fases en esta etapa son: Especificación de los requisitos, diseño e implementación de la herramienta. Finalmente, se realizó la validación de la herramienta con distintas pruebas ransomware y se analizaron los resultados obtenidos en cada ejecución.

En ambas etapas se realizan sesiones semanales de tutorías para el seguimiento adecuado y asignación de nuevas tareas a realizar. En la Tabla 1.1 muestra un resumen del plan de trabajo realizado.

Tarea	Duración	Inicio	Fin
Investigación	100	23/11/17	29/01/18
Desarrollo	130	01/02/18	28/05/18

Tabla 1.1: Plan de trabajo

1.5. Estructura de la Memoria

El resto del trabajo está compuesto por 6 capítulos con la estructura que se comenta a continuación.

En el capítulo 2 se presenta la definición, historia y evolución del malware, los tipos de malware existentes, así como los métodos para realizar análisis forense de malware, hoy en día.

En el capítulo 3 se realiza un análisis del ransomware, desde su aparición hasta la actualidad, explicando los tipos de cifrado que utilizan, su capacidad de depuración y de evitar ser aislado en una máquina virtual.

En el capítulo 4 se presenta la literatura relacionada con este trabajo de fin de grado.

En el capítulo 5 se explica la propuesta de este trabajo para cumplir los objetivos planteados. Se detallan los resultados obtenidos en los experimentos

que evalúan el funcionamiento de la herramienta propuesta.

En el capítulo 6 se presentan las conclusiones y las líneas de investigación posibles.

Finalmente, en el capítulo 8 se realiza un resumen en inglés de la introducción y las conclusiones del presente trabajo.

1.6. División del Trabajo

La distribución del trabajo se llevó a cabo teniendo en cuenta 2 etapas, la investigación de las distintas herramientas y la implementación de estas en el proyecto final.

En la primera etapa se realizó el estudio de distintas herramientas de volcado de memoria, análisis de forense y herramientas de virtualización para el análisis de malware.

En la segunda etapa se llevó a cabo el diseño e implementación de las herramientas estudiadas en la etapa anterior, además de la investigación de distintas literaturas de análisis de imágenes, reconocimiento óptico de caracteres y el aprendizaje del lenguaje de programación Python. Para posteriormente, automatizar estas funcionalidades.

Los autores han realizado el trabajo de forma equitativa en todas las etapas de este. A continuación, se presenta un detalle de las actividades realizadas:

1.6.1. Lenin Benavides Quintana

Especificación de Requisitos:

- Reuniones de definición del proyecto con los directores.
- Investigar los principales ataques de malware, su clasificación según las acciones que realizan analizando algunos ejemplos de cada tipo.

- Investigar los tipos de análisis que se realizan posteriormente a la infección de un malware.
- Investigar la evolución del ransomware, los principales vectores de infección, el proceso de ataque de ransomware, las estrategias de protección utilizadas por el ransomware durante su análisis y las familias de ransomware conocidas.
- Estudio de la literatura relacionada con el análisis forense en equipos infectados con ransomware.
- Estudio de los algoritmos de extracción de características en imágenes digitales.
- Estudio de las técnicas de extracción de descriptores de características SIFT.
- Análisis de las herramientas de volcado de memoria RamCapturer y Winpmem con el objetivo de encontrar la herramienta óptima para ser automatizada.
- Reuniones semanales de seguimiento y control del proyecto con los directores.

Diseño:

- Diseño del módulo de extracción y análisis de la ventana emergente del ransomware.
- Diseño del módulo de búsqueda de ficheros relacionados con el ataque de ransomware.
- Diseño del módulo de automatización del volcado de la memoria RAM del dispositivo infectado.
- Reuniones semanales de seguimiento y control del proyecto con los directores.

Implementación:

- Implementación de las funciones de captura de pantalla del dispositivo víctima y extracción de la ventana emergente de un ransomware.
- Implementación del algoritmo de detección de patrones de una ventana de ransomware apoyado en la técnica de extracción de descriptores SIFT.
- Implementación del módulo de reconocimiento de óptico de caracteres.
- Automatización de la herramienta Dumpit para el volcado de memoria RAM.
- Reuniones semanales de seguimiento y control del proyecto con los directores.

Pruebas:

- Configuración, instalación y puesta en marcha del sistema para las pruebas realizadas con las muestras de ransomware.
- Búsqueda de las muestras de ransomware.
- Análisis de los resultados de las pruebas realizadas.
- Redacción de la memoria

1.6.2. Carlos Roa Medina

Especificación de Requisitos:

- Reuniones de definición del proyecto con los directores.
- Investigar los principales ataques de malware, su clasificación según las acciones que realizan analizando algunos ejemplos de cada tipo.
- Investigar los tipos de análisis que se realizan posteriormente a la infección de un malware.

- Investigar la evolución del ransomware, los principales vectores de infección, el proceso de ataque de ransomware, las estrategias de protección utilizadas por el ransomware durante su análisis y las familias de ransomware conocidas.
- Estudio de la literatura relacionada con el análisis forense en equipos infectados con ransomware.
- Estudio de los algoritmos de extracción de características en imágenes digitales.
- Estudio de las técnicas de extracción de descriptores de características SURF.
- Análisis de las herramientas de volcado de memoria Dumpit y FTK imager con el objetivo de encontrar la herramienta óptima para ser automatizada.

Diseño:

- Diseño del módulo de extracción y análisis de la ventana emergente del ransomware.
- Diseño del módulo de búsqueda de ficheros relacionados con el ataque de ransomware.
- Diseño del módulo de automatización del volcado de la memoria RAM del dispositivo infectado.

Implementación:

- Implementación de las funciones de captura de pantalla del dispositivo víctima y extracción de la ventana emergente de un ransomware.
- Implementación del algoritmo de detección de patrones de una ventana de ransomware apoyado en la técnica de extracción de descriptores SIFT.
- Implementación del módulo de reconocimiento de óptico de caracteres.

- Automatización de la herramienta Dumpit para el volcado de memoria RAM.

Pruebas:

- Configuración, instalación y puesta en marcha del sistema para las pruebas realizadas con las muestras de ransomware.
- Búsqueda de las muestras de ransomware.
- Análisis de los resultados de las pruebas realizadas.
- Redacción de la memoria

2. MALWARE

El software malicioso es cada vez más sofisticado, sus métodos de propagación han mejorado, adaptado y son cada vez más diversos e ingeniosos aumentando el número de usuarios víctimas.

2.1. Definición

Malware es un software malicioso que busca infectar un dispositivo o datos sin el conocimiento del usuario ejecutando acciones no deseadas o dañinas. Está diseñado para extraer información sin autorización o inutilizar los dispositivos infectados. Cualquier aplicación (software) que dañe a un usuario, ordenador o red se puede considerar como malware [7].

2.2. Evolución del Malware

John Von Neumann fue el primero que teóricamente postuló el concepto de virus de ordenador en 1949. En su artículo, discutió la existencia de programas virales. Fue el primer concepto de software malicioso y tomó varios años hasta que apareciera la primera instanciación del concepto de Von Neumann [8]. En la Figura 2.1 se observa algunos de los principales virus con sus respectivas fechas.

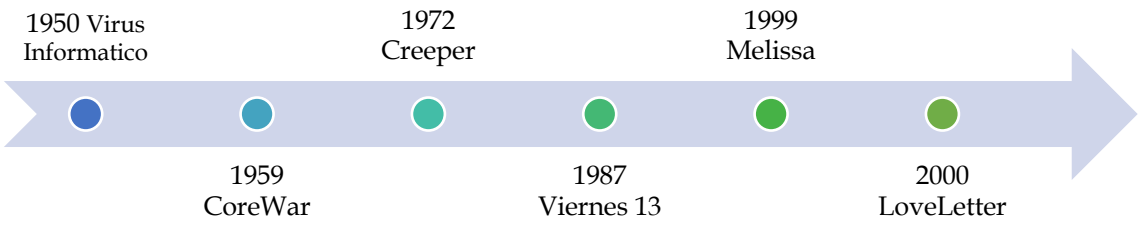


Figura 2.1: Historia de malware

En 1959 se desarrolla un juego en los laboratorios de Bell Computer llamado

CoreWar inspirado en la teoría de Von Neuman. El juego consistía en dos programas compitiendo por ver quien ocupa la memoria del oponente [9].

El programa llamado Creeper, desarrollado por Bob Thomas en 1972, fue considerado la primera implementación del concepto de Von Neumann y pudo expandirse a través de ARPARNET. Pero Creeper no era un malware porque no fue diseñado para hacer daño. Creeper puede ser considerado el padre de todos los gusanos y virus. Una década después Fred Cohen en 1984 desarrolló el primer malware que fue diseñado para obtener acceso a un ordenador [10].

El virus de Jerusalén también conocido como viernes 13, fue creado en Israel en 1987 para celebrar el 40 aniversario de la creación del Estado Judío. Obtuvo el nombre de viernes 13 debido a que se activaba cuando coincidiera en el calendario viernes y 13 [11].

Melissa fue unos de los primeros malware en utilizar ingeniería social mediante el envío de correos electrónicos. El mensaje contenía el siguiente texto: "aquí está el documento que me pediste no se lo enseñes a nadie" al cual iba asociado un fichero adjunto con extensión .doc. En el 2000 se dio a conocer el virus LoveLetter con características parecidas a Melissa, pero con una mayor complejidad, considerado perteneciente a las categorías de troyanos y gusanos. LoveLetter modificaba los ficheros del ordenador infectado y se transmitía a través del correo electrónico cuando el usuario abría los archivos [12].

2.3. Clasificación

Con el paso del tiempo la tecnología evoluciona y el malware debe adaptarse a los nuevos cambios. Estas adaptaciones generan variaciones de los mismo y crece su alcance. Existen diferentes tipos de malware, como se puede ver en la Figura 2.2 [13].

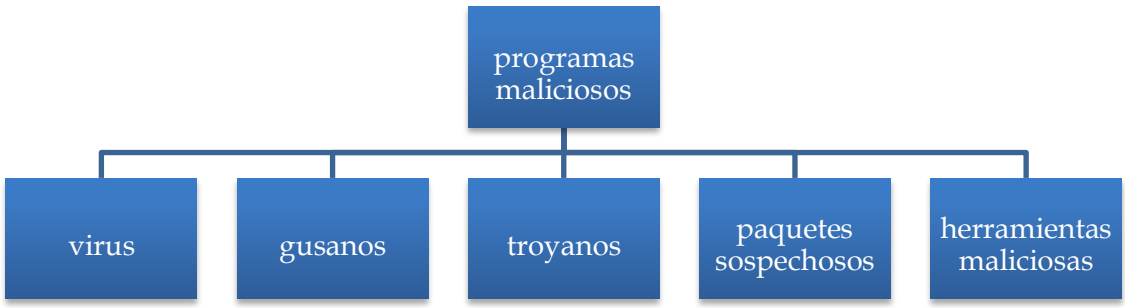


Figura 2.2: Clasificación de malware

- **Virus**: se replican en los recursos del ordenador. Son destructivos, infectan y toman el control de sistemas vulnerables.

- **Gusanos**: intenta obtener las direcciones IP de los equipos en la red. Ralentiza la red y bloquea las comunicaciones. Los gusanos a su vez se clasifican en:

  - ***Net-worn***: se propaga por la red en la que se encuentra el ordenador. No requiere ninguna acción del usuario para expandirse. Busca vulnerabilidades en el software.
  - ***P2P-worn:*** se expande vía peer-to-peer compartiendo archivos en la red.
  - ***IM-worn:*** se expande vía mensajería instantánea. Envía un enlace a una lista de contactos.

- **Troyanos**: efectúa acciones sin que el usuario se dé cuenta, obtienen o modifican datos y los envían a los cibercriminales. Los troyanos a su vez de dividen en:

  - ***Exploit***: aprovecha fallos de seguridad en las comunicaciones para entrar en el equipo.
  - ***Rootkits:*** son programas maliciosos que ocultan ciertos objetos o actividades en el sistema.
  - ***ransomware:*** cifra los datos para que no puedan ser utilizados por el usuario, hasta que se pague un rescate.

- **Paquetes sospechosos**: son archivos comprimidos que van acompañados de un archivo cifrado para evitar que sean detectados. Los principales paquetes sospechosos son:

  - ***Multipacked:*** son archivos comprimidos que utilizan varios tipos de compresión.
  - ***RarePacker:*** son archivos comprimidos que utilizan tipos de compresión muy raros.
  - ***SuspiciousPacker***: han sido diseñados específicamente para proteger a códigos maliciosos contra un antivirus.

- **Herramientas maliciosas**: son herramientas diseñadas para crear virus, gusanos, troyanos. Las herramientas más conocidas son:

  - ***DoS:*** son programas diseñados para enviar muchas peticiones a un ordenador remoto y denegar el servicio. Provoca pérdida de conectividad con la red por el consumo de ancho de banda.
  - ***Flooders:*** se usan para inundar la red con mensajes sin sentido. Esta herramienta puede ser utilizada por los spammers.
  - ***Spoofer:*** suplantación de la identidad a través de la falsificación de los datos en la comunicación.

Otra clasificación de centra en las que son desarrolladas y distribuidas por compañías legítimas, pero en ciertas ocasiones incluyen amenazas para los usuarios:

- **Riskware**: son programas legítimos que pueden ocasionar daño si son utilizados por usuarios maliciosos. Puede eliminar, bloquear, modificar o copiar datos sobre ordenadores o redes.

- **Pornware**: son programas que muestran material pornográfico en un dispositivo. Uno de sus propósitos es publicar sitios web y servicios pornográficos sujetos a tarifas.

- **Adware**: es un software publicitario, que abre ventanas emergentes que pueden ser peligrosas.

2.4. Análisis Forense de Malware

El análisis forense de malware proporciona la capacidad de analizar y comprender el funcionamiento del código malicioso (troyanos, virus, rootkits etc.) para poder evaluar los daños causados y valorar la intenciones y capacidades del atacante.

Conocer la estructura, funcionamiento e interacción del malware, aportará una valiosa información, no solo para el diseño y desarrollo de contramedidas eficaces, sino también para ayudar a conocer el origen de un ataque y evaluar la capacidad de detección de los sistemas de la organización, a la hora de tomar las acciones de respuesta necesarias y adecuadas. El grado de complejidad de las técnicas y el nivel de conocimiento necesario para analizar malware es proporcional al nivel de sofisticación de éste. Estas técnicas, conocidas como técnicas de análisis y reingeniería de malware, pretenden facilitar la adquisición de conocimiento sobre el mismo de una manera sistemática y metodológica.

Los beneficios que se obtendría de sus conocimientos son:

- Conocer el origen de un ataque e identificar al intruso.
- Evaluar la capacidad de detección de malware de los sistemas de protección de las organizaciones.
- Evaluar los daños causados por la intrusión y acciones del malware.
- Descubrir otras máquinas que han sido afectadas por el mismo malware.
- Identificar la vulnerabilidad que fue aprovechada por el malware, para obtener la actualización del software que la mitigue, si está disponible.

- Obtención de datos necesarios para poder implementar las defensas necesarias para mitigar y neutralizar los daños producidos por el malware particular analizado, por ejemplo, reglas de cortafuegos, de sistemas de detección de intrusiones tipo red y host y antivirus.

- Determinar el nivel de sofisticación y complejidad del malware.

Para alcanzar dichos beneficios se exige la implementación de un entorno de pruebas o laboratorio aislado de otras redes para poder asegurar la no propagación del malware o sus efectos a las redes de producción o incluso Internet. Este entorno debe disponer de la capacidad de realizar líneas base de la configuración de los equipos, clasificación y ejecución del malware, así como la recogida y el análisis de datos, además del análisis dinámico en un ambiente que simule su entorno real y el análisis estático de los ficheros malware.

Una metodología para el análisis de malware que se podría seguir seria la que se presenta en la Figura 2.3:

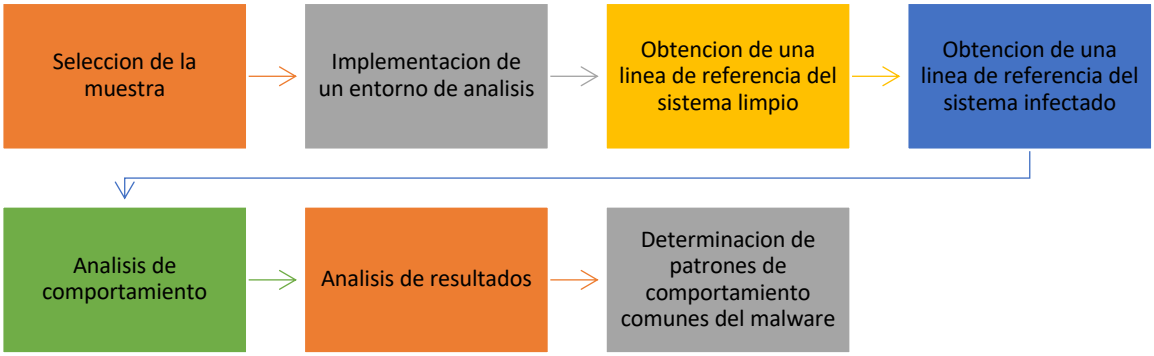


Figura 2.43: Metodología de análisis de malware.

Muchas veces solo se tiene la muestra del malware, teniendo que utilizar varias herramientas y realizar varias acciones para saber cómo es el funcionamiento del software malicioso.

Hay dos tipos de análisis que permiten a los analistas entender rápidamente y de forma detallada los riesgos y las intenciones de una muestra de malware. En la Figura 2.4 se muestra los diferentes tipos de análisis.

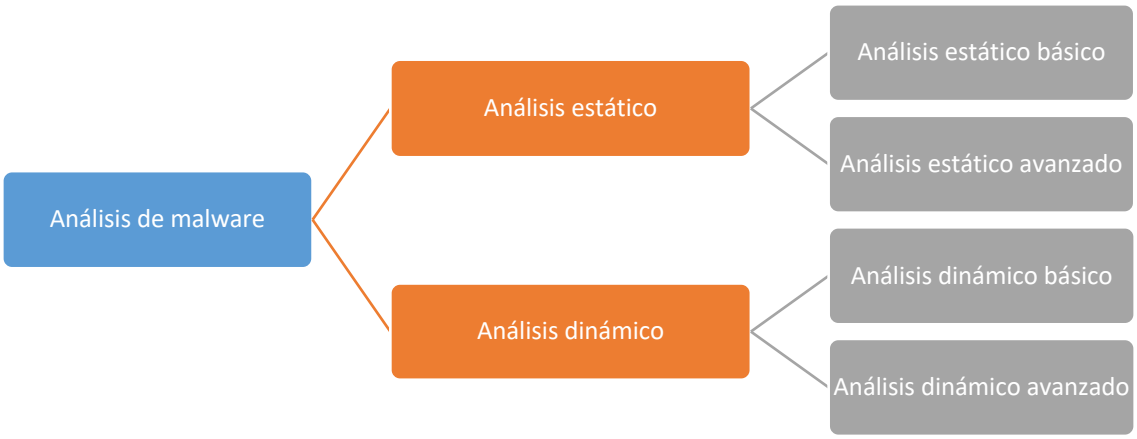


Figura 2.4: Tipos de análisis

- **Análisis estático:**
  - **Análisis estático básico:** se basa en el análisis del archivo sin llegar a ejecutarlo, lo que hace que sea un análisis rápido, pero no tan efectivo.
  - **Análisis estático avanzado:** consiste en realizar ingeniería inversa cargando el archivo ejecutable en un desensamblador y posteriormente analizar las instrucciones para determinar cuál es el funcionamiento del malware. Hacer este análisis requiere tener conocimiento de ensamblador, análisis de códigos y conceptos de sistemas operativos de Windows.
- **Análisis dinámico:**
  - **Análisis dinámico básico:** se ejecuta el archivo y se observa su comportamiento en el sistema, con el objetivo de saber qué acciones realiza. Este análisis se tiene que hacer en un ambiente controlado donde se vaya a ejecutar el malware.
  - **Análisis dinámico avanzado:** se utilizan depuradores para tener conocimiento de cómo se ejecuta el malware. Esta técnica es muy útil para obtener información.

2.4.1. Análisis Estático

Es una técnica que analiza el malware sin ejecutarlo. Para ello estudia su código ensamblador, dependencias y cabeceras entre otras cosas. Este análisis se realiza en dos etapas, la obtención de la muestra y el análisis del archivo. Como se puede observar en la Figura 2.5.

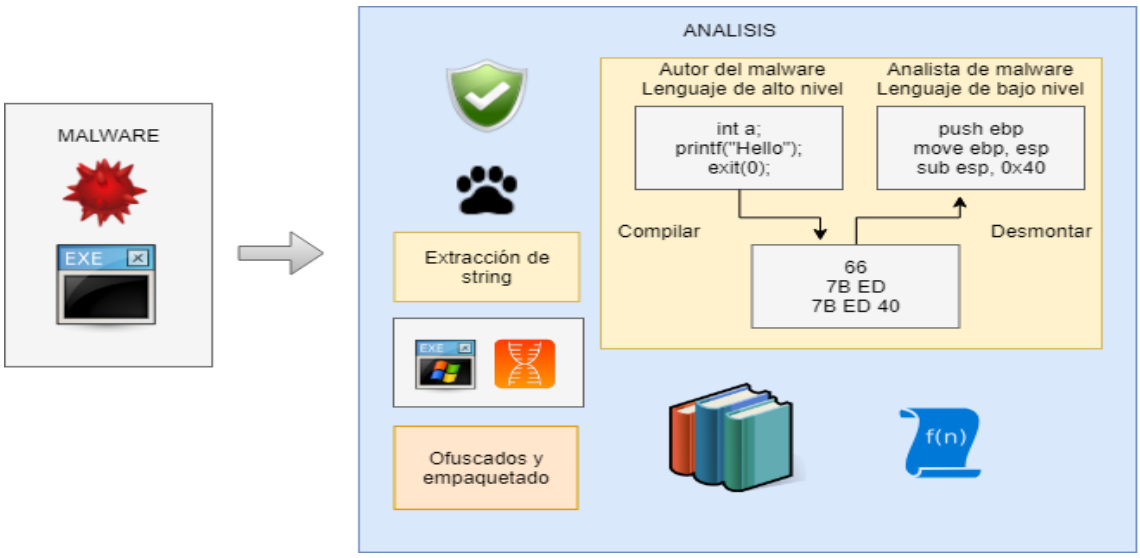


Figura 2.5: Etapas del análisis estático.

Los tipos de análisis estático son:

- **Escaneo con antivirus:** se puede analizar el archivo mediante un antivirus, ya que puede haber sido identificado como malware. Hay que tener en cuenta que no siempre es efectivo ya que pueden tener diferentes firmas, no estar en sus bases de datos o heurísticas que son incapaces de detectar nuevos malwares.

- **Huella dactilar:** los archivos tienen un hash que los identifican, se puede averiguar si ya han sido identificados por otros analistas previamente o mediante búsquedas online.

- **Búsqueda de cadenas de texto:** es interesante buscar cadenas de texto en el ejecutable, ya que podrían indicar algún tipo de mensaje, conexiones URL etc.

- **Ofuscado, empaquetado y cifrado:** el malware utiliza técnicas que dificultan su análisis y para ello aplican la ofuscación con el objetivo de pasar desapercibidos o el uso del empaquetado que utiliza la compresión de programas. O algunas veces cifran datos para que los analistas no puedan acceder y saber su cometido.

- **Estructura del archivo ejecutable:** los archivos tienen una estructura, la cual contiene información como: tipo de aplicación, librerías utilizadas, espacio requerido. Esta información es útil para el analista.

- **Listar librerías y funciones:** esto permite conocer a qué API del sistema operativo está llamando, lo que da una proximidad acerca del funcionamiento del fichero. Hay dos tipos de enlazado: el estático y el dinámico. Dependiendo del enlazado varía el análisis. Es de principal interés el enlazado dinámico para los analistas.

- **Ensamblado y desensamblado:** el malware suele ser escrito en un lenguaje de alto nivel, pero al compilarlo éste ya no es legible. Se desmonta para analizar su código ensamblador, con el objetivo de saber qué instrucciones utiliza y cuál es su objetivo. Después este código se puede poner en un lenguaje de más alto nivel para que sea más comprensible para el analista.

2.4.2. Análisis Dinámico

En esta técnica de análisis se examina la muestra durante su ejecución en un ambiente real o virtual. Permitiendo observar el comportamiento del objeto, como: el tráfico de red, procesos que se ejecutan, APIs usadas y los archivos que son creados o eliminados.

El análisis dinámico se puede dividir en tres fases: la primera es la obtención del malware que puede ser cualquier tipo de archivo o página web. La segunda fase es la preparación del entorno (puede ser un ordenador real, sandBox o una máquina virtual) donde será ejecutado el malware. La última fase, es el análisis

mediante herramientas, que permitan visualizar procesos, comparar registros, crear una falsa red, gestionar el tráfico de la red y utilizar depuradores para saber el funcionamiento del malware. Como se presenta en la Figura 2.6:

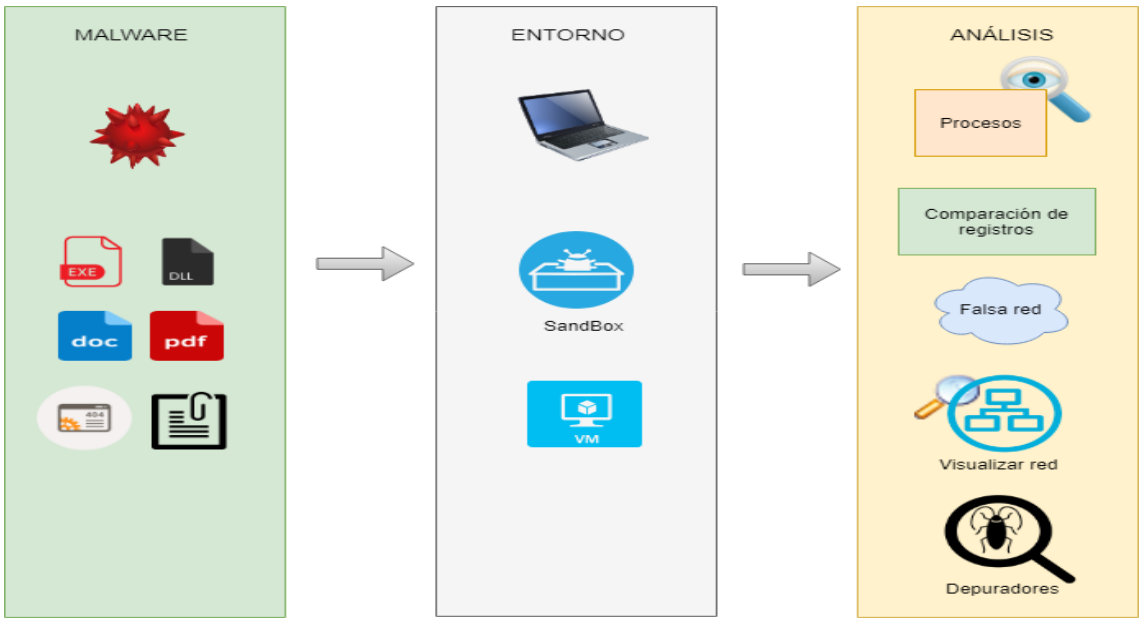


Figura 2.6: Fases del análisis dinámico.

Los pasos que se podrían seguir serían los siguientes ya que hay variedad de alternativas a la hora de hacer el análisis dinámico:

- **Preparación del entorno sanbox:** a través de este entorno se podrá ejecutar malware para conocer su comportamiento, pero no se exponen otros recursos de red, ni la máquina anfitriona. Además, se puede revertir a su estado original, para volver a ejecutar el malware desde una línea base.

- **Establecer la línea base de configuración del sistema:** define el estado original de la máquina, en un estado que esté libre de toda infección y guardar esa condición en un snapshot.

- **Preparación de Sniffer de red:** se configura un sniffer de red para capturar el tráfico de la maquina laboratorio.

- **Preparación de un servicio falso DNS:** con un servicio de falsificación de DNS

se puede hacer que todas las peticiones DNS devuelvan una IP determinada (normalmente la IP local) y se registran dichas peticiones, lo que permite tener un listado de los sitios a los que se conecta.

- **Guardar el estado del registro y el sistema de archivos:** se guarda el estado del registro y el sistema de ficheros, haciendo un hash a cada fichero, lo que permite comparar modificaciones tras ejecutar el malware.

- **Ejecutar el malware:** se ejecuta el malware para analizar su comportamiento.

- **Captura del tráfico de red con el malware en ejecución:** mientras el malware se ha ejecutado se captura el tráfico de red y se almacena en un fichero para analizarlo posteriormente.

- **Comparación del estado del sistema con la línea base:** se vuelve a capturar el estado del registro y el sistema de ficheros, para compararlo con el estado anterior y poder descubrir que ficheros o claves de registro se han modificado.

- **Identificación de conexiones realizadas:** se identifican las conexiones realizadas por el malware, ya sea por petición DNS o por acceso directo a una IP.

- **Capturar el tráfico desde una maquina host**: antes de ejecutar el malware se ha capturado el tráfico de red mediante un sniffer, desde la máquina virtual a la maquina anfitrión. Esto permitirá saber si el malware está intentando propagarse por la red anfitrión.

3. RANSOMWARE

El término ransomware viene de la unión de dos palabras "ransom" que significa rescate y la palabra "ware" que proviene del término software. Por tanto, ransomware no es más que un software malicioso que pide un rescate a cambio de devolver el control del dispositivo o de los datos que contiene el equipo infectado [16].

Cuando se es víctima de un ataque de ransomware, las opciones que se tienen son muy limitadas:

1. Pagar el rescate.
2. Restaurar los datos desde una copia de seguridad.
3. Formatear el dispositivo y perder los datos.
4. Intentar descifrar los datos.

Existen muchas versiones de este tipo de software malicioso, pero la mayoría de ellos ejecutan un cifrado de los datos del dispositivo, para restringir el acceso al usuario. Posteriormente piden un rescate económico para volver a recuperar la información. Sin embargo, el pago no se efectúa en la mayoría de los casos.

A pesar de tratarse de un software tan complejo, requiere de la acción del usuario para ejecutarse y tomar el control del dispositivo. De momento la forma más eficaz de evitar este tipo de software malicioso es la acción preventiva del usuario frente a este tipo de malware.

3.1. Evolución

Se presentan diferentes versiones sobre el inicio de ransomware. La manera que inicio, el dinero pedido como rescate para recuperar los datos y el lugar donde surgió.

Glassberg hizo referencia que ha existido por muchos años, pero no se refirió una fecha exacta [17]. Kharraz, Robertson, balzaroti, Bilge Kirda expusieron que pudo aparecer a partir de 2004, pero no fue más significativo hasta después de diez años más tarde [18]. Salvi and Kerkar indicó que apareció en 1989 donde un malware reemplazaba el archivo autoexec.bat por un archivo diferente. Explicaba que el ordenador estaría bloqueado hasta que se recibiera un pago de $189 dólares americanos [19]. Se puede concluir que empezó a tomar forma entre el 2012-2014.

Con respecto al origen, Glassberg sugiere que se origino en Rusia y el este de Europa [17]. O'Gorman y McDonald empezó en Rusia en 2009, y se extendió por Europa y estados unidos en 2010 [20].

Con la aparición de bitcoin, a principios de 2009, los cibercriminales han visto que el uso de técnicas de cifrado asimétrico junto con el cobro del rescate mediante bitcoin, resuelven los problemas de anonimato que en años anteriores no poseían [21].

En la Figura 3.1 se presenta las familias más representativas que han sido desarrolladas desde sus orígenes, así como su evolución en el tiempo [19]:

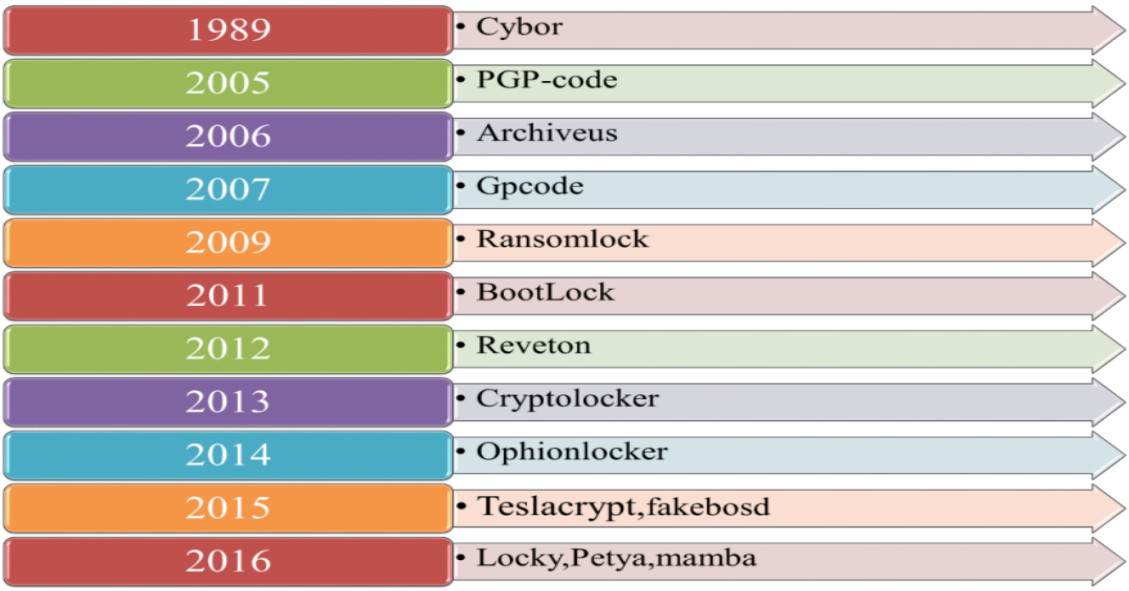


Figura 3.1: Ransomware a lo largo de la historia

- **Trojan archiveus**: cifra los archivos del usuario y se le exige que compre productos de una farmacia online a cambio de obtener la clave.

- **PGPCode**: es un troyano que cifra archivos y se propaga vía email como un fichero adjunto.

- **RansomLock**: al encender el ordenador muestra una ventana con el mensaje "bloqueando el acceso a los datos" y luego pide un rescate para desbloquearlo.

- **BootLocker**: cifra los datos del ordenador infectado e instala un bootlocker en el disco duro para que cuando el usuario encienda el ordenador aparezca un mensaje de que sus datos han sido cifrados.

- **Reveton**: es conocido como el troyano de la policía, se le muestra un mensaje que dice que ha violado alguna ley y seguidamente bloquea la pantalla con el mensaje.

- **Cryptolocker**: cifra archivos y pide un rescate. Derivó en una familia de ransomware como CryptDefense, TorrentLocker, CTB-Locker, Criptowall, Teslacrypt y AlphaCrypt. Utiliza cifrado asimétrico.

- **OphionLocker**: si el rescate no se paga en tres días la clave será eliminada. Utiliza la red anónima TOR y cobra en bitcoin.

- **Teslacryp**: ataca archivos relacionados con juegos de ordenadores como partidas de juego guardadas, perfiles de usuario.

- **Locky**: suele llegar como mensaje no deseado al correo electrónico con un fichero adjunto, normalmente un documento Word con macros que serán ejecutados cuando se habiliten, momento en que empezara la infección.

3.2. Características y Funcionamiento

Este software malicioso busca distintos tipos de archivos que considera importantes. Generalmente son ficheros de textos (PDF, DOC, etc.) los cifra y, a continuación, envía las instrucciones a seguir para recuperar los archivos cifrados por correo electrónico, mediante una ventana emergente o en el propio fondo de pantalla. La Figura 3.2 muestra las etapas del proceso de infección [20].

Figura 3.2: Etapas del Proceso de Infección

- **Distribución:** etapa donde el ransomware se trasmite por medio de internet a determinados objetivos gracias al uso de diversas técnicas de ingeniería social.

- **Infección:** estado en el cual el ransomware ha sido ejecutado o descargado en la máquina víctima.

- **Comunicación:** se establece intercambio de información entre el ordenador infectado y el cibercriminal.

- **Búsqueda de archivos:** busca archivos relevantes en el ordenador víctima con el objetivo de "secuestrar" dicha información.

- **Cifrar:** una vez se han obtenido lo archivos, se aplican diversas técnicas para hacer incomprensible dicha información.

- **Rescate:** el cibercriminal impone una recompensa al usuario víctima para poder recuperar su información.

La Figura 3.3, muestra una captura de pantalla de uno de los mensajes que un ransomware presenta a la víctima una vez que sus ficheros han sido cifrados.

Figura 3.3: Mensaje de ransomware

En el mensaje se puede leer lo siguiente:

- Pague $10,99 a través de western unión de lo contrario seguirán viendo esta pantalla.

- Se eliminará un archivo de su disco duro cada 30 minutos. Los archivos eliminados se restaurarán, cuando se haya pagado e introducido el código de desbloqueo adecuado.

3.2.1. Distribución

Una de las formas de propagación más común es a través del correo electrónico. Suele usar varias técnicas de ingeniería social para conseguir que el usuario confié en el mensaje y así conseguir su objetivo, la ejecución del software.

Algunas de las técnicas de ingeniería social más utilizadas son [22]:

- **Ficheros ejecutables con iconos**: es tan simple como poner la imagen de alguna aplicación familiar al usuario a un archivo ransomware y hacerle creer que es un archivo con una extensión conocida. Si el usuario no se percata de eso lo descarga y se ejecuta, entonces es cuando el software obtiene los permisos que necesita para acceder al sistema.

- **Ficheros ofimáticos con macros**: muchos atacantes solo crean documentos ofimáticos (documentos Word, Excel, etc.) y embeben código malicioso para ejecutar alguna acción dañina. La acción más común suele ser descargar y ejecutar un binario que permita el control remoto del ordenador.

- **Uso de carácter RLO (Right to Left Override)**: el carácter denominado RLO ha sido diseñado para soportar lenguajes escritos de derecha a izquierda como el hebreo. Los atacantes se han aprovechado del mismo para invertir el orden de visualización de los últimos caracteres que conforman el nombre de un fichero junto con su extensión dando así la impresión de que la extensión es de alguna aplicación conocida por el usuario. Por ejemplo, "Factura_2016_xcode.exe" se convertirá en "Factura_2016_exe.docx".

- **Suplantación de identidad**: En muchas ocasiones se utiliza él envió de correos electrónico con nombres de empresas conocidas, pero utilizando remitentes que no son pertenecientes a dichas empresas. Un caso se presentó en 2013, en el que el atacante suplanto la identidad de la empresa eléctrica Endesa, una empresa proveedora de electricidad. El usuario recibió un correo electrónico de Endesa, notificando una factura, al realizarse la descarga de la factura, el usuario no es consiente que en realidad está obteniendo un software malicioso. La parte interesante de esta nueva campaña ransomware es que la mayoría de los dominios que albergan los scripts maliciosos se basan en el popular CMS Joomla. Un ejemplo de correo electrónico malicioso recibido se muestra en la Figura 3.4.

Figura 3.4: Factura falsa de Endesa

En la figura se observa que el correo tiene un enlace con el literal "Consulta tu factura y consumo" que al acceder a él se redirige a una página en la que tras unos segundos de espera se descarga de un fichero .zip (comprimido) llamado endesa_factura.zip [23]. Como se puede apreciar en la Figura 3.5. Dentro hay un fichero de extensión JS (JavaScript) con el código malicioso.

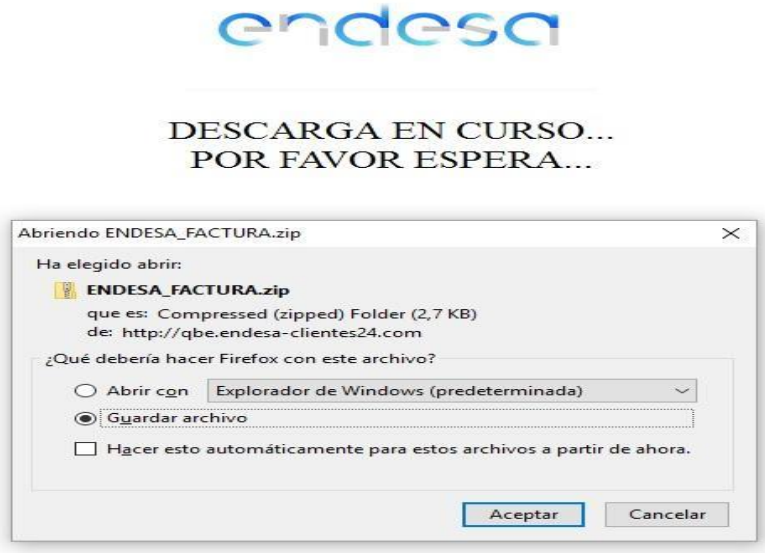


Figura 3.5: Solicitud de descarga de fichero

3.2.2. Cifrado

La etapa de cifrado es crucial dentro de este tipo de malware. Es la característica del ransomware que lo diferencia del resto de malwares. Utiliza dos tipos de cifrado [24]: simétrico y asimétrico.

3.2.2.1. Cifrado Simétrico

El cifrado simétrico utiliza la misma clave para cifrar y descifrar el mensaje. La clave debe ser conocida por el emisor y por el receptor. La comunicación de la clave entre emisor-receptor es el punto débil de este tipo de cifrado, debido a que en dicha comunicación la clave puede ser interceptada con facilidad. En la Figura 3.6 se puede observar el proceso de cifrado simétrico.

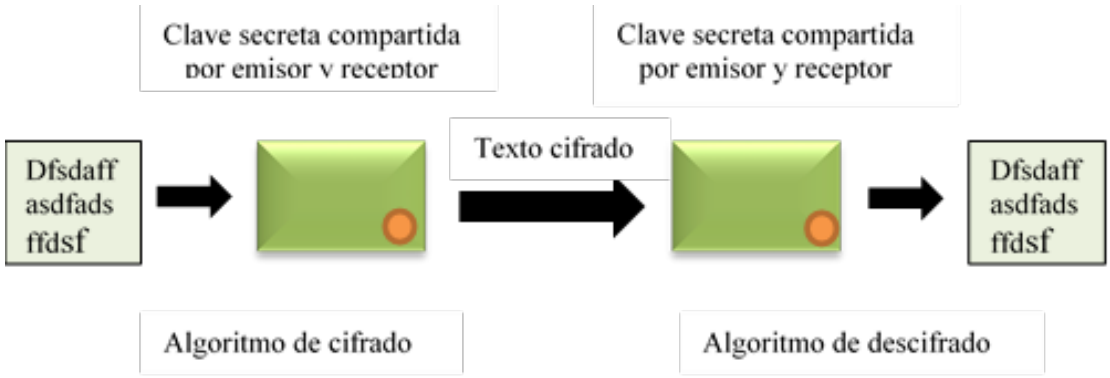


Figura 3.6: Cifrado simétrico

Dentro del cifrado simétrico los más utilizados son los siguientes algoritmos:

- **DES (Data Encryption Standard)**: este tipo de cifrado llegó en bloque a dominar los algoritmos de cifrado simétrico especialmente en aplicaciones financieras.

- **AES (Advanced Encryption Standard)**: es un cifrado simétrico en bloques creado con el objetivo de reemplazar al DES como cifrado estándar, para poder abarcar un campo más amplio de aplicaciones.

3.2.2.2. Cifrado Asimétrico

A diferencia del cifrado simétrico el cifrado asimétrico, se basa en el uso de dos claves. Una clave pública que será la que se envié a quien quiera enviar información cifrada y una clave privada que nunca se debe difundir y sirve para descifrar el mensaje. El descubrimiento de este cifrado originó una revolución en la historia de la criptografía. Se puede apreciar su funcionamiento en la Figura 3.7.

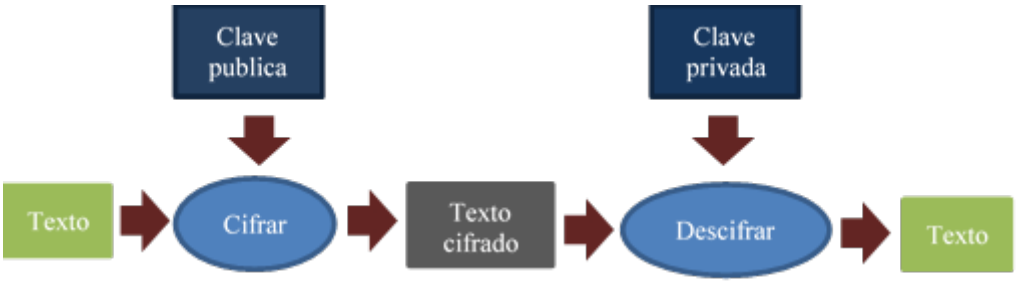


Figura 3.7: Cifrado asimétrico

Se considera que el cifrado asimétrico aumenta la confidencialidad y la autenticidad de la información al ser casi imposible obtener la clave privada, aunque sea interceptada la clave pública. El cifrado más utilizado por el ransomware es el cifrado asimétrico debido al poder que le da sobre su víctima.

3.2.3. Estrategias de protección utilizadas por los ransomware

A la hora de analizar este tipo de malware, los cibercriminales usan técnicas para que sea más difícil comprender su funcionamiento. Por ejemplo, cuando se lanza una muestra de ransomware en un entorno controlado, estos softwares maliciosos saben interpretar que están siendo analizados y modifican su comportamiento. En los siguientes apartados se explican algunas técnicas utilizadas.

3.2.3.1. Anti-Debugging

Son técnicas utilizadas por un software malicioso cuando están bajo el control de un depurador. Utilizan esta técnica para que los analistas de malware tarden más tiempo en analizar su comportamiento. El malware al saber que está siendo analizado por un depurador, puede cambiar su modo de ejecución normal o modificar el código para causar algún daño, interfiriendo con el análisis. Estas técnicas son [14] :

- **Detectar e identificar un depurador en Windows**: los malware utilizan técnicas para no permitir ser analizados por un depurador. Estas técnicas son: Windows APIs, comprobación de estructuras, buscar en el sistema residuos del depurador. Los analistas ponen puntos de ruptura para saber su comportamiento y analizar su funcionamiento. Cuando el malware detecta la ejecución de un depurador este cambia su comportamiento.
- **Interferir con el funcionamiento de los depuradores**: estas técnicas tratan de interrumpir el funcionamiento de los depuradores solamente si se sabe que se está ejecutando uno. Hay diferentes técnicas para hacer interferir el depurador como son: retornos TLS, excepciones e interrumpir inserciones.
- **Explotar vulnerabilidades del depurador:** el malware explota las vulnerabilidades del depurador para prevenir ser depurado.

3.2.3.2. Anti-Maquina-Virtuales

Son técnicas que son utilizadas por el malware para saber si están siendo ejecutados en una máquina virtual. Si el malware detecta que está siendo ejecutado en una máquina virtual puede cambiar su manera de actuar o inclusive no se ejecuta al detectar el análisis. Las técnicas más utilizadas son [14]:

- **VMware artefactos**: al ejecutar una máquina virtual se generan varios artefactos, el malware puede averiguar si éstos están presentes en el sistema para saber si se ejecuta está en una maquina real o en un entorno virtual.

- **Usar la píldora roja anti-VM**: consiste en comparar un byte determinado con 0xFF para detectar la firma de MVware. Esta técnica solo funciona con un procesador.
- **No usar la píldora**: Windows no utiliza la LDT en cambio la máquina virtual si da soporte. El LDT sobre el S.O anfitrión será cero en cambio en la máquina virtual será distinto de cero.
- **Puerto de comunicación I/O**: comparar un número mágico con el de un puerto de comunicación para identificar el uso de la máquina virtual.
- **Vulnerabilidades de las máquinas virtuales**: las vulnerabilidades de las máquinas virtuales pueden hacer que el sistema anfitrión se detenga o inclusive ejecute un código en él. Muchas vulnerabilidades fueron encontradas en las carpetas compartidas o en las herramientas de arrastrar y soltar.

3.3. Clasificación

El ransomware se puede clasificar según las acciones que efectúa en los dispositivos. Son [25]:

- **Ransomware que cifran**: cifra archivos y carpetas. El usuario se percata cuando trata de abrir los archivos y no puede.

- **Ransomware bloqueantes**: bloquea la pantalla del ordenador mediante un mensaje y pide un pago. Los archivos no son cifrados.

- **Ransomware master boot record (MBR)**: es la parte del disco duro donde se encuentra el arranque del sistema operativo, el ransomware cambia el estado de inicio mostrando otro tipo de mensaje.

- **Ransomware de servidores web**: tienen como objetivo atacar servidores web y cifrar sus archivos.

- **Ransomware de teléfonos móviles**: Suelen estar en las aplicaciones que se descargan.

3.4. Familias de Ransomware

En todas las familias de ransomware el comportamiento es similar, la diferencia está en la forma de pago y las acciones que realiza en la victima. El crecimiento del ransomware, se debe a que los autores usan polimorfismo creando variaciones de cada tipo. Hay más de 50 familias en circulación.

Las 10 familias más populares actualmente son (ver Figura 3.8) [26]:

- **Tescryp**: cifra los archivos del ordenador y dirige al usuario a una página con instrucciones para saber cómo desbloquearlos. El pago se hace mediante bitcoin. Se propaga vía webs comprometidas, utilizando angler exploit kit para encontrar vulnerabilidades y poder ejecutar su software. Cifra documentos de texto, pdf, archivos relacionados con videojuegos con AES.

- **Crowti**: infecta a través de documentos adjuntos. Cifra los archivos y luego cobra un rescate por ellos. Los archivos no son renombrados durante el cifrado.

- **Fakebsod**: bloquea el buscador web mostrando un mensaje para llamar a un número. No deja cerrar el navegador web.

- **Brolo**: bloquea el navegador web mostrando un mensaje de cobro de una multa.

- **Locky**: no permite el uso del PC o acceder a sus datos. El usuario ve un mensaje que contiene información acerca de abonar un pago a un hacker. El ciberdelincuente infecta un archivo ofimático, después lo envía al usuario y al descargarlo se instala ransomware. Este suele llegar como spam al email normalmente en un documento Word.

- **Teerac**: cifra archivos y se muestra una página web que pide una cuota para desbloquear los archivos. Este puede ser instalado en el PC por otro malware.

- **Critroni**: bloquea archivos inutilizándolos. Después muestra un mensaje que solicita dinero para desbloquear los archivos. Se descarga por spam o alertas falsas de actualizaciones gratuitas como java o flash player. El pago se hace mediante la red Tor.

- **Reveton**: bloquea el ordenador y muestra un mensaje en toda su pantalla. Se hace pasar por el FBI o la policía, para intimidar al usuario y convencer de pagar multa para desbloquearlo. Se infecta típicamente visitando una página hackeada.

- **Cerber**: este es un ransomware como servicio y se propaga a través de emails, exploit kits o documentos adjuntos. Utiliza los algoritmos RSA y RC4 para cifrar. El mensaje es mostrado en una página web o documento plano.

- **Exxroute**: se instala en el ordenador mediante descargas, enlaces a webs. Bloquea el acceso al ordenador y a los archivos. Se puede pagar mediante MoneyGram, Ukase, MoneyPak y bitcoin.

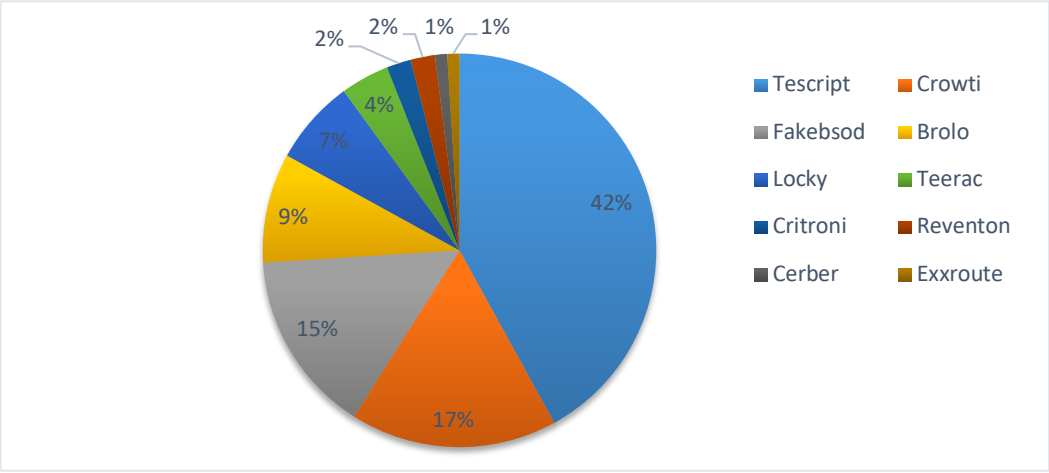


Figura 3.8: Top 10 de familias de ransomware

3.5. Ransomware Como Servicio

El ransomware como servicio nace a partir de su masificación para obtener dinero fácil sin casi enfrentar consecuencias gracias al anonimato que brindan las

distintas formas de cobro y la propagación que utilizan.

Los desarrolladores de estas plataformas son organizaciones criminales que proveen servicios de desarrollo de ransomware que pueden ser usados por usuarios sin conocimientos técnicos. El comprador del malware será el encargado de difundirlo y obtendrá beneficios económicos mediante el pago del rescate de las víctimas. Un porcentaje de ese beneficio económico es para los desarrolladores de dicho ransomware.

En la Figura 3.9 se puede observar el formulario de compra de un ransomware en una página web de la dark net [27].

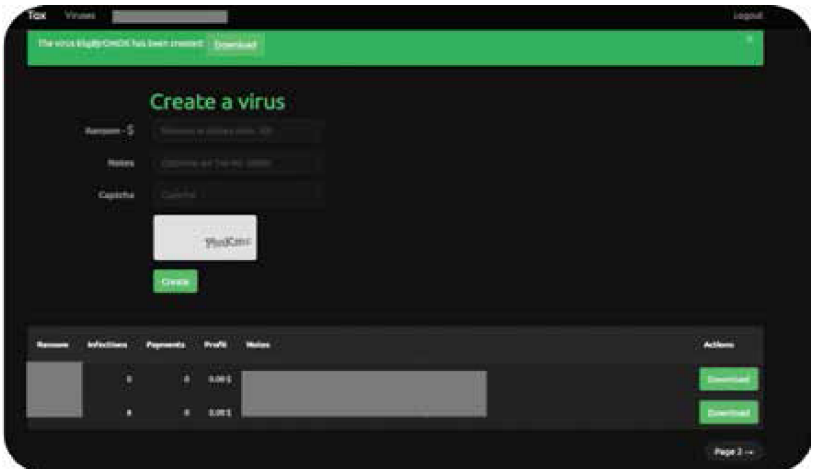


Figura 3.9: Kit de creación de ransomware tox

Los criminales han desarrollado un modelo de negocio muy rentable y sin mayores riesgos, puesto que el usuario a cargo de la propagación (el comprador) es quien, en primera instancia, se considera responsable.

Algunas de las configuraciones que brinda el software malicioso adquirido en estas plataformas criminales son [28]:

- Tiempo que tarda en aparecer la pantalla bloqueada.
- Capacidad de CPU.
- Tiempo de espera hasta que el servidor despierta al ransomware.

- Orden de aparición de la pantalla bloqueada (antes o después de cifrar).

El usuario interesado en obtener este tipo de servicio puede ver toda la información acerca de las ganancias que puede llegar a conseguir de ese ransomware y así poder elegir el que mejor se adapte a sus intereses (ver Figura 3.10). Los cibercriminales por lo general eligen atacar organizaciones críticas como bancos, hospitales y agencias de gobierno.

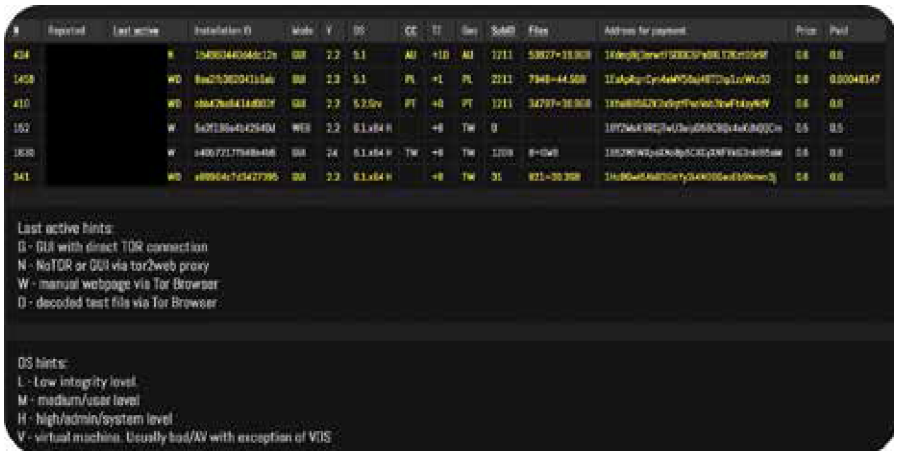


Figura 3.10: Tox información

4. ESTADO DEL ARTE

Se ha realizado un análisis de la literatura existente relacionada con las dos grandes temáticas del trabajo:

a) Técnicas de extracción de información de malware.
b) Técnicas de clasificación de malware.

A continuación, se presentan las investigaciones más relevantes.

En [29] los autores presentan una herramienta detectora de ransomware, llamada GreatEatlon. GreatEatlon utiliza técnicas de análisis estático para reconocer abusos de administración de dispositivos móviles y extrae información de flujo de datos necesaria para detectar usos maliciosos de las APIs criptográficas.

La herramienta Heldroid analiza las tres características principales de un ransomware: compuesto por un texto, cifrado, y analizador de bloqueo [30]. Se mejoran varios aspectos:

- En la herramienta de análisis estático, en la que basa el detector de cifrado.
- Se aumentaron las fuentes de obtención de muestras para no depender de una carpeta local.
- Se mejoró la heurística en el método de invocación a través de patrones de reflexión.
- Por último, se implementó un pre filtró para reducir el consumo de recursos innecesarios.

Se lleva a cabo un procesamiento de imagen para extraer texto. En particular se añadió un escaneo de imágenes en los recursos de la aplicación y aplica una transformación para optimizar el reconocimiento óptico de caracteres. El texto extraído es corregido automáticamente por un comprobador de deletreado, para quitar posibles errores durante el reconocimiento óptico de caracteres.

Seguidamente, el texto corregido es procesado con un analizador de texto, que consulta un clasificador, como resultado devuelve un valor si este es muy superior a un umbral establecido previamente indica entonces que el texto es considerado peligroso, indicando probablemente que es un ransomware.

Los experimentos realizados analizan calidad de la imagen escaneada, probaron un dataset de imágenes amenazantes. La herramienta extrajo y clasificó el texto como amenazante dentro de un grupo de imágenes con un tipo de letras originales. Se hicieron pruebas con otro tipo de letras y se extrajeron los símbolos simples, pero falló al reconocer símbolos más complejos. Sin embargo, se concluyó que las fuentes delgadas, escritas a mano o de otro tipo, son más difíciles de leer para los usuarios por lo que los atacantes suelen utilizar fuentes legibles lo que permite que la herramienta se comporte de la manera esperada.

En [31] los autores desarrollan un framework modular llamado BitIodine, para analizar el blockchain (en español cadena de bloques) del bitcoin. Esta herramienta agrupa direcciones pertenecientes a un mismo usuario o grupo de usuarios, además de clasificarlos y etiquetarlos. Posteriormente, se visualiza la información extraída de la red de bitcoin. El diseño de la herramienta se basa en varios bloques: bloque de agrupación, bloque de rastreo, bloque de grafos, bloque clasificador y bloque exportador. La herramienta desarrollada ha sido probada en casos reales para averiguar la relación entre dos direcciones pertenecientes a una misma persona. Otra utilidad fue consultar una transacción hecha en un día con un valor en específico, devolviendo la dirección bitcoin. El último experimento analizó el ransomware Criptolocker en el que cuantificaron en número de rescates pagados y la información recolectada de la víctima. Se recogen varias direcciones pertenecientes a Criptolocker. Dichas direcciones se introducen en la herramienta y se encuentra que pertenecen a varios grupos de usuarios, que tienen 2118 direcciones del que se identifican 771 rescates.

En [32] se desarrolló una aplicación basada en Volatility framework con el objetivo de automatizar la herramienta de análisis de la memoria RAM y detectar

la presencia de un posible malware generando un informe. La herramienta tiene como argumento la ruta donde está el volcado de memoria para su ejecución. Primero, busca la información de la imagen que detalla las características del sistema operativo y su arquitectura. Volatility llama a esto profile y es obligatorio para ejecutar cualquier comando en la herramienta. Después de tener el profile, el siguiente paso es analizar el malware. El funcionamiento de la herramienta es el siguiente:

En primer lugar, comprueba las conexiones abiertas en el momento que fue extraída la memoria volátil. Los detalles de las conexiones abiertas son muy útiles en el análisis de malware que son basados en redes. La mayoría de estos malware necesitan conectarse al centro de comando y control para ejecutar la siguiente orden o para enviar información específica, como detalles de contraseñas o archivos. De los pasos anteriores, se obtiene una lista de los procesos que se comunican con una dirección IP y los puertos usados para la comunicación. Si se encuentra alguna conexión abierta se recupera fácilmente el identificador del proceso, que inicio la comunicación. Hay posibilidades de que ese proceso pueda no ser legítimo y pueda realizar actividades maliciosas. Es importante comprender que puede ser difícil para los investigadores comprobar y analizar cada proceso ejecutado, ya que cada uno puede conectarse a una dirección IP para confirmar si es un malware o no. A continuación, la herramienta descarga automáticamente el archivo ejecutable que inició la comunicación. Por último, la herramienta comprueba si el archivo ejecutable puede ser un malware. Para ello, lo envía a VirusTotal para comprobar si es un malware o no. Con el resultado de VirusTotal ForMaLity pide un reporte del análisis y lo almacena en un archivo.

En [33] se llevó a cabo el estudio de las metodologías de análisis de malware existentes enmarcadas en 2 técnicas de análisis. Se aplicaron y complementaron dichas metodologías para realizar un análisis orientado al malware del tipo ransomware. Se destaca la comparación de una maquina infectada con el estado de una maquina limpia, permitiendo de esta manera establecer patrones comunes de comportamiento en equipos infectados con malware del tipo

ransomware. Se estudian las metodologías MARE y SANS con el objetivo de unificar y complementar dichas tecnologías para obtener una más eficiente.

Se realizó el análisis de tres muestras de malware de tipo ransomware, en entornos controlados con distribuciones distintas de Windows. En la etapa de análisis de las muestras el autor utilizo aplicaciones web o diferentes herramientas que se ejecutan en un ordenador. Las aplicaciones de análisis de malware en la web utilizadas son las siguientes, Payload Security, VirusTotal y Malwr. Las herramientas locales que se utilizaron fueron seis, SysInspector, Process Explorer, Process Monitor, RegShot, Disk Pulse, md5deep. Finalmente, concluyó que el porcentaje de uso del disco se ve incrementado cuando el sistema está bajo la influencia de ransomware. La muestra Zerolocker utiliza un proceso legítimo del sistema cipher.exe. La variación de paquetes enviados y recibidos de la línea base con respecto a un equipo infectado, no significa que necesariamente exista un centro de comando y control (C&C).

En [34] se especifica la creación de un entorno virtual, con el objetivo de realizar un análisis dinámico de distintas muestras de malware ejecutándose sobre Windows 7 para entender el ciclo de vida del malware. En este trabajo se han utilizado las siguientes herramientas: VMware Workstation, Cuckoo Sandbox y Wireshark. VMware Workstation es la herramienta utilizada para crear el entorno virtual, necesario para permitir el análisis del malware de forma controlada. Cuckoo Sandbox fue utilizado para realizar el análisis dinámico del malware, para efectuar su análisis de forma automática, disminuyendo mucho el trabajo, el tiempo empleado y la complejidad que conlleva el análisis dinámico. Wireshark es utilizado por el autor para identificar los servidores a los que el malware se conecta a través de peticiones DNS, así como las peticiones realizadas por el malware a servidores remotos. Se concluye que los resultados de los análisis del Cuckoo Sandbox no son definitivos, necesitando otras herramientas para constatar la veracidad de los resultados y así poder descartar falsos positivos.

En [35] se analizaran las características estáticas que se pueden obtener de los ficheros ejecutables para realizar una clasificación de los malware de MALICIA que aumentara su índice de clasificación actual. Se utilizaron varias herramientas como PEID para ver el detalle de las características internas de un fichero ejecutable, así como herramientas en línea como VirusTotal y TotalHash que se obtienen de las muestras de los malware los hashes ImpHash y Pehash. Con estas características estáticas de los PE se intentaba mejorar el índice de clasificación de MALICIA. Como resultado se obtuvo una mejora de un 10,76% en la clasificación (paso de un 87,24% a un 98%). La métrica F-Measure agrupa la precisión y e-recall permitió identificar la técnica de clasificación con mejores resultados. Se concluyó que el clustering de referencia era el más adecuado.

En [36] se demostró que el control de versiones en la nube es capaz de revertir eficazmente los efectos del cifrado sin tener que recurrir al pago del rescate exigido. Se probó con un tipo de ransomware Hidden Tear que el control de versiones en una estación remota es una herramienta útil, para evitar pagar el rescate de los archivos. Se hizo una prueba con una carpeta con archivos, sincronizada con Dropbox y se observó que el ransomware cifro todos los archivos de la carpeta. Sin embargo, mediante el control de versiones se pudo recuperar los archivos de la carpeta. Se concluyó que un sistema de almacenamiento y control de versiones externo que realice copias de seguridad en tiempo real es una alternativa de defensa ante el ransomware.

En [37] se realiza un análisis dinámico de seis tipos de malware: troyanos, gusanos, virus, troyanos espía, puertas traseras y rootkits. Se empleó el modelo de lenguaje de n-gramas en las llamadas al sistema de tipo WinAPI para cada muestra. Se utilizó el método de máquina de soporte vectorial (SVM) con kernel polinomial como algoritmo de aprendizaje automatizado para predecir la clasificación de malware. Se realizaron experimentos con 900 muestras de 6 diferentes tipos de malware considerando diferentes familias de cada tipo. Las herramientas utilizadas fueron: WinAPI, Cuckoo SandBox, N-gramas, máquina de soporte vectorial, Weka. Como parte del pre procesamiento de los datos se

obtuvieron los 5-gramas por cada muestra de los seis tipos de malware incluyendo el whiteware. Se obtuvieron 85,013 5-gramas, de los cuales, 63,204 son únicos. Con estos últimos se generó una tabla binaria en representación de la presencia de cada 5-grama para cada muestra de clase. Dicha tabla sirvió como entrada para el algoritmo de clasificación SVM. Se concluyó que los gusanos con un mayor número de 5-gramas, tienen una entropía muy alta. Como consecuencia, tienen procesos con mayor cantidad de llamadas del sistema haciendo más difícil su detección. Análogamente los Rootkits con el menor número de 5-gramas indica que la variación entre cada muestra de este tipo es muy pequeña con respecto a sus 5-gramas, haciéndolos más fácil de detectar.

5. TÉCNICA DE EXTRACCIÓN AUTOMÁTICA DE INFORMACIÓN DE MALWARE

El proceso se lleva a cabo a partir de la obtención de la captura de pantalla, a continuación, se hace el recorte de la ventana emergente, sobre el cuál se realiza el reconocimiento de patrones, además se lleva a cabo el OCR y, por último, siempre se realiza la búsqueda de archivos junto con el volcado de memoria.

5.1. Herramientas Forenses para la Extracción de Información.

En esta sección se presenta el análisis de las herramientas y las técnicas forenses utilizadas por los analistas para la extracción de información valiosa del dispositivo infectado en un ataque informático con malware.

5.1.1. Herramientas de Volcado de Memoria

La memoria volátil es una parte fundamental dentro de una investigación forense ya que posee información sobre el estado de la máquina en el momento de la infección. Es una manera de observar posibles indicios de la presencia de un malware. Las herramientas más utilizadas de volcado de memoria son:

- **Dumpit**: herramienta desarrollada por Monsol, extrae los datos que se encuentran en memoria RAM, en la información extraída están procesos, registros etc. Es una herramienta muy sencilla de ejecutar que se mantiene en continuo desarrollo. Su última versión Dumpit 2016. Su ejecución genera dos archivos, uno es archivo con extensión .dmp que contiene el volcado de la memoria y el otro con extensión .json que contiene información acerca de la arquitectura de la máquina anfitriona. Esta información es necesaria para que herramientas de análisis como Volatility conozcan la arquitectura del ordenador. Dumpit funciona para la mayoría de las versiones de Windows se ejecuta por consola de comandos, sin embargo permite automatizar el proceso

sin que el usuario interfiera.

- **RamCapturer**: están disponibles compilaciones separadas de 32 y 64 bits. Los volcados de memoria capturados con esta herramienta se puede analizar con otra herramienta de la misma empresa. Es compatible con todas las versiones y ediciones de Windows, incluidos XP, Windows 7, 8, 10, 2003 y 2008 server. La herramienta no requiere instalación. Esta herramienta genera una ventana, para ejecutar el volcado haciendo que intervenga el usuario. Extrae el volcado inclusive si está protegido por un sistema activo anti-depuración o anti-dumping.

- **Winpmem**: es una herramienta que forma parte de un framework llamado recall de código abierto que sirve para la extracción de la memoria volátil. Se puede encontrar en GitHub, está desarrollada en Python. Funciona en arquitecturas x86 y x64. Puede generar un archivo con formato crash dump (dmp) o raw dump (raw).

- **FTK imager**: es una herramienta muy utilizada tanto para la extracción de la memoria volátil como para el análisis de memoria. Tiene un entorno grafico que permite su uso de una manera más comprensible para el usuario.

5.1.2. Técnicas de Procesamiento de Imágenes

Los algoritmos más utilizados para la extracción de puntos de interés invariantes son los siguientes: SIFT y SURF.

Se trata de algoritmos de visión artificial encargados de extraer características distintivas de las imágenes, permitiendo encontrar puntos de coincidencia de una imagen en otra.

Las principales diferencias entre estos algoritmos son:

- El algoritmo de SURF es ligeramente más rápido que SIFT a la hora de detectar puntos de interés.

- SIFT supera a SURF en el número de características detectadas extrayendo el doble de puntos de interés.

- SURF calcula un vector descriptor para una determinada posición (x, y) a diferencia de SIFT que puede calcular varios descriptores para una misma posición de la imagen.

- SURF detecta un mayor número de falsos positivos con respecto a SIFT.

De cara a la detección de patrones dentro de una imagen, se prima el poder obtener la mayor cantidad de puntos de interés, así como la mayor información de cada descriptor, frente a la velocidad de su detección. Concluyendo en que el algoritmo que más se adapta a las necesidades de la herramienta es SIFT.

5.1.3. Técnicas de Reconocimiento Óptico de Caracteres

La tecnología de reconocimiento óptico de caracteres, OCR engloba a un conjunto de técnicas basadas en estadísticas, en las formas de los caracteres, transformadas y en comparaciones, que, complementándose entre sí, se emplean para distinguir de forma automática entre los diferentes caracteres alfanuméricos existentes.

Hay tres tipos de funciones que se podrían utilizar en Python para el reconocimiento óptico de caracteres.

- Textract: Si bien existen varios paquetes para extraer el contenido de varios formatos por su cuenta, este paquete proporciona una interfaz única para extraer contenido de cualquier tipo de archivo, sin ninguna marca irrelevante.
- Pytesseract: Reconoce y lee texto embebido en una imagen. Puede leer todo tipo de imágenes. Teeseract funciona mejor cuando hay segmentaciones extremadamente limpias del texto de fondo. Múltiples sistemas operativos.
- Pyocr**:** Es un contenedor de herramientas de reconocimiento óptico de caracteres. Está diseñada para sistemas GNU/Linux. Puede o no funcionar en Windows, MacOSX.

De las tres herramientasse ha seleccionado Pytesseract, porque es el más utilizado. Para llevar a cabo esta funcionalidad, se hizo uso de librerías como PIL para la manipulación de imágenes, además se descargó e instalo la herramienta tesseract-ocr, seguido de la librería Pytesser, que permite a Python trabajar con dicha herramienta.

5.2. Técnica Forense de Extracción Automática de Información de Ransomware.

Las herramientas y técnicas presentadas en la sección 5.1 aportan por separado información en la investigación de un ataque informático. Este trabajo combina dichas técnicas para extraer de forma automática información lo más detallada posible del ransomware involucrado en un ataque informático.

La forma más habitual en la que el cibercriminal informa a un usuario que su dispositivo es víctima de un ataque con ransomware y, que sus datos están secuestrados es a través de ventanas emergentes. La ventana muestra un mensaje en el que se indica que sus datos han sido secuestrados.

En términos generales, la técnica propuesta se compone de tres fases:

1. Extraer información de la ventana emergente del ransomware desde el dispositivo infectado.
2. Realizar una búsqueda de ficheros relacionados con el ataque.
3. Realizar un volcado de la memoria RAM del dispositivo infectado.

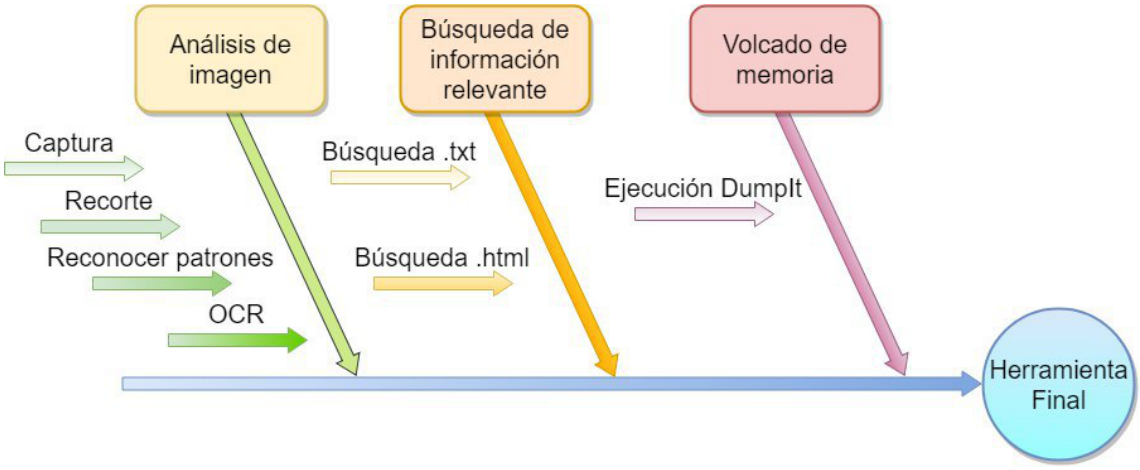


Figura 5.1: Fases de procesamiento de la técnica propuesta

Como se observa en el diagrama de procesos de la Figura 5.1, la primera fase del proceso es identificar la ventana emergente desde una imagen capturada de la pantalla del dispositivo víctima. Dicha captura se realiza directamente desde un dispositivo USB auto ejecutable y se almacena en un fichero con formato PNG para su análisis.

A continuación, se procede a analizar el contenido de la imagen resultante para detectar la ventana emergente del ransomware. Para realizar este proceso, se utilizan técnicas de reconocimiento de patrones para identificar si la ventana pertenece o no a una de las muestras de ransomware almacenadas previamente. Los análisis de las capturas de pantalla se realizan con técnicas de procesamiento de imágenes. Las etapas de análisis de la imagen se pueden ver en la Figura 5.2.

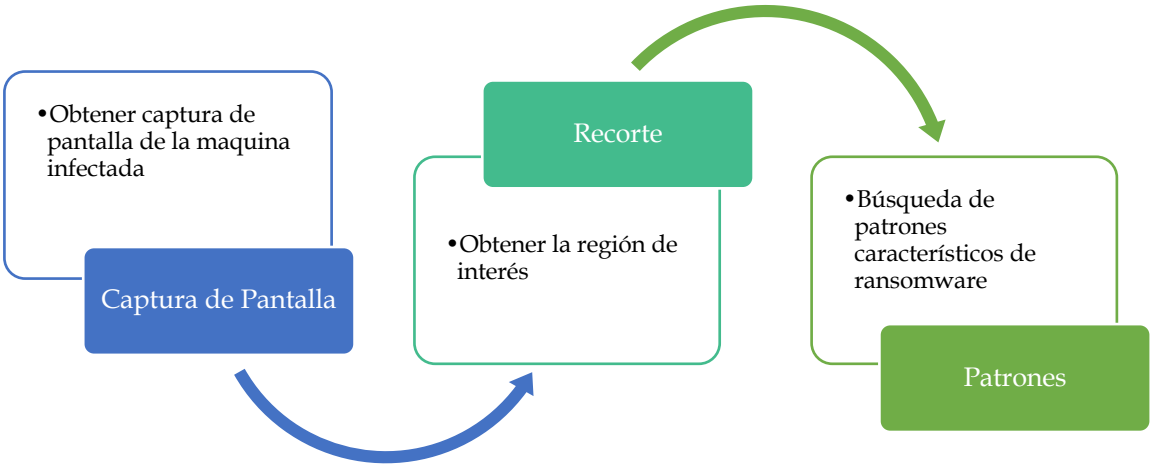


Figura 5.2: Etapas del Análisis de Imagen

A la imagen resultante de la captura se le realiza un proceso de reconocimiento de bordes que detecta los cambios en la intensidad de luz en la imagen. Estos cambios se pueden usar para determinar la profundidad, tamaño, orientación y propiedades de la superficies en una muestra [38]. Se utiliza un método del valor umbral para la separación de regiones de la imagen, en este caso Threshold de la librería OpenCV y FindContours como método de aproximación de contornos. Como resultado de este proceso se obtiene una imagen con los contornos identificados y marcados sobre la que se realiza una búsqueda de figuras rectangulares.

Una vez se ha reconocido el rectángulo, con el área más significativa, se realiza el recorte en la imagen original.

A continuación, se utiliza la extracción de descriptores SIFT para encontrar los puntos de coincidencias de cada patrón ransomware en la imagen extraída. Para obtener los puntos de coincidencias de cada muestra, se utilizan 62 patrones de diferentes ransomware.

Después de varias pruebas realizadas, se estableció un umbral de puntos de coincidencias a superar; en este caso 35 coincidencias por cada patrón dentro de la imagen extraída.

El número de puntos de coincidencia obtenidos por cada patrón se almacena en una variable acumulativa, además de llevar un contador con el número de patrones que han superado el umbral de puntos de coincidencias.

Para que la imagen sea tratada como válida, la variable acumulativa deberá tener como mínimo 500 puntos de coincidencias y que el contador de patrones que superan el umbral debe ser mínimo 5, con esto nos aseguramos que se encuentran puntos de coincidencia suficientes y que estos pertenecen a distintos patrones.

Una vez se ha catalogado la imagen como apta, es decir se ha superado las condiciones impuestas anteriormente, se procede a extraer la información textual de la imagen.

Los procesos realizados internamente se muestran en la Figura 5.3.

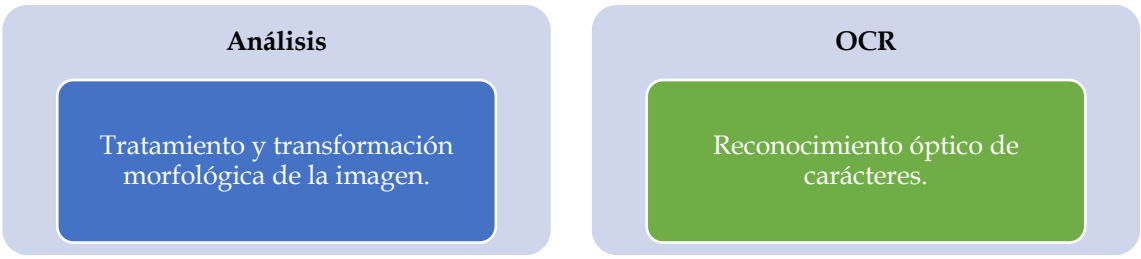


Figura 5.3: Etapas de OCR

Para facilitar y mejorar el reconocimiento óptico de caracteres, primero se lleva a cabo un tratamiento de la imagen, proceso en el cual se realizó un cambio del espacio de color del recorte extraído a escala de grises, y sobre la imagen resultante se aplicó una transformación morfológica en este caso erosión.

Una vez preparada la imagen se lleva acabo el reconocimiento óptico de caracteres (OCR).

La cantidad de información útil extraída de la imagen depende en gran medida del número de elementos que compongan la ventana. Es decir, el número de caracteres reconocidos es menor si la imagen contiene iconos, barras, cuadros

de dialogo, etc. Análogamente, si la imagen contiene un alto porcentaje de texto, el número de caracteres reconocidos será mayor. El texto extraído del contenido de la ventana emergente se almacena en un fichero que será entregado al analista forense.

Dado que la mayoría de los ataques con ransomware agregan un fichero de texto con las instrucciones para recuperar la información secuestrada, en la segunda fase se realiza una búsqueda de la información relacionada con el ataque en los ficheros almacenados en el dispositivo víctima del ataque.

Primero, se realiza una búsqueda de ficheros con extensiones .txt y .html sobre una máquina infectada, que por defecto tendrá todos los documentos cifrados dejando solo aquellos con las instrucciones proporcionadas por el cibercriminal. La búsqueda se realiza de forma recursiva en los directorios más comunes como son: documentos y escritorio.

Los ficheros encontrados, al igual que el resto de información que se obtiene, son almacenados en una carpeta dentro de memoria USB, para su futuro análisis.

Finalmente, la fase 3 realiza el volcado de la memoria RAM del dispositivo infectado utilizando una de las herramientas presentadas en la Sección 5.1.1. La memoria volátil es fundamental en el análisis forense ya que permite observar posibles indicios de la presencia de un malware. Una vez obtenida la memoria volátil, se puede llevar a cabo el análisis de dicha memoria con distintas herramientas. La más utilizada en este proceso es Volatility porque se trata de una herramienta que contiene múltiples aplicaciones integradas en una sola, lo que facilita el trabajo de los analistas. Las memorias volátiles pueden tener distintas extensiones como .dmp, .raw, .mem.

La información extraída en las tres fases se comprime en un archivo.zip que se entrega al analista forense para que realice los procesos de análisis de la información. La Figura 5.4 muestra el diagrama de flujo con el funcionamiento de la técnica.

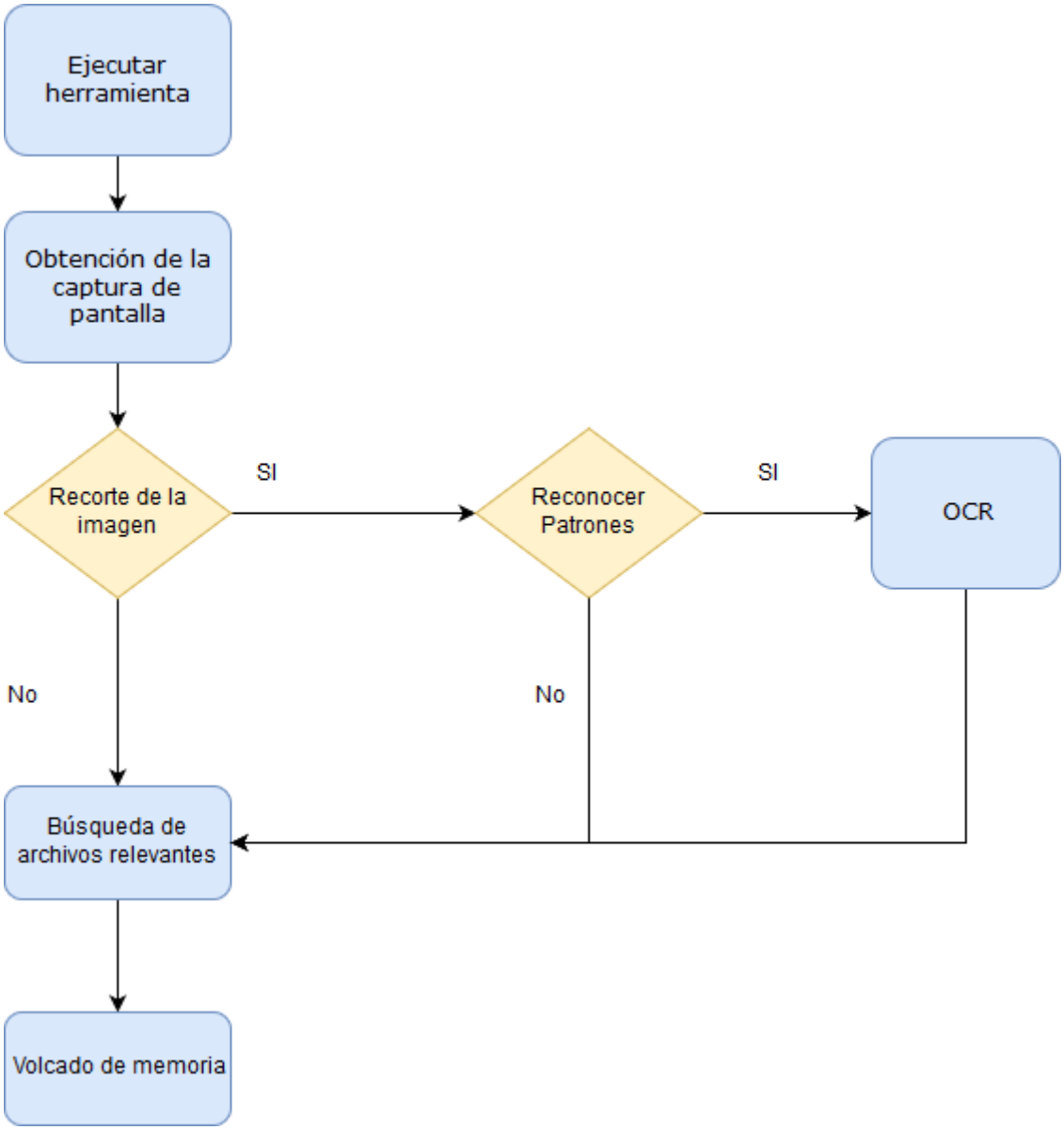


Figura 5.4: Diagrama de Flujo de la Herramienta

5.3. Implementación

El lenguaje elegido para la implementación de la herramienta fue Python, debido al gran número de librerías y funciones que no permite utilizar y que se adaptan a las necesidades de la herramienta y con una comunidad extensa que nos facilitará el alcanzar nuestro objetivo, además se trata de un lenguaje con una curva de aprendizaje bastante adecuada respecto a otros lenguajes que se contemplaron como C++.

Las librerías que se usaron a destacar son:

- Pytesseract: es una librería para el reconocimiento óptico de caracteres (OCR).
- OpenCV: es una biblioteca de funciones para el análisis de imagen.
- PIL: es una librería que permite la edición de imágenes directamente desde Python.

Siendo OpenCV la más usada en la implementación de la herramienta.

5.4. Evaluación de la Técnica

Para evaluar la capacidad de detección de la técnica propuesta se preparó un ordenador de escritorio, como víctima, con las siguientes características: Sistema operativo Windows 7 de 64bits, 4GB de memoria RAM, configuraciones y aplicaciones comunes en usuarios caseros (Microsoft Office, Adobe Reader), para simular un entorno real de la víctima y se generaron ficheros con distintos formatos (texto, imágenes y videos).

La máquina víctima fue infectada por diferentes muestras de ransomware para posteriormente utilizar la técnica propuesta para la extracción de información del ataque. Evaluación de Herramientas de Volcado de Memoria.

El primer grupo de pruebas tuvo como objetivo evaluar las herramientas de volcado de memoria RAM presentadas en la Sección 5.1.1. Los resultados obtenidos se presentan en la Tabla 5.1.

MUESTRAS\HERRAMIENTA	CERBER	LOCKY	TESLACRYPT	VARTESLA	WANNACRY	SAGE	TIEMPO(SEG)	INTERACCION
DUMPIT	✔	✔	✔	✔	✔	✔	50	✔
RAMCAPTURE	✔	✔	✔	✔	✔	✔	58	✘
FTKIMAGER	✔	✔	✔	✔	✔	✔	52	✘
WINPMEM	✔	✔	✔	✔	✔	✔	51	✘

Tabla 5.1: Evaluación de las herramientas de volcado de memoria

En los resultados se observa que todas las herramientas realizaron la extracción de memoria correctamente, diferenciándose en el tiempo de ejecución y las configuraciones de ejecución. La herramienta Dumpit es la herramienta más rápida y no necesita interactuar con el usuario a diferencia de las otras herramientas que muestran algún tipo de interfaz o que no son actualizadas con frecuencia. Estas características la hacen idónea para incluirla en la técnica para que sea automática.

5.4.1. Evaluación del Reconocimiento de Patrones.

En el primer grupo de pruebas se evaluó la capacidad de la técnica en identificar la ventana emergente del ransomware. En las pruebas se utilizaron 62 muestras que representan rasgos característicos de una ventana perteneciente a un ransomware y se analizaron 3 capturas de pantallas. La Tabla 5.2 presenta las configuraciones y resultados del análisis realizado.

ID CAPTURA	RANSOMWARE	RESOLUCIÓN	APTO	COINCIDENCIAS	TIEMPO(SEG)
1	LOCKY	880x600	SI	13	48,76
2	TESLACRYPT	1280x800	SI	12	90,16
3	WANNACRY	1280x800	SI	17	87,92

Tabla 5.2: Análisis de imágenes capturadas en la máquina víctima

Las primeras tres columnas muestran las características de las capturas de pantalla: Identificador, resolución y la muestra de ransomware al que pertenece la ventana emergente. En la columna Apto se muestra si el análisis de la captura de pantalla y la imagen obtenida han sido lo suficientemente buena, para poder buscar patrones. La columna Coincidencias muestra el número de coincidencias que se obtuvo con la ventana emergente con cada muestra. Para que el número de coincidencias devuelto sea tomado en cuenta, este debe superar el umbral definido. El valor del umbral fue definido después de múltiples pruebas con el objetivo de descartar falsos positivos. Los puntos de coincidencia que han superado el umbral se van acumulando, además de ir incrementando un contador de muestras válidas. Una vez terminada la búsqueda de patrones, se

comprueba si el total de puntos de coincidencia acumulados alcanza el umbral establecido y además pertenece a más muestras de las tres utilizadas. Finalmente, se consideró el tiempo que ha tardado la herramienta en realizar todos los procesos.

Para analizar cómo afectan las diferentes configuraciones y sistemas operativos se realizaron 5 capturas de pantalla adicionales. Los resultados de las pruebas se presentan en la Tabla 5.3 (las nuevas capturas de pantalla aparecen sombreadas).

ID CAPTURA	RANSOMWARE	RESOLUCIÓN	APTO	COINCIDENCIAS	TIEMPO(SEG)
1	Locky	880x600	SI	13	48,76
2	TeslaCrypt	1280x800	SI	12	90,16
3	WannaCry	1280x800	SI	17	87,92
4	CryptoLocker	1600x900	SI	18	118,99
5	Cerber	1280x800	SI	7	69,64
6	Variante Tesla	1280x800	SI	7	112,56
7	CTB-Locker	1600x900	NO	9	127,19
8	Sage	1280x800	SI	9	68,48

Tabla 5.3: Segundo Resultado del Análisis de Imagen

Se destaca el caso de la captura 7, debido a que a pesar de haber conseguido realizar el recorte de la imagen, no ha sido capaz de aislar en su totalidad la región de interés del resto del entorno, comprometiendo la privacidad del usuario. Por tanto, el recorte fue rechazado a pesar de conseguir detectar suficientes coincidencias. La Figura 5.5 presenta la captura de pantalla obtenida.

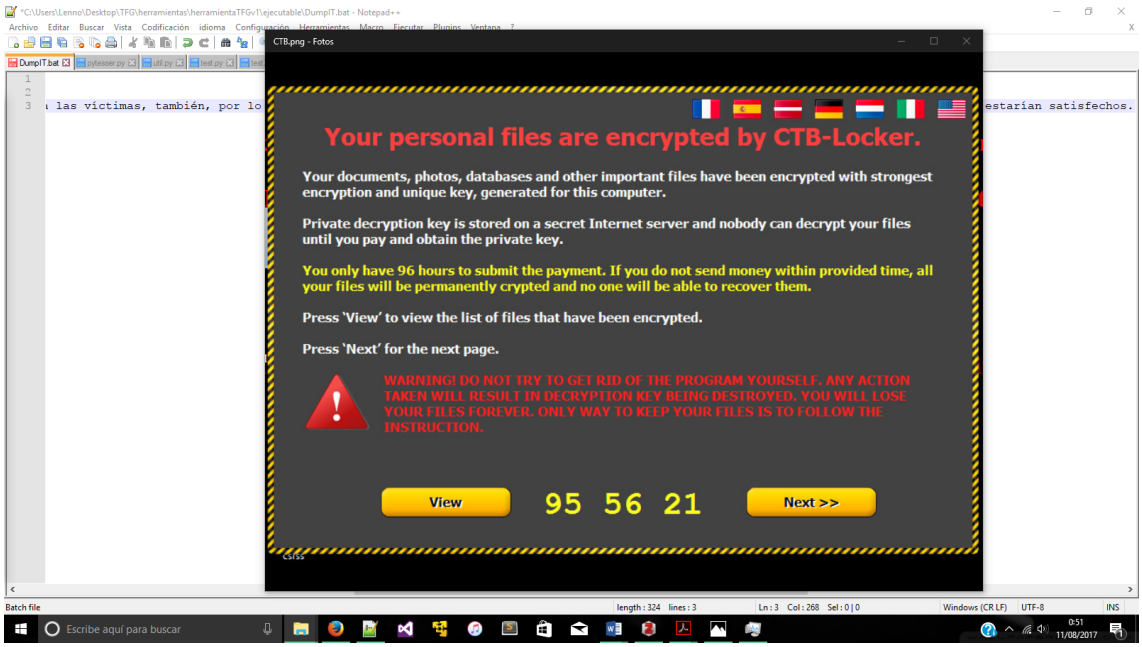


Figura 5.5: Captura de Pantalla CTB-Locker

El resultado obtenido después del recorte se puede apreciar en la Figura 5.6 donde se distingue que no aisló la región de interés.

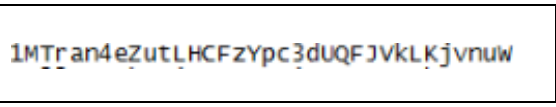
Figura 5.6: Recorte de la Región de Interés

5.4.2. Evaluación del Reconocimiento Óptico de Caracteres.

Para evaluar la efectividad del reconocimiento de caracteres se aplicó dicha funcionalidad sobre las ventanas emergentes de distintas muestras de

ransomware, centrándonos en características clave: Correo electrónico, URL, dirección de Bitcoin y dinero pedido como rescate.

En la Figura 5.7 muestra un ejemplo de la imagen de la dirección de bitcoin con el respectivo texto reconocido.

	lMTran4eZutLHCFlYpt3dUQF]VkLKjvnuW

(a) Imagen recortada de la dirección (b) Texto reconocido

Figura 5.7: Reconocimiento de caracteres Dirección de Bitcoin

La comparación de las direcciones se presenta en la Tabla 5.4:

TESLACRYPT DIRECCION BITCOIN			
CARACTERES	**CARACTERES CORRECTOS**	**CARACTERES ERRONEOS**	**ACIERTO**
34	31	3	91%

Tabla 5.4: Análisis de escalabilidad

Este procedimiento se realizó de igual manera con el resto de las características clave y en las distintas muestras de ransomware, obteniendo el resultado que se puede observar en la Figura 5.8.

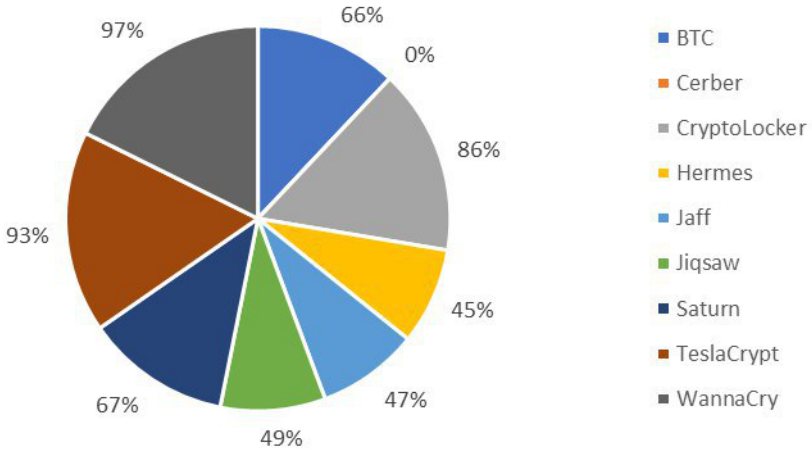


Figura 5.4.28: Resultado de reconocimiento OCR

5.4.3. Evaluación de Funcionalidades de la Herramienta.

Finalmente, se analiza el comportamiento de la técnica de forma automática. El proceso realizado se resume en la Figura 5.9.

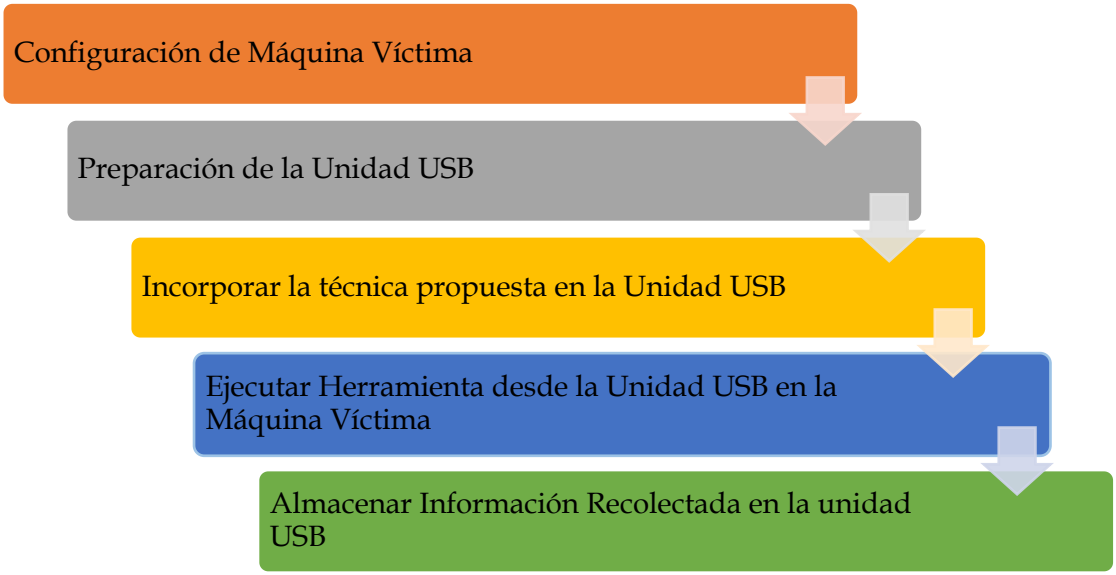


Figura 5.9: Fases de Experimentación

Las pruebas se realizaron con tres muestras de los ransomware más conocidos. La Tabla 5.5 muestra la efectividad en cada una de las fases de la técnica propuesta.

RANSOMWARE	CAPTURA	RECORTE	RECONOCIMIENTO	OCR	BUSQUEDA	VOLCADO
CrytoLocker	✔	✔	✔	✔	✔	✔
TeslaCrypt	✔	✔	✘	✘	✔	✘
WannaCry	✔	✔	✔	✔	✔	✔

Tabla 5.5: Resultados de la Herramienta Final

La herramienta funcionó bien en términos generales, sin embargo, en el caso de TeslaCrypt solo realizó correctamente 3 procesos. En primer lugar, el reconocimiento de patrones dentro de la imagen no fue capaz de identificar la ventana como perteneciente a un ransomware. Esto se debe al nuevo formato de este tipo de ventana y a la falta de patrones para poder identificarlo. Al no encontrar suficientes coincidencias en la imagen, el reconocimiento óptico de

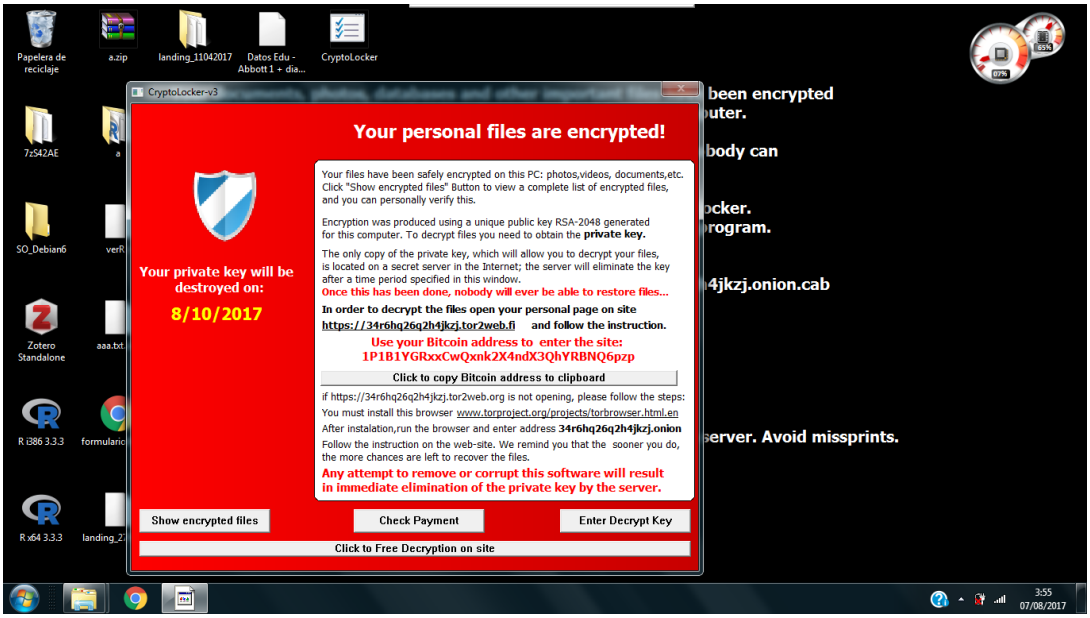
caracteres se descarta. El volcado de memoria también falló debido a que la muestra TeslaCrypt bloqueaba la ejecución de la herramienta DumpIt encargada del proceso. Se observa que el tiempo de ejecución es elevado, pero 5 de los 6 procesos se ejecutan en unos tiempos parecidos a las pruebas realizadas en máquinas no infectadas. El aumento de tiempo se presenta en el volcado de la memoria. Como se puede observar en la siguiente Tabla 5.6.

RANSOMWARE	COINCIDENCIAS	TIEMPO de Ejecución (Seg)
CryptoLocker	13	9,26
TeslaCrypt	5	2,05
WannaCry	12	8,78

Tabla 5.6: Resultados en un Entorno Real

El volcado de memoria es el principal motivo del aumento de tiempo en ejecución y esto se debe principalmente a las características hardware, tanto de la memoria RAM como de la memoria USB donde se ejecuta la herramienta, principalmente nos centraremos en la segunda debido a que la ejecución del volcado de memoria se verá retrasado por el tiempo de escritura de la memoria USB.

La Figura 5.10 se muestra el resultado de la extracción de la ventana emergente de la muestra de CryptoLocker. En la Figura 5.10.a se observa la imagen resultante de la captura de pantalla y en la Figura 5.10.b su respectiva región de interés.



(a) Captura de Pantalla

(b) Región de interés

Figura 5.10: Extracción de ventana emergente

A continuación, se evaluó la técnica aumentando a 13 muestras de ransomware diferentes. Las pruebas se realizaron en entornos reales para analizar su escalabilidad. La Tabla 5.7 muestra el comportamiento de la herramienta con múltiples muestras.

NOMBRE	CAPTURA	RECORTE	OCR	BUSQUEDA	VOLCADO	COINCIDENCIAS
CryptoLocker	✔	✔	✔	✔	✔	✔
TeslaCrypt	✔	✔	✔	✔	✔	✔
WannaCry	✘	✘	✘	✘	✘	✘
Cerber	✔	✔	✔	✔	✔	✔
JigSaw	✔	✔	✔	✘	✔	✘
Vipassana	✔	✘	✘	✘	✘	✘
Hermes	✔	✔	✔	✔	✔	✔
BTCware	✔	✔	✔	✘	✔	✔
Saturn	✔	✔	✘	✔	✔	✘
Locky	✔	✘	✘	✘	✘	✘
Jaff	✔	✔	✘	✔	✔	✘
GandCrab	✘	✘	✘	✘	✘	✘
Crysis	✘	✘	✘	✘	✘	✘

Tabla 5.7: Análisis de escalabilidad

Cómo se observa en la tabla, la muestra del ransomware Wannacry no cifró los archivos del USB obteniendo un resultado positivo en todas sus funcionalidades. En cambio, con la muestra del ransomware Crysis no fue posible ejecutar la aplicación ya que la memoria USB fue cifrada evitando así su ejecución. El comportamiento de la herramienta para cada muestra utilizada se ha clasificado por los fallos más comunes:

- **Fondo de pantalla**: Las muestras de ransomware Vipassana y Locky informan al usuario el ataque sustituyendo el fondo de pantalla. Esto provoca que el proceso de recorte de pantalla no encuentre una región válida protegiendo la privacidad del usuario.

- **Nota de rescate inexistente**: En las muestras de ransomware TeslaCrypt, Jiqsaw, Vipassana y BTCware no se encontraron archivos de rescate en ninguno de los formatos habituales debido a que no los proporcionan.

- **Ventana con pocas características de ransomware conocidos**: Las muestras de ransomware Jaff y Saturn en vez de usar ventanas emergentes proporcionan archivos con extensiones .bmp y .html. Estos archivos se abrieron para realizar el análisis. Como resultado, no hubo suficientes puntos

de coincidencias con respecto a los patrones que suelen tener las ventanas pertenecientes a un ransomware y que están alma en nuestra base de conocimiento. Al no estar seguro de que la ventana pertenece a un ransomware el recorte queda descartado para evitar así comprometer la información delicada del usuario víctima.

- **Cifrado de la unidad de almacenamiento**: Las muestras de ransomware de WannaCry, Crisis y GandCrab cifraron los ficheros de la USB impidiendo la correcta ejecución de la técnica.

6. CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

Hoy en día el análisis de ransomware ha tomado gran relevancia debido al crecimiento y diversidad de este tipo de malware de la necesidad de comprender y analizar su comportamiento.

La herramienta propuesta en este trabajo permite obtener información relevante de un ransomware, dentro de un entorno real sin la necesidad de realizar configuraciones complejas, automatizando funcionalidades estudiadas en diversas literaturas como puede ser el volcado de memoria.

La mayoría de las literaturas investigadas tratan el malware en un entorno controlado, lo que puede alterar su comportamiento real. En muchos de los estudios la creación de los entornos se lleva a cabo mediante el uso de herramientas que necesitan una configuración especifica en cada caso. Limitando estás tareas a un grupo de usuarios con conocimientos suficientes para llevarlo a cabo.

En este trabajo se realizó la búsqueda de patrones en imágenes con el objetivo de obtener información suficiente para determinar si se trata de un ransomware. En el caso que la información sea suficiente se realiza el reconocimiento óptico de caracteres y la extracción en un texto plano, con la finalidad de facilitar su manipulación. En cualquier caso, se hará el volcado de memoria y búsqueda de archivos relevantes. Todas estas funcionalidades se ejecutan desde una memoria USB donde también se almacena la información obtenida.

Para observar el comportamiento de la herramienta se realizaron experimentos con 16 muestras distintas de ransomware en un entorno real. Donde se llegó a la conclusión que la herramienta tiene un comportamiento optimo, en los distintos entornos que fue ejecutada obteniendo los resultados

esperados. También se observó que el tiempo de ejecución de la herramienta aumenta dependiendo la velocidad del USB y del tamaño de la memoria RAM.

6.2. Trabajo Futuro

Como trabajos futuros posibles pueden señalarse los siguientes:

- Ampliar el reconocimiento óptico caracteres dentro de la imagen para aumentar el texto reconocido y obtener más información, así como aumentar la efectividad del reconocimiento en otros idiomas.
- Mejorar el acierto del reconocimiento de la ventana emergente, así como su recorte con distintos métodos y algoritmos.
- Incluir una base de datos con más muestras para no limitar la herramienta, incrementando el reconocimiento de patrones.
- Añadir un servidor donde almacenar los resultados obtenidos por la herramienta y no depender del tamaño de la memoria USB.
- Encontrar la manera de evitar que la memoria USB sea cifrada, permitiendo con ello su ejecución frente a cualquier muestra de ransomware.

RESUMEN EN INGLÉS

7. INTRODUCTION

In its beginnings, the use of the internet was minimal mainly used by industrial sectors, military and research, but little by little has been introduced in the society increasing the number of Internet users. The continued development of the technology and its ease of acquisition attracted a large number of consumers, allowing the Internet to reach all sides.

Technological devices became essential means of storage and communication for the daily use, reaching the point of our most private information on devices with Internet access. Causing a minority Internet group to be interested in exploring such information for the purpose of obtaining a benefit, which leads to the appearance of malicious applications to attack internet-connected devices with access to it.

Within the various types of malicious software that can be found on the internet, one of the most dangerous and in recent years heavily used by cybercriminals is the so-called ransomware. The campaigns of attack of this type of malicious software have been seen in different entities, being these public or private.

In 2016, a campaign was directed at hospitals such as the Hollywood Presbyterian Medical Center in Los Angeles, where ransomware-type malware blocked access to the system, until $ 17,000 was paid as a ransom, FBI Laura Miller reported that she was charged with investigations however police sources explained to the media how long the hospital had paid the ransom before sending aid from the police.

Similarly, in Germany other than hospitals were affected by the operation of computer systems, such as the case of the x-ray system that can not access the data it needed because they were encrypted. Fortunately, the hospitals did not reach the point of paying any ransom because they were able to recover the data

through backup copies [2].

Other notable targets were major media companies such as The New York Times, also prestigious universities such as Calgary in Canada where they paid $ 16,000 to recover emails that were encrypted for a week.

In general, the attacks are intended to disrupt the normal operation of the computer service, as was the case with the San Francisco train ticket vending machines, allowing people to travel without paying for such tickets [4].

Most malicious software is intended to obtain confidential information from both users and large companies. The storage of personal data on the network is increasingly used, highlighting the importance of servers for technology companies, as it is a primary part of its operation, becoming the main target for cybercriminals. Coming to a massive ransomware attack on companies like MongoDB, where 32,000 servers were "hijacked" demanding payment of a bailout through bitcoins to retrieve the information. Several companies were affected like Telefónica and eBay, in addition to the governments that use this service [5].

Nowadays nobody is exempting from a computer attack, in fact 17 American libraries have been compromised with this type of attacks like those of a means of communication BBC [6].

7.1. Motivation

The studies related to obtaining the information and analysis of the behavior of a ransomware, focus on the configuration of controlled environments for their observation and extraction of relevant information, such as memory dump and forensic analysis.

Currently in the market there are tools that focus on testing, ransomware analysis, some focus on memory dump, others on behavioral observation and

there are also tools to automate functions but need qualified staff to carry out These tasks.

The scarcity of tools to unify the various functionalities about obtaining samples of ransomware has generated the interest on the part of the security bodies to acquire a tool capable of automating the processes of sampling of ransomware simplifying and facilitating the management of the tool Without the need to have prior knowledge for power.

7.2. Objectives

This Final Degree Work (TFG) has the following objectives:

- Perform state of the art malware, ransomware, evolution, types, families, performance, how it spreads. Study of ransomware as a business.
- Extract information from the screen capture of a ransomware-infected computer, anonymizing user information.
- Implement a tool that allows the acquisition of information through the message displayed by the ransomware and extract the volatile memory from the computer.

7.3. Work Schedule

The first stage defined the work objectives and research methods for its documentation, assisting tutorials for proper follow-up and assignment of new tasks to be performed. The second stage implements the image analysis tool and the memory dump. The main phases in this stage are: Specification of the requirements, design and implementation of the tool and lastly the tests with different tests of ransomware and results obtained after the execution.

8. CONCLUSIONS AND FUTURE WORK

8.1. Conclusions

The conclusions obtained from this work are as follows:

Today the ransomware analysis has taken on great importance due to the growth and diversity of this type of malware, in addition to the need to understand and analyze its behavior.

The tool proposed in this work allows to obtain relevant information from a ransomware, within a real environment without the need to make complex configurations, automating functionalities studied in various literatures such as memory dump.

Most investigated literature deals with malware in a controlled environment, which can alter its actual behavior. In many of the studies the creation of the environments is carried out by the use of tools that need a specific configuration in each case. By limiting these tasks to a group of users with sufficient knowledge to carry it out.

In this work, we searched for patterns in images with the objective of obtaining sufficient information to determine if it is a ransomware. If the information is sufficient, optical character recognition and extraction in a plain text are performed, in order to facilitate its manipulation. In any case, the memory dump and search of relevant files will be done. All these functions are executed from a USB memory stick where the information obtained is also stored.

To observe the behavior of the tool, experiments were performed with three different samples of ransomware in a real environment. Where it was concluded that the tool has an optimal behavior, in the different environments that was executed obtaining the expected results. It was also observed that the execution

time of the tool increases depending on the USB speed and the size of the RAM.

8.2. Future work

Possible future work may include:

- Expand the optical recognition characters within the image to increase the recognized text and get more information.
- Improve the accuracy of the recognition of the pop-up window, as well as its clipping with different methods and algorithms.
- Include a database with more samples so as not to limit the tool, increasing the pattern recognition.
- Add a server to store the results obtained by the tool and will not depend on the size of the USB memory.

9. REFERENCIAS

[1] (2016) ‘‘Hollywood hospital pays $17,000 in bitcoin to hackers; FBI investigating – LA Times’’. http://www.latimes.com/business/ technology/la-me-ln-hollywood-hospital-bitcoin-20160217-story.html.

[2] R. Millman. (2016) ‘‘Ransomware holds data hostage in two German hospitals’’. https://www.scmagazineuk.com/ransomware-holds-data-hostage-in-two-german-hospitals/article/530494/.

[3] I. Anton, E. David, S. Fedor, P. Santiago. (2016) ‘‘Kaspersky security bulletin 2016. The ransomware revolution’’, Securelist - Information about Viruses, Hackers and Spam. https://securelist.com/kaspersky-security-bulletin-2016-story-of-the-year/76757/.

[4] S. Gibbs, (2016) ‘‘Ransomware attack on san Francisco public transit gives everyone a free ride’’, the Guardian. http://www.theguardian.com/technology/2016/nov/28/passengers-free-ride-san-francisco-muni-ransomeware.

[5] (2017) ‘‘Un enorme ataque de “ransomware” secuestra 32.000 servidores de MongoDB’’, abc. http://www.abc.es/tecnologia/redes/ abci-enorme-ataque-ransomware-secuestra-32000-servidores-mongodb-201701102153_noticia.html.

[6] (2017) ‘‘US libraries hit by ransomware attack’’, BBC News. http://www.bbc.com/news/technology-38731011.

[7] Avast. (2017) ‘‘¿Qué es el malware y cómo eliminarlo? Antimalware’’. https://www.avast.com/es-es/c-malware.

[8] Panda, (2017) ‘‘Virus y Antivirus: Información, historia y evolución-información sobre seguridad-Panda security’’. http://www.pandasecurity.com/spain/homeusers/security-info/classic-malware/.

[9] M. J. Erquiaga, ‘‘Botnets: Mecanismos de control y de propagación’’, en las Actas del XVII Congreso Argentino de Ciencias de la Computación, 2011, pp. 1076-1085.

[10] T. Wüchner, ‘‘Behavior-based malware detection with quantitative data flow analysis’’, Bachelor Thesis, Computer Science, Munchen University, Munchen, German, 2016.

[11] Panda, (2015) ''Los virus más famosos de la historia: Viernes 13 - Panda Security Mediacenter''. http://www.pandasecurity.com/spain/mediacenter/malware/virus-viernes-13/.

[12] L. J. Locher, C. Doyle, C. Amaris, R. Morimoto, y Anonymous, Maximum Windows 2000 Security, Sams Publishing, 2001.

[13] Securelist, (2017), ''The classification tree'', Securelist - Information about Viruses, Hackers and Spam. https://securelist.com/threats/the-classification-tree/.

[14] M. Sikorski, A. Honig, Practical malware analysis: the hands-on guide to dissecting malicious software. USA, San Francisco:no starch press, 2012.

[15] (2015), ''Blog elhacker.NET: Introducción al análisis forense de malware''. http://blog.elhacker.net/2015/02/introduccion-al-analisis-de-malware-herramientas-forense.html.

[16] F. de B. Nafría Oñate, ''Plataformas de ejercicios de ciberseguridad'', Proyecto de Fin de Grado, departamento de Ingeniería Telemática y Electrónica, Universidad Politécnica de Madrid, 2016.

[17] A. Azad. "Ransomware: A research and a personal case study of dealing with this nasty malware". *Issues in Informing Science and Information Technology*, vol. 14, pp. 087-099, 2017.

[18] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, y E. Kirda, ''Cutting the gordian knot: A look under the hood of ransomware attacks'', en Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 3–24, 2015.

[19] M. H. U. Salvi y M. R. V. Kerkar, ''Ransomware: A cyber extortion'', *Asian Journal for Convergence in Technology*, vol. 2, 2016.

[20] G. O'Gorman y G. McDonald, "Ransomware: A growing menace". Symantec Corporation, 2012.

[21] J. A. L. Gamiño, S. F. L. Jiménez, S. I. R. Cacho, y M. P. Gaytán, ''Uso del bitcoin como una propuesta de intercambio monetario en México'', Innovaciones Puntos Clave Para El Desarrollo, pp. 295-309, 2015.

[22] Universidad Complutense, ''Analisis de ransomware''.

[23] Incibe (2016) ''Oleada de ransomware suplantando a Endesa'', INCIBE, 31-may-2016.https://www.incibe.es/protege-tu-empresa/avisos-seguridad/ransomware-Endesa.

[24] M. H. Ligh, A. case, J. Levy, A. walters. The art of memory forensics: detecting malware and threats in windows, linux, and Mac memory. John Wiley & Sons, 2014.

[25] (2017)''The No More Ransom Project''. https://www.nomoreransom.org/ransomware-qa.html.

[26] (2017) ''Ransomware FAQ - Windows defender security intelligence''. https://www.microsoft.com/en-us/wdsi/threats/ransomware.

[27] McAfee (2016) ''McAfee labs quarterly threat report December 2016 -rp-quarterly-threats-dec-2016.pdf''. https://www.mcafee.com/ca/resources/reports/rp-quarterly-threats-dec-2016.pdf.

[28] McAfee (2016) ''Understanding ransomware & strategies to defeat''. https://www.mcafee.com/es/resources/white-papers/wp-understanding-ransomware-strategies-defeat.pdf.

[29] C. Zheng, N. Dellarocca, N. Andronio, S. Zanero, y F. Maggi, ''GreatEatlon: fast, static detection of mobile ransomware'', in Proceedings of the International Conference on Security and Privacy in Communication Systems, pp. 617-636, 2016.

[30] N. Andronio, S. Zanero, y F. Maggi, ''Heldroid: dissecting and detecting mobile ransomware'', in Proceedings of the International Workshop on Recent Advances in Intrusion Detection, pp. 382-404, 2015.

[31] D. NECSTLab, ''BitIodine: Extracting intelligence from the bitcoin network'', in Proceedings of the 18th International Conference on Financial Cryptography and Data Security, FC 2014, Christ Church, Barbados, March 3-7, vol. 8437, p. 457, 2014.

[32] P. H. Rughani, ''ForMaLity: Automated FORensic MAlware Analysis using VolatiLITY'', *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, 2017.

[33] D. F. Arce Villarruel, ''Análisis digital de una infección de malware en sistemas windows'' Proyecto de Fin de Grado, Escuela Politécnica Nacional, Ecuador, Quito, 2016.

[34] T. M. Jumbo Tene,''Metodología para el análisis de malware en un ambiente controlado'' Proyecto de Fin de Grado, Universidad Politécnica Salesiana, Ecuador, Cuenca, 2017.

[35] R. P. Rivera Guevara, ''Análisis de características estáticas de ficheros ejecutables para la clasificación de malware'' Tesis de Máster, Universidad Politécnica de Madrid, España, Madrid, 2014.

[36] H. A. M. García y L. B. C. Us, ‘‘Hidden Tear: Análisis del primer Ransomware Open Source. ’’, en Avances y perspectivas de la innovación, investigación y vinculación, Mérida Yucatán, México, vol. 1, pp. 31–54, 2015.

[37] A. I. Valencia-Valencia y S. N. Galicia-Haro, ‘‘Detección de malware con modelo de lenguaje y su clasificación mediante SVM. ’’, *Research in Computing Science*, vol. 115, pp. 9-18, 2016.

[38] ‘‘Deteccion de Bordes’’. http://www.tecnicaenlaboratorios.com/Nikon/Info_deteccion_de_bordes.html.