Maestría en

CIBERSEGURIDAD

Trabajo Previo a la Obtención de Título de

Magister en Ciberseguridad

AUTORES:

Alberto André Carrasco Ruiz

Dennisse Mishel Guartan Salinas

María Fernanda Pacheco Tulcanaza

Jeniffer Judith Urquiza Olivo

Alexis Israel Veloz Jaya

TUTORES:

Iván Reyes Chacón

Alejandro Cortés López

TEMA:

Perfil Mínimo de Detección en Windows Contra Ransomware Black Basta: Diseño y Análisis en Laboratorio

Quito – Ecuador

09 - 2025

REINVENTEMOS EL FUTURO

Certificación de Autoría

Nosotros, **Alberto André Carrasco Ruiz, Dennisse Mishel Guartan Salinas, María Fernanda Pacheco Tulcanaza, Jeniffer Judith Urquiza Olivo y Alexis Israel Veloz Jaya,** declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

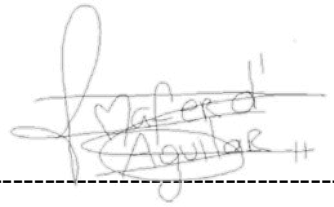
Cedemos nuestros derechos de propiedad intelectual a la Universidad Internacional del Ecuador (UIDE), para que sea publicado y divulgado en Internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.

Firma

Alberto André Carrasco Ruiz

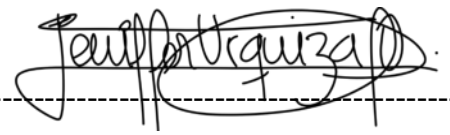
Firma

Dennisse Mishel Guartan Salinas



Firma

María Fernanda Pacheco Tulcanaza



Firma

Jeniffer Judith Urquiza Olivo

Firma

Alexis Israel Veloz Jaya

Autorización de Derechos de Propiedad Intelectual

Nosotros, **Alberto André Carrasco Ruiz, Dennisse Mishel Guartan Salinas, María Fernanda Pacheco Tulcanaza, Jeniffer Judith Urquiza Olivo y Alexis Israel Veloz Jaya,** en calidad de autores del trabajo de investigación titulado ***Perfil Mínimo de Detección en Windows Contra Ransomware Black Basta: Diseño y Análisis en Laboratorio***, autorizamos a la Universidad Internacional del Ecuador (UIDE) para hacer uso de todos los contenidos que nos pertenecen o de parte de los que contiene esta obra, con fines estrictamente académicos o de investigación. Los derechos que como autores nos corresponden, lo establecido en los artículos 5, 6, 8, 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento en Ecuador.

D. M. Quito, septiembre 2025.

Firma

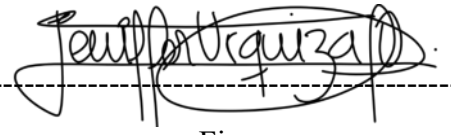
Alberto André Carrasco Ruiz

Firma

Dennisse Mishel Guartan Salinas

Firma

María Fernanda Pacheco Tulcanaza



Firma

Jeniffer Judith Urquiza Olivo

Firma

Alexis Israel Veloz Jaya

Aprobación de Dirección y Coordinación del Programa

Nosotros, **Alejandro Cortés e Iván Reyes,** declaramos que: **Alberto André Carrasco Ruiz, Dennisse Mishel Guartan Salinas, María Fernanda Pacheco Tulcanaza, Jeniffer Judith Urquiza Olivo y Alexis Israel Veloz Jaya,** son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal de ellos.

Alejandro Cortés L.

Maestría en Ciberseguridad

Iván Reyes Ch.

Maestría en Ciberseguridad

Dedicatoria

Este trabajo está dedicado a todas las personas que confiaron en nosotros, nos ayudaron y alentaron en los momentos de mayor exigencia, celebrando el cumplimiento de cada objetivo como suyo. A nuestra familia y seres queridos que nos brindaron apoyo incondicional durante todo este caminar.

Esta dedicatoria es también un reconocimiento al esfuerzo colectivo que convirtió nuestras habilidades en una fortaleza y nuestro trabajo en un verdadero proyecto de equipo.

Agradecimientos

Agradecemos a cada una de las personas que hicieron posible culminar con éxito la realización de este trabajo.

A nuestros docentes, por guiarnos, compartir su conocimiento y experiencia, su acompañamiento fue indispensable para nuestro desarrollo académico. Agradecemos de manera especial el compromiso, dedicación y colaboración de cada uno de los cinco integrantes de este equipo, la predisposición y buena actitud hicieron culminar con éxito esta etapa.

Finalmente, gracias a quienes de una u otra manera, nos brindaron ya sea su apoyo técnico, personal o moral. Este logro es de todos quienes con amor y respeto han dejado su granito de arena y huella en este camino.

Resumen

En este documento se diseña y propone un Perfil Mínimo de Detección (PMD) para el ransomware Black Basta en sistemas operativos Windows. La investigación se desarrolla mediante un laboratorio aislado compuesto por un servidor Ubuntu, un servidor de análisis y múltiples máquinas Windows configuradas como víctimas. Para la recolección de telemetría y evidencia forense se emplean herramientas como Sysmon, Procmon, Regshot, Process Hacker, Velociraptor, y utilidades de adquisición de memoria, permitiendo obtener observaciones precisas desde múltiples capas del sistema. Al ejecutar el ransomware se generó un grupo de artefactos forenses, por citar, eventos de Sysmon, lista de procesos, servicios, capturas en tiempo real, entre otros. Tras analizarlos se comprobó que eran artefactos característicos de Black Basta, tales como el uso de APIs de enumeración de volúmenes, operaciones multihilo para acelerar el cifrado, manipulación del Wallpaper, creación del archivo readme.txt, uso de rutas extendidas, y presencia de metadatos estructurados en los archivos cifrados. Por otro lado, el análisis estático, dio a conocer una arquitectura criptográfica robusta y un esquema híbrido basado en RSA y técnicas de empaquetamiento.

La correlación de evidencia estática y dinámica permitió la elaboración de un perfil mínimo de detección relacionado a Black Basta, el cual incluye artefactos, patrones de comportamiento y modificaciones relevantes observadas durante su ejecución. Este perfil, ofrece una base metodológica sólida para apoyar la identificación de actividad maliciosa asociada al ransomware en sistemas Windows, y constituye un insumo técnico replicable para futuras investigaciones orientadas al fortalecimiento de la defensa y detección temprana de amenazas.

Palabras clave: Ransomware, artefactos, Black Basta, análisis, cifrado, telemetría.

Abstract

In this document, a Minimum Detection Profile (MDP) is designed and proposed for the Black Basta Ransomware in Windows operating systems. The research is carried out through an isolated laboratory composed of an Ubuntu server, an analysis server, and multiple Windows machines configured as victims. For telemetry collection and forensic evidence acquisition, tools such as Sysmon, Procmon, Regshot, Process Hacker, Velociraptor, and memory acquisition utilities are used, allowing precise observations to be obtained from multiple layers of the system. When the ransomware was executed, a set of forensic artifacts was generated, including Sysmon events, process lists, services, real time captures, among others. After analyzing them, it was confirmed that they were characteristic artifacts of Black Basta, such as the use of volume enumeration APIs, multithreaded operations to accelerate encryption, wallpaper manipulation, creation of the readme.txt file, use of extended paths, and the presence of structured metadata in the encrypted files. Moreover, the static analysis revealed a robust cryptographic architecture and a hybrid scheme based on RSA and packing techniques.

The correlation of static and dynamic evidence enabled the development of a minimum detection profile related to Black Basta, which includes artifacts, behavioral patterns, and relevant modifications observed during its execution. This profile provides a solid methodological foundation to support the identification of malicious activity associated with the ransomware in Windows systems and constitutes a reproducible technical input for future research aimed at strengthening defense and early threat detection.

Keywords: Ransomware, forensic artifacts, Black Basta, analysis, encryption, telemetry.

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Capítulo 1

1. Introducción

1.1. Definición del Proyecto

Este proyecto de investigación tiene como propósito diseñar y proponer un perfil mínimo de detección del ransomware Black Basta en sistemas operativos Windows, con el objetivo de fortalecer la prevención de ataques mediante un enfoque técnico y experimental. Para ello, se llevará a cabo un análisis forense, estático y dinámico de muestras de malware en un entorno de laboratorio controlado, que permita identificar y caracterizar los artefactos digitales más relevantes como eliminación de shadow copies, creación de notas de rescate, uso de herramientas, modificaciones de registro y comportamiento de cifrado y exfiltración.

Con base en los indicadores obtenidos, se diseñará un perfil mínimo de detección, que incluirá reglas o umbrales, tales como la encriptación masiva de archivos o notas de rescate en corto periodo, uso de herramientas inusuales o eliminación de backups.

El desarrollo del proyecto se enmarca en una metodología rigurosa que contempla la planificación, ejecución en laboratorio y análisis de evidencias, con miras a generar un resultado viable y aplicable a entornos reales de seguridad informática. Este documento guía, además sirve como base para asegurar que el estudio sea coherente, factible y científicamente sustentado.

1.2. Justificación e Importancia del Trabajo de Investigación

La creciente sofisticación y persistencia del ransomware Black Basta ha evidenciado la necesidad urgente de diseñar estrategias de detección más efectivas en entornos Windows. Desde su aparición en abril de 2022, Black Basta ha afectado a más de 500 organizaciones a nivel mundial, incluyendo sectores críticos como salud, infraestructura y educación, mediante técnicas avanzadas de doble extorsión, exfiltración de datos y cifrado con extensiones sospechosas. Su

capacidad para evadir herramientas de seguridad tradicionales, como antivirus y EDR, al deshabilitarlas durante el proceso de infección, resalta la necesidad de enfoques de detección basados en artefactos y comportamientos específicos del malware. Además, la utilización de herramientas legítimas como Rclone y PowerShell para la exfiltración de datos y la ejecución remota, junto con la modificación de configuraciones del sistema, subraya la importancia de identificar patrones conductuales característicos. En este contexto, el diseño de un perfil mínimo de detección, fundamentado en el análisis detallado dentro de un laboratorio controlado, representa una estrategia prometedora para la identificación temprana de ataques por parte de Black Basta. Este enfoque no solo contribuirá al fortalecimiento de las defensas en sistemas Windows, sino que también proporcionará una base sólida para futuras investigaciones y el desarrollo de soluciones de seguridad más adaptativas frente a amenazas emergentes.

1.3. Alcance

El presente estudio se enfocará en el análisis estático, dinámico y forense del ransomware Black Basta en sistemas operativos Windows 10 y 11 los cuales representan las versiones de uso predominante en la actualidad. Se realizará una evaluación detallada de las técnicas y herramientas empleadas por esta variante de malware dentro de un entorno controlado de laboratorio. La inspección incluye el comportamiento en el sistema, artefactos generados y métodos de evasión. El análisis se limitará a muestras específicas de Black Basta obtenidas de fuentes confiables y actualizadas, y no se incluirán otras variantes de ransomware en este estudio. Además, la investigación no abarcará la implementación de soluciones de mitigación o la evaluación de su efectividad en entornos reales. El objetivo principal será proporcionar un perfil mínimo de detección basado en artefactos y comportamientos observados durante el

análisis, que pueda servir como referencia para futuras investigaciones y desarrollos en la detección de Black Basta en sistemas Windows.

1.4. Objetivos

1.4.1. Objetivo General. Desarrollar o diseñar un perfil mínimo de detección para Black Basta en Windows, fundamentado en análisis de artefactos y la emulación de TTPs (Tácticas, Técnicas y Procedimientos) en laboratorio aislado, con criterios de calidad centrados en detección temprana y baja tasa de falsos positivos.

1.4.2. Objetivo Específico. Analizar muestras del ransomware Black Basta mediante técnicas estáticas, dinámicas y forenses con el propósito de identificar sus características principales.

Comparar distintos artefactos encontrados con respecto a otros procedentes de muestras adicionales de ransomware Black Basta para determinar cambios importantes en el comportamiento.

Determinar limitaciones y falsos positivos encontrados en el desarrollo del análisis.

Diseñar un perfil mínimo del ransomware mediante el uso de las características encontradas en la revisión de los artefactos.

Capítulo 2

2. Revisión de Literatura

2.1. Estado del Arte

La exploración de ransomware en el sistema operativo de Windows va desde métodos comunes basados en firmas hacia enfoques más especializados como telemetría y la creación de perfiles mínimos de detección, lo cual ayuda a identificar fases críticas del ataque. Análisis actuales indican que la correlación de fuentes como Sysmon, registros de Windows, volcados de memoria y telemetría EDR puede elevar significativamente la detección temprana frente a variantes modernas (Jamil Ispahany, 2024).

2.1.1. Ransomware Black Basta: Técnicas vs Evidencia en Laboratorio. Cuando se habla específicamente de Black Basta existen TTPs (tácticas, técnicas y procedimientos) característicos: cifrado rápido, eliminación de shadow copies, uso de herramientas de movimiento lateral, la manipulación de artefactos notorios como: notas de rescate y extensiones con otro nombre. Esto se logra comprobar con el análisis experimental, donde se evidencia que Black Basta ejecuta vssadmin para suprimir copias de volumen; posteriormente, crea artefactos visuales como el fondo de pantalla usado a modo de vector para presionar a la víctima. También realiza enumeración de árboles de directorio, y ejecuta operaciones de escritura de alta entropía seguidas por renombramiento masivo (doble extensión); finalmente, crea archivo readme.txt que se abre con notepad.exe de manera forzada y automática. Todo lo mencionado coincide con descubrimientos de laboratorios y descripciones oficiales (Cybersecurity & Infrastructure Security Agency (CISA), 2024).

2.1.2. Limitaciones Presentes en la Literatura Actual. Dentro de las limitaciones se puede mencionar las siguientes:

- Son escasos los perfiles mínimos de detección creados y validados en laboratorio para el ransomware Black Basta, en su gran mayoría los escritos dan a conocer las tácticas y técnicas, pero no especifica los eventos mínimos a recolectar para tener una detección confiable. La validación que se realiza en las investigaciones llega a ser redundante, ya que, es extraño encontrar laboratorios que no sean determinísticos; por lo que, este laboratorio tiene como meta obtener un perfil de detección específico y confiable ejecutando distintas muestras, aun si el comportamiento cambia en su ejecución.
- Estudios recientes evidencian que combinar señales como la creación de procesos, la ejecución de comandos, cambios en el sistema de archivos, y entropía de archivos permite caracterizar de manera robusta una fase de cifrado activa (Cybersecurity & Infrastructure Security Agency (CISA), 2024). No obstante, es limitado el número de investigaciones que logran obtener y comparar estos indicadores de múltiples muestras reales, lo cual introduce un importante vacío en la definición de perfiles de detección.

En conclusión, la literatura actual tiende a la construcción de perfiles de detección que indiquen los eventos a capturarse, cómo se correlacionan y hace énfasis en qué los umbrales son vitales para distinguir actividad legítima de comportamiento malicioso; dando a notar la falta de metodologías estandarizadas. Bajo esta perspectiva, el establecimiento de un Perfil Mínimo de Detección en Windows contra Black Basta, sustentado en análisis de laboratorio y contrastado con la literatura técnica, aporta un valor relevante para el avance de prácticas de defensa basadas en comportamiento.

2.2. Marco Teórico

2.2.1. ¿Qué es un Ransomware?. Un ransomware es un tipo de software malicioso denominado malware que se caracteriza por la captura de documentos, archivos o del propio

dispositivo, con la finalidad de solicitar un pago a cambio del rescate de información o el desbloqueo del equipo. Muchos de estos ataques vienen con la divulgación de datos considerados confidenciales en sitios de la dark web, siendo esta una de las amenazas impuestas por parte de los atacantes (Kosinski, 2024).

En la actualidad las entidades han tenido que recurrir a la inversión de equipo y la implementación de normas de seguridad como copias de seguridad múltiples de sus servidores para la prevención y respuesta ante un ataque, sin embargo, así como los protocolos de seguridad han ido evolucionando también lo han hecho los ataques, ya que, cada vez apuntan a tácticas de extorsión más elaboradas y específicas.

En general existen dos tipos principales de ransomware, los de cifrado y no cifrado. Los del tipo cifrado son los más comunes y como lo dice su nombre se caracterizan por cifrar los archivos individuales del dispositivo infectado a cambio de un pago, dentro de esta categoría existen otras derivadas, por ejemplo: Leakware, wipers, scareware, entre otras. Los ransomware del tipo no cifrado son aquellos que bloquean al dispositivo infectado deteniendo todas las funciones del equipo. La determinación del tipo de ransomware con el cual se está tratando es importante para determinar las acciones que se debe ejecutar y calcular las consecuencias de este (Kosinski, 2024).

2.2.2. Ransomware Black Basta. Black Basta es un ransomware que opera bajo un modelo RaaS, es decir, un modelo en el que un atacante adquiere código ransomware de otros ciberdelincuentes especializados en el desarrollo y venta de este tipo de malware, llegándose a convertir en un tipo de afiliados que utilizan dicho código para ejecutar sus propios ataques (Jim Hodsworth, 2024).

Detectado por primera vez en el año 2022, maneja una técnica de doble extorsión, donde se exige a la víctima un pago por el descifrado de archivos y otro pago por evitar la divulgación de la información capturada. Este tipo de ransomware utiliza tácticas de ingeniería social para llegar a las víctimas, tales como el phishing y el email bombing, en las cuales, al lograr ingresar a los dispositivos, el objetivo principal es explotar las vulnerabilidades encontradas en los equipos. Como forma de ejemplo, existen registros de casos en los que se utiliza la plataforma Microsoft Teams para contactar con la victima ofreciendo algún tipo de soporte en el que se requiere la descarga de ciertas herramientas en donde se esconde este malware (Paliwal, 2025).

Una vez dentro de la red, Black Basta se caracteriza por ejercer tácticas de movimiento lateral, la misma que consiste en métodos utilizados por los atacantes para moverse entre una aplicación o dispositivo vulnerable hacia otro en busca de la escala de privilegios que le permita alcanzar su objetivo.

2.2.2.1. Herramientas Utilizadas por Black Basta. Black Basta dentro de los equipos utiliza las vulnerabilidades de las herramientas que encuentra en la red con distintos fines, una de estas son Power Shell, Cobalt Strike, Mimikatz y otras de acceso remoto. A continuación, se detalla la herramienta y el propósito por el cual es utilizado:

- **Acceso remoto:** Windows Management Instrumentation (WMI), WinSCP.
- **Movimiento lateral:** WMI, BTSAdmin, PsExect y SoftPerfect.
- **Acceso a credenciales y escalabilidad de privilegios:** Mimikatz.
- **Desactivar protecciones de seguridad:** WMI, PowerShell.
- **Exfiltración de datos:** WinSCP, WMI, RClone.

Con la exfiltración de los datos Black Basta procede al cifrado y a la eliminación de instancias de volumen con vssadmin.exe que impiden la recuperación del sistema (CISA, 2024).

2.2.2.2. Cifrado de Archivos que Utiliza Black Basta. Para el cifrado de archivos, Black Basta utiliza el algoritmo ChaCha20, el cual es un cifrado de flujo de 256 bits que se encarga de codificar toda la data seleccionada de tal forma que se vuelve ilegible para aquellos que no tienen la clave. En este caso los archivos son cifrados bajo una clave pública RSA-4096 que viene en el ejecutable, se agrega la extensión ". basta" y una nota de rescate en un ".txt" (Crawford, 2023).

2.2.2.3. Técnicas TTP Utilizadas por Black Basta.

- **T1059.003 - Command and Scripting Interpreter: Windows Command Shell.-** Esta técnica es identificada porque el atacante usa el intérprete de comandos del sistema operativo, como por ejemplo cmd.exe para poder enviar instrucciones de forma directa. Mediante el intérprete lanza comandos que modifican el sistema, eliminan copias de seguridad, cambian archivos de direcciones o ejecutan programas. Se usa porque genera menor sospecha y evade soluciones de antivirus y EDR (Command and Scripting Interpreter: Windows Command Shell (T1059.003). MITRE ATT&CK., 2024).

- **T1562.001 - Impair Defenses: Disable or Modify Tools.-** Con esta técnica las herramientas de seguridad son deshabilitadas o modificadas para así evadirlas. El atacante busca modificar claves de registro, usar binarios legítimos, detener servicios. En este caso para Black Basta intenta deshabilitar Windows Defender. Con la aplicación de esta técnica se aumenta la probabilidad de que el ransomware logre cifrar archivos sin ser detectado, y reduce la capacidad de recuperación o intervención del sistema (Brumm, 2021).

- **T1204 – User Execution.-** Para comenzar con la infección es necesario que el usuario ejecute de forma manual un archivo o aplicación maliciosa. Esta técnica usa ingeniera social con el objetivo de que la víctima descargue el programa y lo ejecute. Black Basta utiliza

esta técnica porque reduce la necesidad de vulnerabilidades explotables, basta con que la víctima confíe y ejecute el archivo (Brinkmann, 2016).

- **T1490 - Inhibit System Recovery. -** Esta técnica tiene como objetivo eliminar la capacidad de recuperación del sistema como: copias de sombra, servicios de respaldo o puntos de restauración. Se realiza por medio de comando nativos, tiene la finalidad impedir que la víctima pueda recuperar la información y obligarla a pagar rescate, ya que depende del atacante para la restauración de datos cifrados (Brandt A, 2020).

- **T1486 - Data Encrypted for Impact. -** Con el objetivo de pedir rescate se cifran los datos en los sistemas de la víctima. El ransomware recorre el sistema, identifica archivos, aplica una rutina criptográfica (en el caso de Black Basta aplica criptografía híbrida con RSA y ChaCha/ChaCha20), renombra archivos y genera extensiones específicas.

- **T1070 - Indicator Removal on Host.-** Técnica usada para borrar, modificar o manipular evidencia que permita rastrear la actividad del atacante en el sistema. Se pueden eliminar logs, archivos temporales hasta el punto de destruir archivos. Se realiza con el fin de dificultar la investigación posterior, evadir la respuesta a incidentes y ocultar rastros que permitan identificar la cadena completa de ejecución.

- **T1489 - Servicie Stop.-** Consiste en detener servicios esenciales del sistema operativo o de aplicaciones con el fin de eliminar obstáculos para la operación maliciosa. Esto se realiza mediante la ejecución de comandos o scripts para detener servicios relacionados con antivirus, copias de seguridad, bases de datos o sistemas críticos. Tiene la finalidad de garantizar que no existan bloqueos que interrumpan el cifrado, especialmente en servicios que mantienen archivos abiertos.

2.2.3. Artefactos en Windows. En la informática forense se utiliza el termino artefacto para referirse a todo tipo de evidencias o rastros dejados tras una actividad o un evento, los mismos que son fundamentales, ya que permiten determinar distintos detalles de dicha actividad, como el método, la herramienta, e incluso mediante esta huella, llegar a identificar el autor, la línea de tiempo y el propósito de la actividad.

En los sistemas operativos Windows se pueden considerar como artefactos a los objetos generados tras la realización de una actividad como la ejecución de un programa o un aplicativo, estos suelen ser logs, metadatos, archivos de caché, LNK, listas de accesos directos, entre otros, los cuales se puede almacenar en sitios como la papelera de reciclaje (INCIBE, 2025).

2.2.4. Artefactos de Black Basta. El ransomware Black Basta deja múltiples artefactos a su paso en los sistemas operativos Windows que pueden ser utilizados para análisis, entre los principales son:

Tabla 1

Artefactos principales de Black Basta

Artefacto	Descripción
.basta	Archivos cifrados y renombrados.
readme.txt	Nota de rescate.
.jpg	Imágenes de fondo de escritorio.
.ico	Íconos para archivos renombrados.
vssadmin.exe	Ejecutables para la eliminación de copias de seguridad.
netscan.exe, rclone.exe	Ejecutable para la recopilación de información.

Nota. Datos tomados del Cybersecurity and Infrastructure Security Agency (2024).

Existen otros tipos de artefactos que se esconden bajo nombres de utilidades inofensivas dentro directorios del sistema y que su función es hacer el llamado hacia herramientas de Black Basta (CISA, 2024).

2.2.5. El Análisis Forense en la Detección de Artefactos. Ante la inminente escalada de ataques producidos por ransomware hacia entidades de distinta índole, ha surgido la necesidad de implementar mejoras en los mecanismos de defensa como el uso de antivirus, firewalls y automatizaciones en actualizaciones del sistema operativo, además de buenas prácticas por parte del equipo de trabajo, sin embargo, estas no son garantía de protección y la comprensión de la amenaza se convierte en la mejor herramienta de defensa, la cual se la puede encontrar mediante el análisis forense.

El análisis forense permite la búsqueda, recuperación e identificación de datos, preparación y preservación de evidencias valiosas que contribuyen con la comprensión del comportamiento de un ransomware. Cada uno de los puntos que implica el análisis forense nos analizar los artefactos y nos acercará a nuestro objetivo de lograr diseñar un perfil mínimo del ransomware Black Basta (Franco, 2022).

Cuando se realiza un análisis forense en equipos con sistema operativo Windows los artefactos que se deberían revisar con relevancia son (Ibáñez, 2025):

- **Registros de eventos del sistema o "Event Logs":** algunos ficheros importantes a revisar son: System.evtx, Security.evtx y Application.evtx.
- **Hives del registro de Windows:** algunos ficheros importantes a revisar son: DEFAULT, SAM, SYSTEM, SOFTWARE, SECURITY.
- **Información del perfil de usuario, configuraciones, accesos y actividades:** algunos ficheros importantes a revisar son: NTUSER.DAT y USRCLASS.DAT. Para las

actividades del usuario se puede encontrar información en Jump Lists, MRU (Most Recently Used), ActivitiesCache.db.

- **Actividades de archivos y File System:** registro MFT (Master File Table), Journaling, $LogFile, y VSS (Volume Shadow Service).
- **Información de configuración de red, TCP/IP, firewall, RDP, Prefetch, DNS Caché, ARP, netstat:** hive del registro SYSTEM.
- **Servicios y puertos**
- **Volcados de memoria:** hiberfil.sys, pagefile.sys, archivos.dmp
- **Historial, cookies y caché del navegador**
- **Programas y persistencia:** revisión de Run keys, Prefetch Files, Amchache.hve
- **Archivos temporales y eliminados:** $Recycle.bin, Local/Temp, Thumbcache.

2.2.6. Herramientas de Análisis Dinámico y Forense. A continuación, se brinda una breve descripción de algunas herramientas que son implementadas en actividades de análisis forense de artefactos:

2.2.6.1. Virtual Box. Es un software de visualización y ejecución que permite la posibilidad de utilizar varios sistemas operativos tipo Windows, Linux, BSD, entre otros, todos dentro de un mismo ordenador como máquinas virtuales, facilitando la creación de entornos de desarrollo o laboratorios de prueba aislados (softonic, s.f.).

2.2.6.2. Kali Linux. Es una distribución de Linux gratuita basada en Debian que posee múltiples herramientas, configuraciones y scripts diseñados para llevar a cabo actividades de análisis forense, ingeniería inversa, detección y explotación de vulnerabilidades (g0tmi1k, 2025).

2.2.6.3. Ubuntu. Es una distribución GNU/Linux gratuita basada en Debian que puede instalarse de forma sencilla en equipos de escritorio o servidores, comúnmente empleado en el

ámbito educativo y entornos de desarrollo. Su interfaz gráfica es amigable, similar a la de Windows. A través de la terminal de Ubuntu, se pueden ejecutar comandos potentes para análisis forense. (Gobierno de Canarias, s.f.)

2.2.6.4. Sysmon. System Monitor o conocido por su abreviatura Sysmon, es un controlador de dispositivo que supervisa y registra la actividad del sistema dentro del registro de eventos de Windows, tales como creación de procesos, conexiones de red y cambios o modificaciones de archivos. Este tipo de registros permite la detección de actividades maliciosas, además del análisis y comprensión de una posible amenaza (Russinovich & Garnier, Learn Microsoft, 2024).

2.2.6.5. PowerShell. Es una automatización de tareas desarrollada por Microsoft que consiste en una consola de línea de comandos y un lenguaje de scripting, donde es posible automatizar y configurar la administración de sistemas, ejecutar scripts, manipular archivos, procesos y registros, además de interactuar con otros servicios (Learn Microsoft, 2025).

2.2.6.6. Wireshark. Es un analizador de protocolos de red gratuito que visualiza y examina el contenido de los paquetes existentes junto con su contenido a través de la captura de tráfico que se produce en una red, mostrando detalles como puertos, dirección IP, direcciones MAC, protocolos, horarios de actividad e incluso en ciertos casos gracias a la información que proporciona es posible determinar ubicaciones (INCIBE, 2025).

2.2.6.7. Velociraptor. Es una plataforma avanzada que permite el seguimiento de la línea de tiempo de un ataque, su inicio, situación actual y futuras amenazas, convirtiéndose en una herramienta poderosa para la búsqueda y extracción de evidencias, automatización de tareas forenses y respuesta a incidentes (Velociraptor, 2025).

2.2.6.8. Winpmem. Es una herramienta de código correspondiente al proyecto Rekall utilizada para adquirir un volcado de memoria RAM (Random Access Memory) de un equipo Windows. El proceso de volcado de memoria o "physical memory dump" es un copiado bit a bit del contenido de la RAM en un fichero binario de un dispositivo externo conectado al equipo (Ibáñez, 2025). Winpmem genera como resultado un archivo en formato AFF4, por omisión; sin embargo, permite usar otros formatos (entre ellos RAW) compatibles con otras herramientas (por ejemplo, Volatility puede analizar formatos RAW). El formato AFF4 es un archivo zip que contiene la RAM volcada, pero separada en ficheros e indexada (SOTHIS, 2025).

2.2.6.9. Autoruns. Es una herramienta gratuita de Microsoft de la suite Sysinternals, la cual entre sus funciones permite ver qué programas se inician al arrancar Windows para detectar presencia de malware, identificar tareas programadas dudosas, deshabilitar entradas innecesarias y exportar resultados para el posterior análisis forense. Esta herramienta entrega información que incluso no es visible a través del administrador de tareas nativo en Windows (Russinovich M. , Learn Microsoft, 2024).

2.2.6.10. Procmon. Process Monitor es una herramienta avanzada de supervisión para Windows que pertenece a la suite Sysinternals, cuya función permite monitorear la actividad del sistema de archivos en tiempo real, así como los procesos y subprocesos (Russinovich M. , Learn Microsoft, 2024). Process Monitor es muy usada para la búsqueda de malware y solución de problemas en sistemas Windows ya que facilita el filtrado de información por cualquier columna o campo de datos, y permite el seguimiento de procesos y sus relaciones con otros gracias a la herramienta de árbol de procesos.

2.2.6.11. Regshot. Esta utilidad de código abierto LGPL se encuentra alojada en el repositorio SourceForge para su descarga, permite capturar instantáneas del Registro de

Windows para poder comparar información antes y después de la ejecución de un programa o la instalación de un software determinado. De esta forma, se logra un seguimiento de la actividad para conocer si las claves y valores de los registros fueron añadidos, modificados o eliminados durante la ejecución de programas (en este caso, ejecución del ransomware). Su funcionamiento es sencillo, ya que a través de las opciones que presenta la ventana del programa se puede ejecutar “1st Shot”, después de algún cambio se procede con “2nd Shot”, y finalmente, la opción “Compare” generará un archivo con extensión .txt donde se indica un resumen de los cambios (Hendrikx, 2014).

2.2.6.12. Volatility. Es un framework de código abierto gratuito compatible con diversos sistemas operativos, utilizado para la extracción y el análisis de datos de memoria volátil. Esta herramienta contribuye en la identificación, recuperación y estudio de archivos eliminados o perdidos, haciendo posible la investigación de incidentes producidos por ataques de malware (VOLATILITY, s.f.).

2.2.6.13. Autopsy. Es una plataforma de código abierto diseñada para facilitar tareas de análisis forense, entre sus funciones se encuentra el recuperar, examinar y analizar archivos, correos electrónicos e historial web, además de contribuir con la detección de artefactos de malware y evidencias de ataques (AUTOPSY, 2025).

2.2.6.14. Reglas YARA. Son expresiones lógicas conformadas por cadenas, patrones, expresiones regulares y condiciones, que en conjunto construyen un criterio para detectar y analizar muestras de malware. Comúnmente son utilizadas para comparar en archivos y procesos, patrones sospechosos, siendo implementadas en el área de la ciberseguridad y análisis forense. Existe una plataforma de código abierto gratuito llamado YARA, disponible para su descarga y uso, dando a disposición algunas reglas YARA y permitiendo construir nuevas reglas,

sin embargo, existen otras herramientas que ya poseen integrados reglas YARA como lo son Autopsy y FTK (YARA, 2022).

2.2.6.15. Windows Process Hacker. Es una herramienta de administración de procesos para Windows y de análisis avanzado. Su función principal es inspeccionar, monitorear manipular procesos, hilos, servicios y uso de memoria (Russinovich & Solomon, 2017). Esta herramienta interactúa directamente con las APIs internas del sistema operativo, especialmente con el kernel el cual brinda la facilidad de acceder a información en tiempo real sobre actividades como la creación y terminación de procesos. En el ámbito forense es importante porque facilita la identificación de anomalías de comportamiento, inyecciones de código y ejecución de procesos encubiertos. A través de sus paneles de monitoreo, logs y capacidades de intervención directa, ofrece un entorno completo para estudiar en detalle la interacción entre aplicaciones y el sistema operativo (Finkel, 2007).

2.2.7. Herramientas de Análisis Estático. A continuación, se brinda una breve descripción de algunas herramientas que son implementadas en actividades de ingeniería inversa para análisis estático de malware:

2.2.7.1. Detect It Easy. DIE es un programa multiplataforma cuyo objetivo es determinar el tipo de archivo al que corresponde un fichero, admite más de 200 tipos de archivos para analizar. Esta herramienta puede ser usada en equipos con sistema operativo Windows, Linux, y MAC (KaliTools, 2021). Su instalación es sencilla a través del comando “sudo apt install detect-it-easy” en equipos con sistema operativo Linux. Se sugiere instalar tanto la versión CLI como la GUI. Al analizar un archivo la herramienta es capaz de detectar el compilador, enlazador, firmas, porcentaje de entropía, entre otras características del archivo (Brinkmann, 2016).

2.2.7.2. Ghidra. Esta herramienta de desensamblaje utilizada en ingeniería inversa fue desarrollada por NSA y lanzada en el año 2019, permite realizar análisis estático del malware debido a que no ejecuta código, sino que mapea el código de ensamblaje del malware; de esta manera el usuario puede avanzar y retroceder en el código sin afectar el sistema de archivos del equipo en el cual se está ejecutando el análisis (Fox, 2023).

2.2.8. ¿Qué es "Baseline" y "Snapshot"?. Un baseline se describe como un registro del estado actual del sistema que realiza el analista. Se utiliza como referencia del estado inicial o normal de un sistema para compararlo posteriormente cuando ha sucedido un incidente, en este caso una infección.

En este proyecto un baseline es de utilidad para comparar y analizar qué archivos han cambiado tras la ejecución del ransomware; de la misma manera, ayudan a verificar qué claves de registros se modificaron, si se detectan nuevos procesos, o cualquier alteración a través del tiempo. El baseline debe generarse antes del ataque, una vez que se hayan instalado todas las herramientas y se haya asilado la red completamente. Después de tener el baseline, es importante ejecutar un snapshot.

Aunque los términos "baseline" o "snapshot" parecen significar lo mismo, no lo son, sino que en análisis forense se los usa en forma complementaria. A diferencia del baseline, un snapshot es una captura o fotografía del estado de un sistema virtualizado en un tiempo determinado, congelando la RAM, el estado del disco, las configuraciones del equipo, entre otras características, y se lo realiza a través de una opción dentro del hipervisor empleado para la virtualización. Un snapshot es útil para retroceder el estado de la máquina virtual al punto exacto en que fue tomada su instantánea, en este caso antes de la infección; dicho de otra manera, permite la rápida restauración del entorno virtual en el cual se simulan incidentes de seguridad.

2.2.9 Servidores Web y Arquitecturas de Manejo de Peticiones. De forma general los servidores web implementan modelos de concurrencia que pueden basarse en procesos, hilos o eventos. Es una aplicación que se encarga de recibir, procesar y responder solicitudes HTTP. Algunos modelos de procesamiento que se usan son los siguientes: Multiproceso, Multihilo, Basado en eventos.

2.2.9.1. Nginx. Es un servidor web y proxy reverso de alto rendimiento caracterizado por su arquitectura *event-driven* y *non-blocking*. Su diseño permite manejar miles de conexiones simultáneas utilizando pocos recursos. A continuación, se enlistan algunas de sus características:

- Balanceo de carga.
- Servidor de archivos estáticos.
- Proxy reverso para HTTP, HTTPS, SMTP.
- Integración con aplicaciones externas mediante FastCGI, uWSGI, SCGI o gRPC.
- Manejo de conexiones con latencia baja.
- De gran ayuda para arquitecturas de microservicios y contenedores.

2.2.9.2. Fcgiwrap. Es un programa ligero que trabaja como un wrapper entre un servidor web y scripts CGI convencionales presentando como aplicaciones FastCGI para permitir su ejecución eficiente y segura desde servidores que no implementan directamente el antiguo modelo CGI (Shklar & Rich). Se ejecuta como un proceso el cual recibe peticiones FastCGI, por cada una de estas prepara un entorno, invoca el script CGI correspondiente y devuelve la salida al servidor el cual a su vez la reenvía al cliente HTTP. Al ser un programa ligero no gestiona pools complejos ni reintentos lo que facilita su integración en entornos limitados.

Capítulo 3

3. Desarrollo

3.1. Desarrollo del Trabajo

Con la finalidad de obtener un perfil mínimo de detección del ransomware Black Basta basado en su comportamiento dentro de un sistema operativo Windows, se replicó mediante un laboratorio virtual un escenario muy similar a un entorno real en donde Black Basta operaría; de esta forma, se obtuvieron los artefactos necesarios para el análisis.

El laboratorio creado no solo permite reconstruir el ataque de manera didáctica y segura, sino que sirve como fundamento para generar un **Perfil Mínimo de Detección (PMD)**, el cual se construye a partir de señales forenses reales obtenidas durante la ejecución controlada del ransomware. La construcción del laboratorio mencionado se detalla a largo de esta sección.

3.1.1. Escenario Planteado Para Infección del Usuario. En un día laboral ordinario, un empleado recibe un correo aparentemente legítimo con un enlace para descargar un “Instalador actualizado de Microsoft Teams”. Ante la urgencia de conectarse a una reunión accede al sitio falso, sin notar los indicios de suplantación, exponiéndose a una cadena típica de tácticas MITRE ATT&CK: **Reconocimiento (T1592)**, **Ingeniería social (T1566.002)**, **Suplantación de sitios web (T1584.004)** y **Ejecución de binarios maliciosos (T1204.002)**.

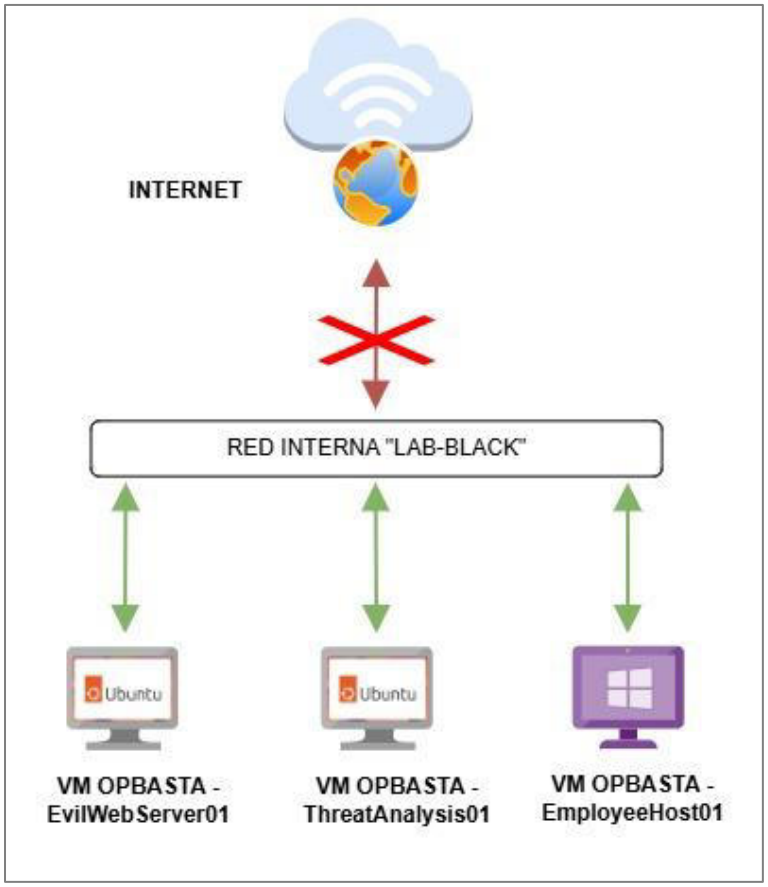
Cuando el usuario da clic en “Descargar”, obtiene un ejecutable que parece legítimo, pero que corresponde a una muestra del ransomware Black Basta. Al ejecutar el archivo descargado, los siguientes procesos maliciosos se despliegan en segundos: eliminación de copias de volumen (T1490), modificación de configuraciones del sistema (T1112), cifrado masivo de archivos (T1486) y creación de notas de rescate (T1491). Mientras el ransomware se despliega sobre el

equipo de la víctima herramientas como Sysmon, Procmon y Velociraptor registran cada evento clave para su posterior análisis.

3.1.2. Topología del Entorno de Laboratorio. Para el análisis forense del ransomware Black Basta, se diseñó un entorno controlado y completamente aislado con el objetivo de simular un escenario de ataque realista sin comprometer sistemas externos. Todas las máquinas virtuales fueron desplegadas en el software VirtualBox, utilizando una única red interna denominada **LAB-BLACK**, lo que garantizó el confinamiento total del tráfico. Con la ayuda de esta herramienta se levantaron 3 máquinas virtuales que simulaban los siguientes componentes:

Figura 1

Topología del Entorno de Laboratorio (VMs en Red Interna)



- **VM OPBASTA - EvilWebServer01:** Servidor malicioso (OS: Ubuntu) que aloja payload disfrazado como software legítimo. Contendrá diferentes muestras de Ransomware "Black Basta".
- **VM OPBASTA - ThreatAnalysis01:** Servidor de análisis (OS: Ubuntu). Centro de análisis forense y telemetría, control del ataque y recolección.
- **VM OPBASTA - EmployeeHost01:** Representa la máquina víctima (OS: Windows 10 / 11), simula empleado el cual descarga un instalador (muestras de Black Basta) fraudulento desde web maliciosa.

3.1.3. Componentes del Laboratorio y Justificación Técnica. Cada uno de los componentes implementados en este laboratorio fueron creados con el propósito de simular un escenario lo más real posible. A continuación, se explica brevemente la función que tiene cada componente.

3.1.3.1. VM OPBASTA - EvilWebServer01 (Servidor Malicioso). Con el objetivo de reproducir un escenario realista de infección por ransomware, mediante técnicas de ingeniería social, se implementó un servidor web virtualizado que simula un portal legítimo de descarga de software corporativo (en este caso, Microsoft Teams). Este servidor forma parte fundamental del entorno controlado del laboratorio, ya que permite replicar el vector inicial de compromiso asociado a múltiples campañas de distribución de ransomware, particularmente las relacionadas con T1566.002 (Phishing vía archivo malicioso) y T1204.002 (User Execution) dentro del marco MITRE ATT&CK.

La implementación del servidor web no solo permite simular de manera fiel la fase inicial de una infección típica por ransomware, sino que introduce un mecanismo flexible para analizar

de forma sistemática múltiples ejecuciones del malware bajo condiciones homogéneas. Esto mejora significativamente la calidad del proceso analítico, puesto que:

- Los datos recolectados por Sysmon, Procmon, Velociraptor provienen de muestras diferentes, pero bajo el mismo entorno operativo.
- Los patrones de comportamiento recurrentes pueden ser identificados con mayor claridad, fortaleciendo la validez del PMD resultante.
- La infraestructura creada es fácilmente ampliable, pudiendo incorporar más muestras en el futuro sin alterar el diseño general del laboratorio.

En conjunto, el servidor web malicioso se convierte en un componente esencial del laboratorio, asegurando realismo, control, seguridad y capacidad de reproducción, cuatro elementos clave para la construcción de un Perfil Mínimo de Detección robusto y sustentado empíricamente.

Adicionalmente, se pueden mencionar dos capacidades críticas que ofrece el servidor web para el desarrollo del laboratorio:

- **Distribución controlada y segura de múltiples muestras del ransomware Black Basta:** La plataforma aloja varias muestras del malware, seleccionadas y almacenadas previamente en un entorno aislado. Mediante un mecanismo de descarga aleatoria vinculado al botón principal del sitio falsificado, el entorno permite que cada ejecución del laboratorio utilice una muestra distinta del ransomware; lo cual, facilita el análisis comparativo del comportamiento entre diferentes variantes o builds de Black Basta; permite observar cómo pequeñas diferencias en los binarios pueden modificar la telemetría registrada; y, enriquece el proceso de elaboración del Perfil Mínimo de Detección (PMD) al basarse en múltiples ejecuciones no determinísticas.

- **Eliminación de la necesidad de descargas externas y aumento del aislamiento del laboratorio:** El laboratorio opera en un entorno totalmente desconectado de Internet, garantizando que las muestras del ransomware no se extraigan, filtren ni propaguen accidentalmente. El servidor web cumple así un doble propósito, primero, actuar como fuente interna de distribución de malware, reduciendo el riesgo operativo y elevando la seguridad del experimento; y segundo, permite reproducir tantas ejecuciones como sea necesario sin depender de repositorios externos como MalwareBazaar o VX-Underground. Esto convierte al laboratorio en un entorno autónomo, repetible y seguro, condiciones indispensables para experimentación con malware real a nivel académico.

3.1.3.2. VM OPBASTA - ThreatAnalysis01 (Máquina de Análisis). Esta máquina permite la instalación de herramientas de recolección (Velociraptor en modo Servidor), centralización, procesamiento, correlación y análisis de la evidencia extraída durante la ejecución del ransomware realizada en la máquina víctima. Además, permite llevar a cabo análisis estático a través de las herramientas Ghidra y Detect It Easy.

Por lo tanto, la relevancia de VM OPBASTA - ThreatAnalysis01 dentro del proyecto puede resumirse en los siguientes puntos clave:

- **Centraliza el análisis:** Toda la evidencia que se recolecta de la máquina víctima durante el ejercicio de análisis dinámico se combina con los resultados de un análisis estático para obtener hallazgos fáciles de correlacionar.
- **Permite organización del flujo de trabajo:** Esto reduce errores por corrupción de la evidencia y alteración de la línea temporal; mejora la calidad del análisis.

3.1.3.3. VM OPBASTA - EmployeeHost01 (Máquina Víctima). Esta máquina representa el activo víctima dentro del laboratorio simulado de infección y análisis de ransomware Black Basta. Su función central es proporcionar un entorno controlado que emule fielmente el comportamiento de un equipo corporativo estándar operado por un usuario final, el cual es objetivo habitual en campañas reales de ingeniería social.

Este host constituye el punto de ejecución del payload malicioso, permitiendo analizar en profundidad la progresión del ataque desde la etapa inicial hasta el cifrado, los cambios realizados en el sistema operativo, los mecanismos de persistencia, los indicadores de comportamiento relevantes para la construcción del Perfil Mínimo de Detección (PMD) del ransomware y la telemetría generada por herramientas como Sysmon, Procmon y Velociraptor.

La correcta implementación de esta máquina virtual garantiza que los resultados generados en el laboratorio sean fiables, reproducibles y comparables, condiciones indispensables para el estudio del comportamiento del ransomware real. Esta VM constituye el eje central del laboratorio, debido a que representa al equipo víctima en un escenario de infección realista por ransomware. Su diseño y preparación no responden únicamente a criterios funcionales, sino a la necesidad de construir un ambiente altamente controlado apto para el análisis forense.

Desde una perspectiva metodológica, se convierte en la fuente primaria de evidencia empírica para la formulación del Perfil Mínimo de Detección (PMD). Cada acción realizada por el ransomware (procesos creados, archivos modificados, llaves de registro alteradas, hilos lanzados, conexiones de red establecidas) queda registrada mediante las herramientas previamente instaladas: Sysmon, Procmon, Process Hacker, Regshot y el agente de Velociraptor.

Estas herramientas mencionadas permiten visualizar los diferentes planos de interacción del malware con el sistema operativo, aportando así una visión multidimensional del ataque, condición imprescindible para identificar los patrones de conducta más estables y robustos entre diferentes ejecuciones.

Por lo tanto, la relevancia de VM OPBASTA - EmployeeHost01 dentro del proyecto puede resumirse en los siguientes puntos clave:

- **Simulación realista del endpoint corporativo comprometido:** La máquina reproduce fielmente un entorno similar al que emplearía cualquier empleado de una organización. Esto permite observar cómo el ransomware se comporta en un entorno estándar, capturar telemetría auténtica (libre de alteraciones propias de sistemas extremadamente endurecidos) e identificar vectores, artefactos y persistencias que solo emergen en entornos genuinos.
- **Captura integral de la actividad del ransomware:** Al combinar herramientas complementarias (Sysmon para telemetría profunda, Procmon para eventos a nivel kernel, Regshot para cambios en el registro, etc.), el sistema es capaz de registrar la secuencia completa del ataque (Kill Chain), modificaciones estructurales al sistema, artefactos persistentes y efímeros, e indicadores de compromiso que serían difíciles de detectar de forma aislada. Todo esto permite obtener un registro exhaustivo de eventos que sirven para fundamentar el PMD con evidencia sólida.
- **Entorno seguro y controlado para ejecutar malware real:** La máquina está configurada para operar aislada de Internet, evitando la filtración accidental del malware. No se encuentra activado Windows Defender ni protecciones adicionales, permitiendo observar el comportamiento natural del ransomware sin interferencias, y con herramientas instaladas en

directorios estratégicos que minimizan el riesgo de cifrado accidental de los instrumentos de análisis; lo cual permite realizar múltiples ejecuciones, comparar variaciones de comportamiento y estudiar el impacto del ransomware en diferentes escenarios de forma segura.

- **Generación de telemetría para Velociraptor:** La máquina víctima no solo ejecuta el malware, sino que se integra con el servidor de Velociraptor; esto permite correlacionar eventos a nivel de endpoint con los eventos a nivel de red o EDR, además, almacena toda la telemetría en un repositorio central para su análisis. Con la información que se recolecta de este equipo se construyen reglas de detección basadas en rasgos invariantes del ransomware. EmployeeHost01 es el punto donde inicia toda la cadena de análisis.
- **Repetibilidad y consistencia de ejecuciones:** Gracias al snapshoting y la automatización del entorno cada ejecución del ransomware puede repetirse bajo condiciones idénticas; lo cual permite analizar múltiples muestras de Black Basta sin alterar la integridad del sistema. También se puede decir que aporta rigor científico al proyecto, garantizando que los resultados no dependen de factores externos o al azar.

3.1.4. Herramientas Implementadas. En la siguiente tabla se enlista las herramientas utilizadas en cada máquina.

Tabla 2

Herramientas implementadas en cada máquina virtual del Laboratorio

VM OPBASTA - EvilWebServer01	VM OPBASTA - ThreatAnalysis01	VM OPBASTA - EmployeeHost01
Nginx	Velociraptor (Servidor)	Sysmon
Fcgiwrap	Ghidra	Regshot
	Detect It Easy (DIE)	Procmon

Process Hacker
Velociraptor (Agente)
Winpmem

Nota. En cada columna se enlistan los nombres de las herramientas instaladas de acuerdo con el propósito de cada máquina.

3.1.4.1. Scripts Utilizados Para Recolección de Evidencias. Como valor agregado, y a fin se ejecutar pruebas simultáneas de las muestras del ransomware Black Basta para estudiar su comportamiento, se montó un laboratorio paralelo fiel al diseño original propuesto en la sección 3.1.2. Topología del Entorno de Laboratorio.

En este laboratorio paralelo, se automatizó la recolección de evidencias a través de scripts PowerShell personalizados para capturar artefactos volátiles y no volátiles generados por el ransomware Black Basta, y transferirlos de forma que la evidencia no se altere. Los scripts creados son:

- **Script para captura de archivos PML:** Este script automatiza el uso de Process Monitor (Procmon.exe), una herramienta esencial para capturar actividad de bajo nivel. De forma general realiza las siguientes instrucciones: verifica la presencia de Procmon, genera un nombre para un archivo de extensión .pml usando marca de tiempo, inicia Procmon de manera oculta, espera 60 segundos hasta que se ejecute el comportamiento inicial malware y capturarlo, y al final, detiene Procmon.

```
$procmon = "C:\Tools\Procmon.exe"
if (-not (Test-Path $procmon)) {
    Write-Error "Procmon.exe no encontrado en $procmon. Ajusta la
ruta."
    exit 1
```

```
}
$time = (Get-Date).ToString("yyyyMMdd_HHmmss")
$outPml = "\\10.10.10.20\forensic\procmon_$time.pml"
# Start Procmon capturing quietly and write to backing file
Start-Process -FilePath $procmon -ArgumentList
"/BackingFile",$outPml,"/AcceptEula","/Quiet" -WindowStyle Hidden -PassThru |
Out-Null
Start-Sleep -Milliseconds 500
Write-Output "Procmon started. Run the ransomware NOW. Will stop
capture automatically in 60 seconds."
# Wait enough time for ransomware to run and generate activity
Start-Sleep -Seconds 60
# Stop Procmon (safe) and ensure file flushed
Start-Process -FilePath $procmon -ArgumentList "/Terminate" -Wait -
WindowStyle Hidden
Write-Output "Procmon stopped. PML should be at $outPml"
```

- **Script para captura de volcado de memoria:** Este script implementa una técnica de intercepción de ejecución, para evitar que Black Basta cifre la información capturada por la herramienta Winpmem antes de que finalice su volcado de memoria. El script está basado en una ventana crítica de 1.2 segundos, obtenido como resultado del análisis iterado del comportamiento del ransomware. En el tiempo mencionado, se descubrió que era el punto exacto en el que Black Basta había completado su rutina de inicialización, pero aún no iniciaba el bucle masivo I/O que bloquea el sistema. Por lo tanto, este script permite obtener el archivo MEM_1200ms.raw, que contiene el volcado íntegro del sistema con el proceso blackbasta.exe activo y con su carga útil expuesta para el análisis posterior.

```
# Configuración de Camuflaje y Red
$SafeToolPath = "C:\Windows\sysdump.exe"
```

```
$MalwareName = "blackbasta"
$RemoteShare = "\\10.10.10.20\forensic"
# Definición de API Nativa para Suspensión de Hilos (P/Invoke)
$code = @"
using System;
using System.Runtime.InteropServices;
public class Utils {
    [DllImport("kernel32.dll")] public static extern IntPtr
OpenThread(int dwDesiredAccess, bool bInheritHandle, int dwThreadId);
    [DllImport("kernel32.dll")] public static extern int
SuspendThread(IntPtr hThread);
    [DllImport("kernel32.dll")] public static extern int
CloseHandle(IntPtr hObject);
}
"@
Add-Type -TypeDefinition $code
# Bucle de Detección y Respuesta
while ($true) {
    $target = Get-Process -Name $MalwareName -ErrorAction
SilentlyContinue
    if ($target) {
        $id = $target.Id
             # VENTANA CRÍTICA: 1.2 Segundos
        # Tiempo calculado para permitir el "Unpacking" en memoria
        Start-Sleep -Milliseconds 1200
                # 1. Congelamiento del Malware (Stop Execution)
        $target.Threads | ForEach-Object {
            $hThread = [Utils]::OpenThread(0x0002, $false, $_.Id)
            [Utils]::SuspendThread($hThread)
```

```
                [Utils]::CloseHandle($hThread)
            }
             # 2. Conexión Segura y Volcado
            net use Z: $RemoteShare /user:Guest "" | Out-Null
            $out = "Z:\MEM_1200ms_$(Get-Date -Format 'HHmmss').raw"
            Start-Process -FilePath $SafeToolPath -ArgumentList "$out" -
Wait
            break
        }
        Start-Sleep -Milliseconds 50
    }
```

- **Script para captura de EVTX:** Este script implementa una técnica de captura de logs de eventos registrados en Windows, antes de que Black Basta termine de cifrar los archivos o alterar registros internos. De manera general, este script crea una carpeta de destino con marca de tiempo, define un conjunto de logs como objetivo, identifica el nombre real de cada log del sistema, ejecuta la muestra del ransomware y espera 8 segundos para que Black Basta se ejecute; al finalizar, exporta cada log detectado.

```
$sample = "C:\Users\isra\Downloads\m1\blackbasta.exe"
$destRoot = "\\10.10.10.20\forensic"
$timestamp = Get-Date -Format "yyyyMMdd_HHmmss"
$dest = Join-Path $destRoot "evtx_$timestamp"
New-Item -ItemType Directory -Path $dest -Force | Out-Null
# Logs que se quieren obtener, pero con búsqueda flexible
$targets = @(
    "Application",
    "Security",
    "System",
```

```
    "Sysmon",
    "PowerShell"
)

Write-Host "Detectando nombres reales de logs..."
$available = wevtutil el
$resolved = @{}
foreach ($t in $targets) {
    $match = $available | Where-Object { $_ -match $t }
    if ($match) {
        Write-Host "? Encontrado para '$t': $match"
        $resolved[$t] = $match
    } else {
        Write-Host "? No existe ningún log que coincida con '$t'"
    }
}
Write-Host "`nEjecutando BlackBasta..."
Start-Process $sample
Start-Sleep -Seconds 8
Write-Host "`nExportando logs existentes..."
foreach ($key in $resolved.Keys) {
    foreach ($logName in $resolved[$key]) {

        $san = $logName.Replace("/", "_")
        $outfile = Join-Path $dest "$san.evtx"
        try {
            wevtutil epl "$logName" "$outfile"
            Write-Host "? Exportado -> $logName"
        }
```

```
            catch {
                Write-Host "? Falló -> $logName"
            }
        }
    }
    Write-Host "`nExportación completada en: $dest"
```

3.1.5. Muestras Black Basta. Con la finalidad de lograr obtener la mayor cantidad de evidencias que permitan cumplir con los objetivos planteados, se decidió trabajar con las siguientes cinco muestras del ransomware Black Basta:

Tabla 3

Lista de Hash SHA-256 de cinco muestras de ransomware Black Basta Utilizadas

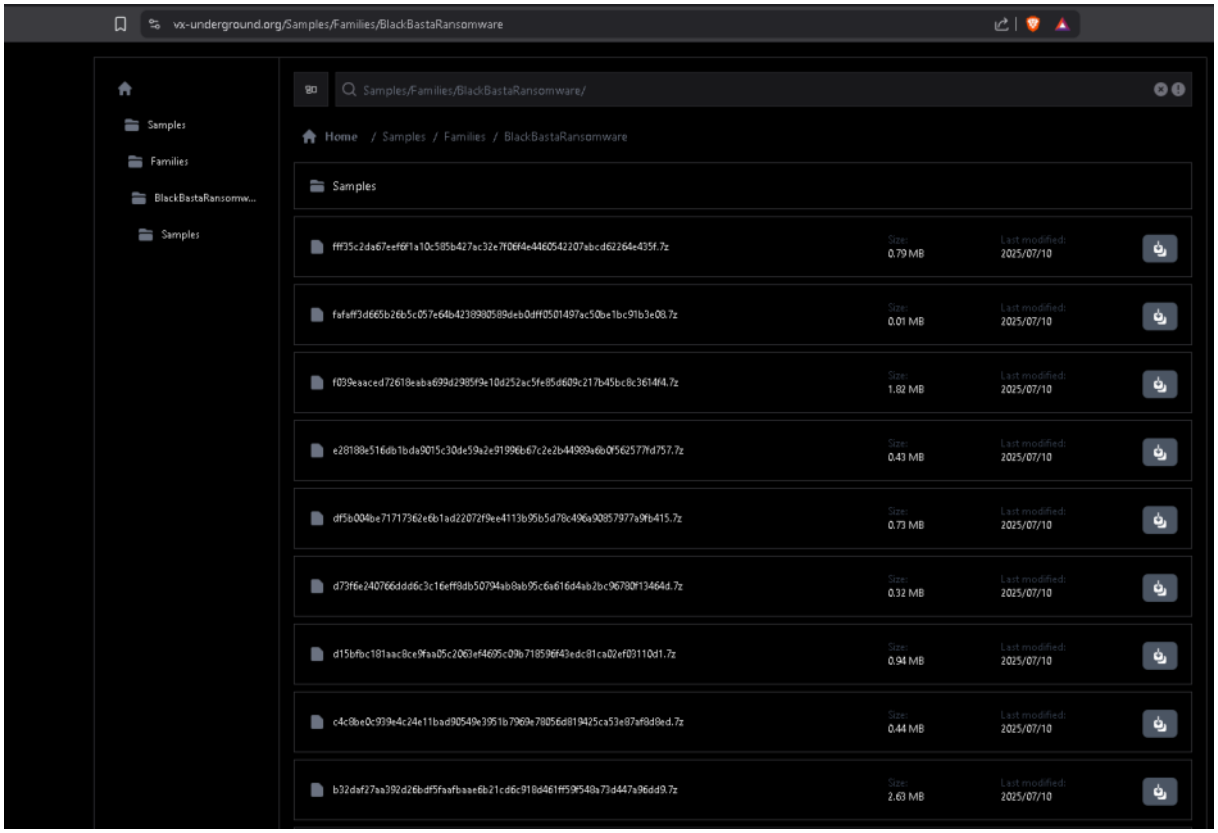
Nº Muestra	Hash SHA-256 de Muestra Del Ransomware Black Basta
M1	df5b004be71717362e6b1ad22072f9ee4113b95b5d78c496a90857977a9fb415
M2	e28188e516db1bda9015c30de59a2e91996b67c2e2b44989a6b0f562577fd757
M3	f039eaaced72618eaba699d2985f9e10d252ac5fe85d609c217b45bc8c3614f4
M4	fafaff3d665b26b5c057e64b4238980589deb0dff0501497ac50be1bc91b3e08
M5	fff35c2da67eef6f1a10c585b427ac32e7f06f4e4460542207abcd62264e435f

Nota. En la primera columna se indexa el número de muestra y en la segunda columna se indica el nombre del archivo de muestra usado correspondiente a su Hash SHA-256.

Estas cinco muestras fueron extraídas del sitio web vx-underground.org de manera aleatoria y considerando que la fecha sea lo más reciente posible.

Figura 2

Selección de muestras de ransomware Black Basta desde el sitio web vx-underground.org



3.1.6. Medidas de Seguridad Implementadas.

Como medidas de seguridad para la contención del ransomware Black Basta se diseñó un entorno controlado y completamente aislado de máquinas virtuales, sin comprometer sistemas externos, y sin acceso a Internet. Todas las VMs fueron desplegadas en el software VirtualBox, utilizando una única red interna denominada LAB-BLACK, lo que garantizó el confinamiento total del tráfico.

En el equipo físico que hospeda las máquinas virtuales se procedió a instalar un antivirus de pago, como buena práctica de seguridad, para asegurar la protección de este equipo en tiempo real, en caso de algún error humano.

En la configuración de las máquinas virtuales que conforman el entorno de laboratorio se desactivaron integraciones con VirtualBox como: Portapapeles compartido, Drag & Drop, Carpetas compartidas, Features Guest Additions.

Para el análisis de las evidencias, se utilizó una VM Ubuntu y no se empleó el propio Host.

3.1.7. Despliegue del Laboratorio. Como primera instancia, se crearon las máquinas virtuales a utilizar, ya descritas en apartados anteriores.

A continuación, se muestra el paso a paso de la descarga, instalación y configuración de cada máquina junto con sus respectivas herramientas.

3.1.7.1. VM OPBASTA - EvilWebServer01 (Servidor Malicioso). Para la simulación de la web fraudulenta se usó una máquina virtual con OS Ubuntu Server. Esta máquina fue nombrada como OPABASTA – EvilWebServer01.

Figura 3

Instalación del sistema operativo Ubuntu para VM OPBASTA - EvilWebServer01

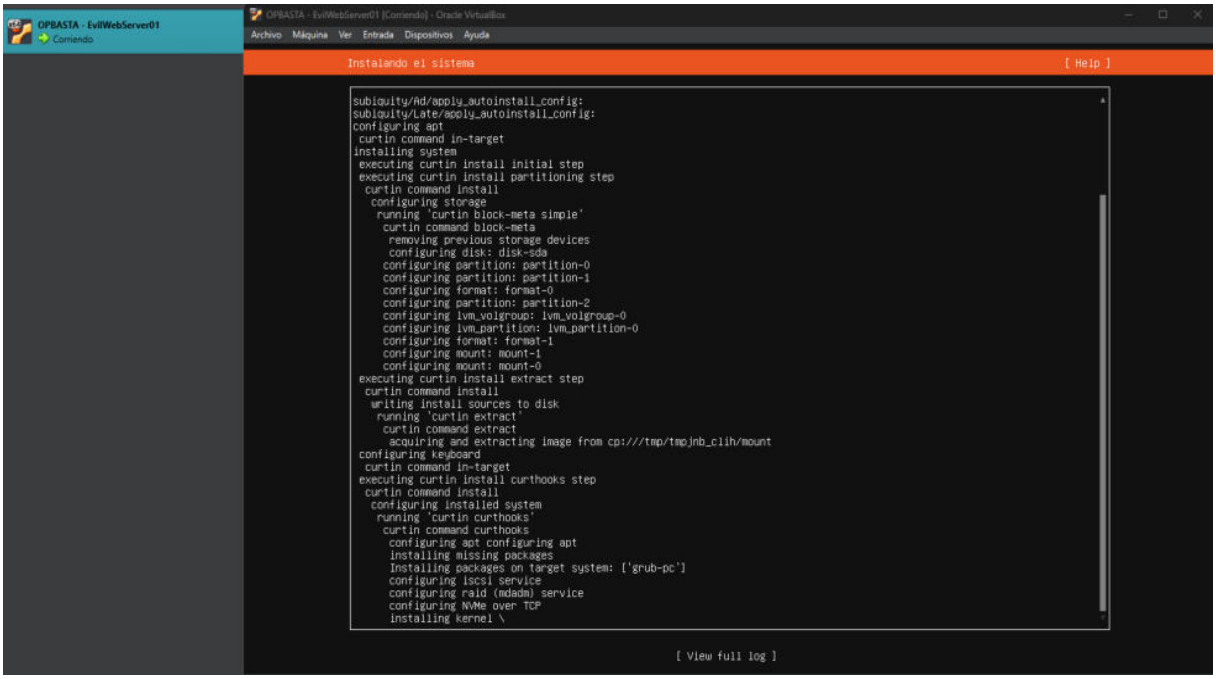
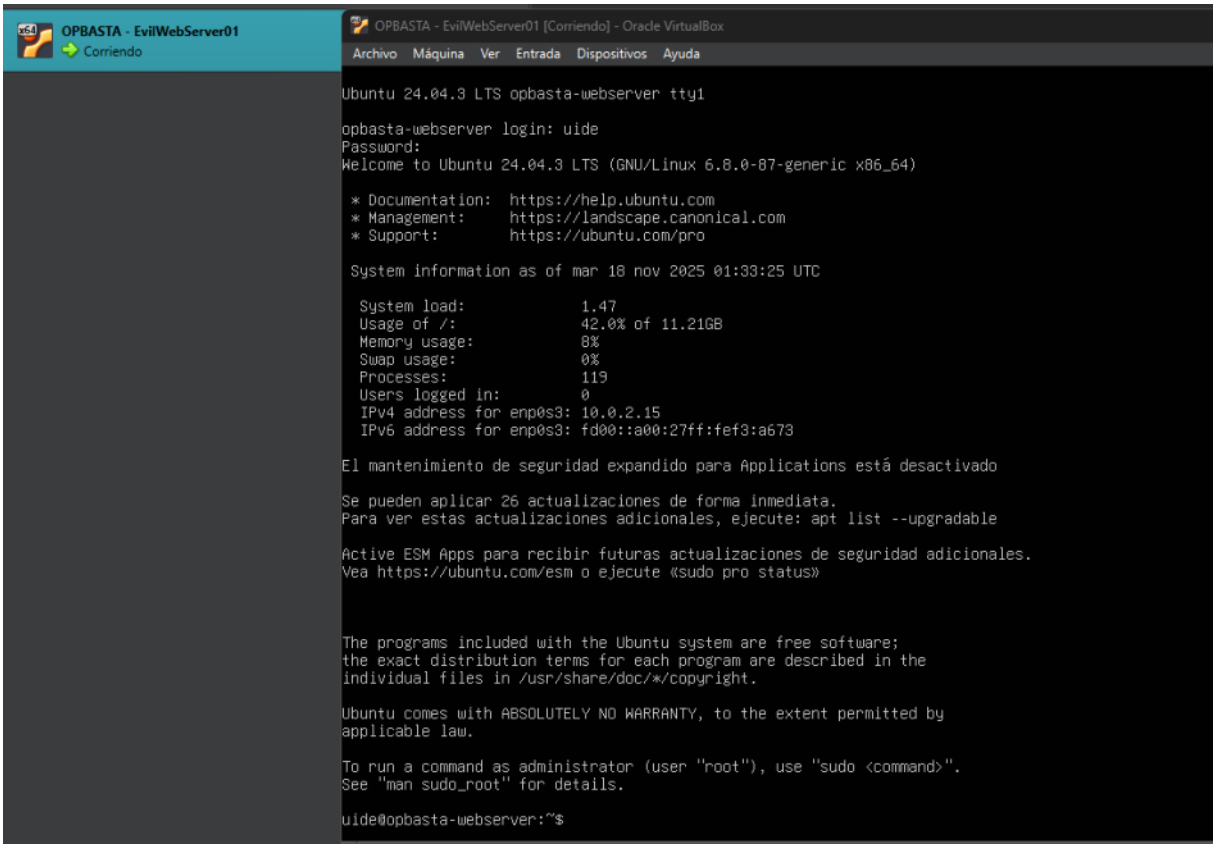


Figura 4

Instalación finalizada del sistema operativo Ubuntu para VM OPBASTA - EvilWebServer01

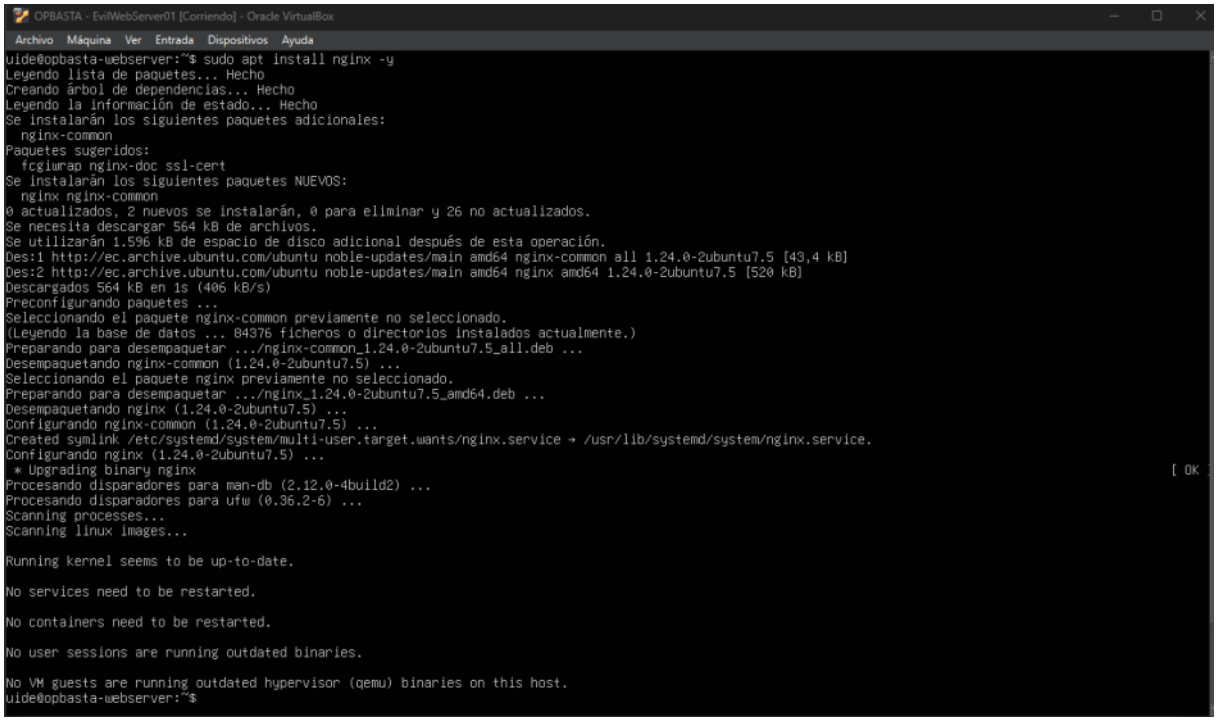


A continuación, se instaló **nginx** para alojar los recursos necesarios de la web maliciosa simulada, utilizando los siguientes comandos:

```
sudo apt update

sudo apt install nginx -y
```

Figura 5

Instalación de herramienta nginx en VM OPBASTA - EvilWebServer01

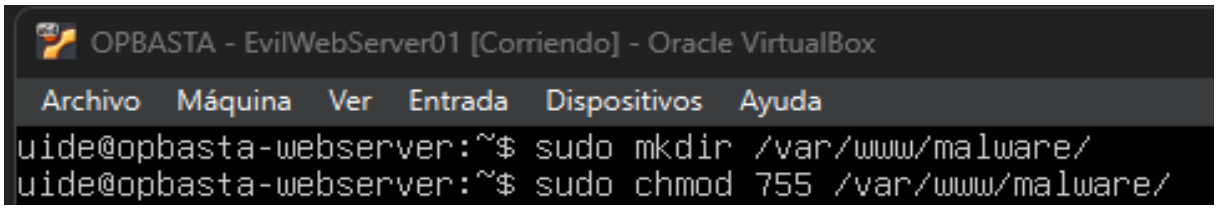


Luego, se creó el directorio en donde se ubican las muestras del malware “Black Basta”, utilizando los siguientes comandos:

```
sudo mkdir -p /var/www/malware/

sudo chmod 755 /var/www/malware/
```

Figura 6

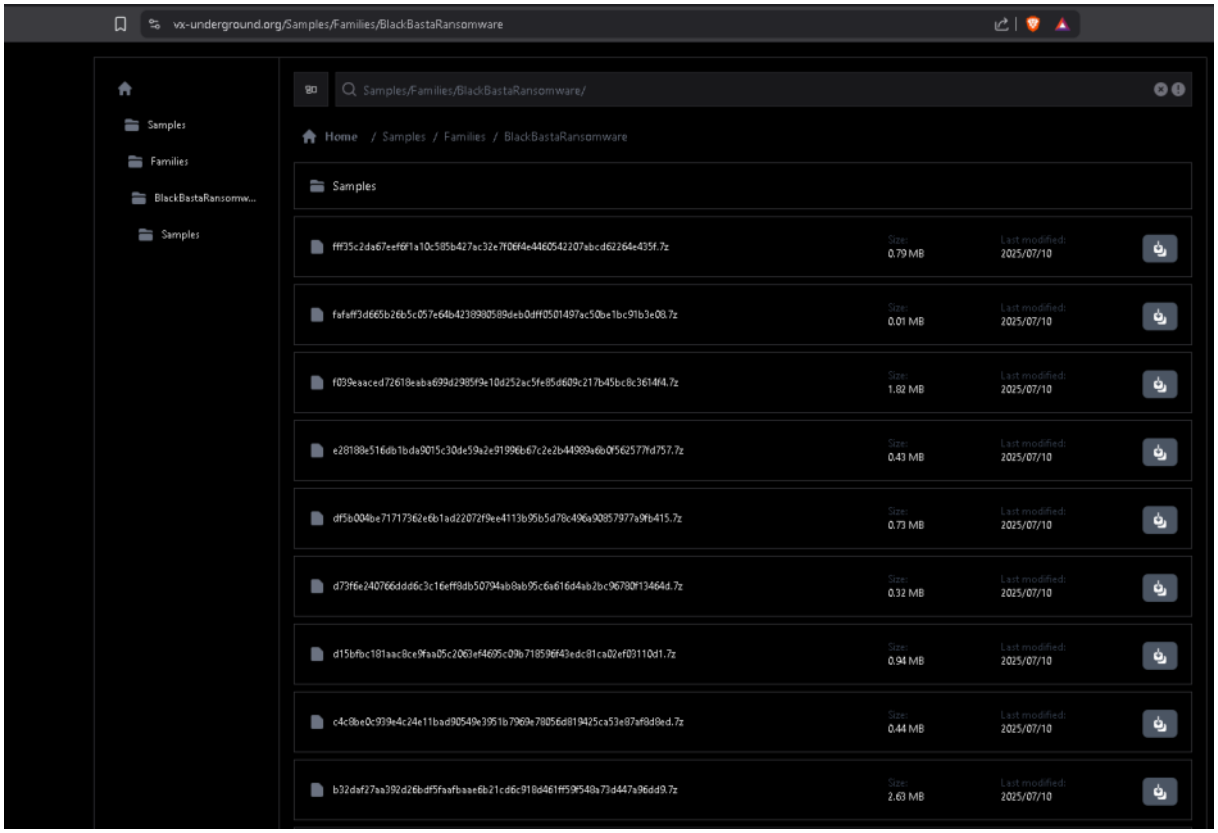
Creación de directorio que aloja las muestras



Se procedió con la elección aleatoria de muestras de malware “Black Basta” usando la famosa web “VX-Underground”.

Figura 7

Repositorio de muestras de Black Basta



Lo siguiente fue transferir los archivos al servidor ubicándolos en el directorio anteriormente creado. Adicionalmente, se renombran estos archivos para que la simulación sea mucho más legítima.

Figura 8

Archivos de muestras transferidos de directorio

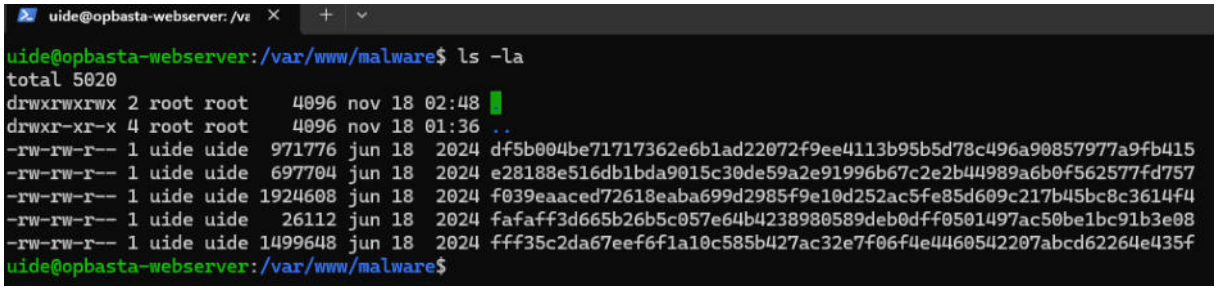
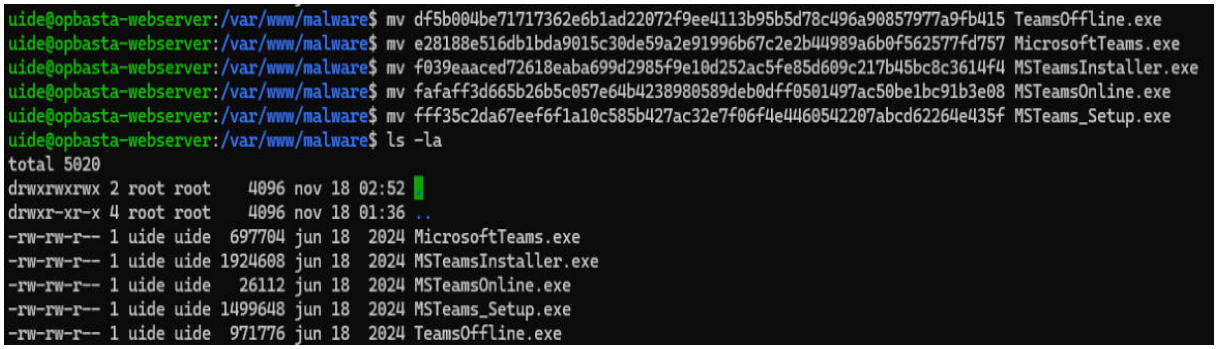


Figura 9

Archivos de muestras renombradas en el directorio creado

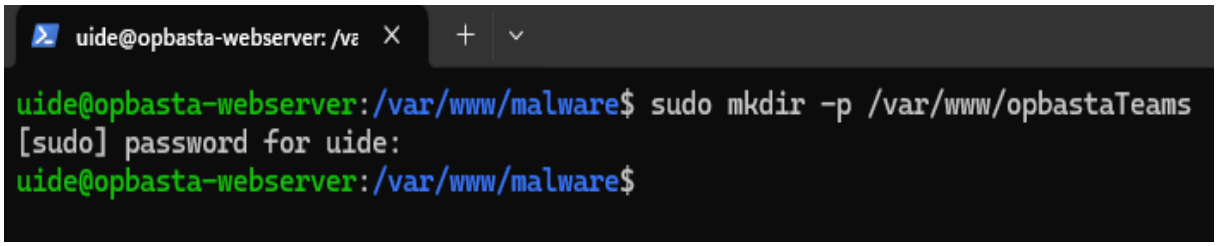


Seguidamente, se creó el directorio raíz de la web falsificada, utilizando el comando:

```
sudo mkdir -p /var/www/opbastaTeams
```

Figura 10

Creación de directorio destinado a la web maliciosa



Se usó la extensión “SingleFile” disponible en Chrome para descargar una versión idéntica del front de la página web a falsificar; en este laboratorio se usó la interfaz de descarga de Microsoft Teams. Una vez que se descargó el index.html, se movió la ubicación hacia el directorio raíz de la web.

Figura 11

Interfaz de Microsoft Teams a falsificar

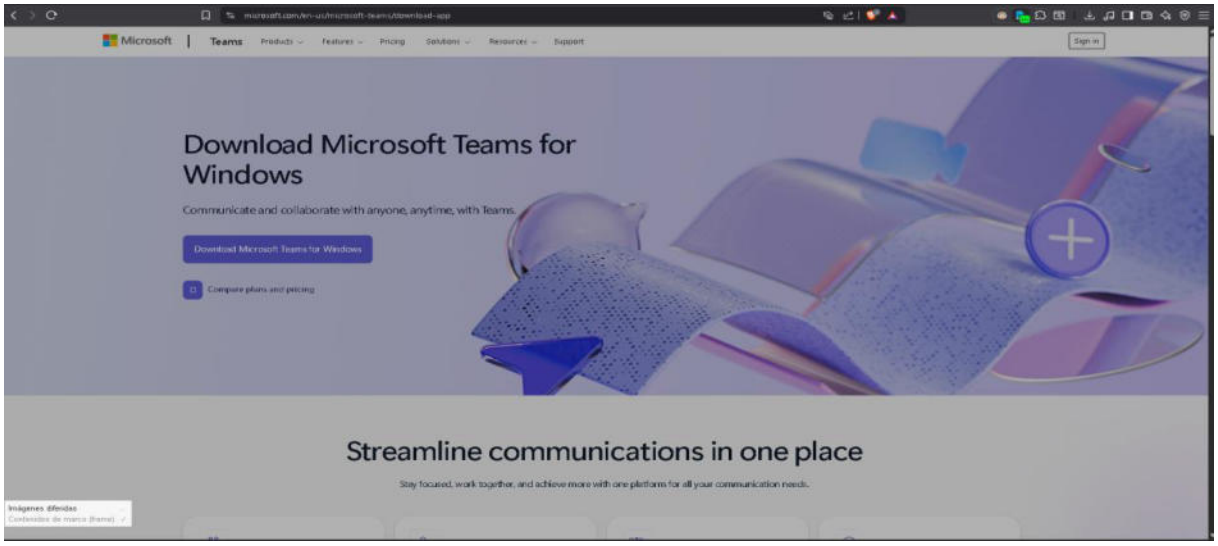
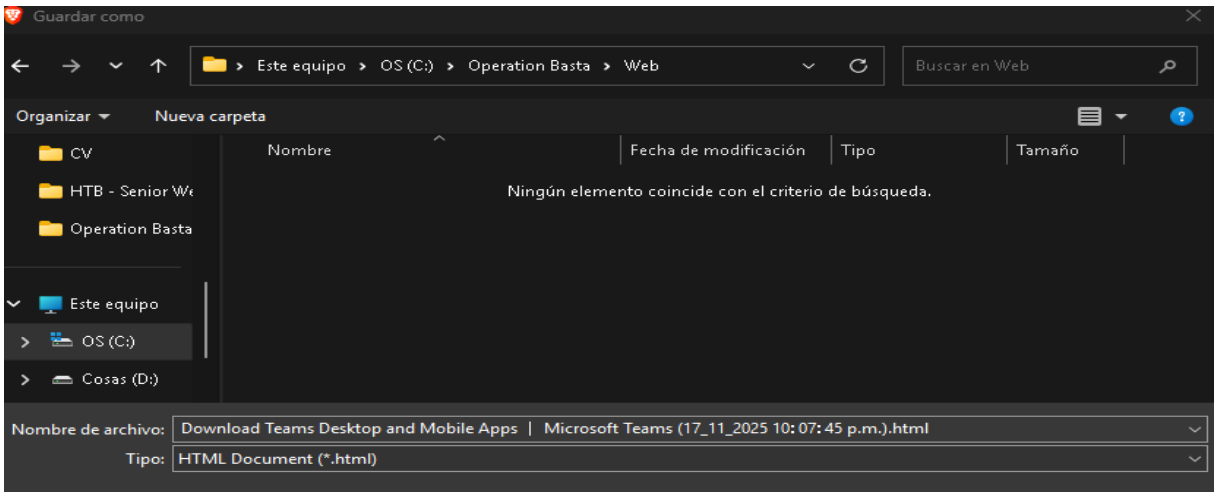


Figura 12

Descarga en .html de la Interfaz de Microsoft Teams



Como siguiente paso, se editó el index.html descargado y se localizó dentro del código el botón de descarga en donde se hace referencia al ejecutable de Microsoft Teams. El enlace al cual dirige este botón fue modificado dirigiéndolo hacia el endpoint personalizado (posteriormente se explica la configuración).

Figura 13

Modificación del index.html de Microsoft Teams

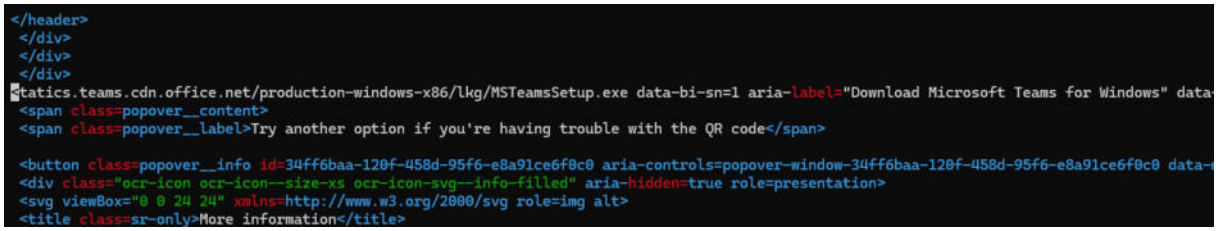
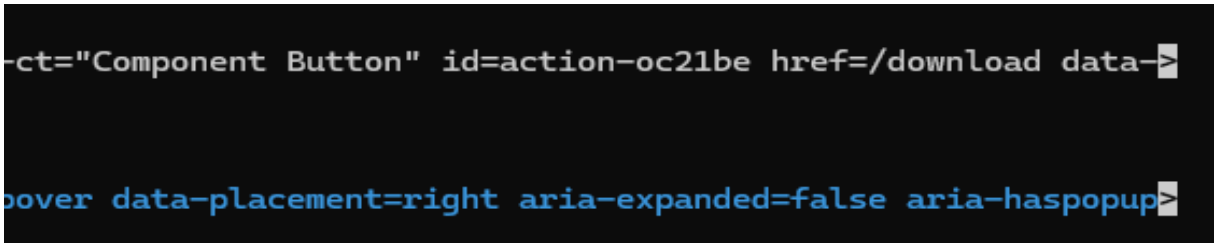


Figura 14

Modificación del botón de descarga de Microsoft Teams



Luego, se creó un script en bash, el cual se encarga de elegir una muestra aleatoria del ransomware de las cinco que se han descargado e incluido en el trabajo; para editar el contenido del script en la terminal se ejecutó el código: `sudo nano /usr/local/bin/random_basta.sh`

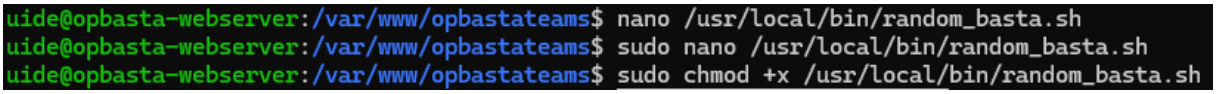
El código que contiene el script es el siguiente:

```
#!/usr/bin/env bash

MALWARE_DIR="/var/www/malware"

# Seleccionar archivo aleatorio

FILE=$(ls "$MALWARE_DIR" | shuf -n 1)

FULLPATH="$MALWARE_DIR/$FILE"

# Enviar cabeceras CGI válidas

echo "Content-Type: application/octet-stream"

echo "Content-Disposition: attachment; filename=\"$FILE\""

echo ""

# Enviar el contenido binario
```

```
cat "$FULLPATH"

sudo chmod +x /usr/local/bin/random_basta.sh
```

Figura 15

Creación del Script para la selección de las muestras aleatorias de Black Basta



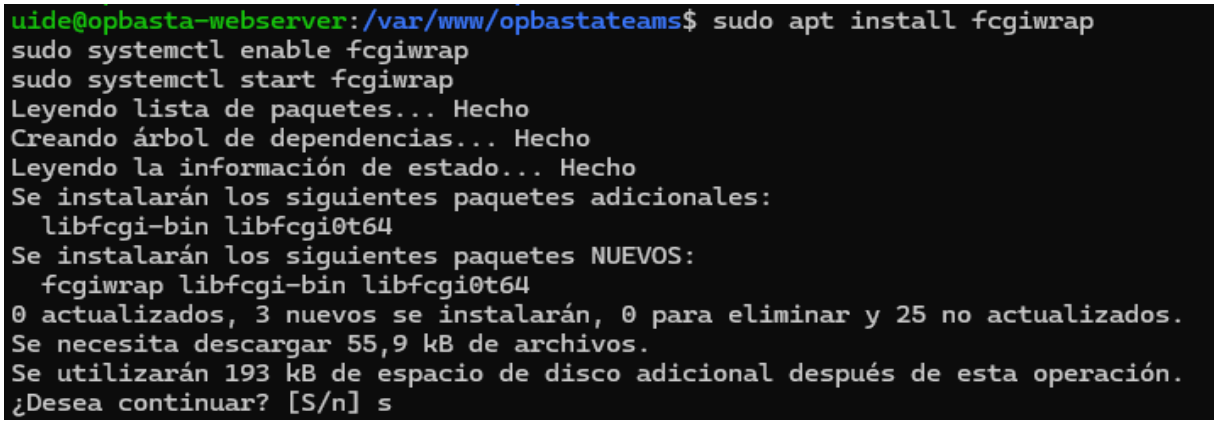
Finalmente, se configuró el servidor **nginx** para servir la página web falsa junto con el malware, el cual será descargado de forma aleatoria eligiendo entre las cinco muestras.

El primer paso de esta configuración fue instalar la herramienta **fcgiwrap,** la cual facilita la descarga; los comandos utilizados son:

```
sudo apt install fcgiwrap

sudo systemctl enable fcgiwrap

sudo systemctl start fcgiwrap
```

Figura 16

Instalación de herramienta fcgiwrap



Como segundo paso, se creó el archivo del sitio que se va a servir, con la configuración correspondiente, incluyendo el endpoint de la descarga: `sudo nano /etc/nginx/sites-enabled/opbastateams`

El código que contiene el script es el siguiente:

```
server {
    listen 80;
    server_name _;
    root /var/www/opbastateams;
    location / {
        try_files $uri $uri/ =404;
    }
    # Endpoint para la descarga aleatoria
    location /random-download {
        fastcgi_pass unix:/run/fcgiwrap.socket;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME /usr/local/bin/random_basta.sh;
    }
    # Servir archivos malware
    location /malware/ {
        alias /var/www/malware/;
        autoindex on;
    }
}
```

Como tercer paso, se reinició el servicio de **nginx** a través de los comandos:

```
sudo nginx -t
sudo systemctl reload nginx
```

Figura 17

Reinicio del servicio nginx

Con esta configuración el servidor web se encuentra listo; por lo tanto, se ejecuta un test utilizando la máquina Windows víctima y se comprueba que se descarga un archivo aleatorio de las muestras de Black Basta que se han subido.

Figura 18

Creación del servidor Web malicioso finalizado

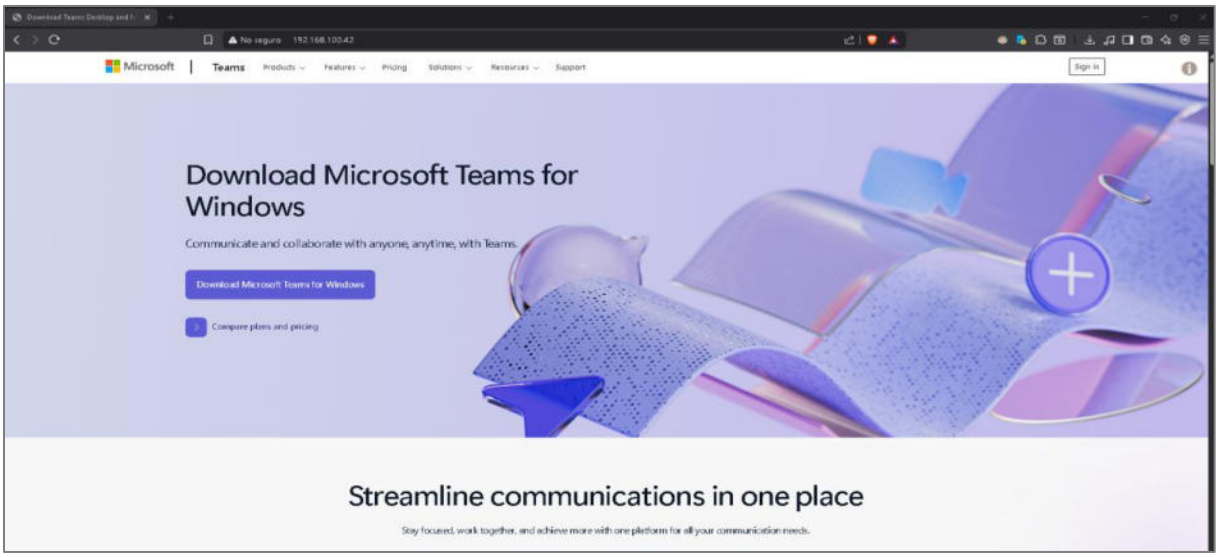


Figura 19

Comprobación de descarga de la muestra Black Basta renombrada

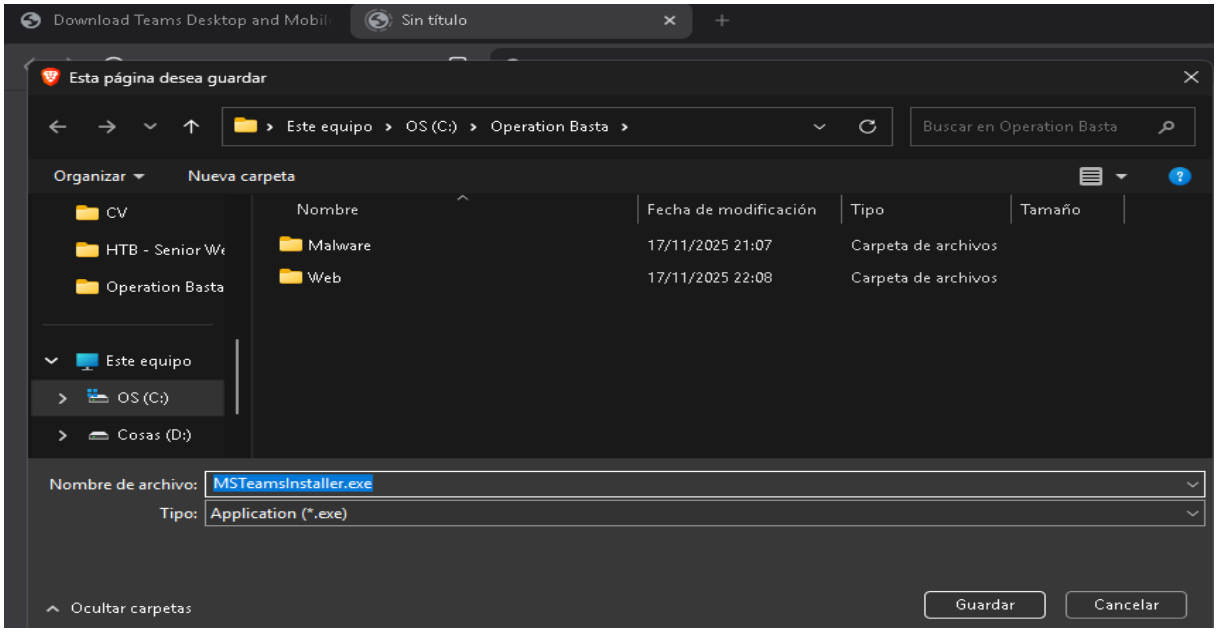
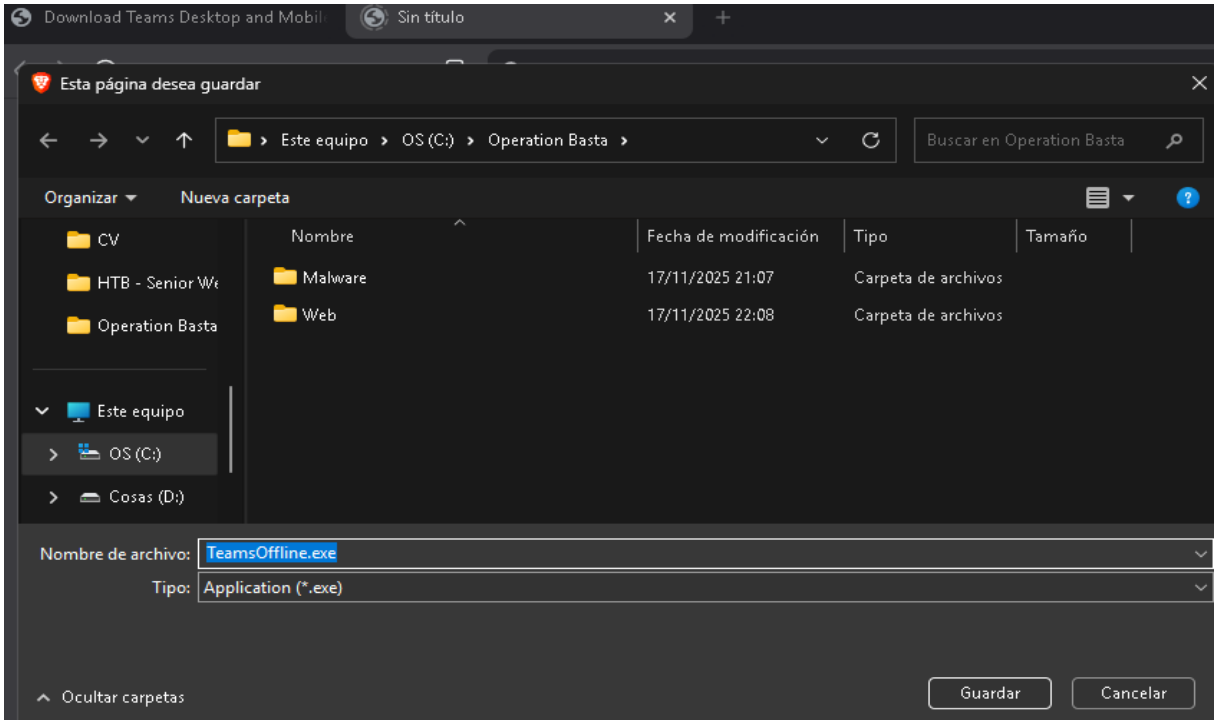


Figura 20

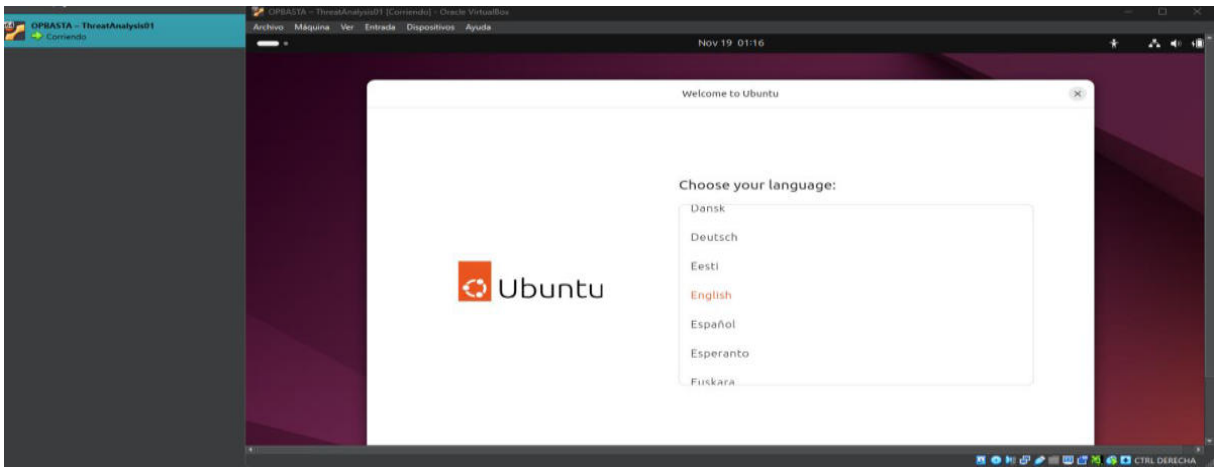
Segunda comprobación de descarga de la muestra Black Basta renombrada



3.1.7.2. VM OPBASTA - ThreatAnalysis01 (Máquina de Análisis). En esta máquina virtual se descargó Ubuntu 22.04.4 LTS Desktop (64-bit) desde la web oficial *https://ubuntu.com/download/desktop* y se creó la VM usando VirtualBox.

Figura 21

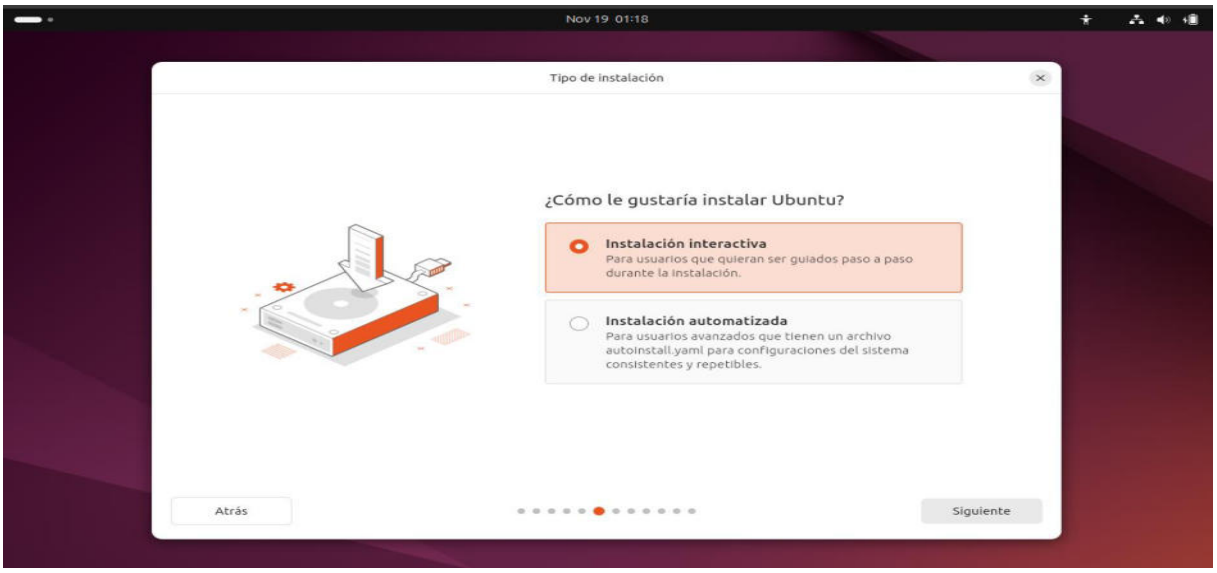
Descarga del sistema operativo Ubuntu para VM OPBASTA - ThreatAnalysis01



Luego, se inició el proceso de instalación del OS Ubuntu seleccionando la opción de "Instalación Interactiva".

Figura 22

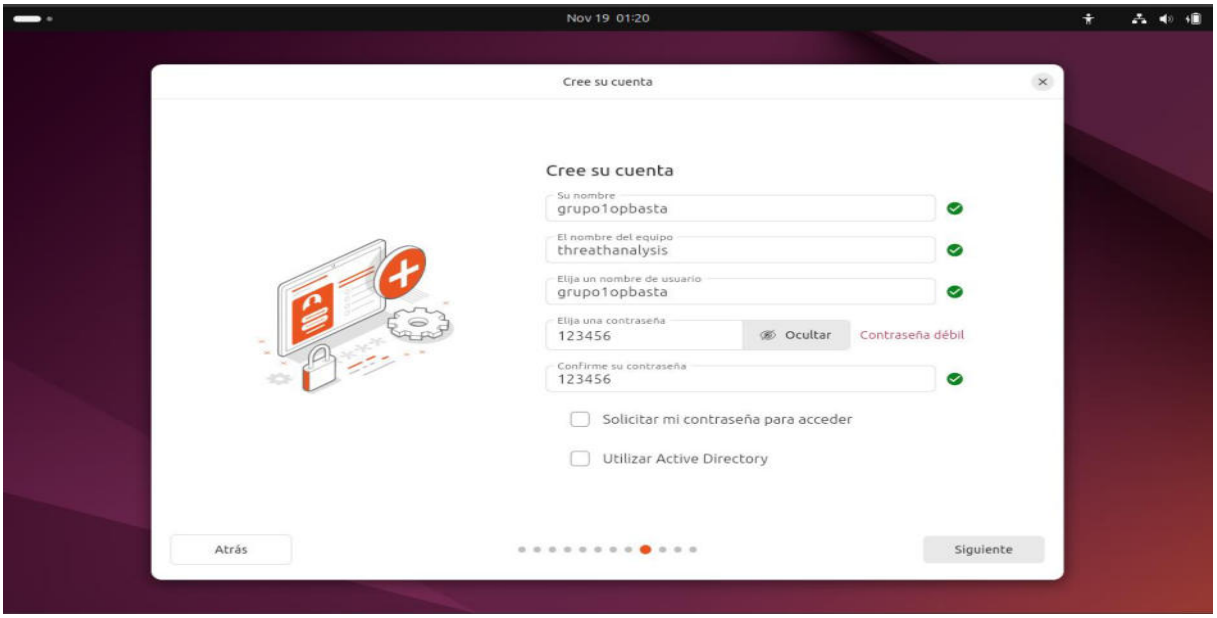
Configuración del modo de instalación del sistema operativo Ubuntu



Seguidamente, se asignó el nombre de equipo, usuario y contraseña.

Figura 23

Configuración de usuario y contraseña del SO Ubuntu



Finalmente, se revisaron las configuraciones antes de dar clic en “Instalar”.

Figura 24

Configuración de características del SO Ubuntu

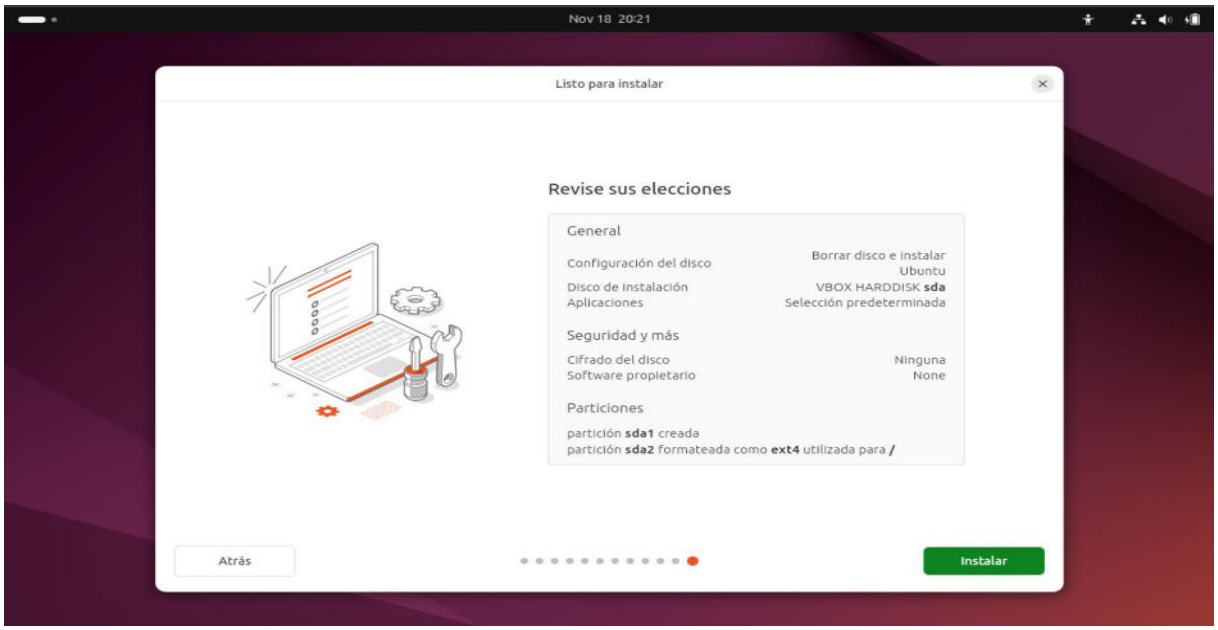
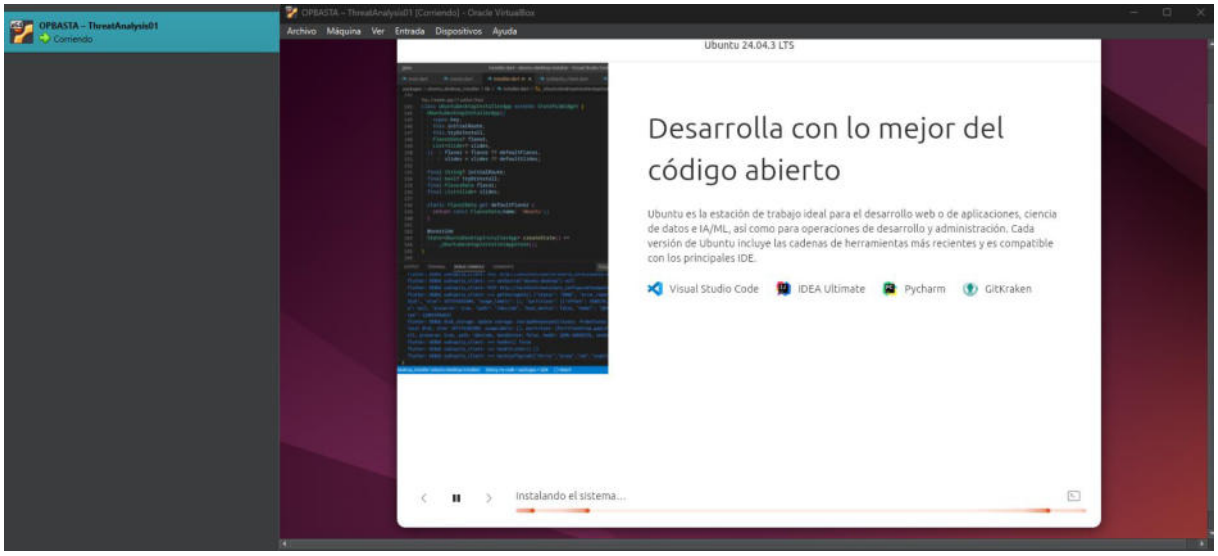


Figura 25

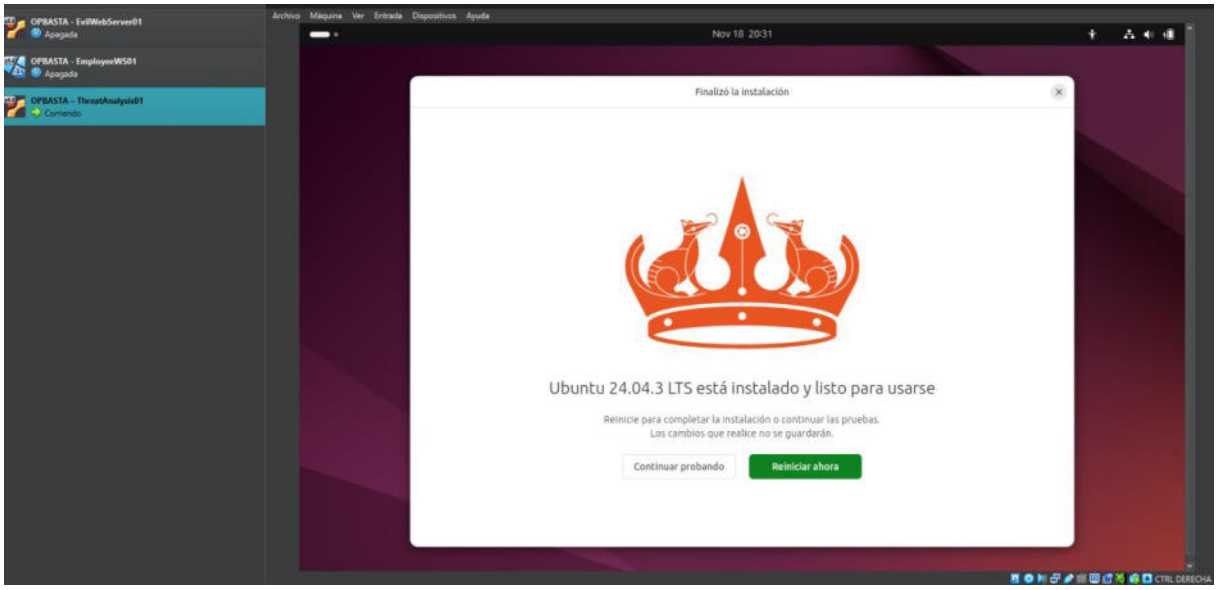
Proceso de instalación del SO Ubuntu



Al terminar la instalación, se reinició la máquina virtual.

Figura 26

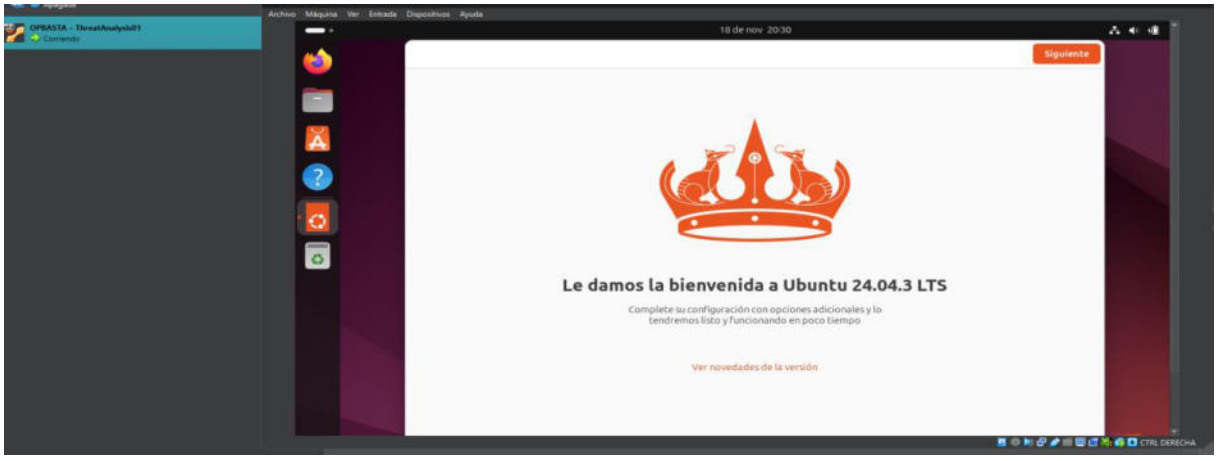
Mensaje de reinicio al culminar la instalación



Tras el reinicio, se visualizó el mensaje de bienvenida a Ubuntu.

Figura 27

Sistema operativo Ubuntu instalado en VM OPBASTA - ThreatAnalysis01



A continuación, se abrió una terminal de Ubuntu para iniciar con la instalación de las herramientas de invitado de VirtualBox y así obtener un mejor rendimiento y experiencia en el manejo de la máquina virtual. Primero, se instalaron las siguientes dependencias:

```
sudo apt update
```

```
sudo apt install bzip2 tar

sudo apt install -y build-essential dkms gcc make perl linux-headers-
$(uname -r)
```

Figura 28

Instalación de dependencias para obtener mejoras en VM OPBASTA - ThreatAnalysis01

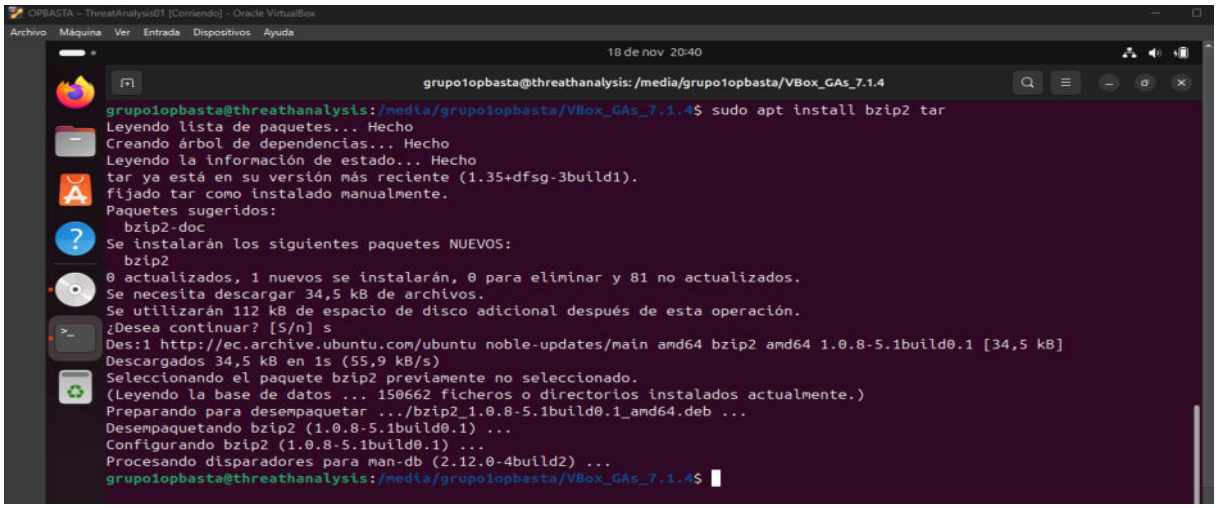
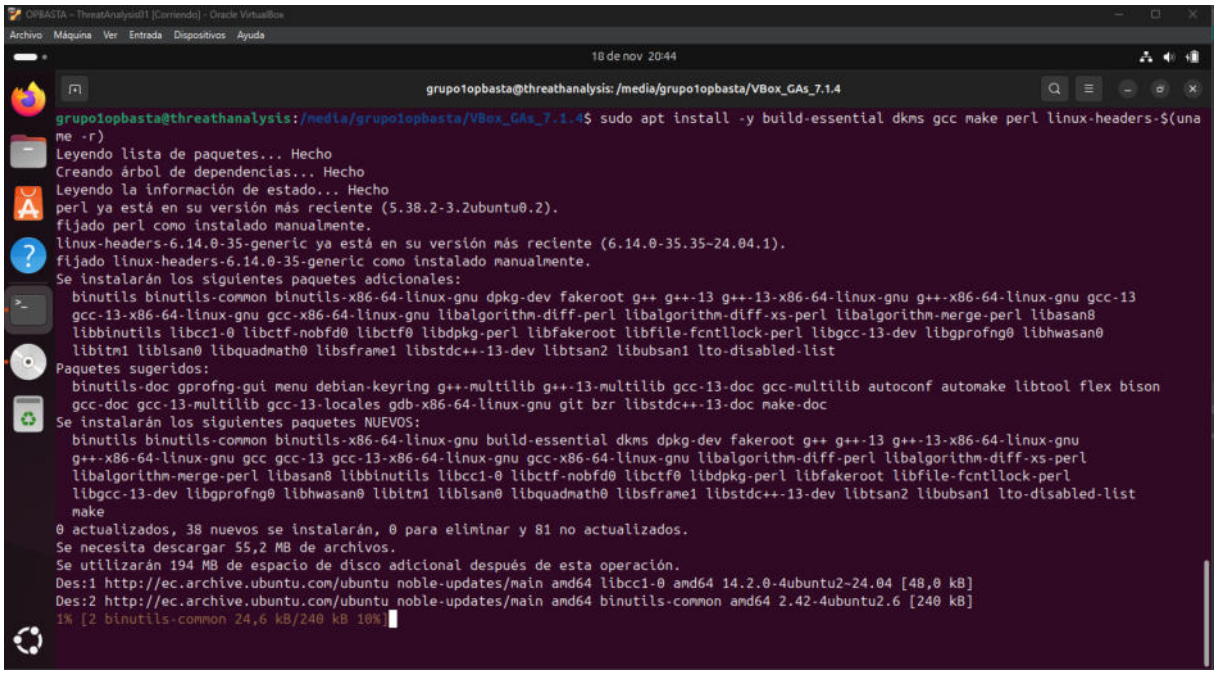


Figura 29

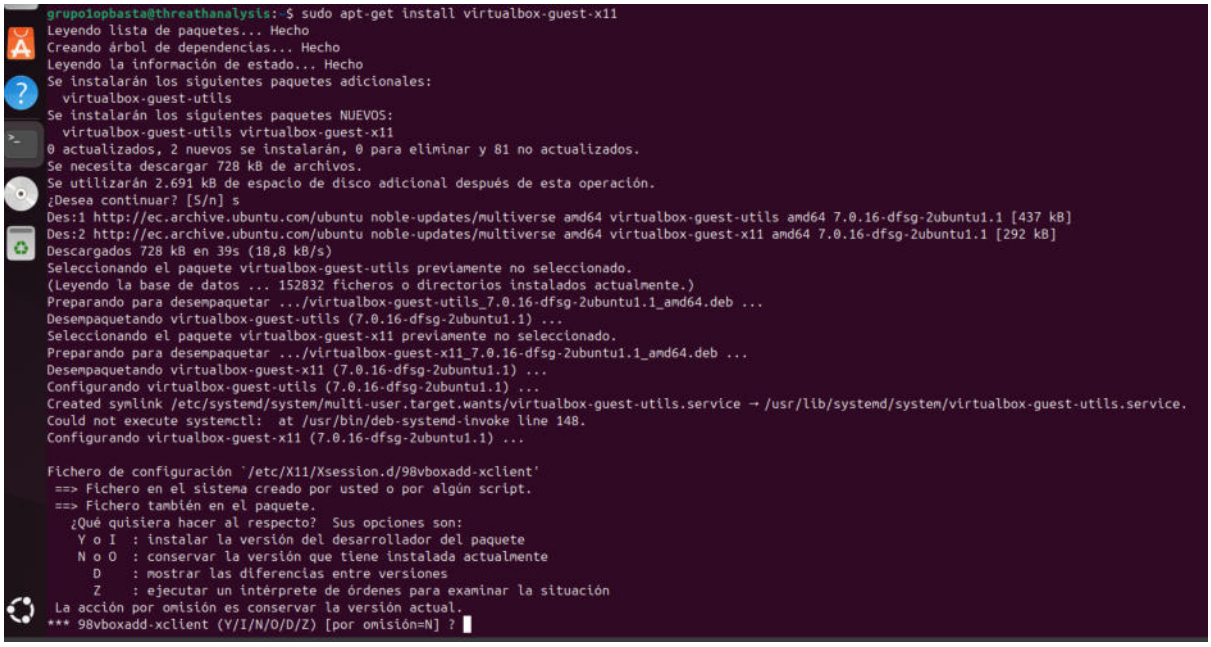
Continuación del proceso de instalación de dependencias



Seguidamente, se ejecutó el siguiente comando: `sudo apt-get install virtualbox-guest-x11`

Figura 30

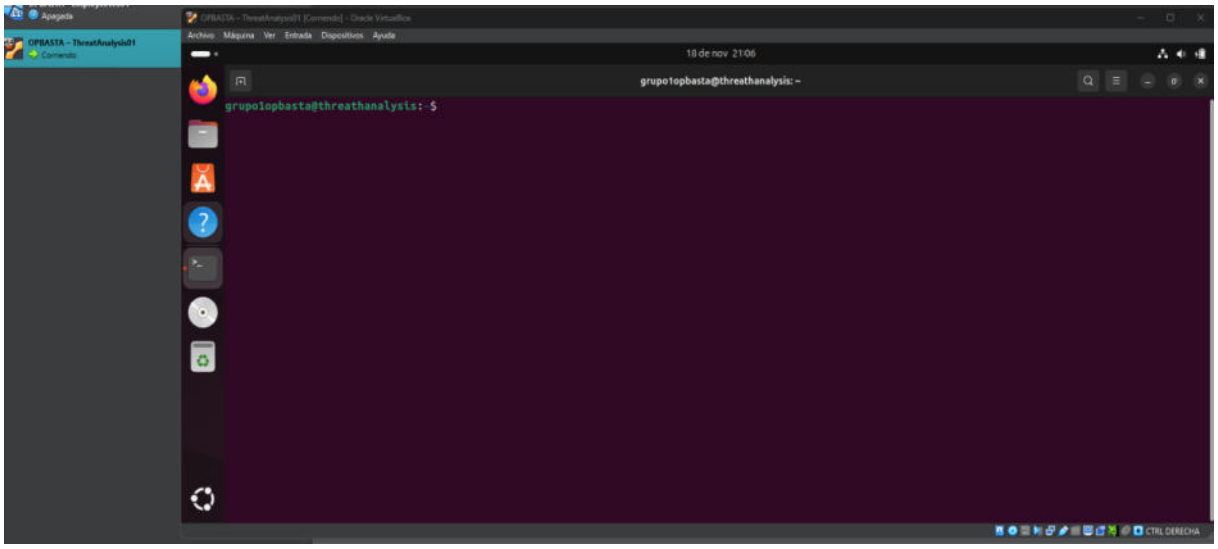
Instalación de mejoras de integración



Luego, se reinició la máquina para que se actualicen los nuevos controladores.

Figura 31

Equipo listo para usarse

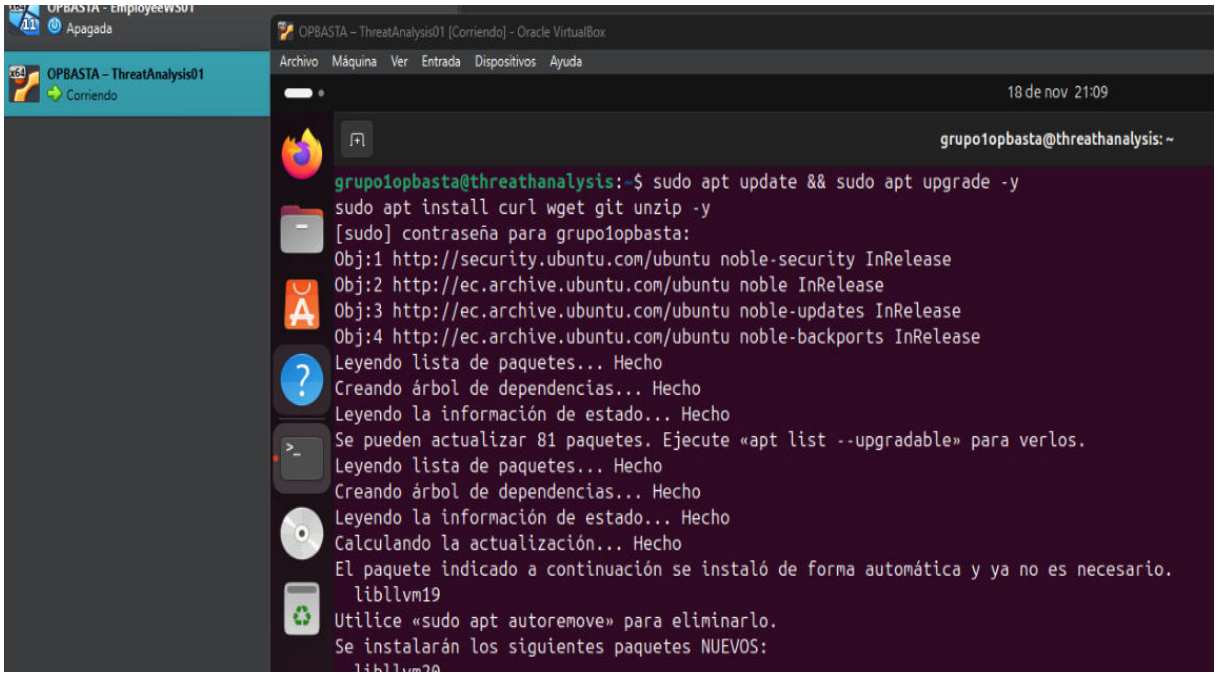


Se procedió con la instalación de todas las herramientas necesarias. Como primera herramienta, se instaló Velociraptor, el cual actúa como un “mini EDR”, ya que su objetivo principal es recolectar toda la data posible de cada movimiento realizado en el equipo Windows víctima. Los siguientes comandos fueron utilizados para instalar dependencias necesarias:

```
sudo apt update && sudo apt upgrade -y
sudo apt install curl wget git unzip -y
```

Figura 32

Actualización de paquetes en repositorios

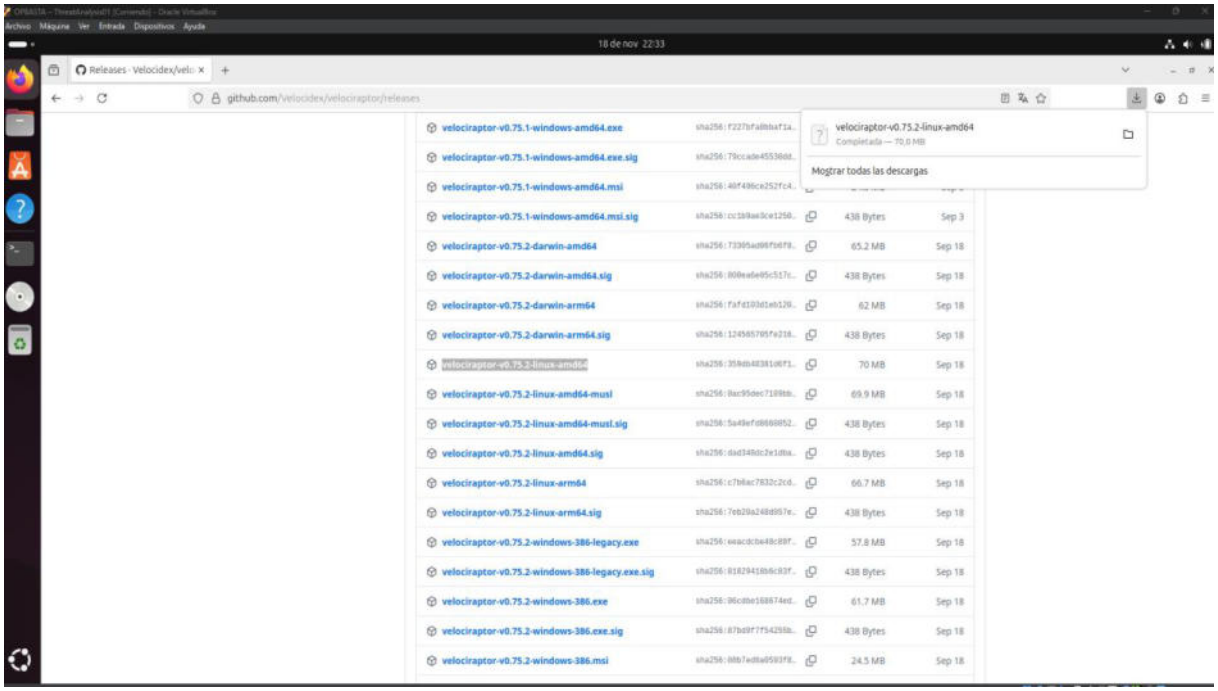


Para obtener el instalador de “Velociraptor”, se descargó la última versión publicada para sistemas operativos Linux, a través del siguiente enlace:

https://github.com/Velocidex/velociraptor/releases

Figura 33

Descarga de la última versión de Velociraptor para Linux

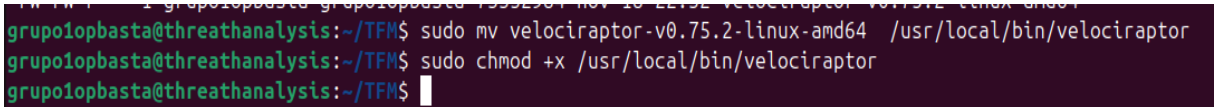


Luego de la descarga, se movió el binario de Velociraptor a una de las rutas del PATH para poder invocarlo desde cualquier directorio en consola, se usaron los comandos:

```
sudo mv velociraptor-v0.75.5-linux-amd64 /usr/local/bin/velociraptor
sudo chmod +x /usr/local/bin/Velociraptor
```

Figura 34

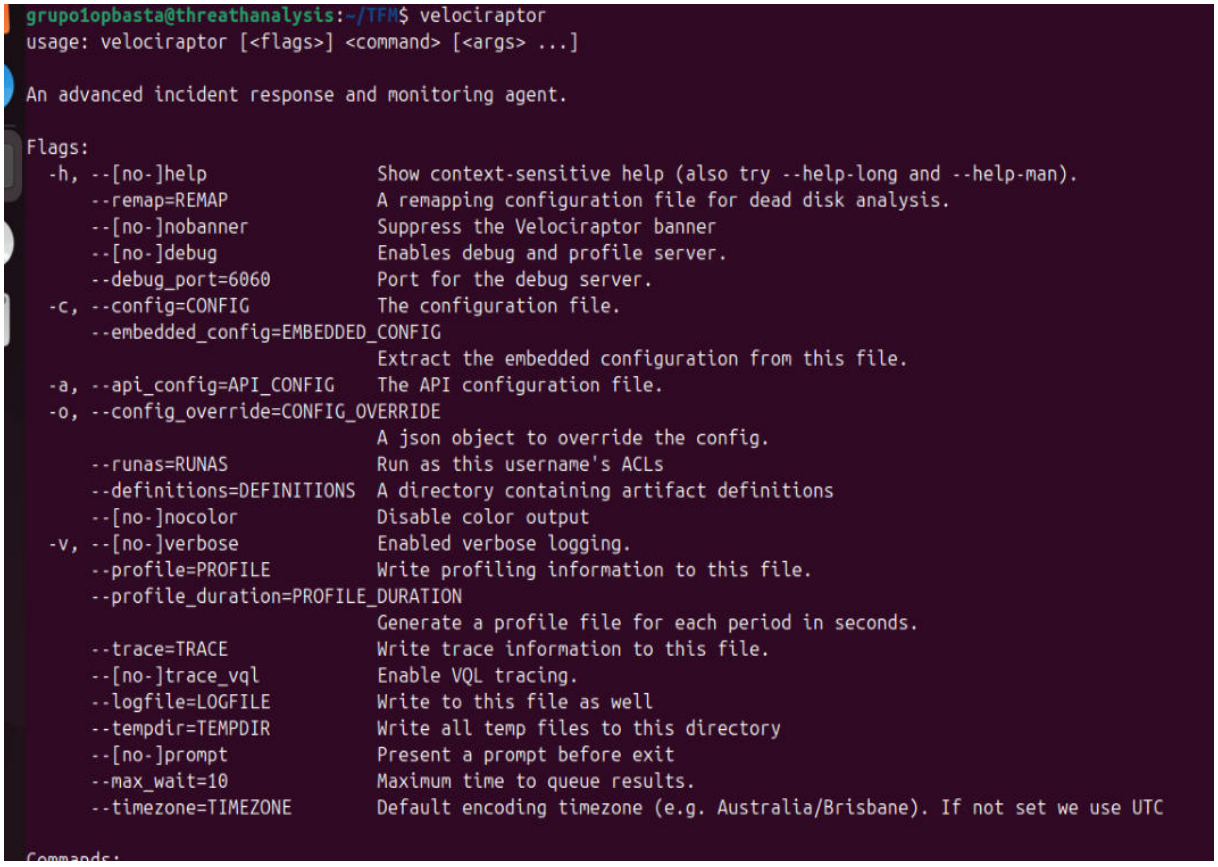
Redirección de la carpeta donde se invoca Velociraptor



Una vez completado el paso anterior, se ejecutó el binario de Velociraptor.

Figura 35

Ejecución del binario de Velociraptor



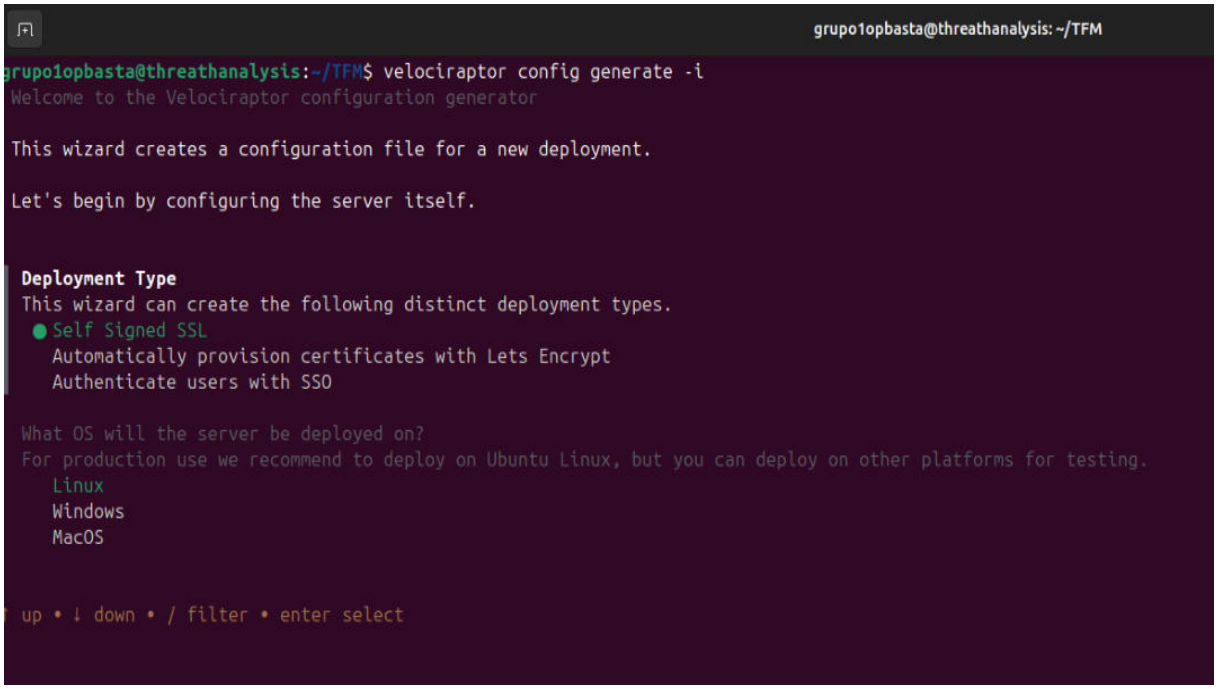
Inmediatamente, se configuró Velociraptor para que funcione como servidor y reciba toda la información que se recolecte desde un agente. Se utilizó el siguiente comando:

```
velociraptor config generate -i
```

Tras la ejecución del comando, se desplegó una ventana interactiva con distintas configuraciones a definir, se seleccionó la opción “Self Signed SSL” y Linux”.

Figura 36

Configuración del servidor de Velociraptor

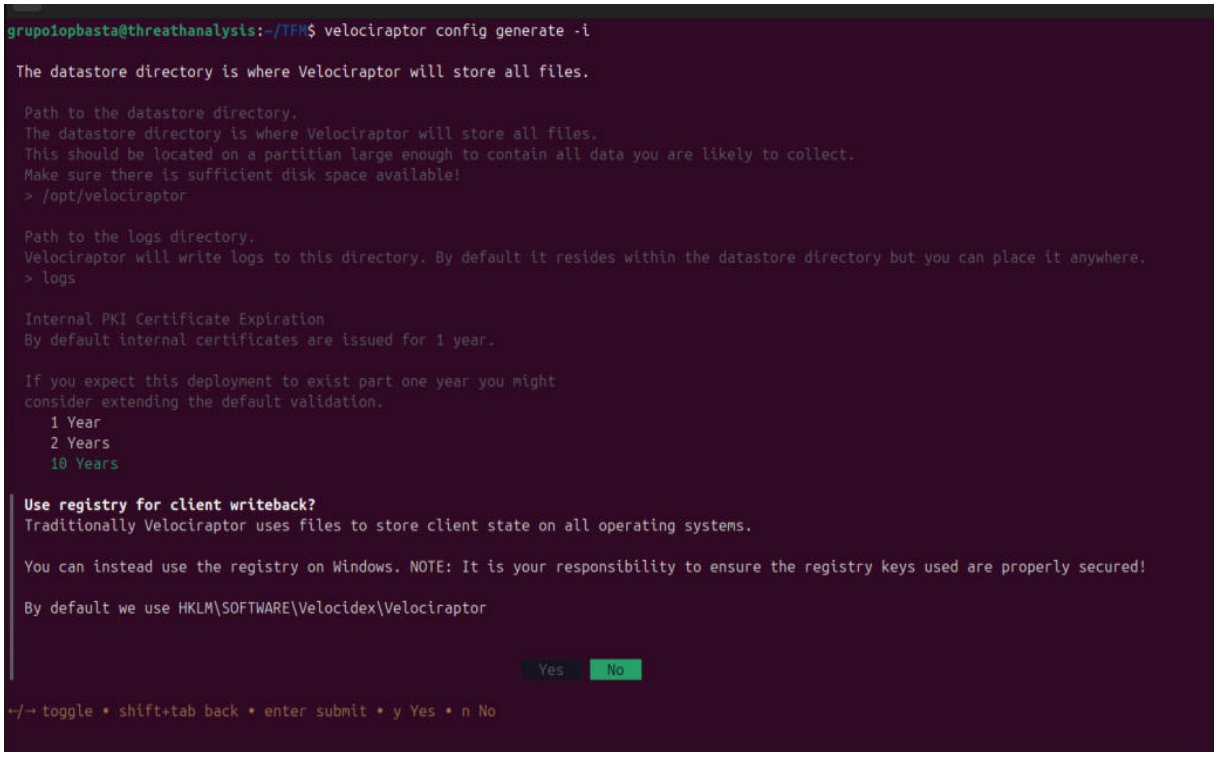


Continuando, se asignó la ruta predeterminada “/opt/velociraptor” como directorio en donde se almacenarán todos los datos. También se eligió al directorio “logs” como ruta predeterminada de almacenamiento de logs.

Además, se seleccionó el valor “10 Años” para expiración de certificado; y, en la última opción donde se pregunta si se desea usar el Registro de Windows para guardar la información de estado del cliente, se eligió “No”, ya que se requiere guardar la información en un archivo normal.

Figura 37

Configuración de rutas, tiempo de expiración y otras opciones del servidor de Velociraptor

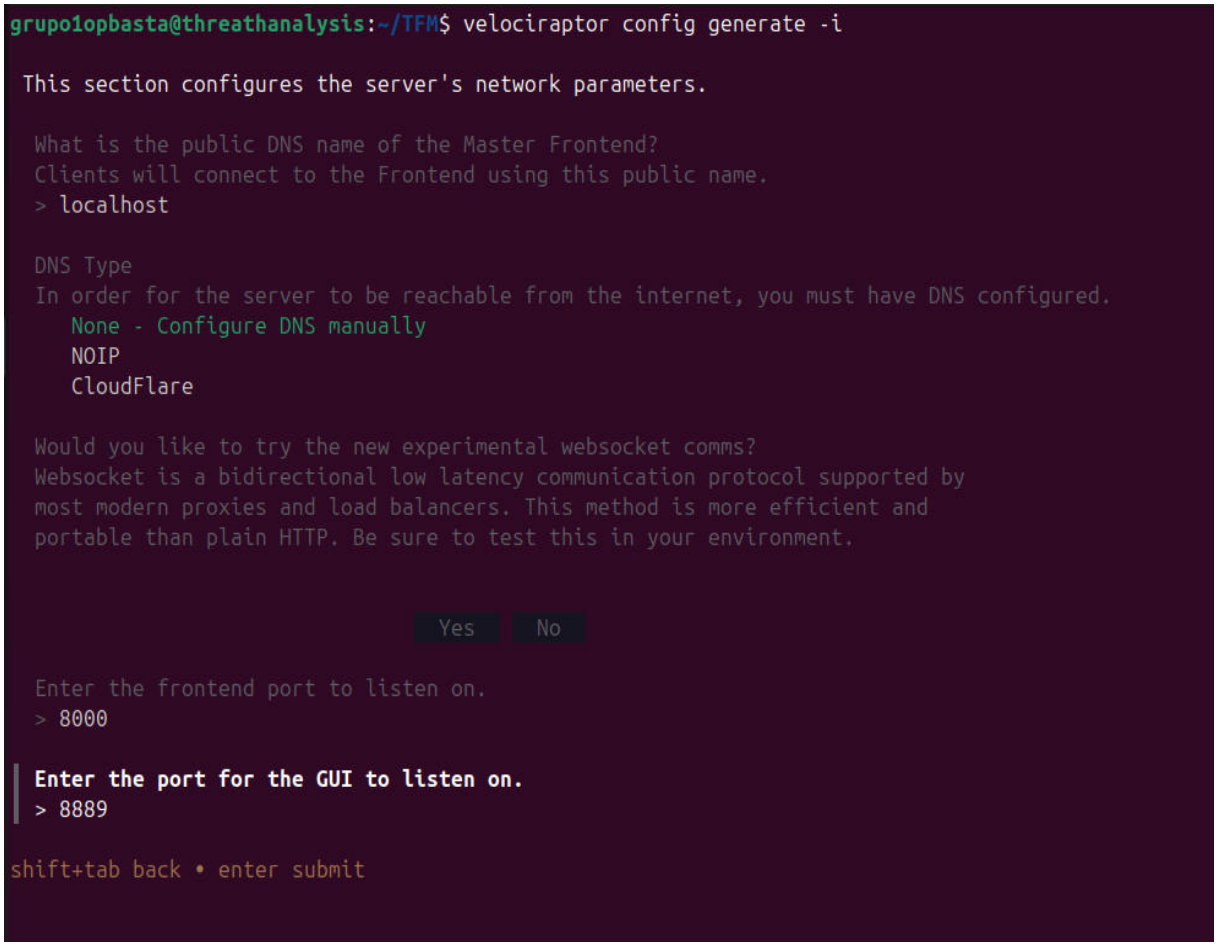


En las opciones siguientes en la ventana interactiva se aceptaron las configuraciones por defecto; es decir:

- Se mantuvo la configuración sugerida para uso de dirección IP localhost.
- Se mantuvo que la configuración del DNS sea manual.
- Se mantuvieron los puertos 8000 y 8889 para la escucha del frontend y de la interfaz gráfica (GUI) respectivamente.

Figura 38

Configuración de características de red y puertos del servidor Velociraptor

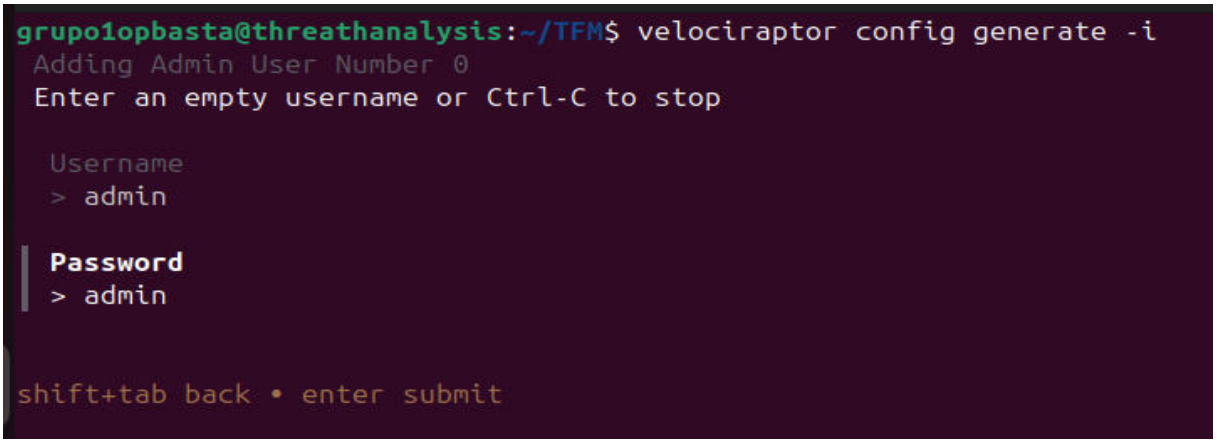


Cerca de finalizar la configuración en la ventana interactiva, se solicitó la creación de un usuario para la autenticación hacia el servicio publicado en el puerto 8889 (GUI). A través de la GUI se permite crear políticas, visualizar los agentes conectados, recolectar artefactos, etc.

Para este laboratorio se creó el usuario “admin” con la contraseña “admin”. Al presionar “Enter”, se mostró la misma ventana para continuar añadiendo otros usuarios, en este caso se presiona “Enter” nuevamente sin introducir nada, ya que no se requiere crear más usuarios administradores.

Figura 39

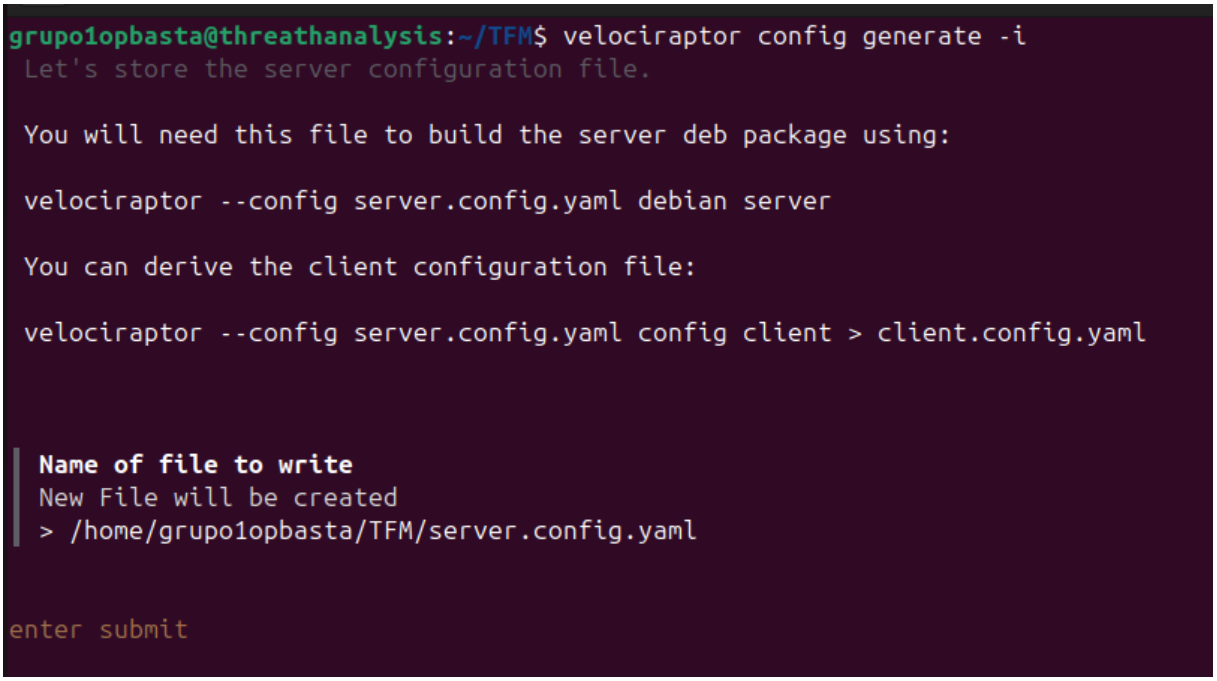
Configuración de usuario y contraseña para autenticación en la GUI de Velociraptor



Para finalizar la configuración en la ventana interactiva, se indicó la ruta en la que se desea guardar el archivo de configuración del servidor de Velociraptor (server.config.yaml), y se presionó “Enter” para guardar.

Figura 40

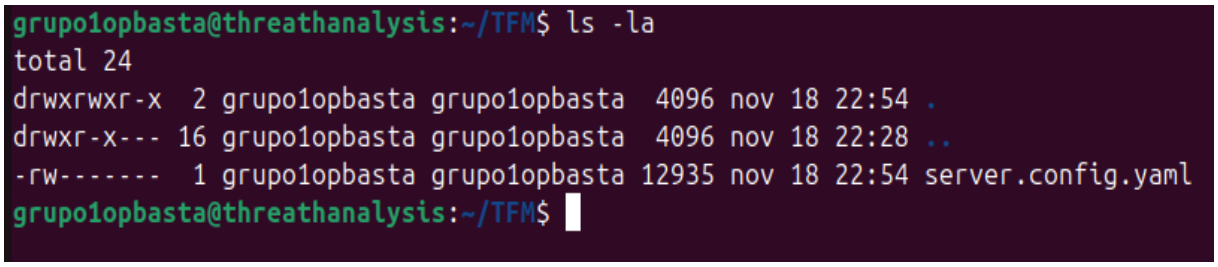
Finalización de la configuración del servidor de Velociraptor



A fin de comprobar que se haya generado el archivo de configuración server.config.yaml en la ruta seleccionada anteriormente, se enlistaron los directorios.

Figura 41

Comprobación de creación del archivo server.config,yaml de Velociraptor

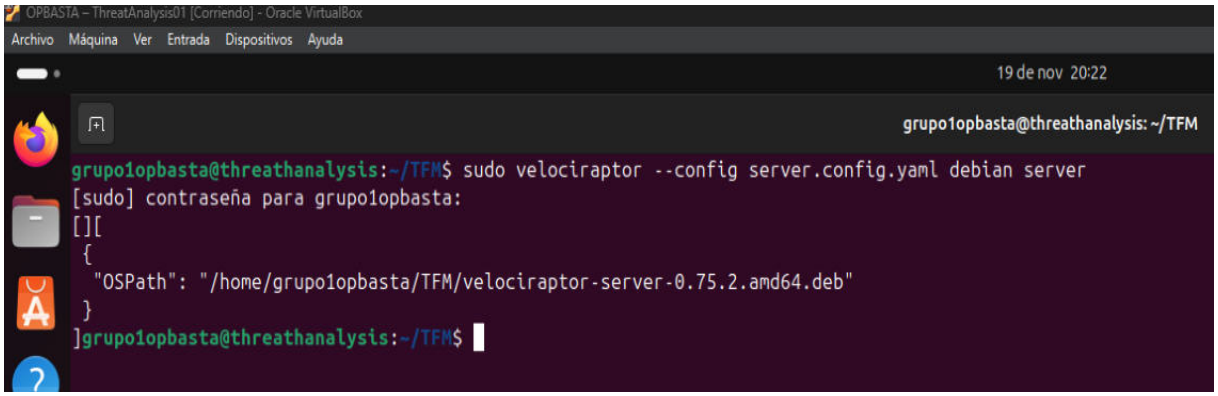


Más adelante, se usó el archivo de configuración generado para crear el servicio de systemd, y como resultado se obtuvo un archivo .deb, el comando usado es:

```
sudo velociraptor --config server.config.yaml debian server
```

Figura 42

Creación del servicio de systemd



Posteriormente, se instaló el paquete .deb generado, utilizando el comando:

```
sudo dpkg -i velociraptor-server-0.75.5.amd64.deb
```

Figura 43

Instalación del paquete .deb de Velociraptor

Luego, se inicializaron los servicios del servidor de Velociraptor usando los siguientes comandos:

```
sudo systemctl enable velociraptor_server

sudo systemctl start velociraptor_server

sudo systemctl status velociraptor_server
```

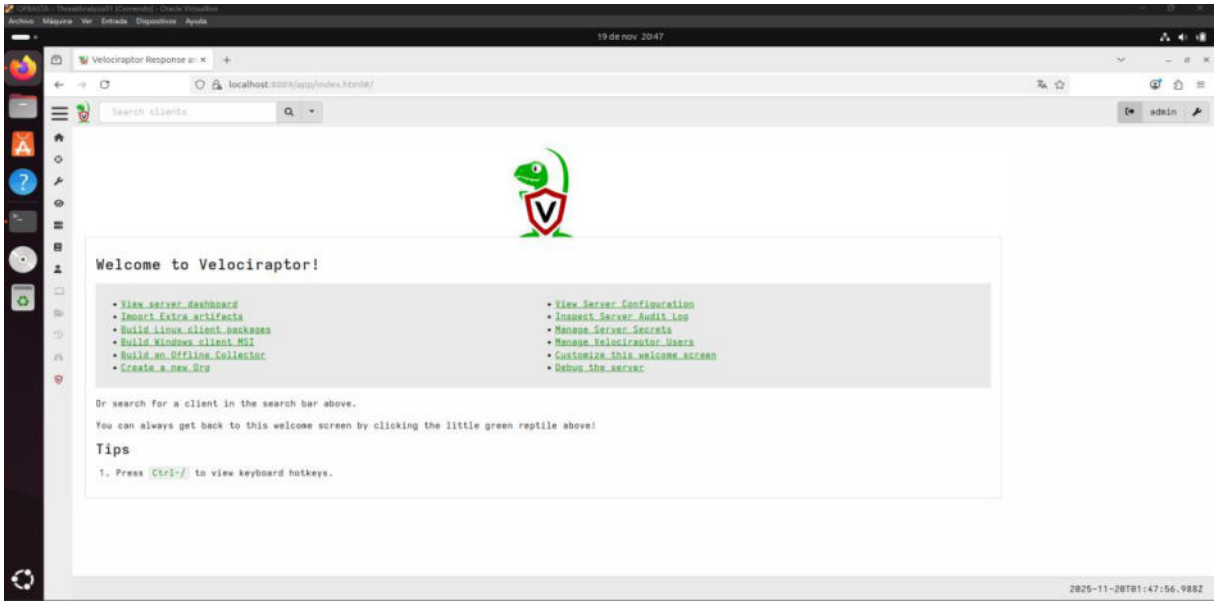
Figura 44

Inicialización de los servicios del servidor de Velociraptor

Con el servidor de Velociraptor ejecutándose en segundo plano, se logró acceder a la interfaz gráfica alojada en el servidor localhost a través del puerto 8889, tal como fue configurado.

Figura 45

Interfaz gráfica de Velociraptor



Como paso adicional, se modificó el archivo de configuración del servidor ya desplegado para que se logre acceder a la GUI desde VM OPBASTA - EmployeeHost01 usando la dirección IP del adaptador de la red correspondiente a la red interna (192.168.56.103). Se utilizó el siguiente comando:

```
sudo nano /etc/velociraptor/server.config.yaml
```

Figura 46

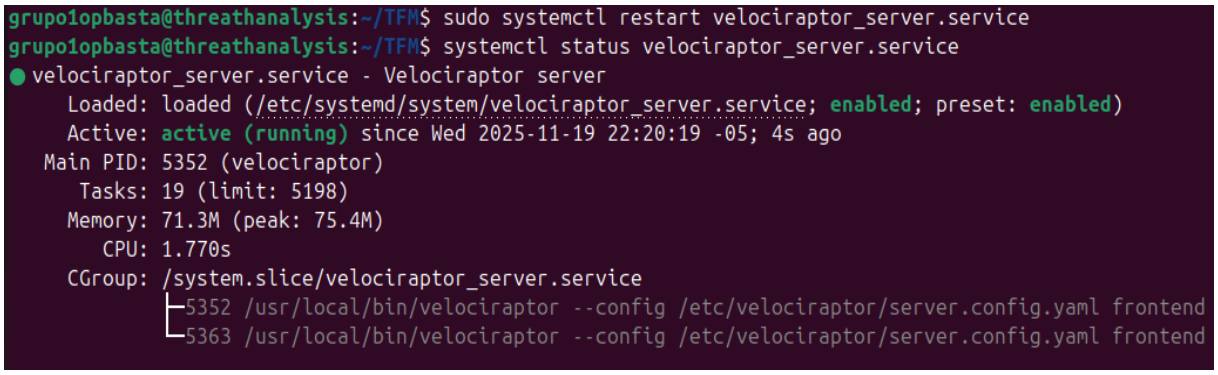
Modificación del archivo de configuración del servidor de Velociraptor

Después de la modificación realizada, se reinició el servicio a través del comando:

```
sudo systemctl restart velociraptor_server.service
```

Figura 47

Reinicio del servidor de Velociraptor



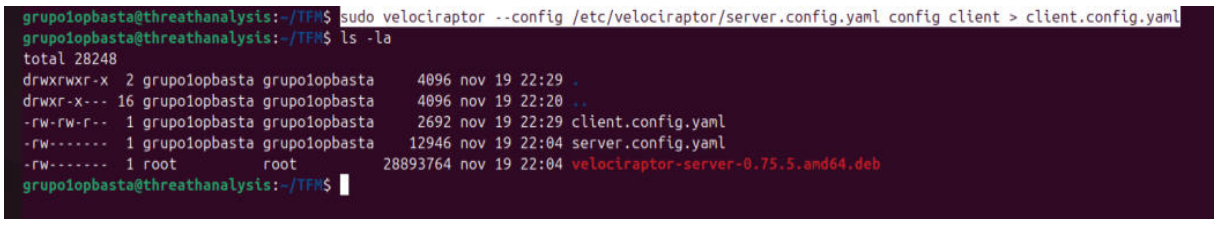
Para generar el agente encargado de recolectar toda la telemetría y artefactos de la máquina Windows víctima (VM OPBASTA - EmployeeHost01), se empleó el siguiente comando:

```
sudo velociraptor --config /etc/velociraptor/server.config.yaml config client > client.config.yaml
```

Al ejecutar el comando indicado, el equipo tomó la configuración asignada en el servidor, y la adecuó para que se ejecute como cliente; de esta manera, se logró establecer la comunicación correctamente.

Figura 48

Creación del agente recolector de artefactos de Velociraptor

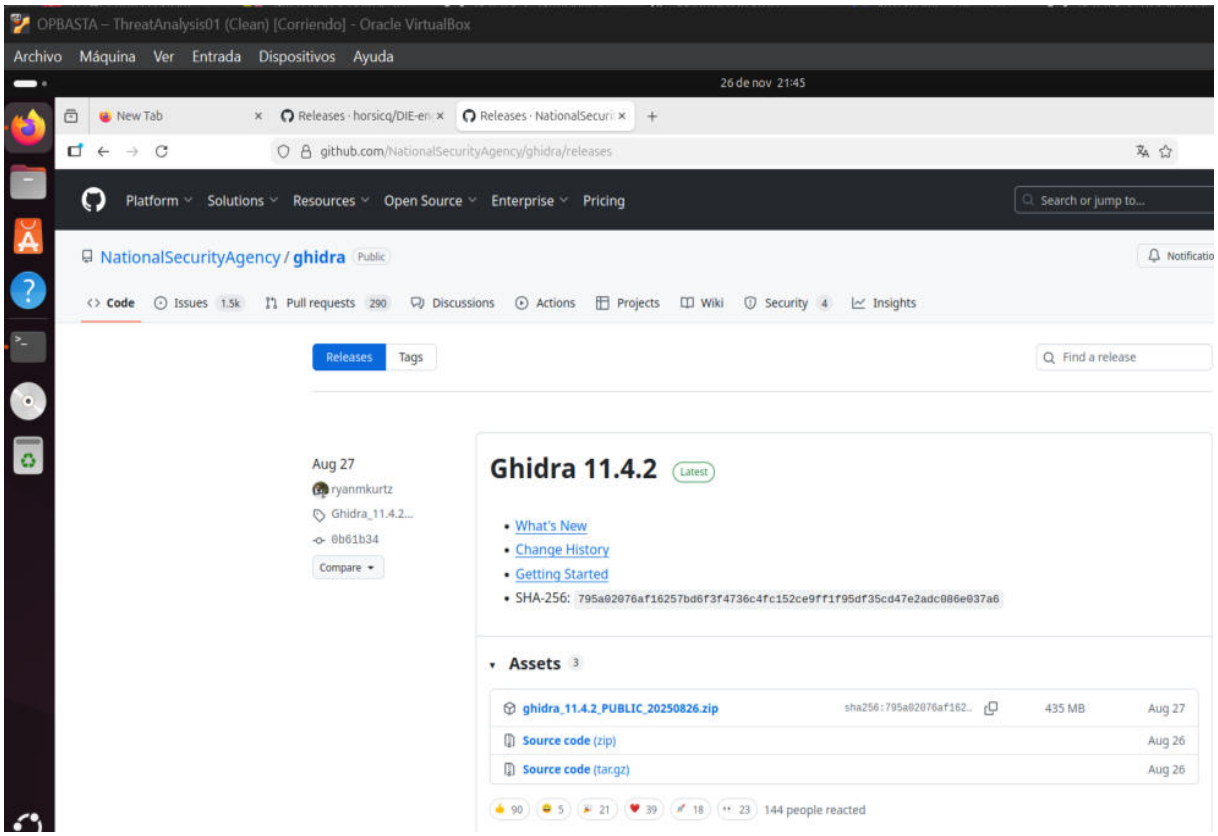


El archivo generado “client.config.yaml” se lo usa al momento de instalar el agente en nuestra máquina víctima VM OPBASTA - EmployeeHost01.

Como segunda herramienta, se instaló Ghidra, la cual más adelante se la usará para ejecutar análisis estático sobre las muestras de ransomware. Esta herramienta ayuda a obtener información de cómo actúa el malware sin ejecutarlo, ampliando los puntos clave del comportamiento que permiten modelar el perfil mínimo de detección de una forma más precisa. Se descargó Ghidra desde el repositorio oficial de la NSA a través del siguiente enlace: *https://github.com/NationalSecurityAgency/ghidra/releases*

Figura 49

Descarga de la última versión de Ghidra desde el repositorio oficial

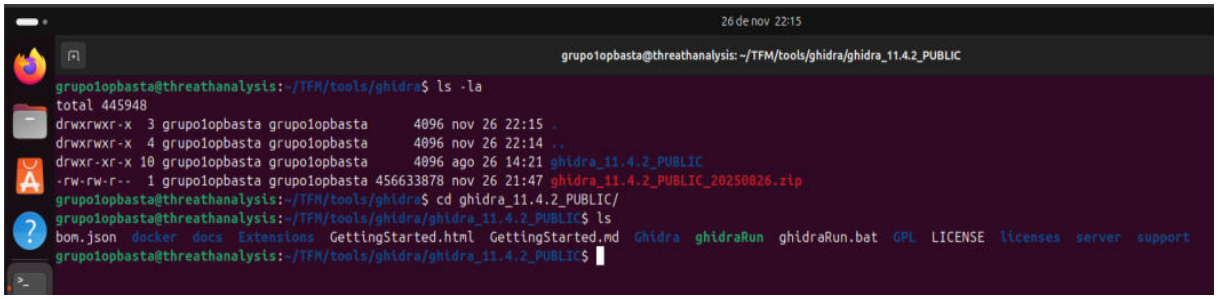


Una vez descargado, se descomprimió el archivo utilizando el comando:

```
unzip ghidra_11.4.2_PUBLIC_20250826.zip
```

Figura 50

Descompresión del archivo descargado de Ghidra



Lo siguiente fue instalar dependencias, en este caso, JAVA.

```
sudo apt install openjdk-21-jdk -y
```

Figura 51

Instalación de dependencias de Java

Continuando, se añadió el entorno de JAVA instalado al PATH, con el uso del comando:

```
export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
```

Figura 52

Establecimiento de la variable de entorno JAVA_HOME

Adicionalmente, se otorgaron permisos de ejecución al archivo “ghidraRun” y finalmente, se lo ejecutó:

```
chmod +x ghidraRun
```

```
./ghidraRun
```

Figura 53

Aceptación de acuerdo de usuario antes de ejecutar Ghidra

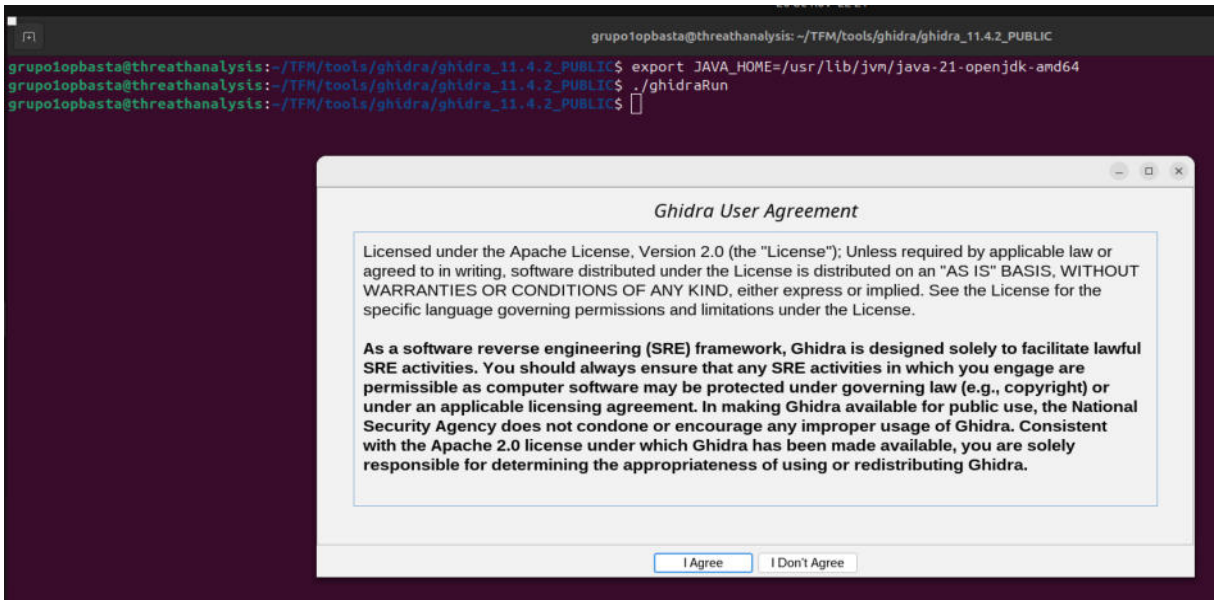


Figura 54

Ejecución de Ghidra en VM OPBASTA - ThreatAnalysis01

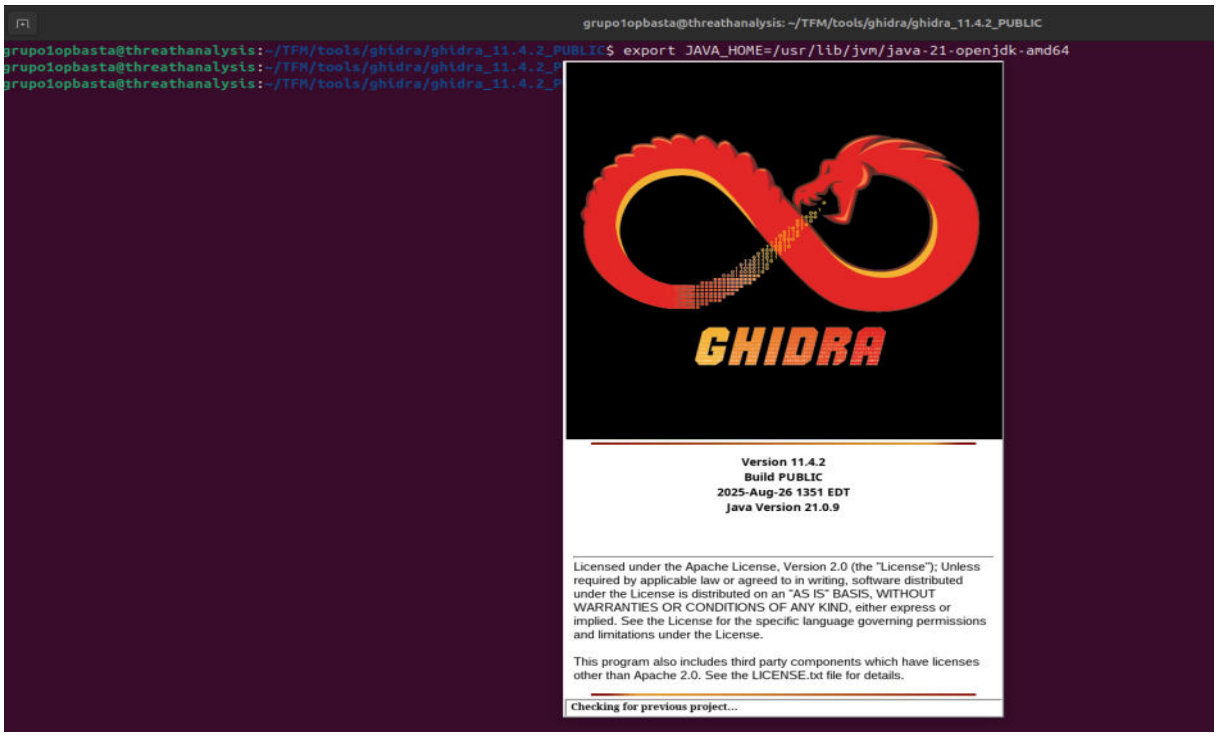


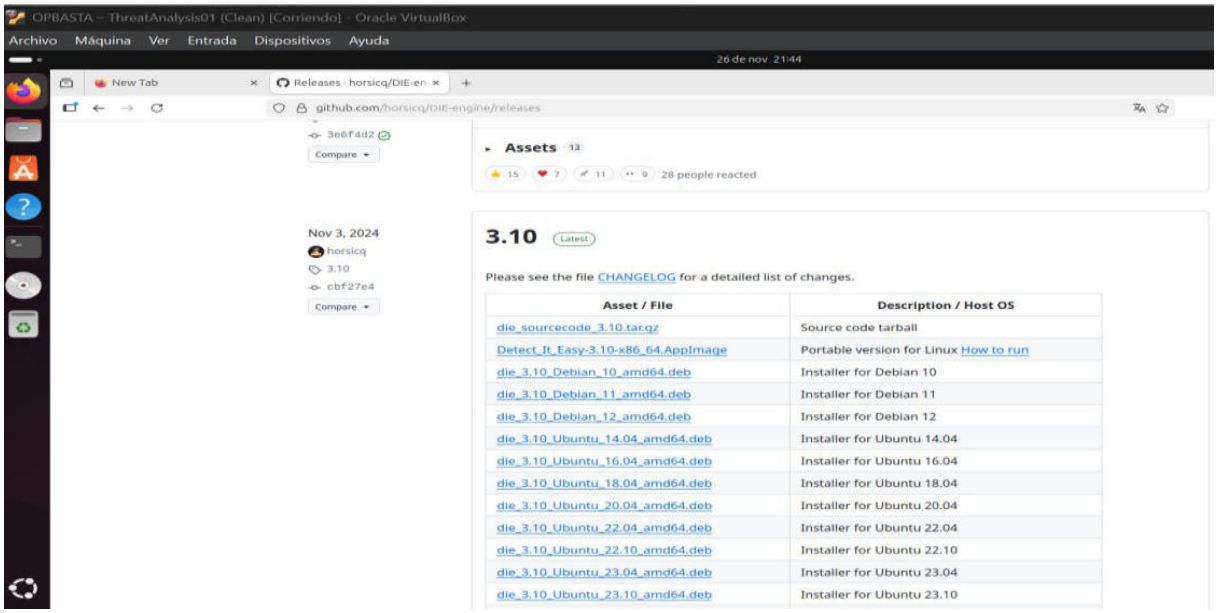
Figura 55

Interfaz de Ghidra abierta en VM OPBASTA - ThreatAnalysis01

Como tercera herramienta, se instaló Detect It Easy (DIE), un software para el análisis estático que permite verificar la información relacionada a un binario en específico. Esta información incluye tipo de binario, entropía, cadenas, compilador, lenguaje, etc. Se descargó el AppImage de DIE desde el repositorio oficial: *https://github.com/horsicq/DIE-engine/releases*

Figura 56

Descarga de herramienta Detect It Easy

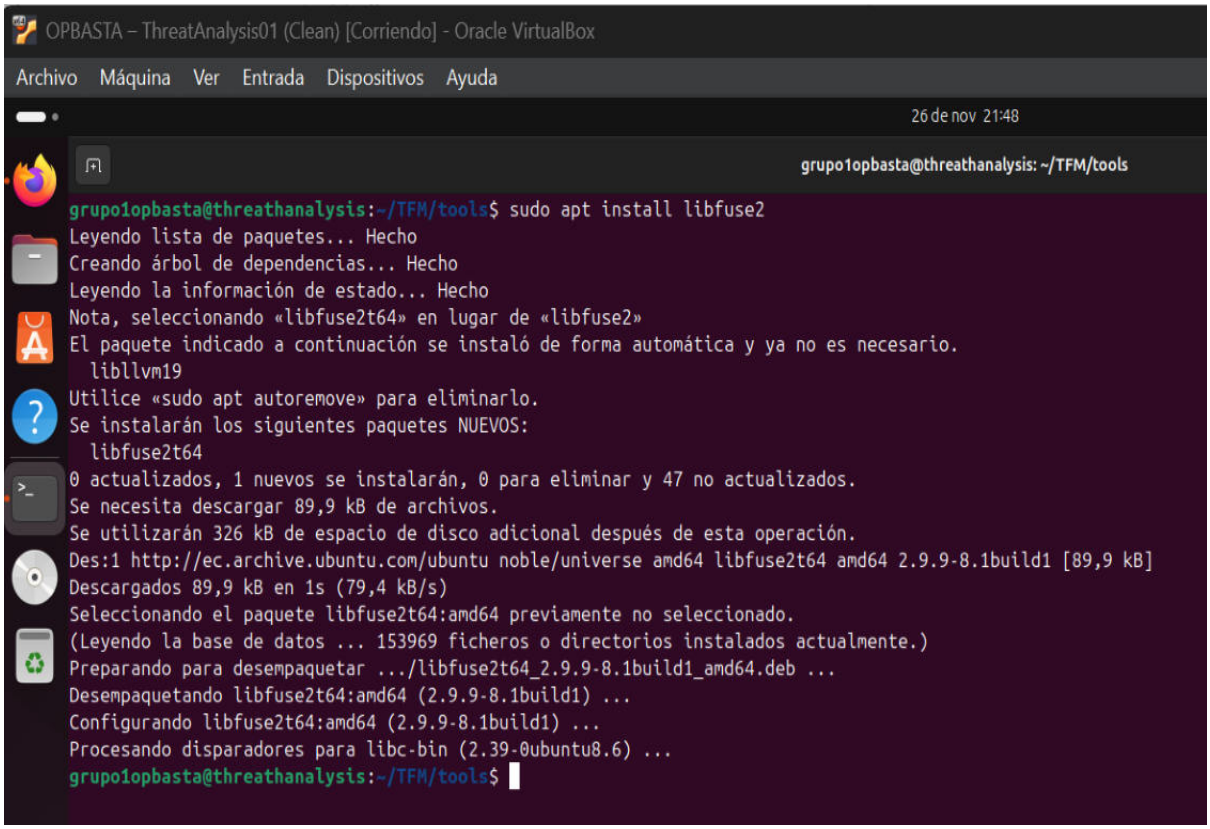


Se continuó con la instalación de dependencias para poder ejecutar el archivo de tipo AppImage. Se utilizó el siguiente comando:

```
sudo apt install libfuse2
```

Figura 57

Instalación de dependencias libfuse2



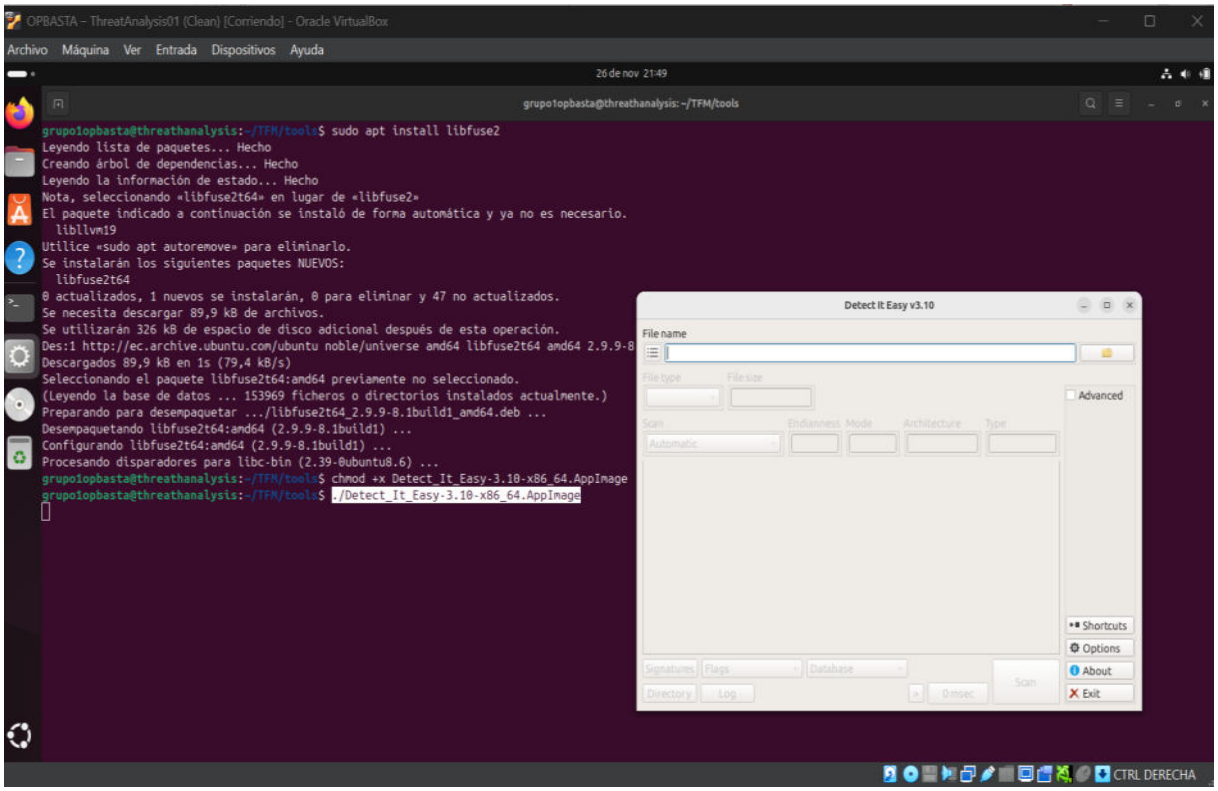
Luego, se otorgaron permisos de ejecución al archivo y finalmente, se lo ejecutó mediante los siguientes comandos:

```
chmod +x Detect_It_Easy-3.10-x86_64.AppImage
```

```
./Detect_It_Easy-3.10-x86_64.AppImage
```

Figura 58

Ejecución de Detect It Easy en VM OPBASTA - ThreatAnalysis01



3.1.7.3. VM OPBASTA - EmployeeHost01 (Máquina Víctima). En este equipo se realizó la descarga e instalación del sistema operativo Windows con ISO de Windows 11 versión 25H2, obtenida desde la página oficial de Microsoft: *https://www.microsoft.com/en-us/software-download/windows11*

NOTA: En este punto se usó una configuración de red en modo adaptador en NAT para descargar desde Internet las herramientas necesarias a fin de adecuar la máquina; posteriormente, se configuró todo el laboratorio en una red interna virtual.

Figura 59

Configuración de la máquina virtual Windows para VM OPBASTA - EmployeeHost01

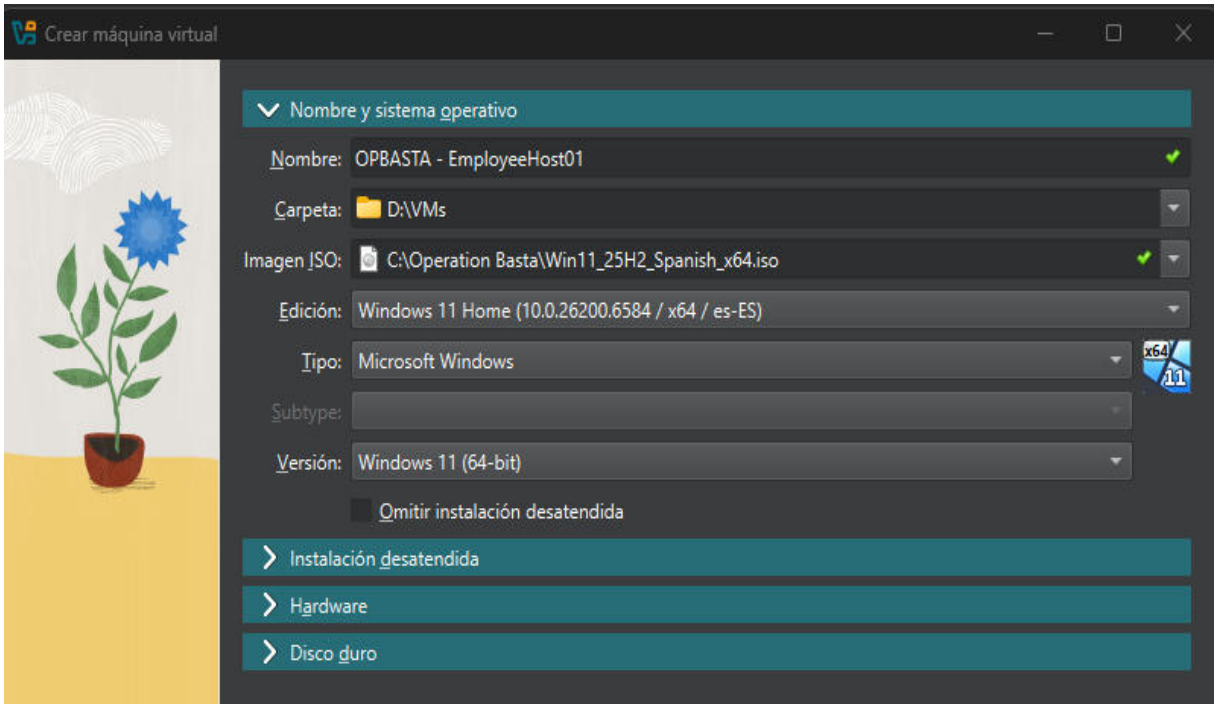
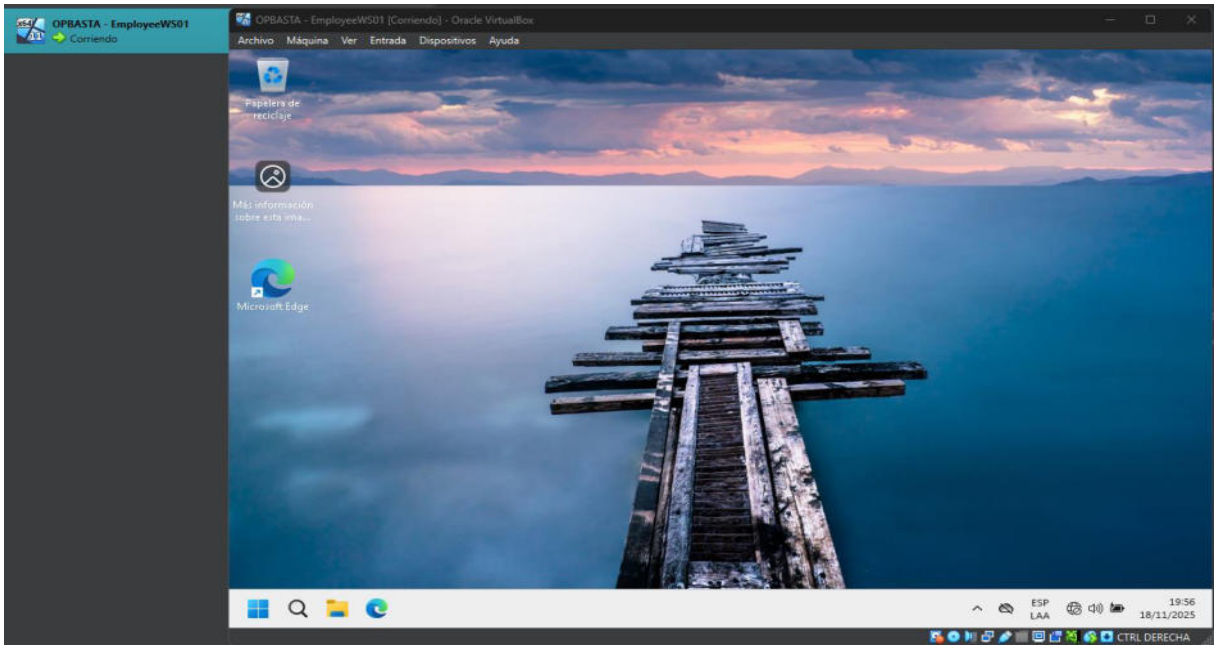


Figura 60

Máquina virtual Windows VM OPBASTA - EmployeeHost01 instalada en VirtualBox



Con el propósito de ser un laboratorio simulado y vulnerable, se desactivaron las actualizaciones automáticas de Windows Update y toda protección por defecto de Windows Defender.

Figura 61

Desactivación de protecciones de Windows en VM OPBASTA - EmployeeHost01

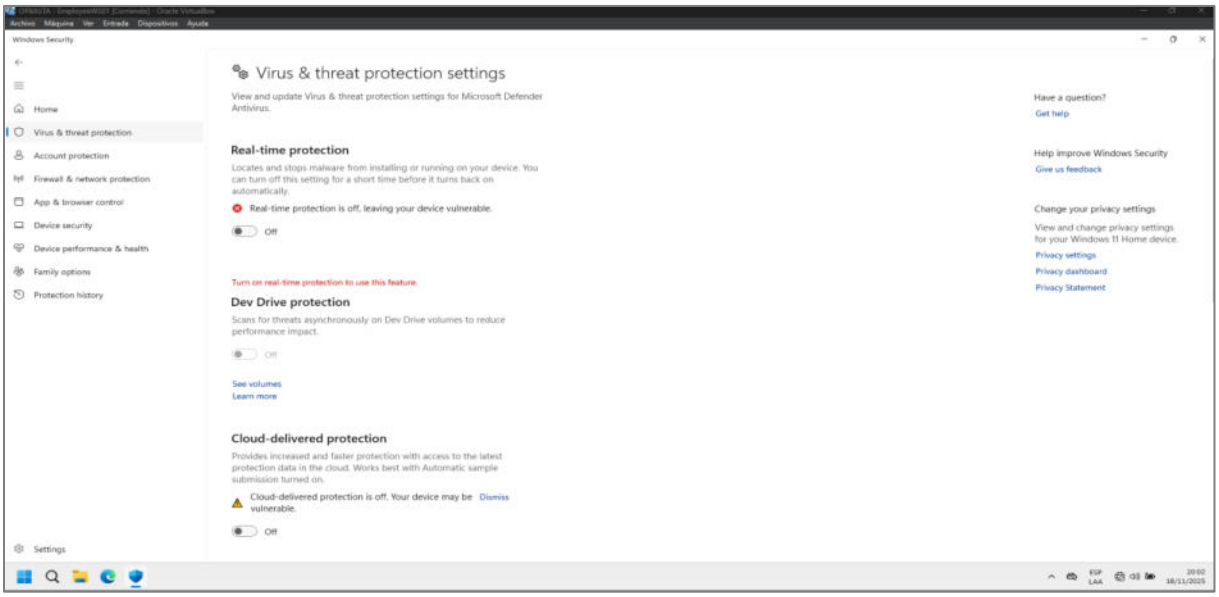
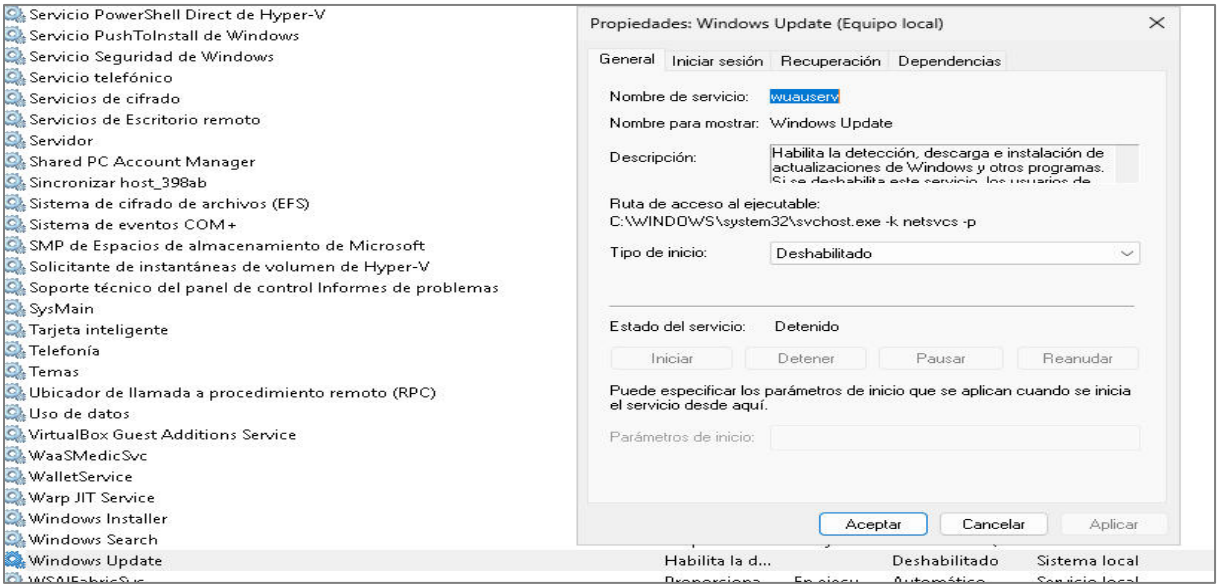


Figura 62

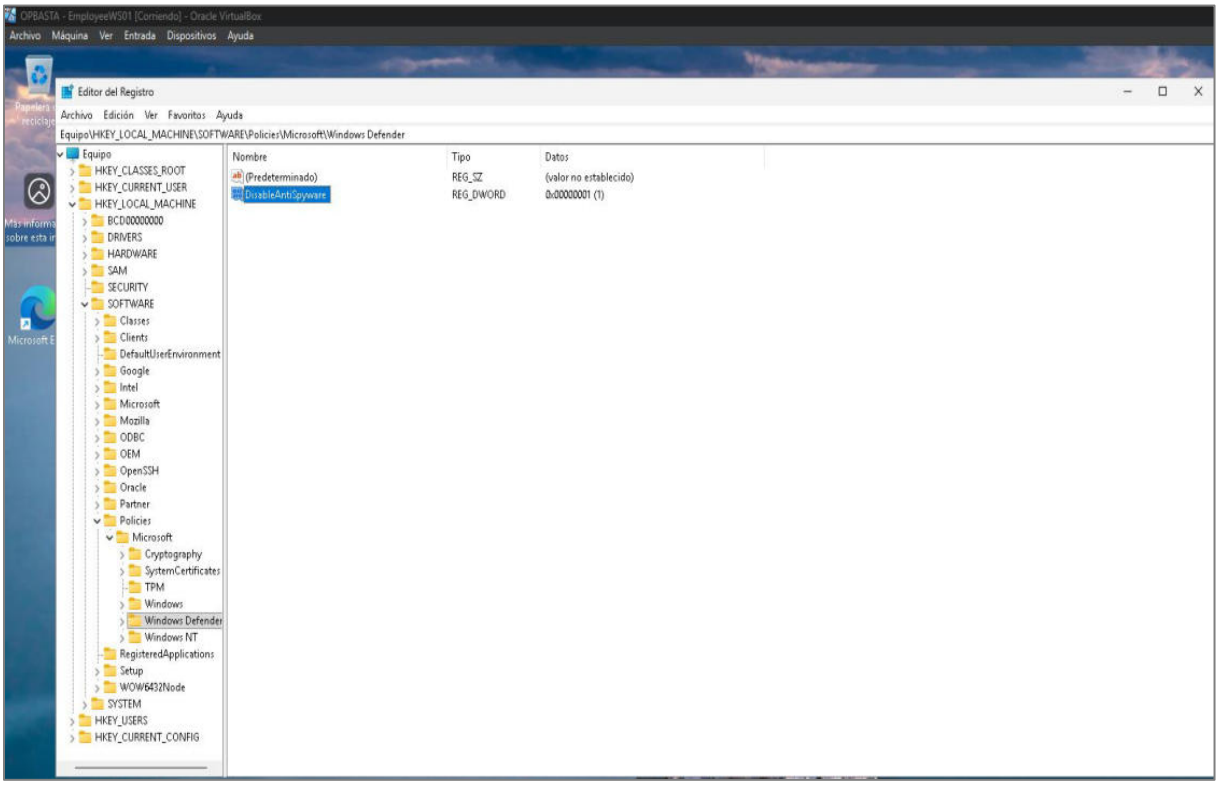
Desactivación de actualizaciones de Windows Update en VM OPBASTA - EmployeeHost01



Para desactivar permanentemente la protección de Windows Defender y que no se vuelva a iniciar cuando se apague el equipo, se modificó el registro "\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender" añadiendo una clave de 32 bits llamada "DisableAntiSpyware" y se asignó el valor "1".

Figura 63

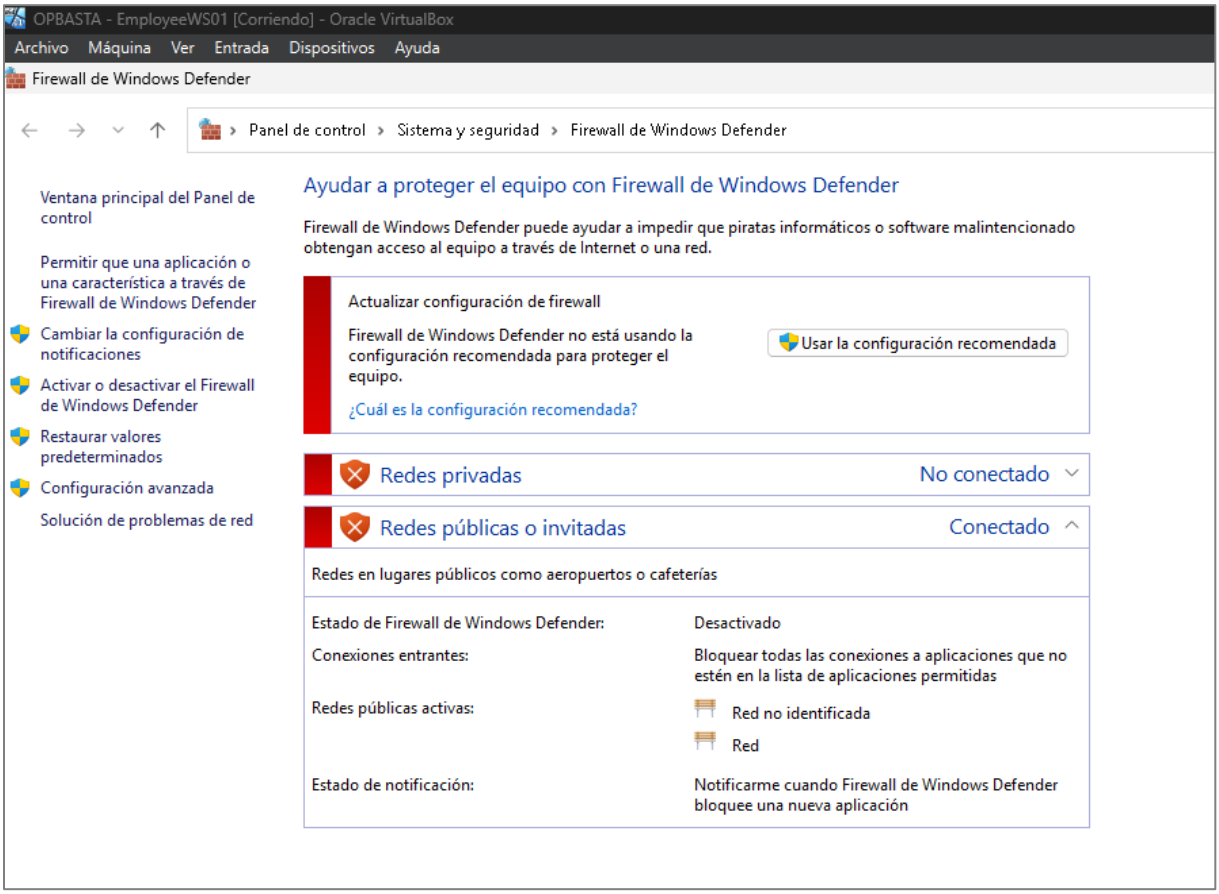
Desactivación de Windows Defender desde el Editor de Registro



Complementariamente, se desactivaron todos los perfiles del Firewall interno de Microsoft para evitar cualquier conflicto de conexión entre las máquinas del laboratorio.

Figura 64

Desactivación del Firewall de Windows Defender

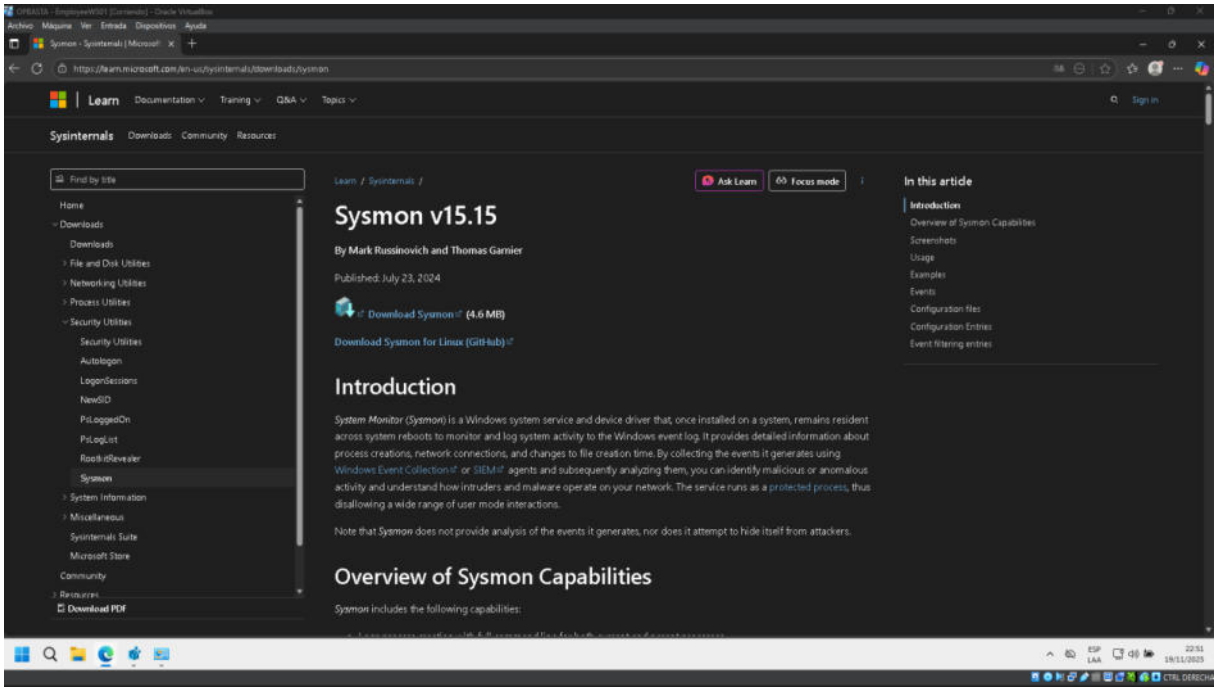


A continuación, se procedió con la instalación de todas las herramientas necesarias para ejecutar un análisis en mayor profundidad cuando el ransomware Black Basta sea ejecutado.

Como primera herramienta, se descargó Sysmon desde la web oficial: *https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon* ya que esta herramienta brinda telemetría en profundidad que permite conocer lo que está sucediendo por detrás en el OS, lo cual por defecto no se guarda en registros de Windows.

Figura 65

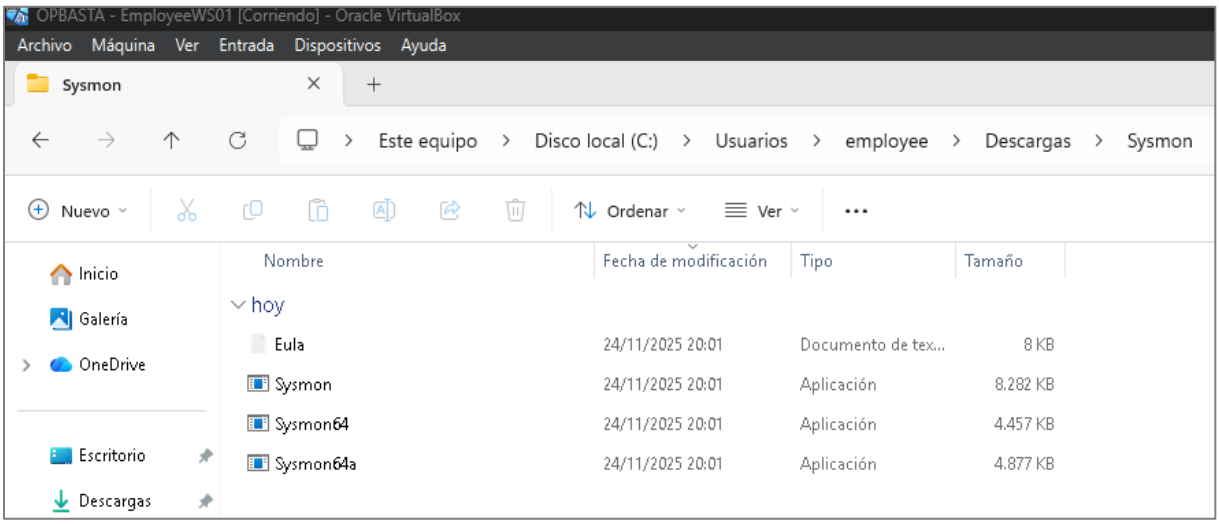
Descarga de herramienta Symon



Luego, se descomprimió el archivo descargado y se distinguieron 3 binarios.

Figura 66

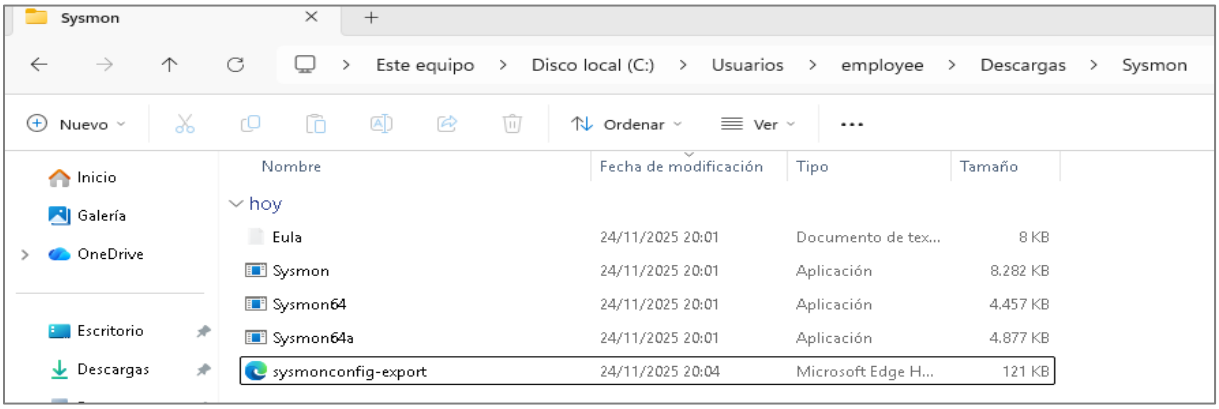
Vista del archivo de Sysmon descomprimido



Antes de la instalación Sysmon se descargó un archivo preconfigurado, el cual ya tiene establecido parámetros para detectar los eventos de seguridad más importantes y evitar ruido; para este proyecto se escogió el archivo más utilizado en la comunidad, el enlace es el siguiente: *https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml*

Figura 67

Descarga del archivo de configuración de Sysmon



Con el archivo de configuración listo, se procedió a instalar Sysmon usando el siguiente comando en Powershell modo administrador:

```
.\Sysmon64.exe -accepteula -i sysmonconfig-export.xml
```

Figura 68

Instalación de Sysmon de acuerdo con archivo de configuración

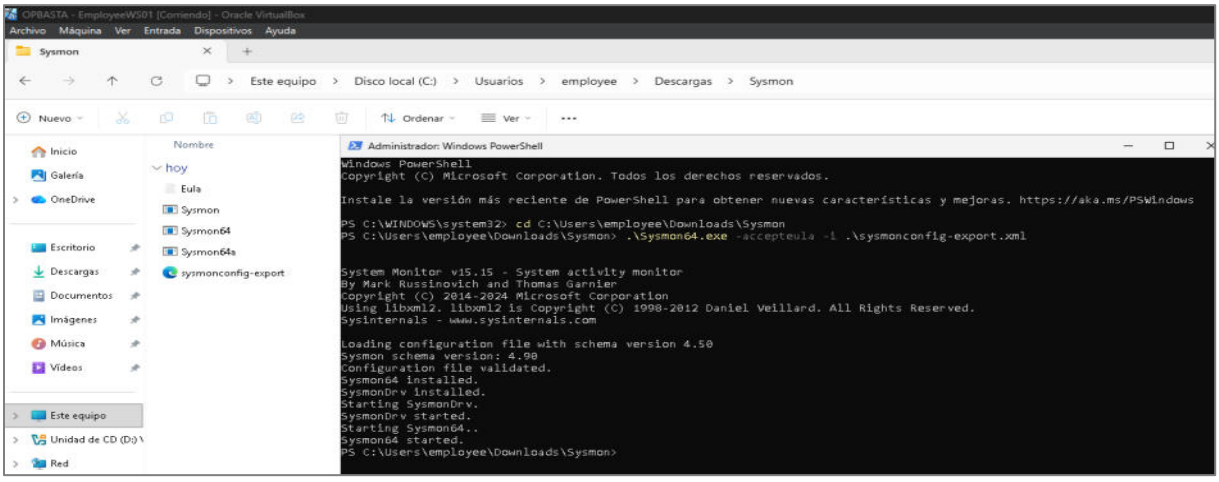
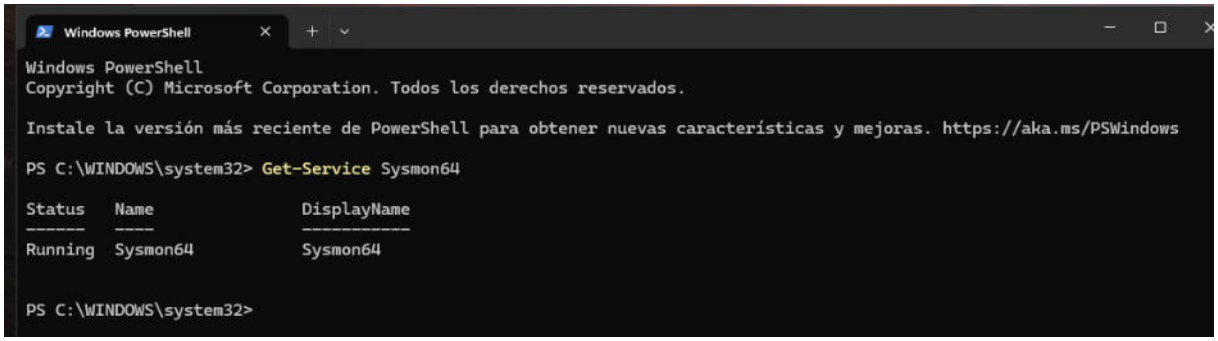


Figura 69

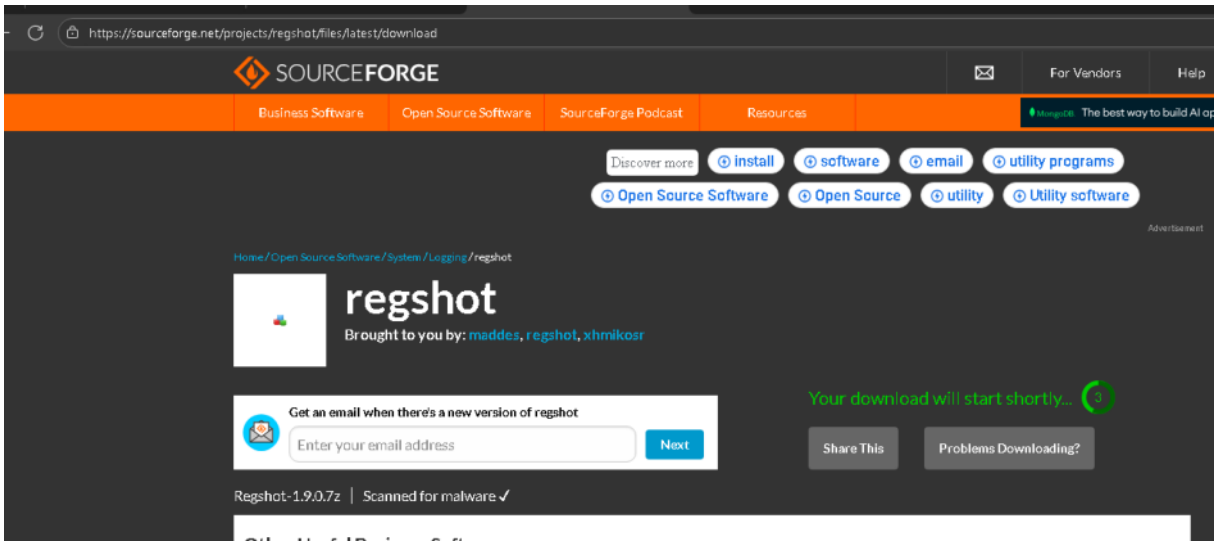
Verificación de instalación de Sysmon



Como segunda herramienta, se descargó Regshot para visualizar y comparar los cambios en Windows a nivel de registros del sistema, por ejemplo: creaciones, modificaciones, eliminaciones de datos realizados por el ransomware al ser ejecutado; el enlace de descarga usado es: *https://sourceforge.net/projects/regshot/*

Figura 70

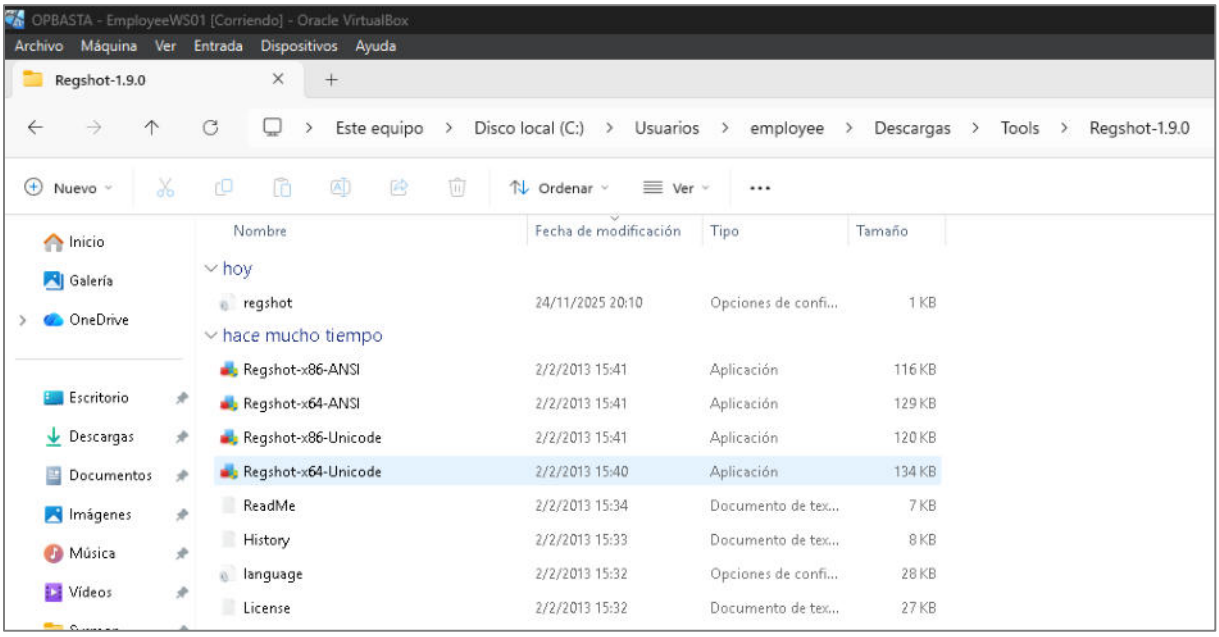
Descarga de herramienta Regshot



Una vez descargado, se descomprimió el archivo y dentro de este se observaron ejecutables listos para ser usados, por lo cual, no fue necesario realizar procesos de instalación.

Figura 71

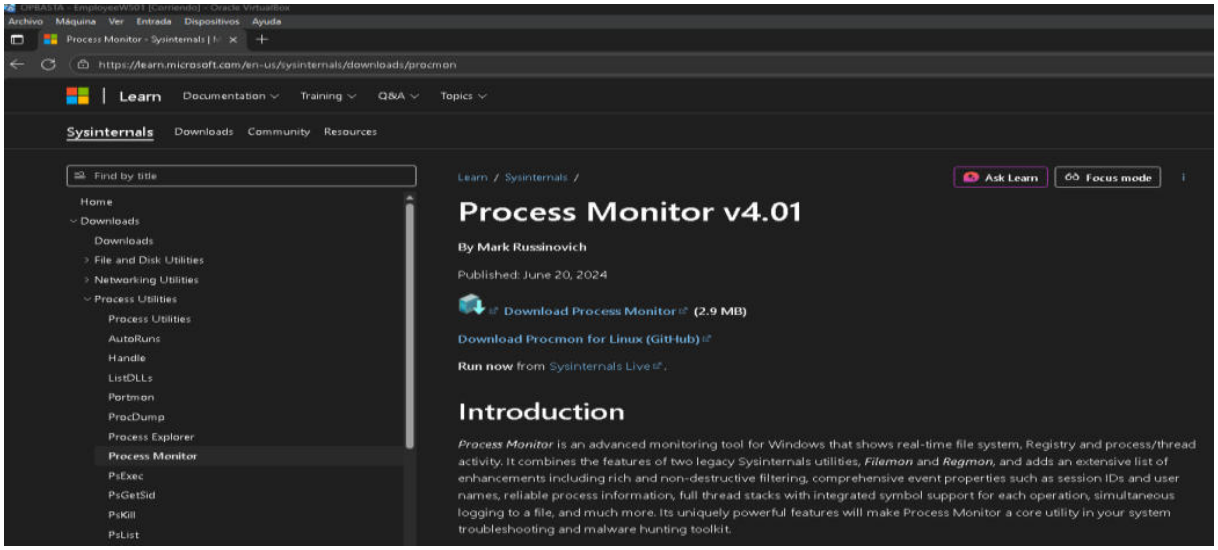
Vista del archivo Regshot descomprimido



Como tercera herramienta, se descargó Procmon desde su página oficial: *https://learn.microsoft.com/en-us/sysinternals/downloads/procmon* para tener visibilidad a nivel de procesos del OS.

Figura 72

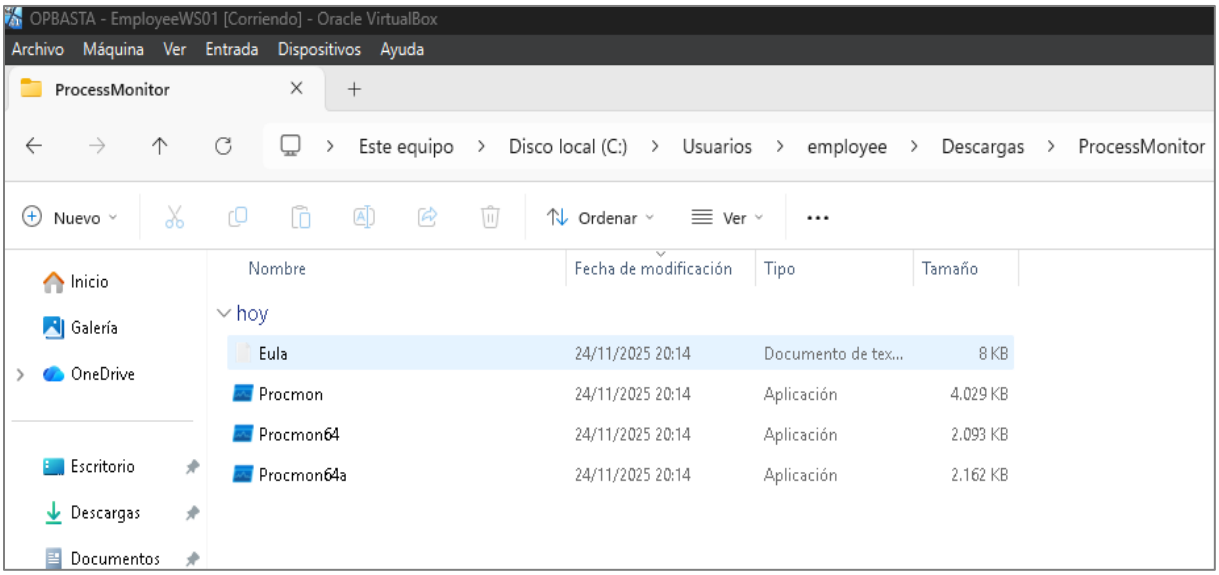
Descarga de herramienta Procmon



Una vez descargado, se descomprimió el archivo y dentro de este se observaron tres ejecutables listos para ser usados, por lo cual, no fue necesario realizar procesos de instalación, debe ser usado al momento del análisis del malware.

Figura 73

Vista del archivo Procmon descomprimido



Como cuarta herramienta, se descargó Process Hacker desde su página oficial: *https://sourceforge.net/projects/processhacker/* debido a que ayuda al análisis de procesos en tiempo real distinguiendo entre procesos padres e hijos, como también los cambios con su origen.

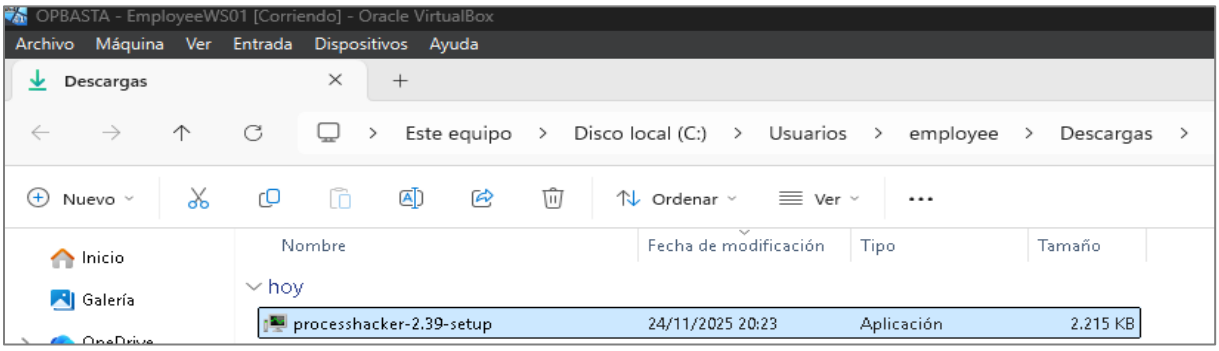
Figura 74

Descarga de herramienta Process Hacker

Una vez descargado, se ejecutó el archivo como administrador para proceder con la instalación de la herramienta.

Figura 75

Ejecución del archivo descargado processhacker-2.39-setup



Durante la instalación se mantuvieron los parámetros predeterminados, y al finalizar se cerró el instalador sin ejecutar la herramienta, ya que será usada al momento del análisis.

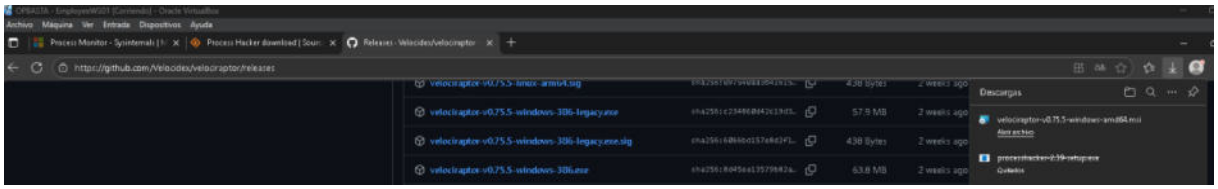
Figura 76

Instalación de herramienta Process Hacker

Como quinta herramienta, se instaló el agente de Velociraptor, el cual hace la función de mini EDR en el laboratorio construido; para esto, se descargó el ejecutable desde la página oficial: *https://github.com/Velocidex/velociraptor/releases/download/v0.75/velociraptor-v0.75.5-windows-amd64.msi*

Figura 77

Descarga de Velociraptor para Windows



Antes de proceder con la instalación del agente, se transfirió el archivo de configuración generado en secciones anteriores desde servidor VM OPBASTA - ThreatAnalysis01. Para la transferencia del archivo se creó un servidor simple con Python.

Figura 78

Creación de servidor para transferencia de archivos entre VM ThreatAnalysis01 y VM EmployeeHost01

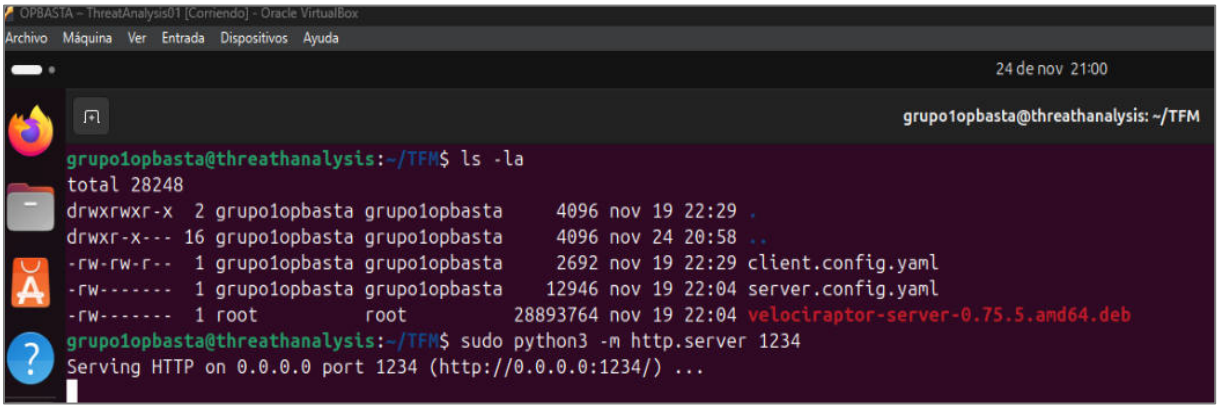


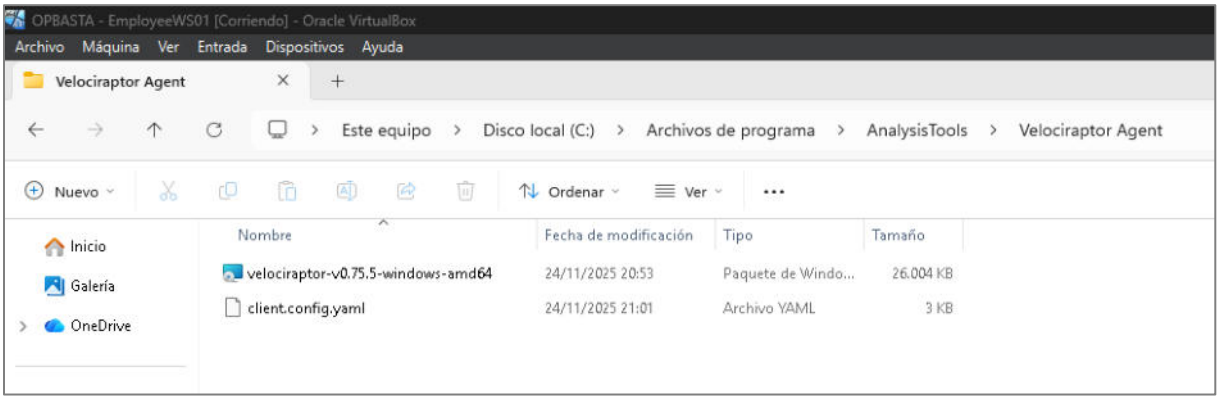
Figura 79

Vista del servidor con los archivos a descargar

Una vez descargado, se ubicó el archivo en el mismo directorio donde se encuentra el instalador.

Figura 80

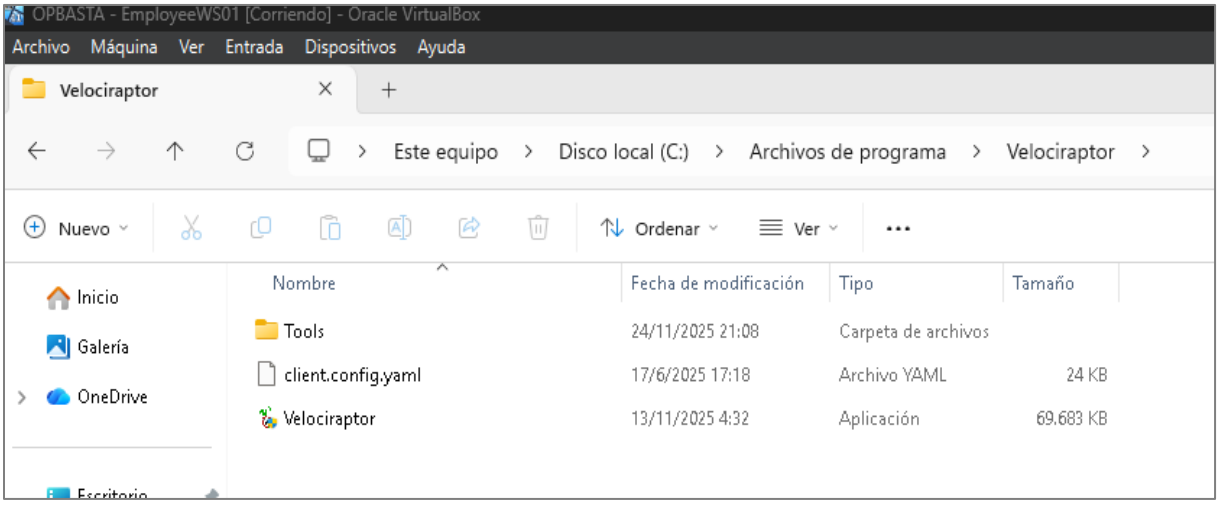
Archivos descargados desde el servidor de transferencia



Seguidamente, se ejecutó el instalador (.msi) y se dio clic en “siguiente” hasta que el proceso de instalación finalice. Para verificar la correcta instalación del cliente de Velociraptor se revisó la ruta “C:\Program Files\Velociraptor”.

Figura 81

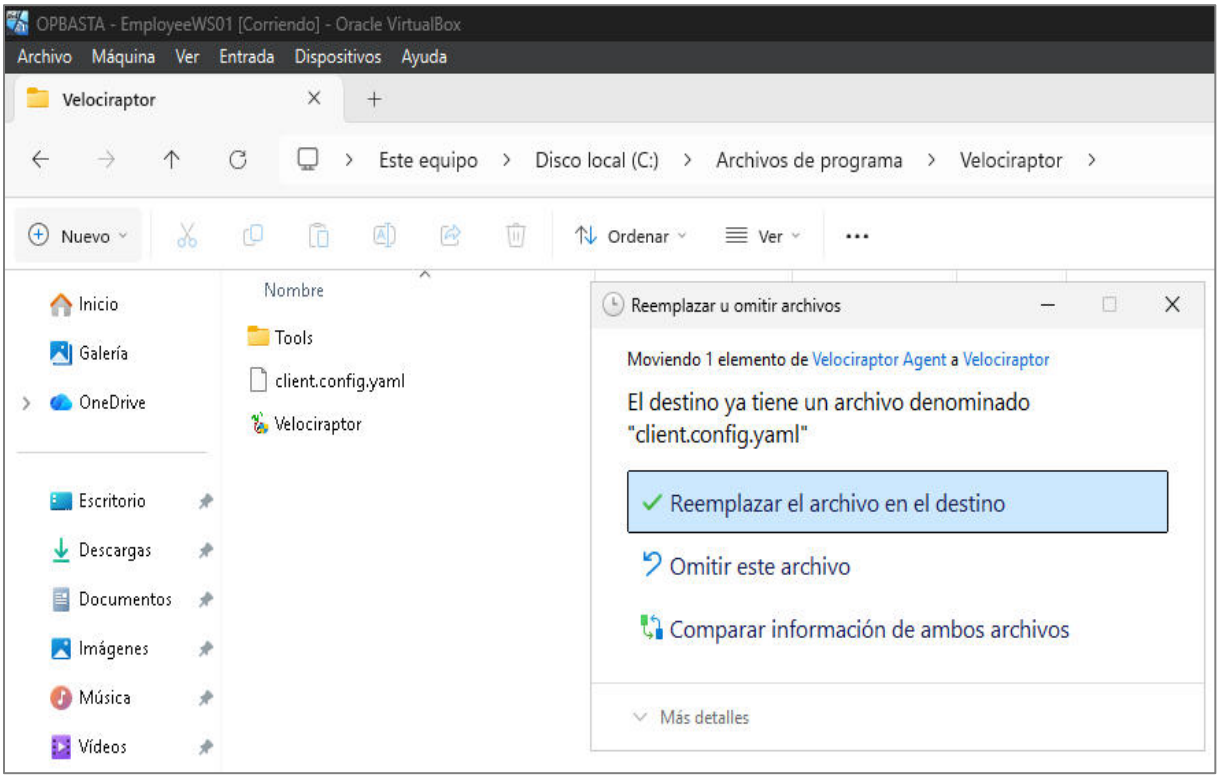
Instalación del cliente de Velociraptor



Al revisar la ruta se observó el archivo “client.config.yaml”; y, como siguiente paso se reemplazó dicho archivo por el descargado desde VM ThreatAnalysis01.

Figura 82

Reemplazo del archivo client.config.yaml



Desde el momento en que el archivo de configuración fue reemplazado, el agente inició su función de reportar toda información hacia el servidor de Velociraptor (instalado en VM OPBASTA - ThreatAnalysis01).

Por lo tanto, al abrir el panel de administración en VM OPBASTA -ThreatAnalysis01 se pudo visualizar, en la pestaña de “Clients”, a la víctima VM OPBASTA - EmployeeHost01 Windows 11.

Figura 83

Visualización del cliente de Velociraptor desde la interfaz gráfica del servidor

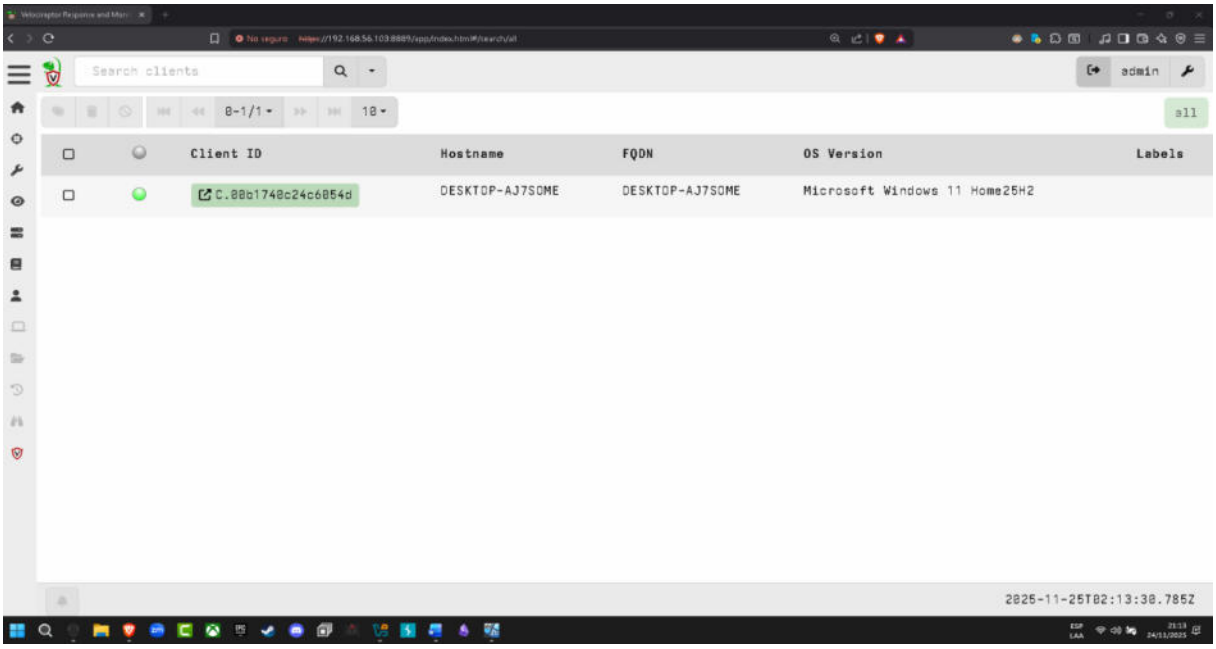
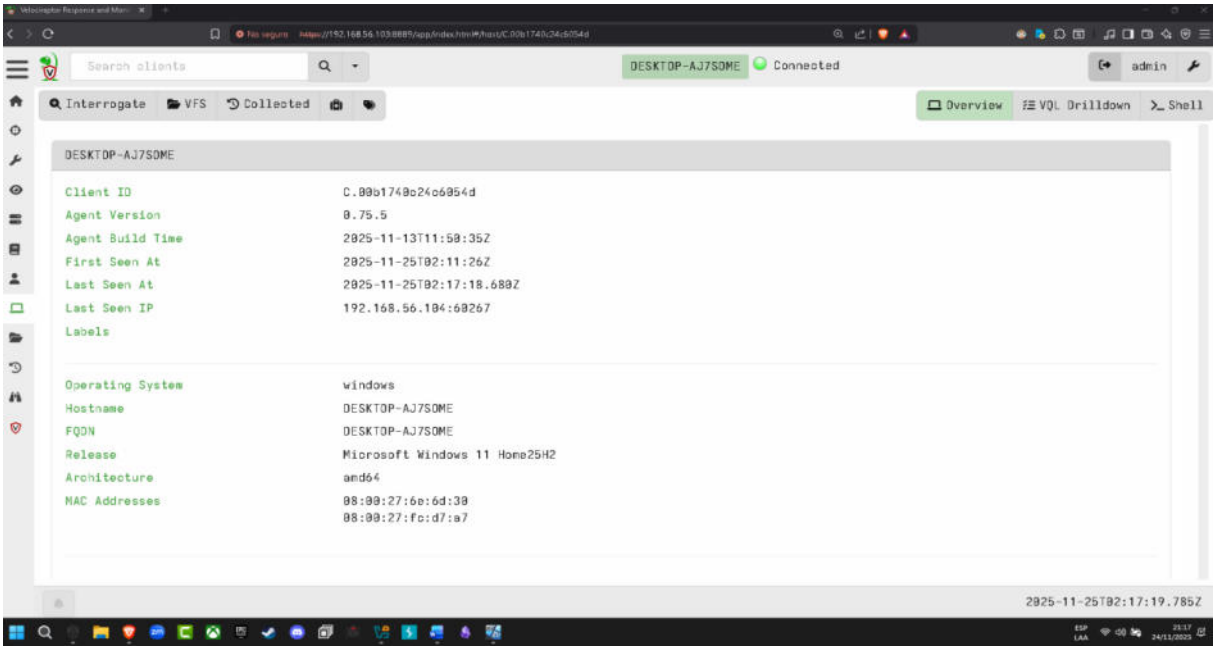


Figura 84

Reporte entregado por el cliente de Velociraptor sobre la máquina Windows

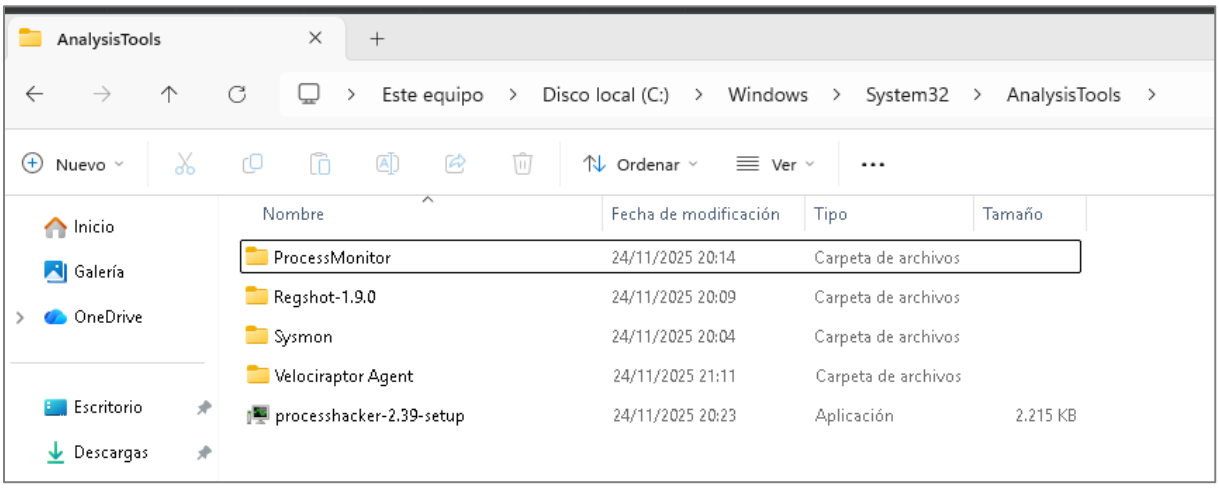


Continuando con la implementación de la máquina víctima VM OPBASTA - EmployeeHost01, una vez que se culminó con la instalación de todas las herramientas de análisis y recolección necesarias, éstas fueron reubicadas hacia el directorio *"C:\Windows\System32\AnalysisTools"* para evitar que el ransomware las encripte.

Usualmente los ransomware evitan cifrar archivos del sistema, como por ejemplo los ubicados en rutas como "Windows" o "System32", esto para evitar que el sistema se rompa y no se pueda visualizar el archivo con el mensaje de extorsión.

Figura 85

Herramientas de análisis reubicadas en AnalysisTools de System32



Para finalizar el despliegue de laboratorio, se realizó un Snapshot de todas las máquinas virtuales que conforman el entorno de pruebas; de esta manera, se podrá recuperar el estado actual de cada una, en caso de que algo inesperado suceda durante la ejecución del malware o el sistema se rompa. Con el snapshot se provee un entorno controlado sobre el cual se puede ejecutar el malware las veces que sean necesarias y realizar el análisis en mayor profundidad para la detección de diferentes comportamientos.

Figura 86

Snapshot de VM OPBASTA - EvilWebServer01 previo a la ejecución del ransomware

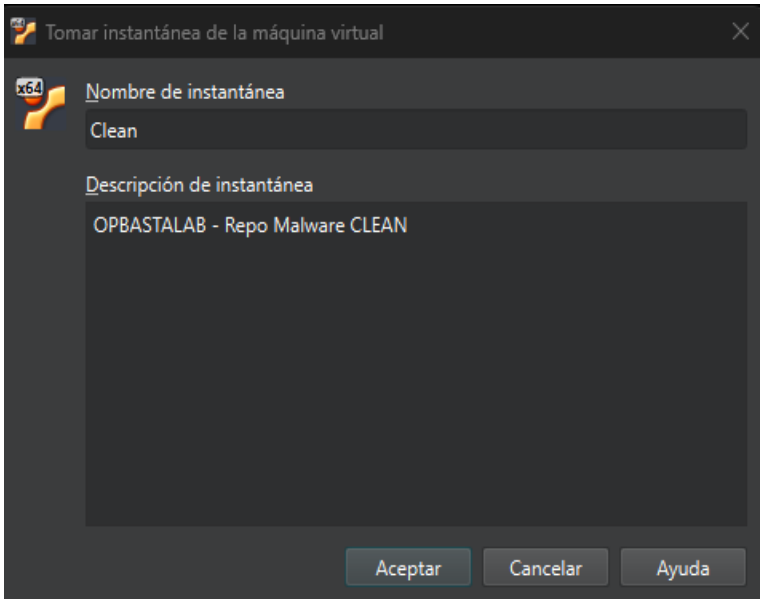


Figura 87

Snapshot de VM OPBASTA - ThreatAnalysis01 previo a la ejecución del ransomware

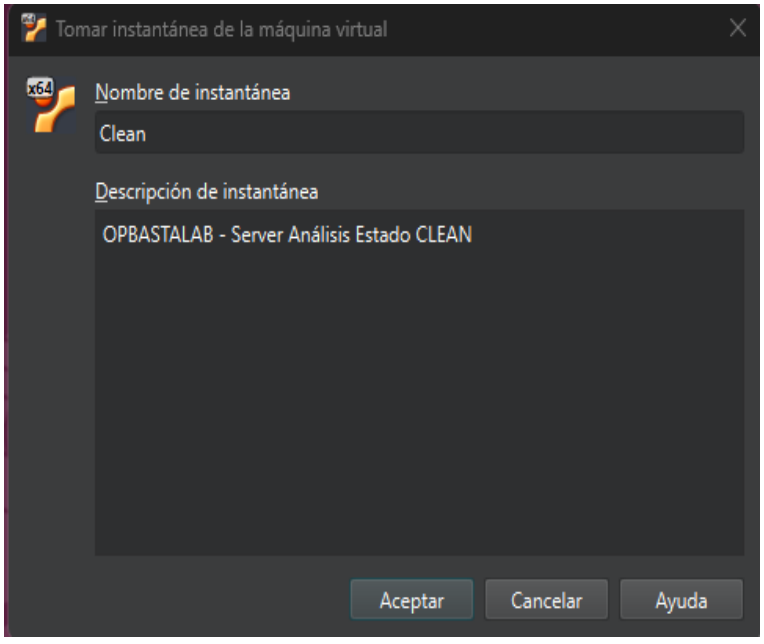


Figura 88

Snapshot de VM OPBASTA - EmployeeHost01 previo a la ejecución del ransomware (parte 1)

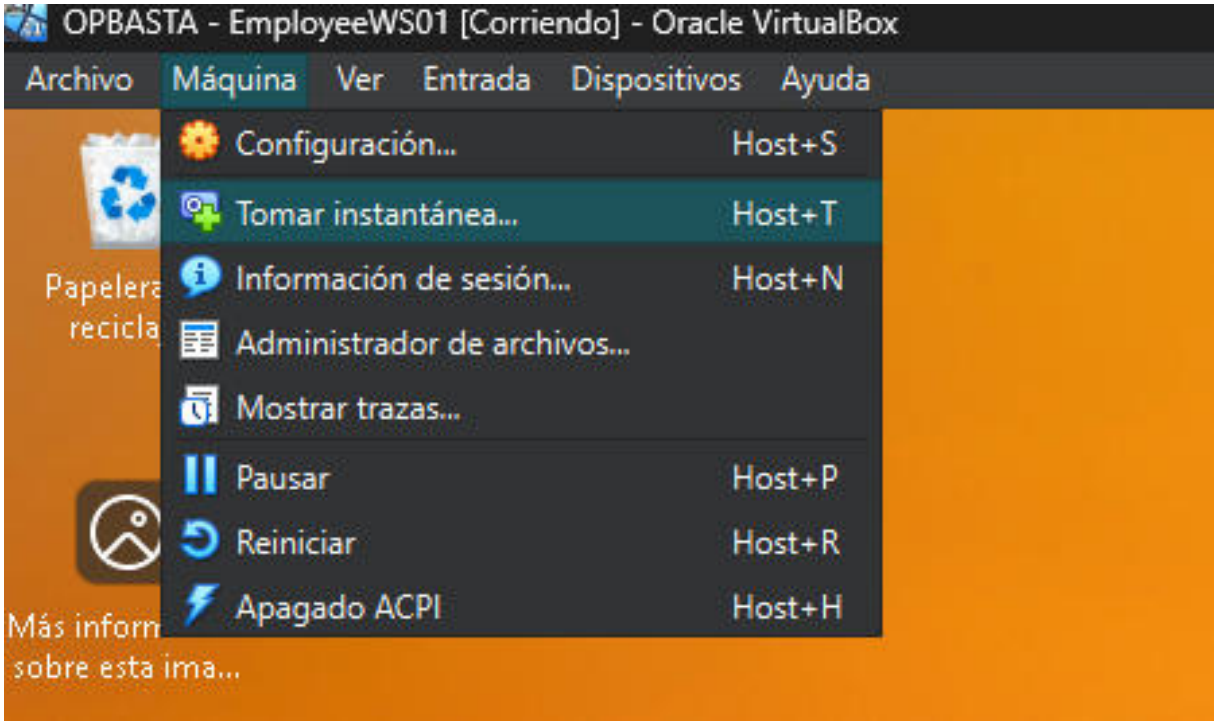
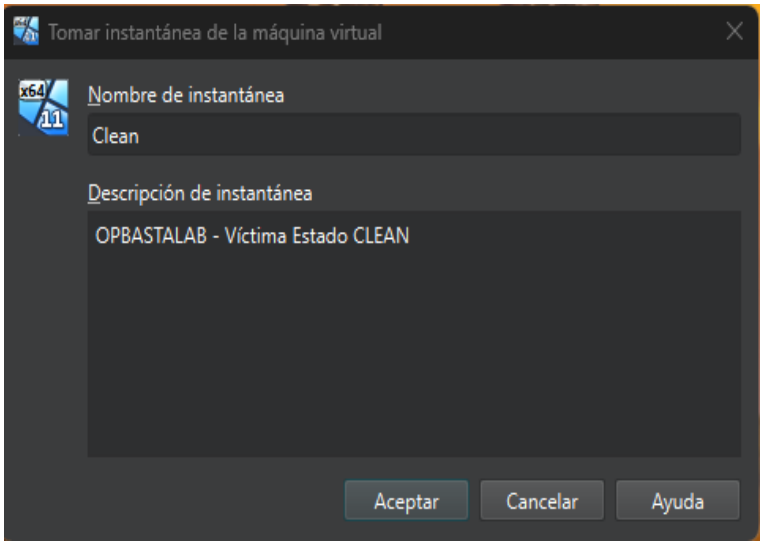


Figura 89

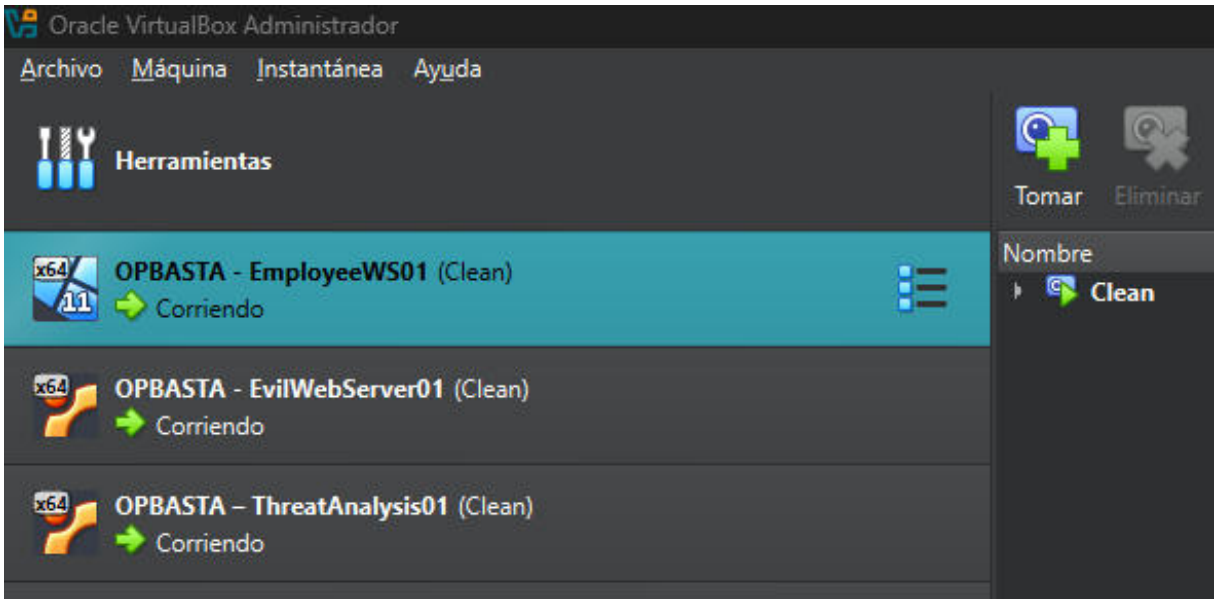
Snapshot de VM OPBASTA - EmployeeHost01 previo a la ejecución del ransomware (parte 2)



Finalmente, se ha preparado el laboratorio de pruebas con tres máquinas virtuales involucradas, que nos permitirán desarrollar el escenario planteado. ¡Laboratorio listo para operar!

Figura 90

Laboratorio completo con las tres máquinas



3.1.7.4. Portal "Operation Basta" - Plataforma Integrada de Distribución Segura de Muestras y Herramientas Forenses (Valor Agregado). Como parte del ecosistema desarrollado para el análisis controlado del ransomware Black Basta, se implementó un portal web interno denominado "Operation Basta". El objetivo de esta implementación como valor agregado fue de centralizar, normalizar y facilitar el acceso a todos los recursos necesarios para la investigación académica realizada en este proyecto.

Este portal actúa como un repositorio seguro y autosuficiente que concentra tanto las muestras reales del ransomware, obtenidas desde fuentes reconocidas como VX-Underground y MalwareBazaar, como también, un catálogo curado de herramientas forenses e instrumentos de ingeniería inversa empleados durante el análisis.

Figura 91

Portal "Operation Basta"- Plataforma Integrada de Distribución Segura de Muestras y Herramientas Forenses

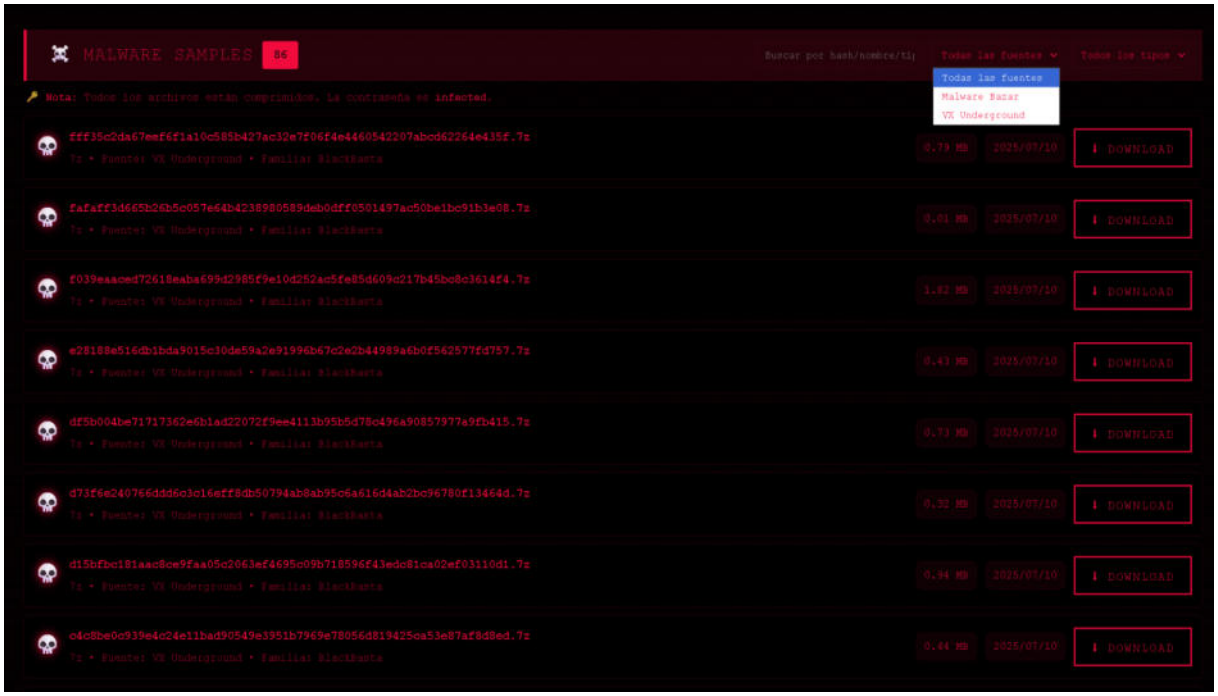
Este servidor web no cumple únicamente un rol de conveniencia; sino que, la finalidad es garantizar que todo el proceso de análisis y el flujo de trabajo se mantengan 100% offline y aislados, evitando dependencias de Internet y reforzando la seguridad operativa del laboratorio. Este almacenamiento local contiene todas las muestras empaquetadas y protegidas con la contraseña estándar "infected", logrando un entorno de investigación reproducible, consistente y reducido a un único punto de acceso seguro dentro del laboratorio.

La plataforma desplegada organiza los recursos en tres secciones fundamentales:

- **Muestras del ransomware Black Basta:** Se catalogaron todas las variantes disponibles públicamente en la Clear web. Cada muestra incluye metadatos claves, tales como, familia, fuente, tamaño, fecha, hash, tipo de comprimido (7z o zip). Al habilitar descargas directas desde el servidor interno, se evita la manipulación de repositorios externos, reduciendo la exposición y manteniendo la integridad del laboratorio.

Figura 92

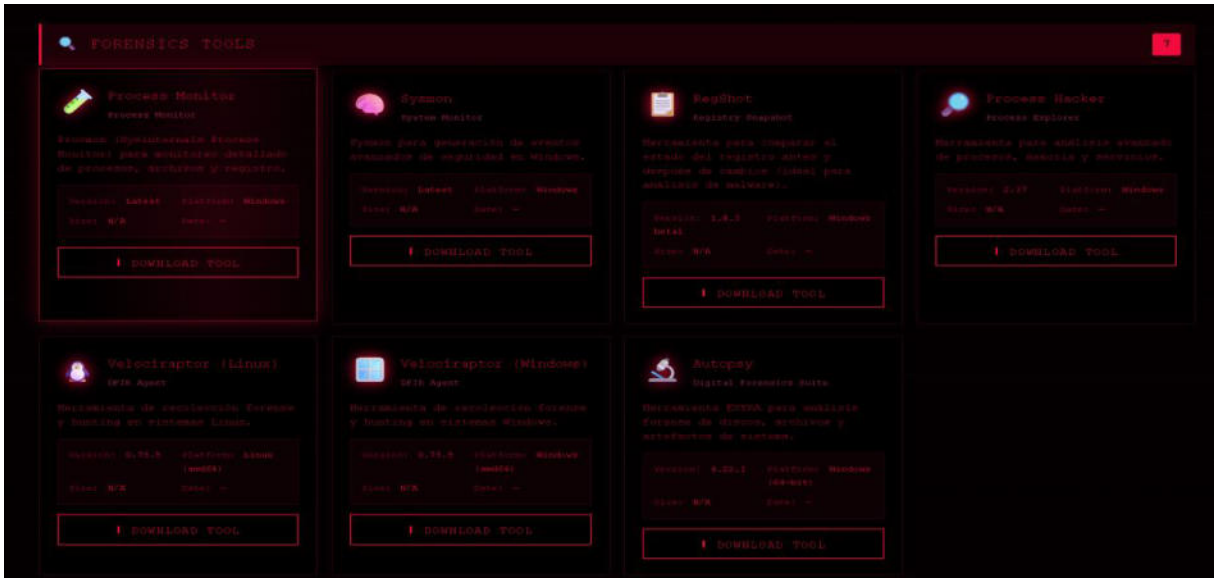
Portal "Operation Basta"- Lista de muestras de malware disponibles a descargar



- **Herramientas Forenses profesionales:** En la plataforma se encuentran disponibles para descarga directa herramientas como Procmon, Sysmon, Regshot, Process Hacker, Velociraptor y Autopsy. Cada herramienta es presentada con su versión, plataforma y descripción técnica, ofreciendo una experiencia tipo "DFIR Toolkit Repository". Esto asegura que cualquier analista dentro del laboratorio trabaje con el mismo conjunto estandarizado.

Figura 93

Portal "Operation Basta"- Sección de herramientas forenses disponibles



- **Herramientas de Ingeniería Inversa:** Adicionalmente, se integraron herramientas críticas como Ghidra, Detect It Easy y x64dbg. El objetivo es proporcionar un entorno completo para análisis estático, dinámico y de empaquetamiento. Todas las herramientas están listas para su uso sin necesidad de conexiones externas.

Figura 94

Portal "Operation Basta"- Sección de herramientas de ingeniería inversa disponibles

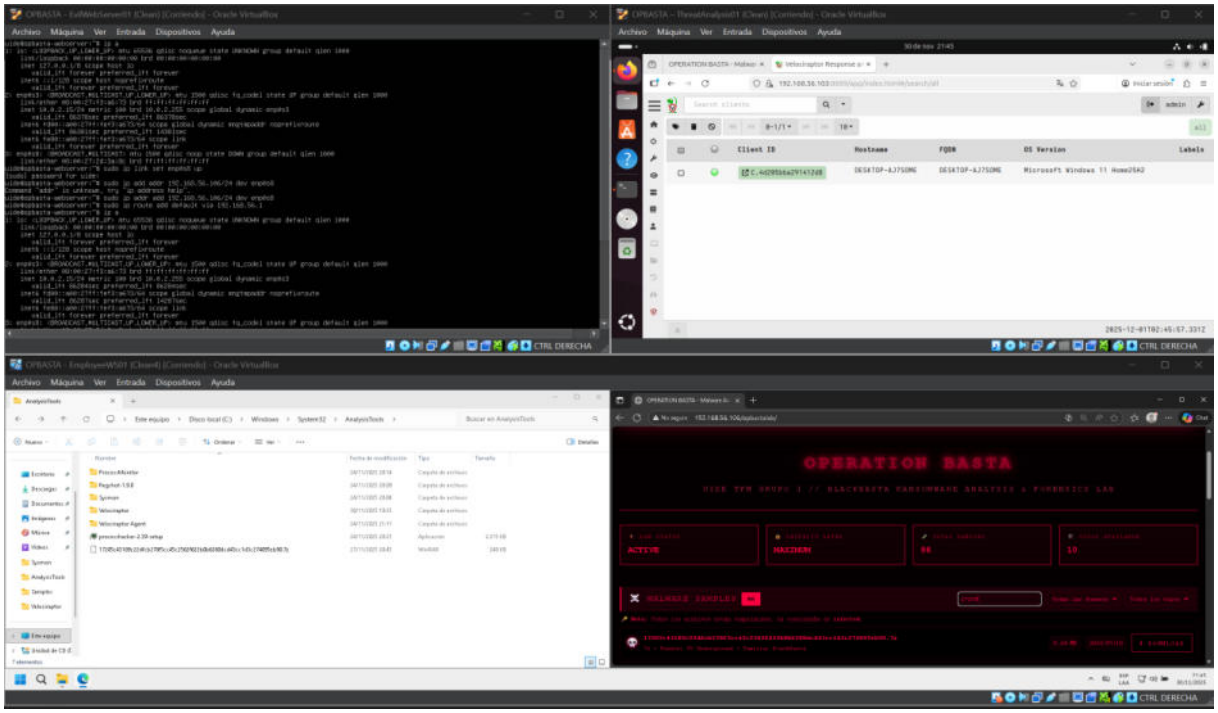
Como se pudo observar en las imágenes anteriores, el diseño visual del portal sigue una estética tipo “Malware Research Lab”, lo cual refuerza su identidad como plataforma interna de análisis avanzado. Además, el portal incluye advertencias explícitas sobre los riesgos y obligaciones legales del manejo de malware real en ambientes no controlados.

En conjunto, el portal “*Operation Basta”* se convierte en un componente diferenciador y de alto valor agregado dentro del laboratorio construido; entre las ventajas se resumen en los siguientes puntos:

- Centralización de recursos,
- Elimina el uso de dependencias externas,
- Estandariza el entorno de trabajo,
- Permite que la experimentación sea segura, repetible y auditada.

Figura 95

Vista del laboratorio completo y operativo



Por todo lo indicado, el portal "*Operation Basta"* se consolida como un ecosistema forense completamente autónomo, alineado con las mejores prácticas internacionales para el estudio técnico del comportamiento del ransomware.

3.2. Ejecución del Ransomware Black Basta en VM OPBASTA - EmployeeHost01

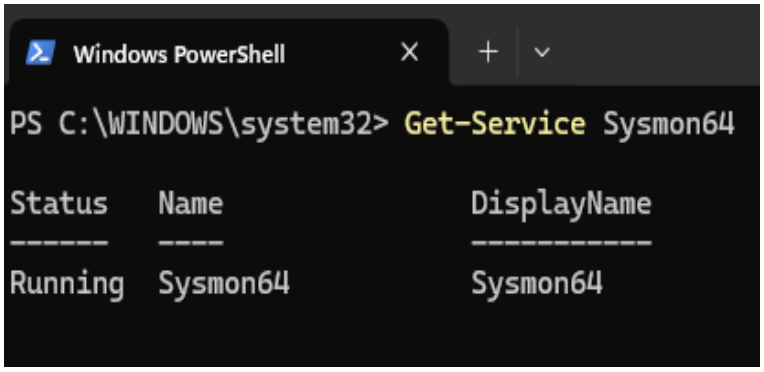
3.2.1. Preparación de VM OPBASTA - EmployeeHost01 Antes de la Ejecución del Ransomware.

En primer lugar, se verificó que Sysmon esté ejecutándose correctamente, para esto desde la ventana de PowerShell con permisos de administrador se ingresó el siguiente comando:

```
Get-Service Sysmon64
```

Figura 96

Verificación de que Sysmon se encuentra activo



A continuación, se ejecutó la herramienta Regshot para crear un snapshot del registro actual del equipo Windows antes de que se ejecute el ransomware. Se dio clic en "1st shot", se esperó a que finalice la captura, y luego, se dio clic en "Shot and Save" para guardar el primer snapshot obtenido.

Figura 97

Primer Snapshot desde Regshot antes de la ejecución del ransomware

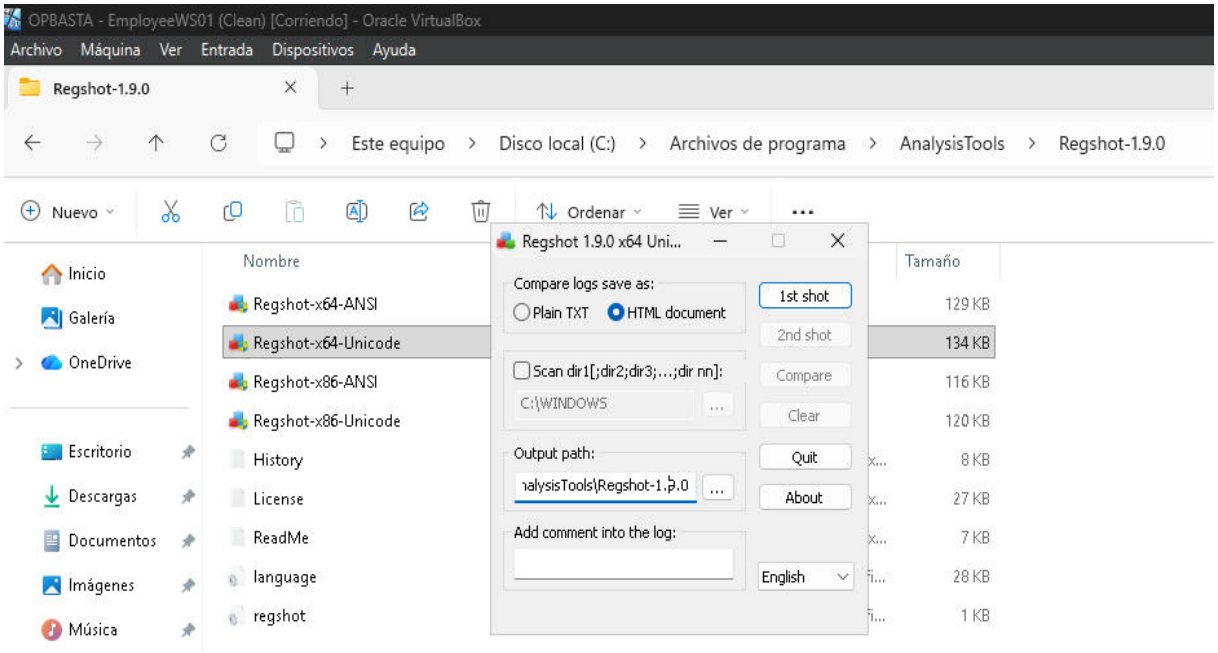


Figura 98

Primer shot de Regshot

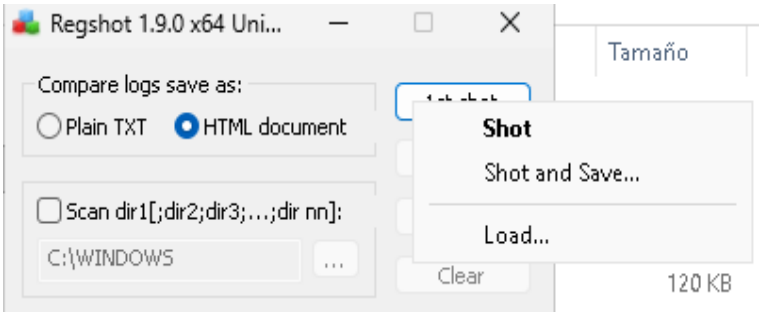
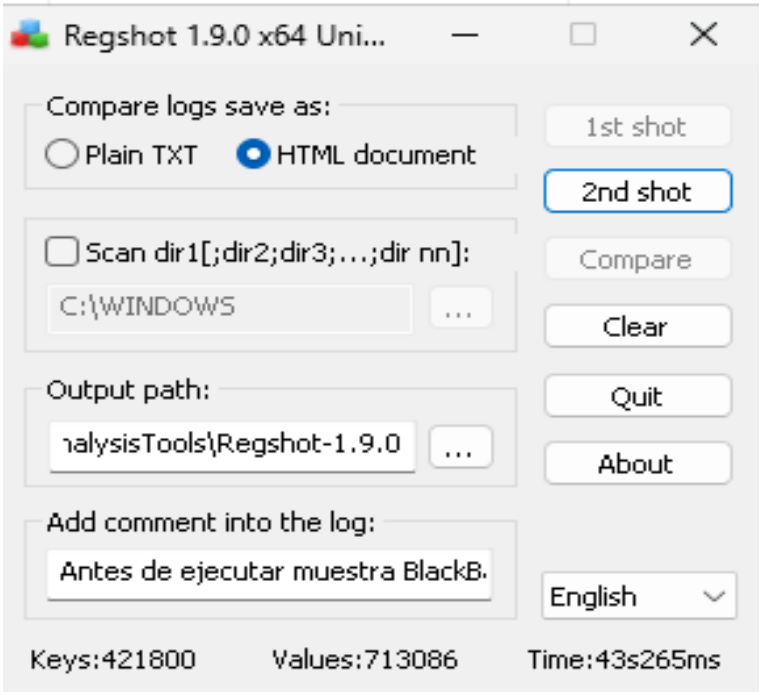


Figura 99

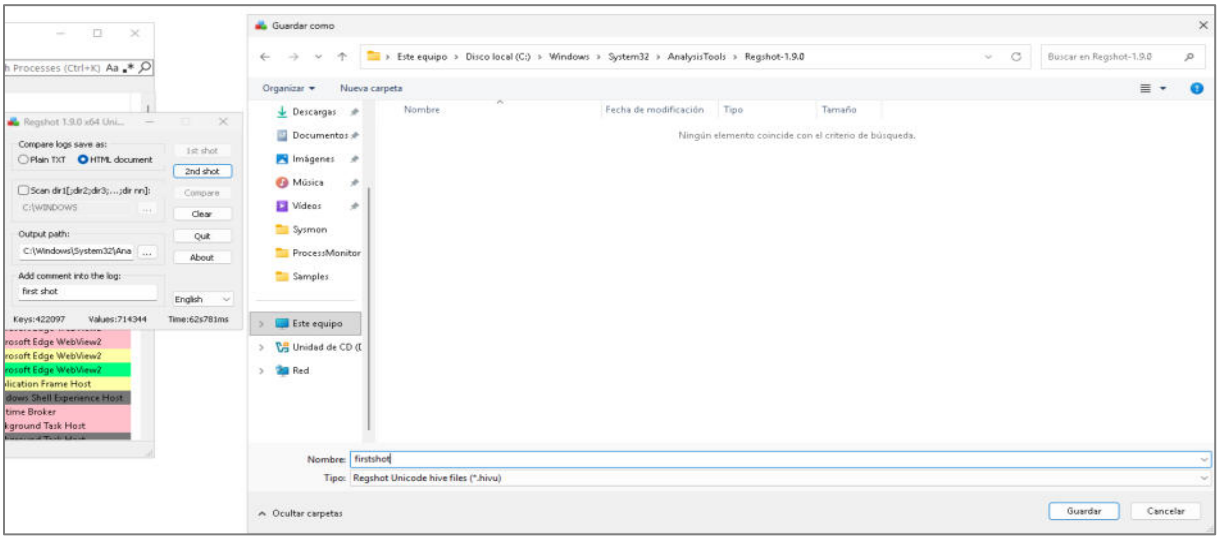
Finalización del primer shot Regshot



Se guardó el primer shot en un directorio seguro de System32 (en donde se alojaron las demás herramientas), para evitar que al ejecutar el ransomware se corrompa.

Figura 100

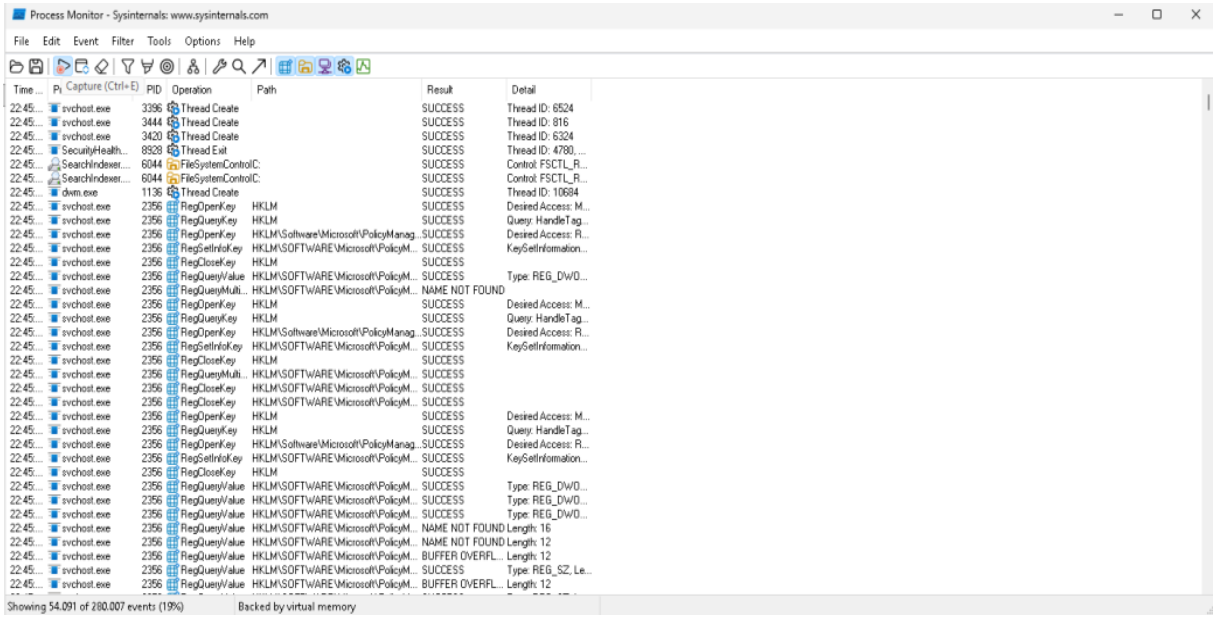
Primer shot de Regshot guardado en directorio dentro de System32



La tercera herramienta que se preparó fue Procmon, se la ejecutó con permisos de Administrador.

Figura 101

Ejecución de Procmon con permisos de administrador

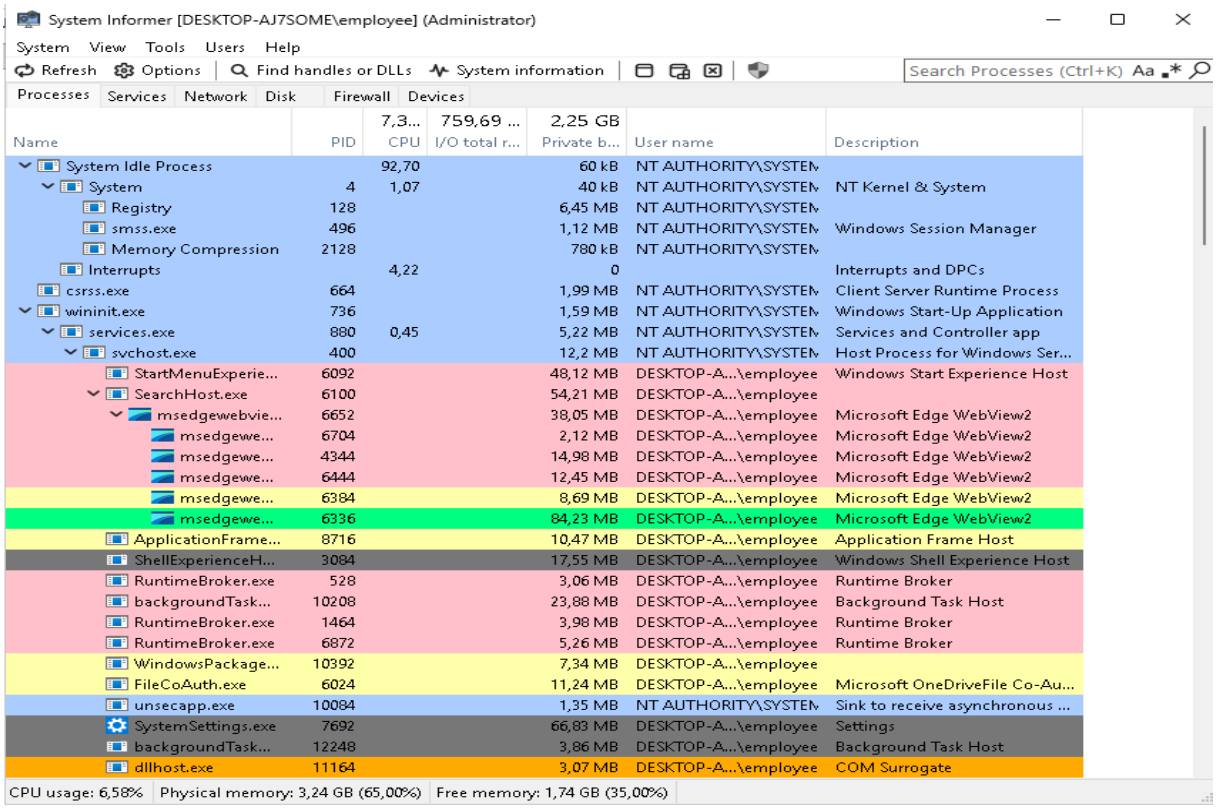


Justo un momento antes de ejecutar el ransomware, se presionó la combinación de teclas “CTRL+E” en Procmon para empezar a capturar todo.

Seguidamente, se preparó la cuarta herramienta Process Hacker, también ejecutada con permisos de Administrador.

Figura 102

Ejecución de Process Hacker con permisos de administrador



3.2.2. Ejecución del Ransomware Black Basta. Se ejecutó el ransomware como Administrador desde la consola de PowerShell.

Figura 103

Ejecución de la muestra de Black Basta desde PowerShell

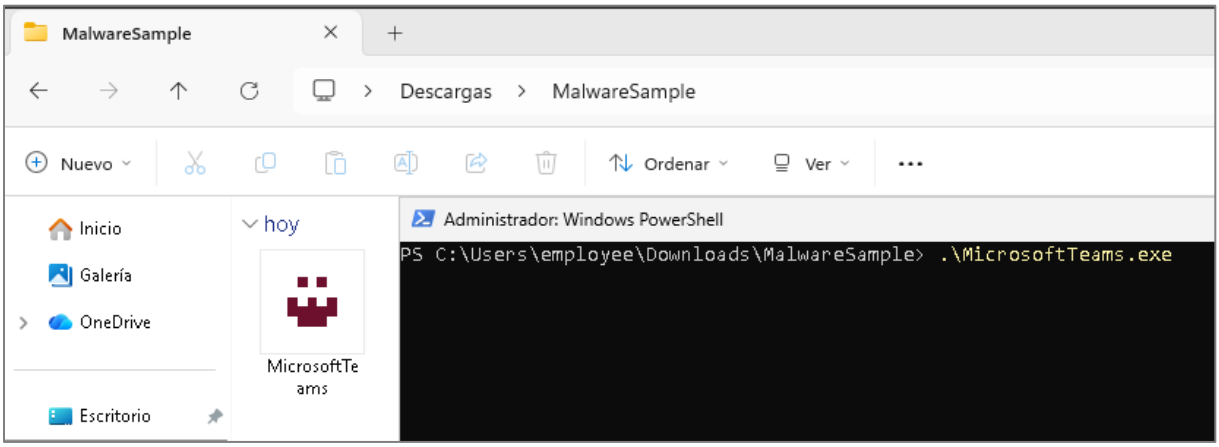
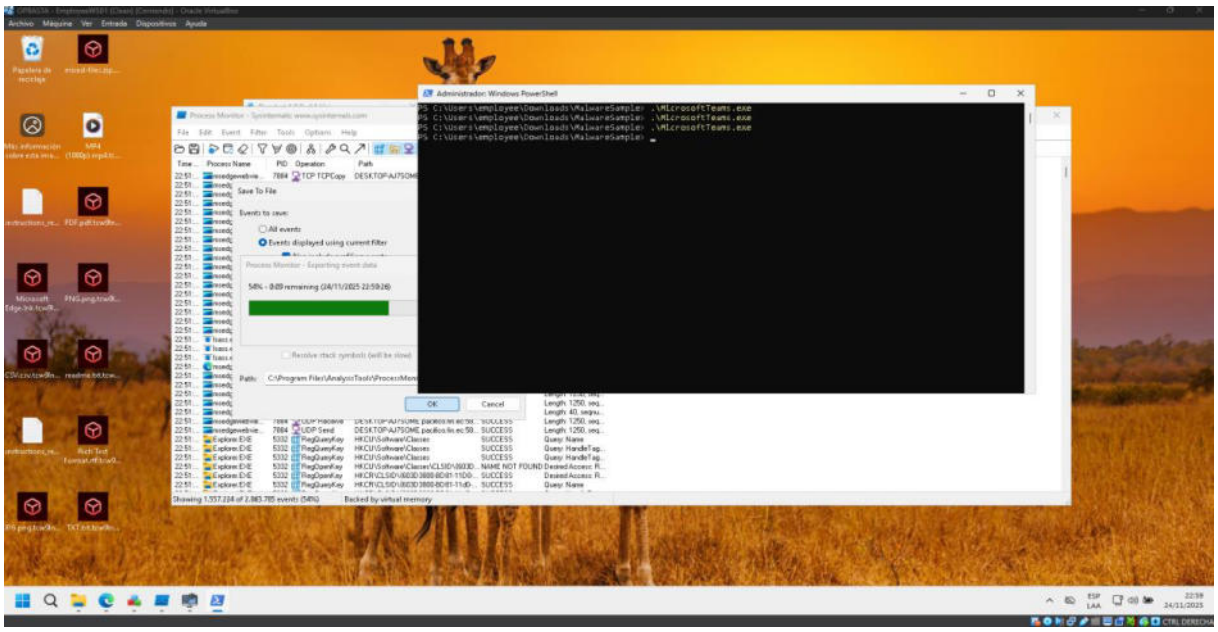


Figura 104

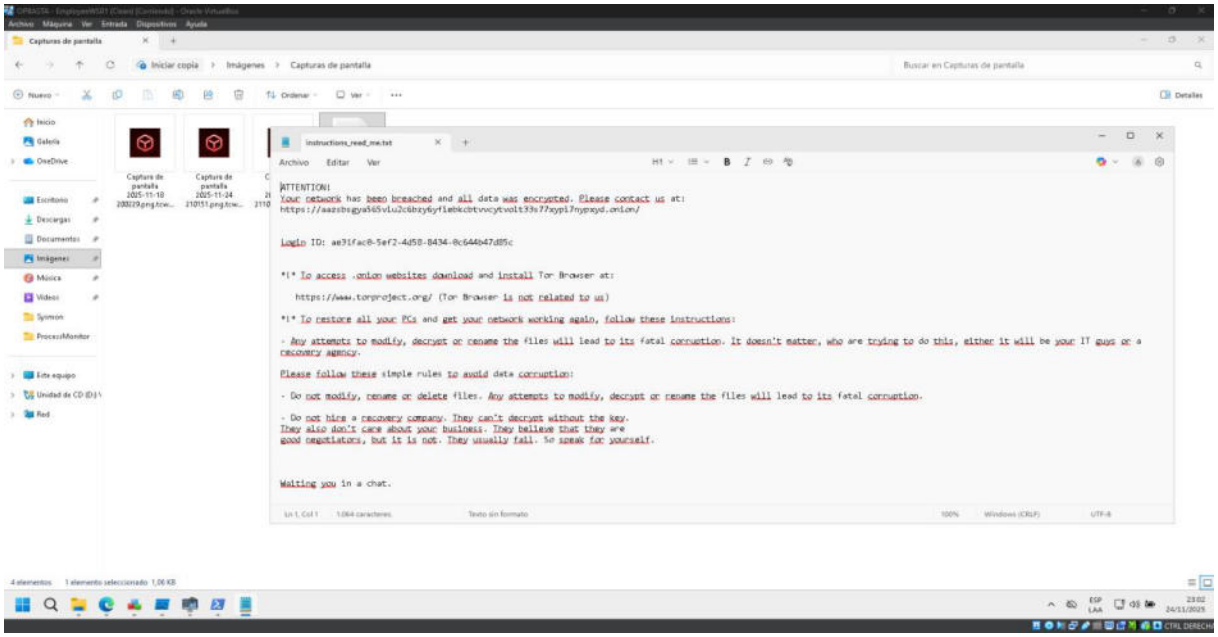
Vista de la ejecución de Black Basta y de la recepción de actividad en las herramientas



Como se observa en la **Figura 104** y **Figura 105**, el ransomware encriptó toda la información del sistema y aparecieron mensajes extraños “readme”.

Figura 105

Archivos encriptados por Black Basta y nota de readme



3.3. Recolección de Evidencias Tras la Ejecución del Ransomware

Se ejecutó Regshot para obtener una segunda captura de sistema infectado. Se dio clic en "2nd shot", y luego en "Shot and Save". También se lo ubicó en la ruta segura junto al primer shot.

Figura 106

Segundo shot posterior a la ejecución de Black Basta

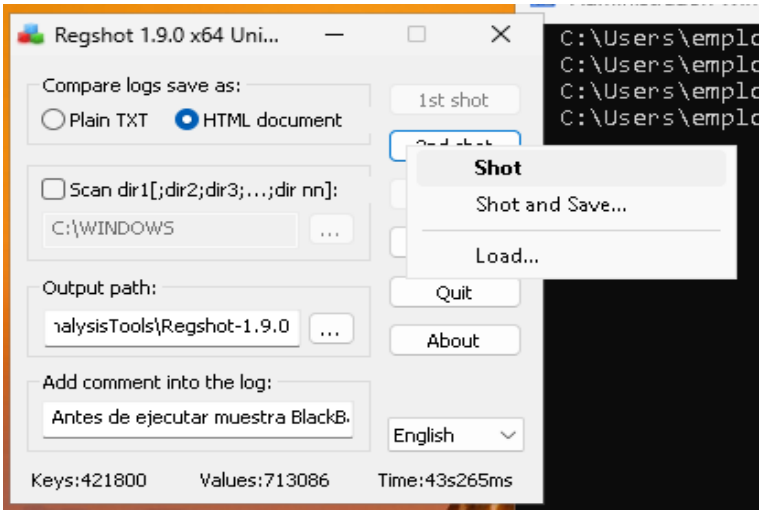
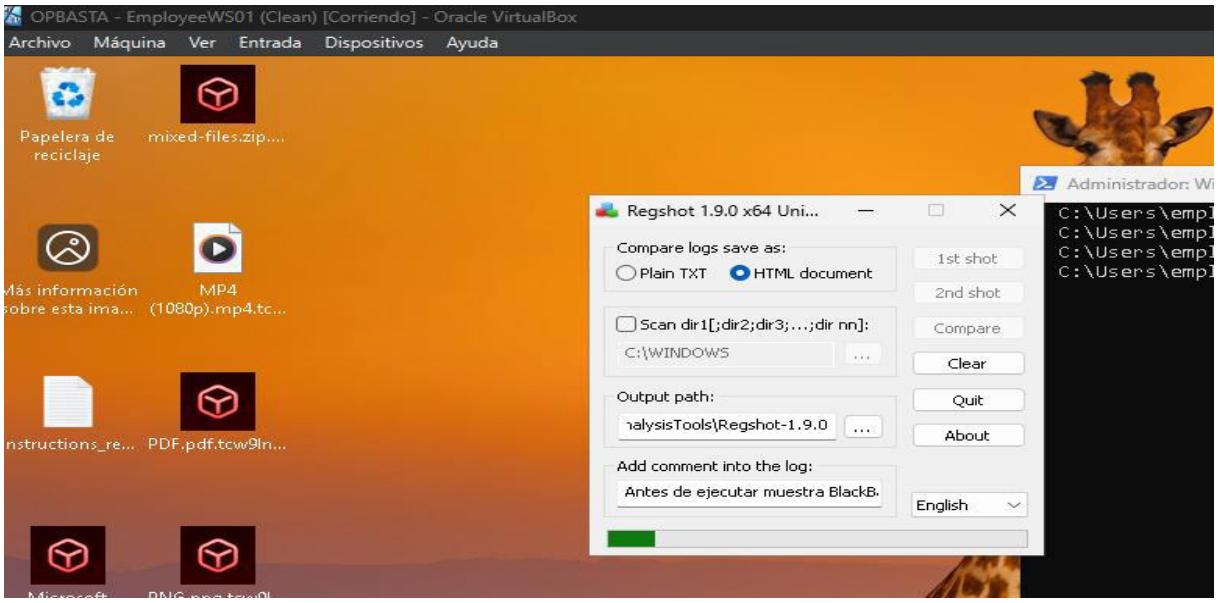


Figura 107

Proceso de comparación de primer y segundo shot obtenidos



En Procmon, se detuvo el monitoreo y se guardó toda la información recolectada en .pml

Figura 108

Registros de Procmon posterior a la ejecución del ransomware

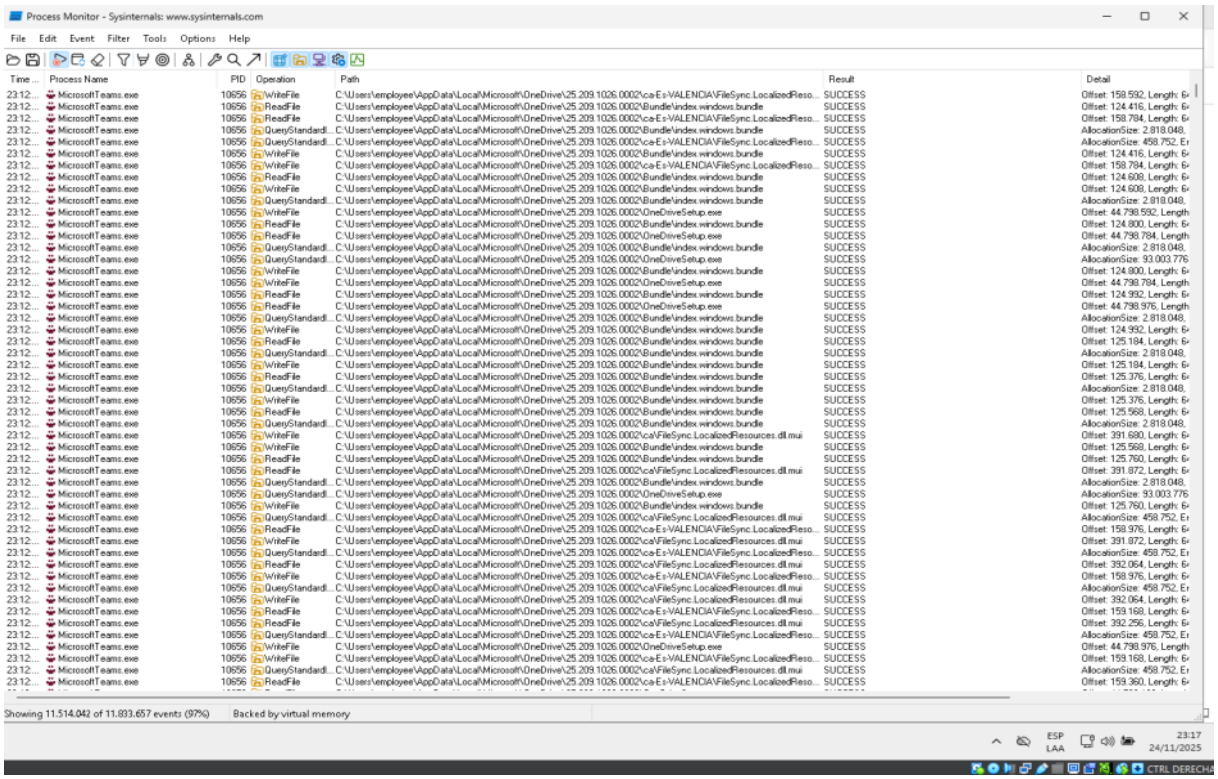
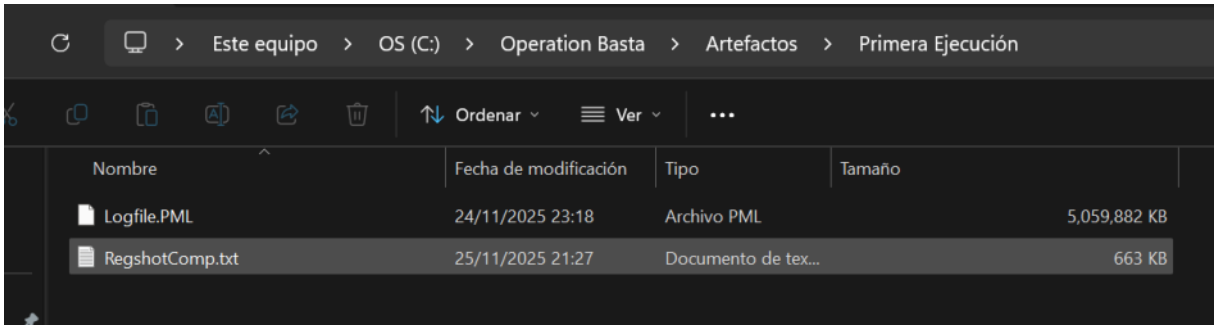


Figura 109

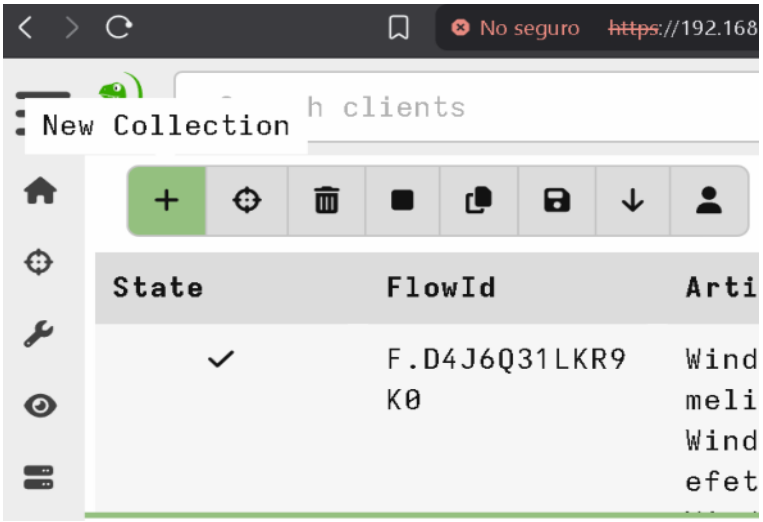
Archivos obtenidos de las herramientas Regshot y Procmon



A través de la GUI del servidor de Velociraptor, instalado en VM OPBASTA - ThreatAnalysis01, se ubicó al cliente y se dio clic en “Collected” y luego en “New Collection”, para empezar a recolectar toda información capturada por el agente de Velociraptor.

Figura 110

Selección del Collector en la interfaz gráfica de Velociraptor

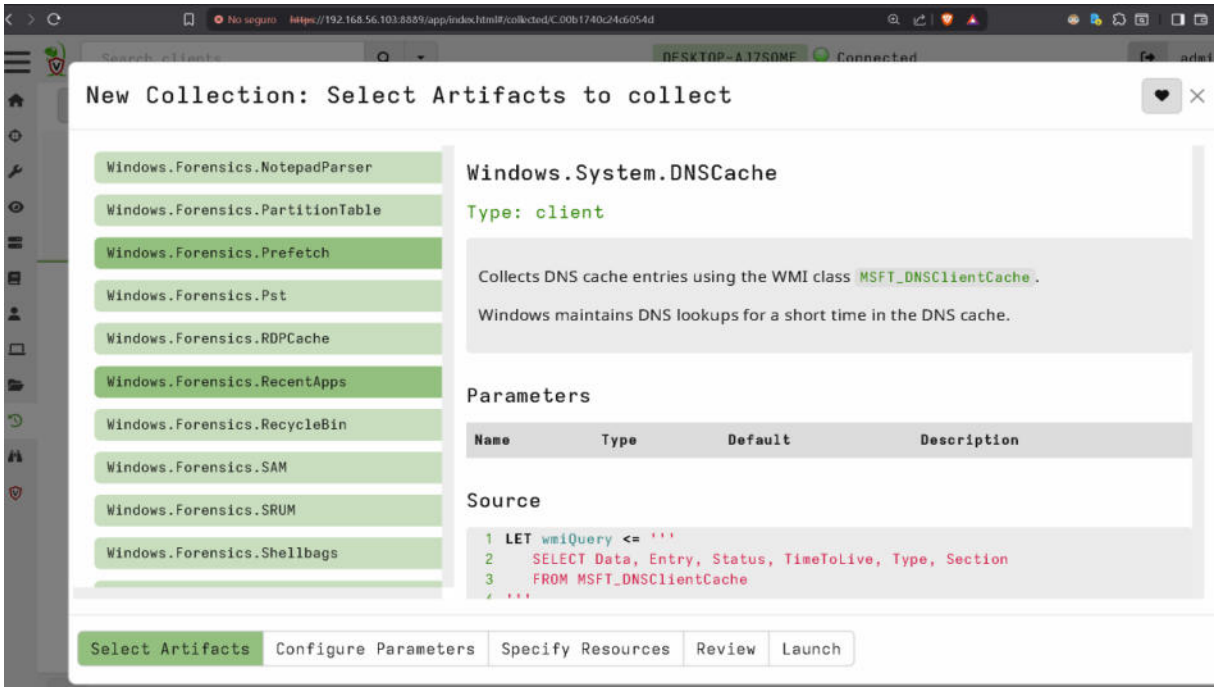


Se seleccionaron los siguientes artefactos a recolectar, los cuales sirven para analizar el comportamiento del Ransomware:

- Windows.Forensics.Timeline
- Windows.Forensics.Prefetch
- Windows.NTFS.MFT
- Windows.Forensics.Usn
- Windows.EventLogs.Evtx
- Windows.System.Pslist
- Windows.System.Services
- Windows.Forensics.RecentApps
- Windows.Attack.ParentProcess
- Windows.System.DNSCache

Figura 111

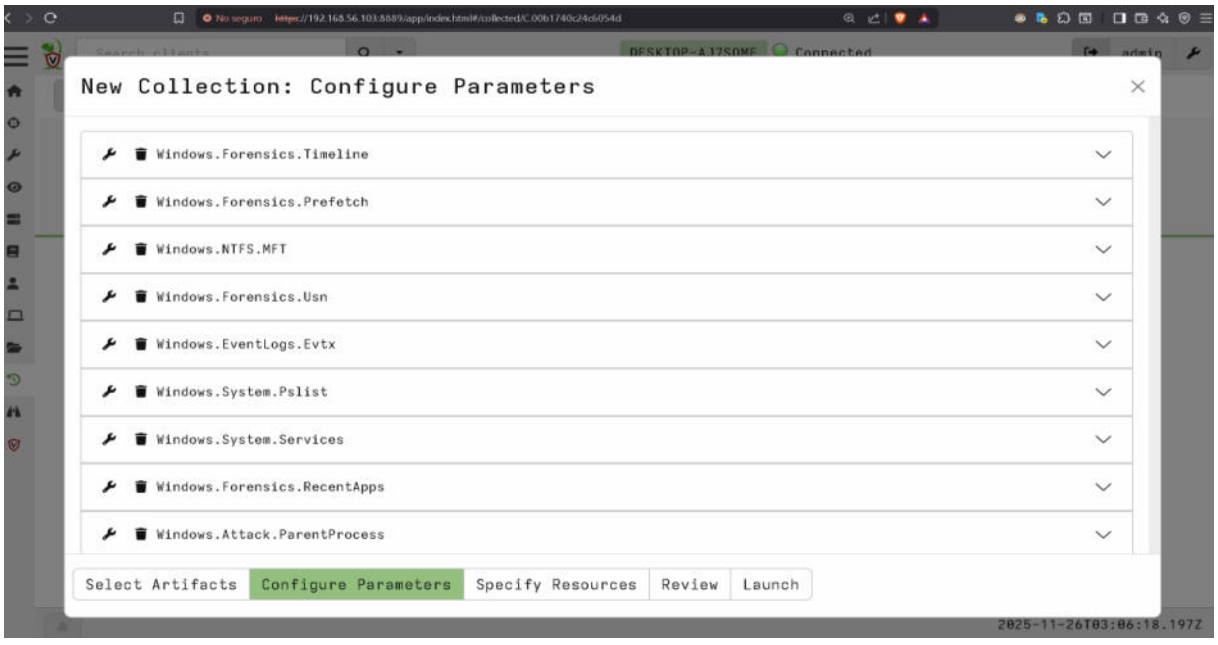
Selección de artefactos que serán recolectados



A continuación, en “Configure Parameters” se mantuvo la configuración por defecto.

Figura 112

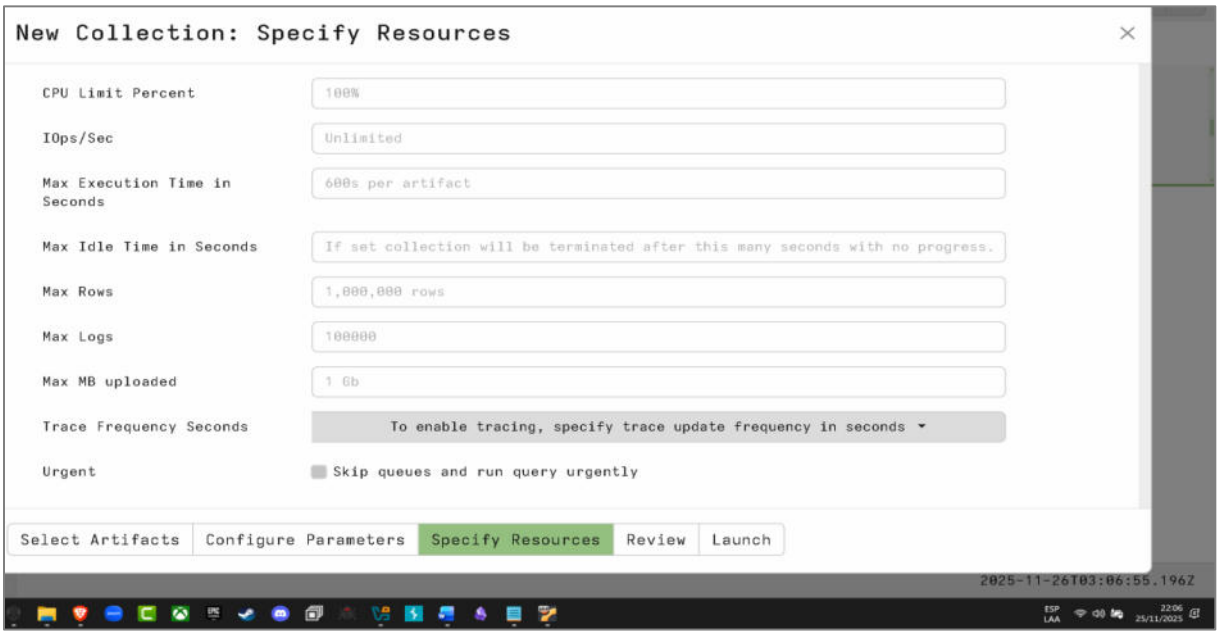
Configuración de los parámetros del colector



Luego, en la pestaña “Specify Resources”, también se mantuvo la configuración predeterminada.

Figura 113

Configuración de Specify Resources del colector



Finalmente, en “Review” se visualizó un resumen de todos los artefactos que se van a recolectar y se dio clic en “Launch” para iniciar la recolección de estos, se esperó unos minutos hasta que el proceso finalice.

Figura 114

Revisión de los artefactos capturados por el colector

Capítulo 4

4. Análisis de Resultados

En esta sección de análisis de resultados se integra y correlaciona los hallazgos encontrados durante el análisis estático, dinámico y de memoria de la muestra del ransomware Black Basta ejecutada en un entorno aislado. Con ayuda de herramientas como: Detect It Easy, Ghidra, Sysmon, Process Monitor y Volatility fue posible evaluar tanto el comportamiento real del ransomware durante la infección como su estructura interna.

Los resultados obtenidos permiten conocer las fases importantes del ataque: inicialización del proceso, eliminación de mecanismos de recuperación, despliegue de artefactos visuales, cifrado multihilo y creación repetitiva de notas de rescate. También, la exploración de artefactos persistentes y volátiles ratifica el uso de técnicas de alto nivel de evasión, eliminación de evidencias y manipulación del sistema.

Esta sección tiene como finalidad, comprender el funcionamiento del ransomware para establecer la base para su perfil mínimo de detección.

4.1. Pruebas de Concepto

4.1.1. Análisis Estático. A fin de entender el funcionamiento del ransomware Black Basta se ejecutó un análisis estático utilizando herramientas de ingeniería inversa como Detect It Easy y Ghidra. A continuación, se detalla lo encontrado:

4.1.1.1. Funcionamiento del Ransomware. La muestra analizada a continuación tiene por nombre su código hash SHA-256.

- **Muestra:**

17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe

- **Lenguaje:** C

- **Fecha de creación:** 2022/02/16
- **Tipo:** Consola
- **Arquitectura:** i386
- **Modo:** 32 bits
- **Entropía:** Alta (82%) posiblemente empaquetado

Figura 115

Análisis estático de la muestra de Black Basta con herramienta Detect It Easy (DIE)

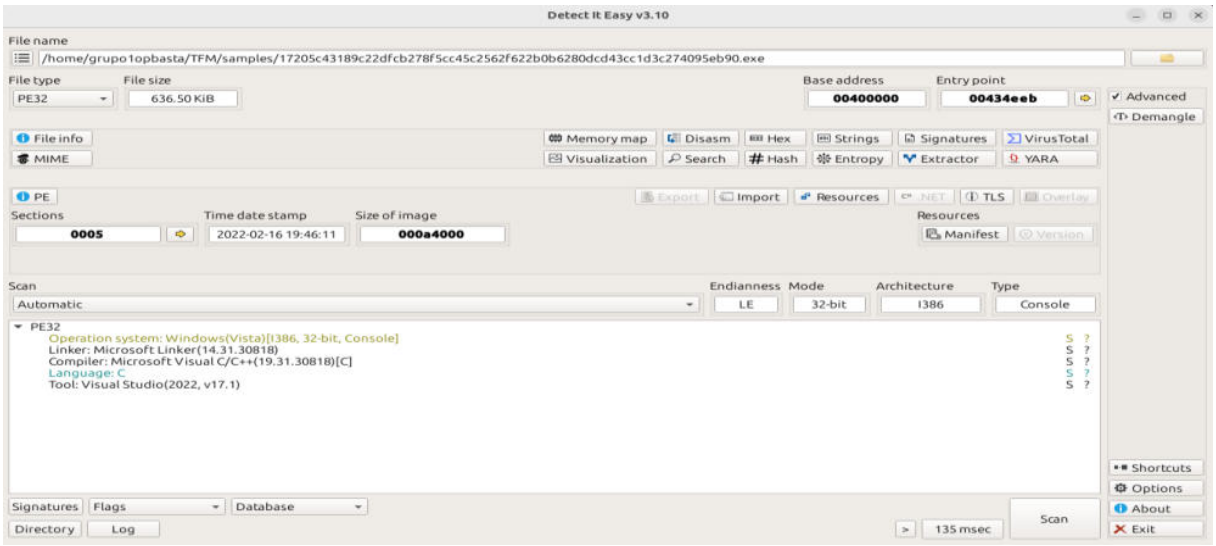


Figura 116

Revisión de la entropía de la muestra de Black Basta en DIE

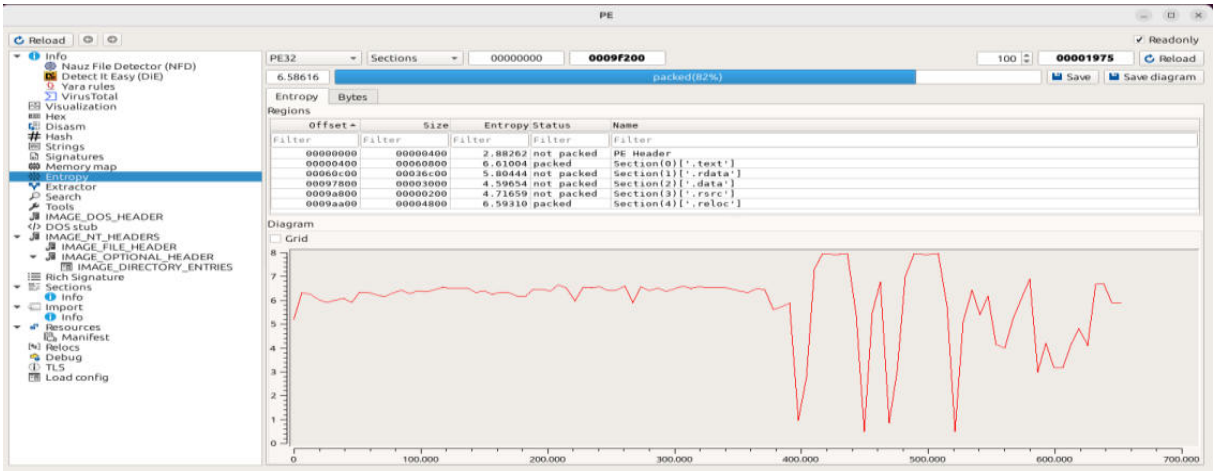
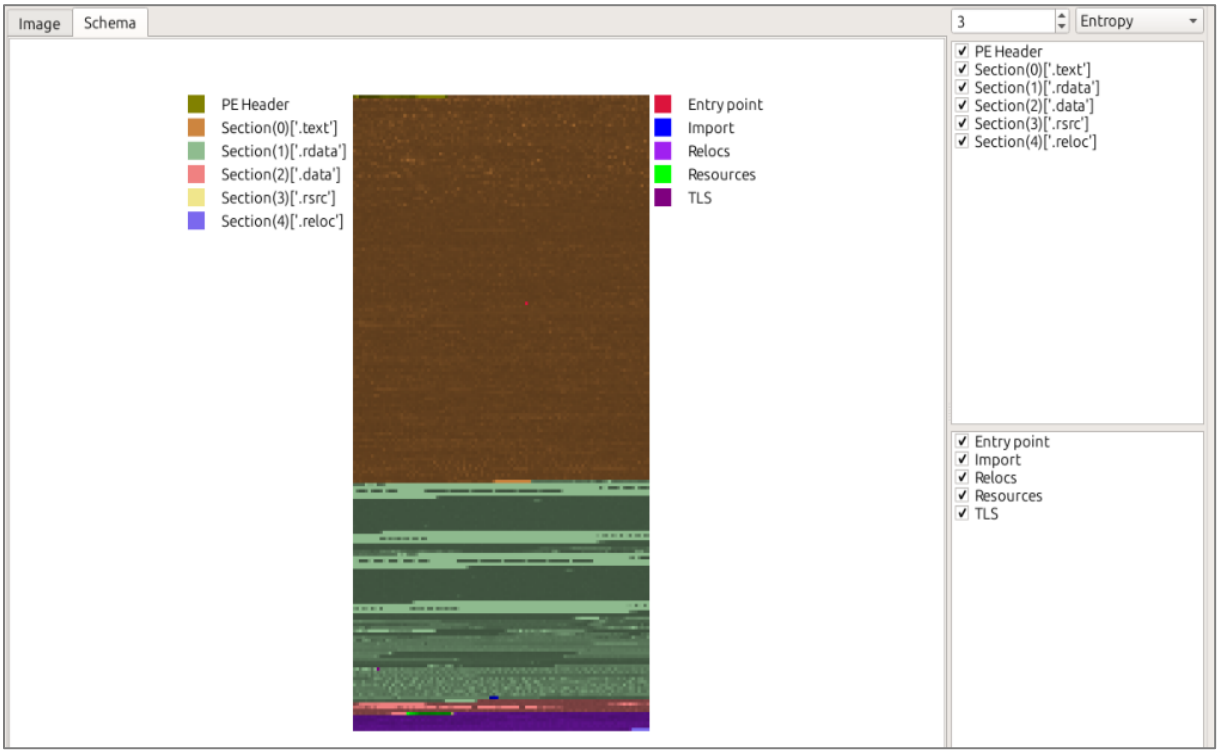


Figura 117

Revisión del porcentaje de empaquetado de la muestra de Black Basta en la herramienta DIE



Como resultado del análisis estático preliminar con Detect It Easy, se revela un binario PE32 compilado con MSVC, que presenta un nivel de entropía del 82%, lo cual evidencia un alto grado de empaquetamiento u ofuscación. Las secciones .text, .data y .rdata presentan patrones homogéneos que corresponden a payload cifrado típico de Black Basta.

El Entry Point apunta a un stub dentro de una región de alta entropía, lo que sugiere que la muestra implementa un mecanismo de self-unpacking en memoria antes de ejecutar la lógica de cifrado.

4.1.1.2. Revisión del Ransomware con la Herramienta Ghidra. Adicionalmente, se importó la muestra del ransomware en Ghidra y se ejecutó el análisis predeterminado.

Figura 118

Análisis de la muestra de Black Basta en la herramienta Ghidra

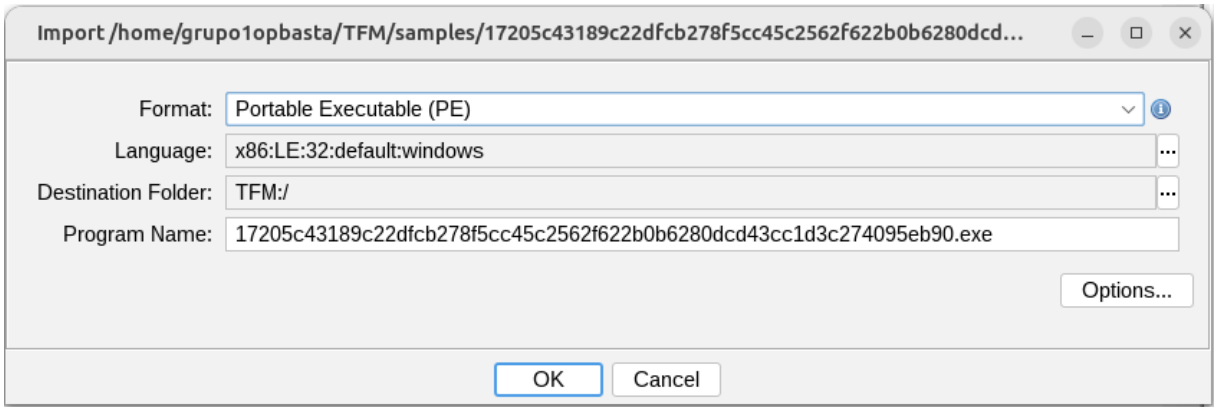


Figura 119

Mensaje de confirmación del análisis de la muestra de Black Basta en la herramienta Ghidra

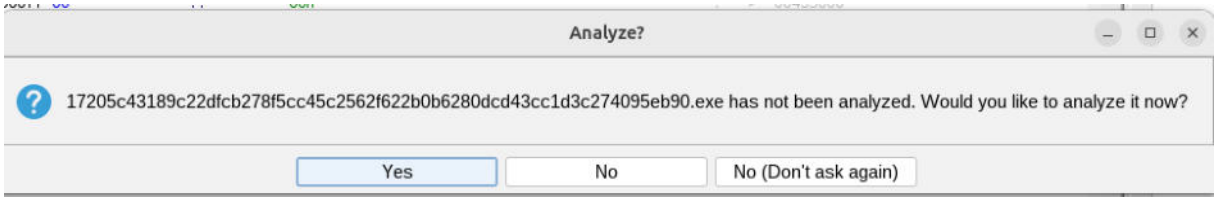
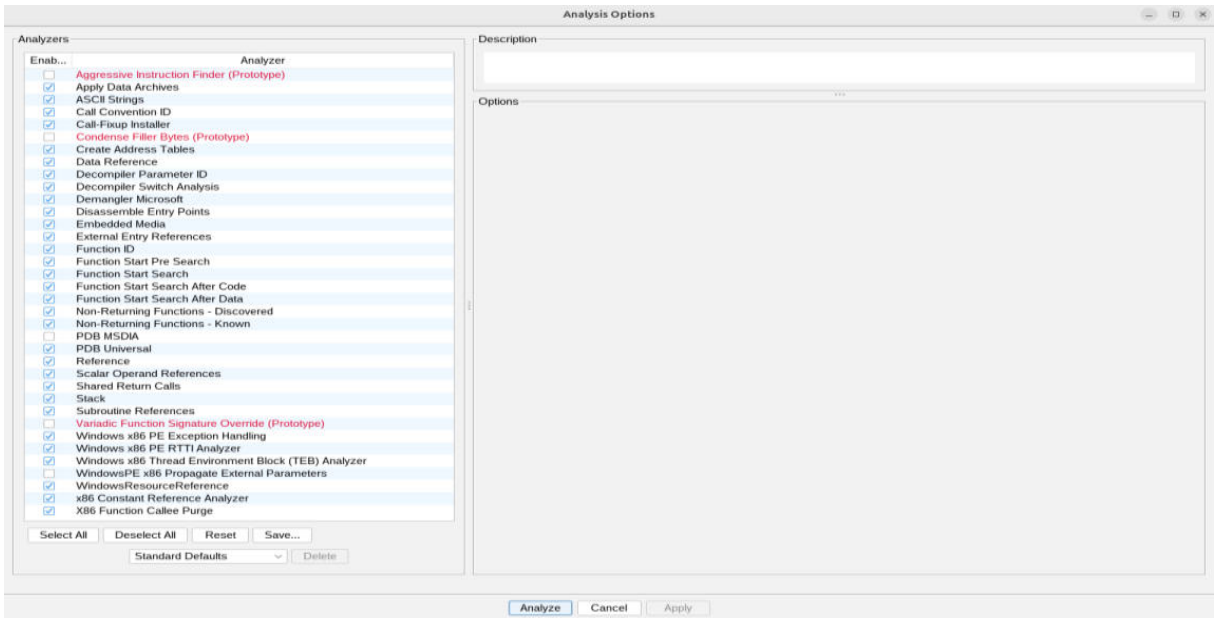


Figura 120

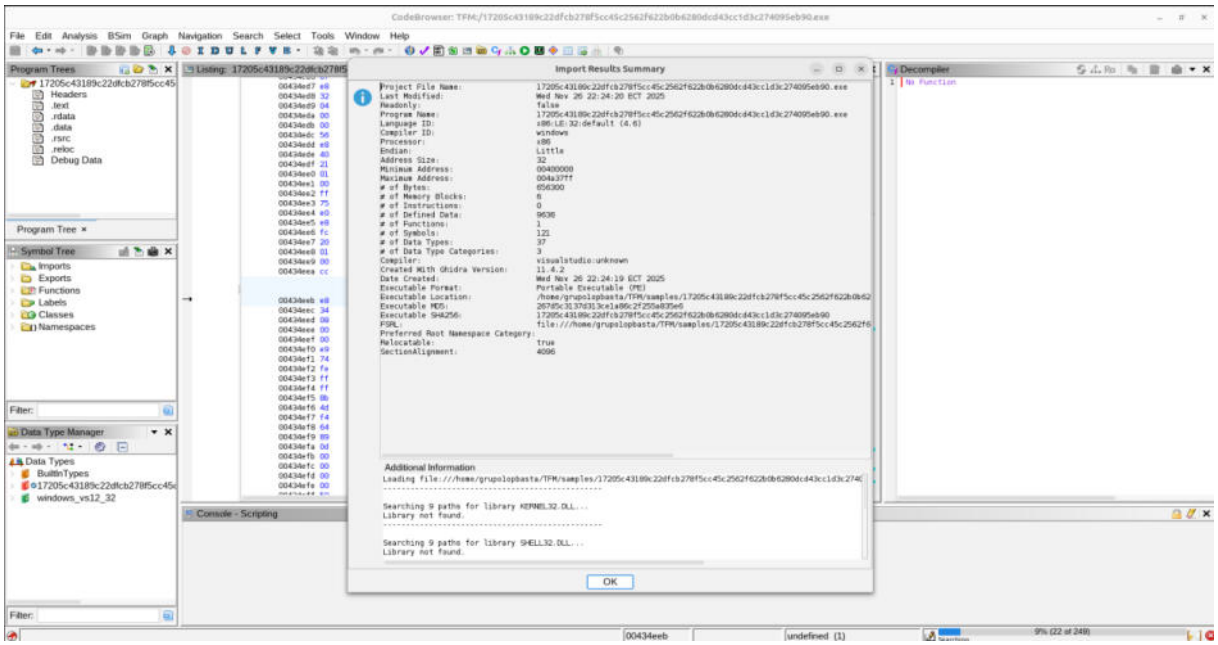
Inicio del análisis de la muestra de Black Basta en la herramienta Ghidra



Se esperó unos segundos a que Ghidra finalice el análisis; en primera instancia mostrará información adicional del binario para complementar lo encontrado con DIE.

Figura 121

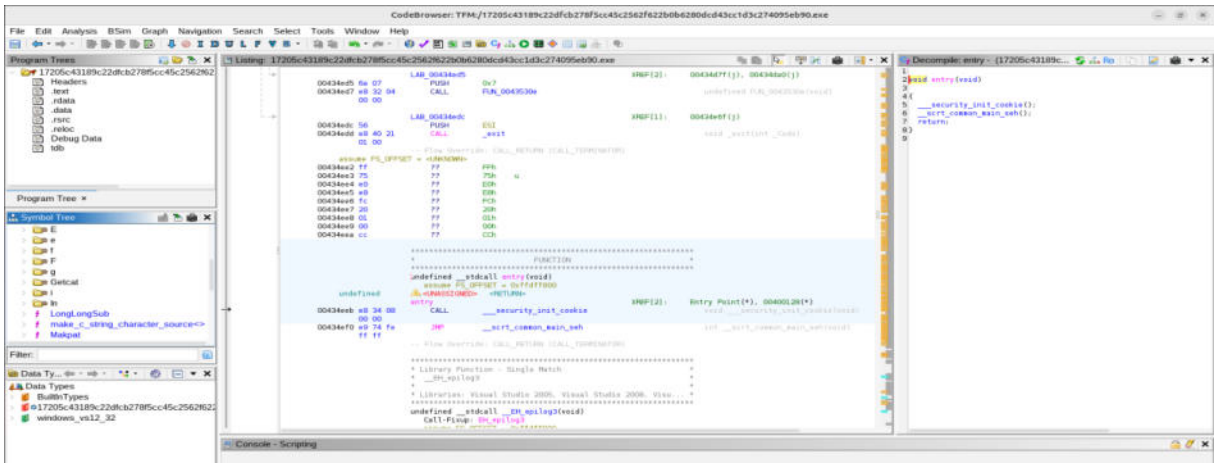
Finalización del análisis de la muestra de Black Basta en la herramienta Ghidra



Terminado el proceso, se procedió con el análisis a profundidad del código descompilado de cada una de las funciones que detectó Ghidra.

Figura 122

Código descompilado de Black Basta en la herramienta Ghidra



4.1.2. Análisis Dinámico. A fin de entender el funcionamiento del ransomware Black Basta observando su comportamiento en tiempo real, se ejecutó un análisis dinámico utilizando herramientas de análisis forense como Sysmon, Procmon, Process Hacker, entre otras, para recolección de artefactos útiles. A continuación, se detalla lo encontrado.

4.1.2.1. Comportamiento en Tiempo Real de Black Basta. Al ejecutar la muestra en el entorno de laboratorio, se corroboró el comportamiento previsto durante el análisis estático, y otros nuevos detalles observados:

- **Encriptado de información de forma rápida:** el malware hace uso de múltiples hilos para realizar varias tareas al mismo tiempo, de manera que todos los archivos se observaron totalmente encriptados en un lapso de aproximadamente 2 minutos.

Figura 123

Estado inicial de los archivos antes de la ejecución de Black Basta

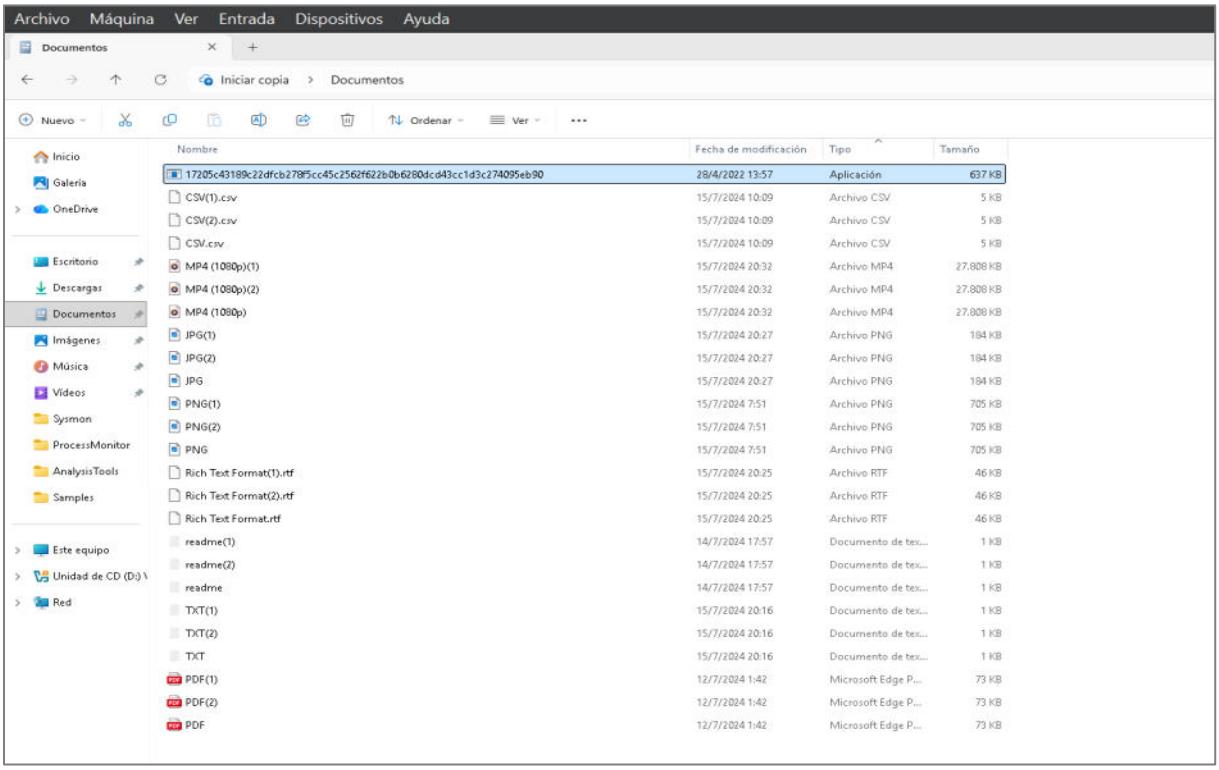


Figura 124

Revisión de archivo readme.txt generado por Black Basta

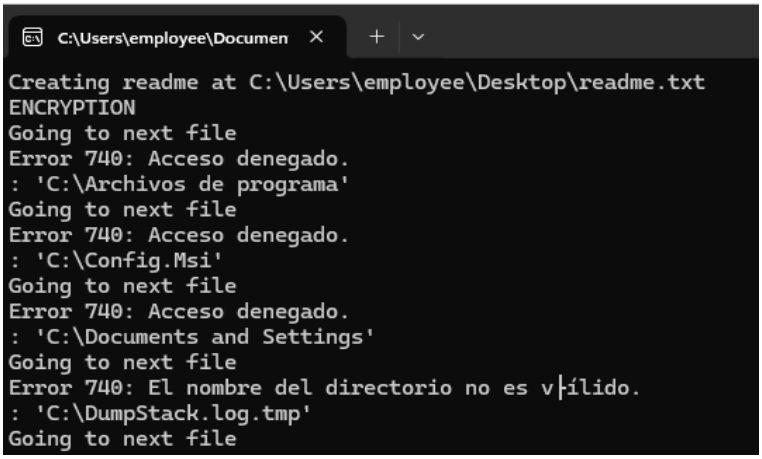


Figura 125

Encriptación de los archivos realizada por Black Basta

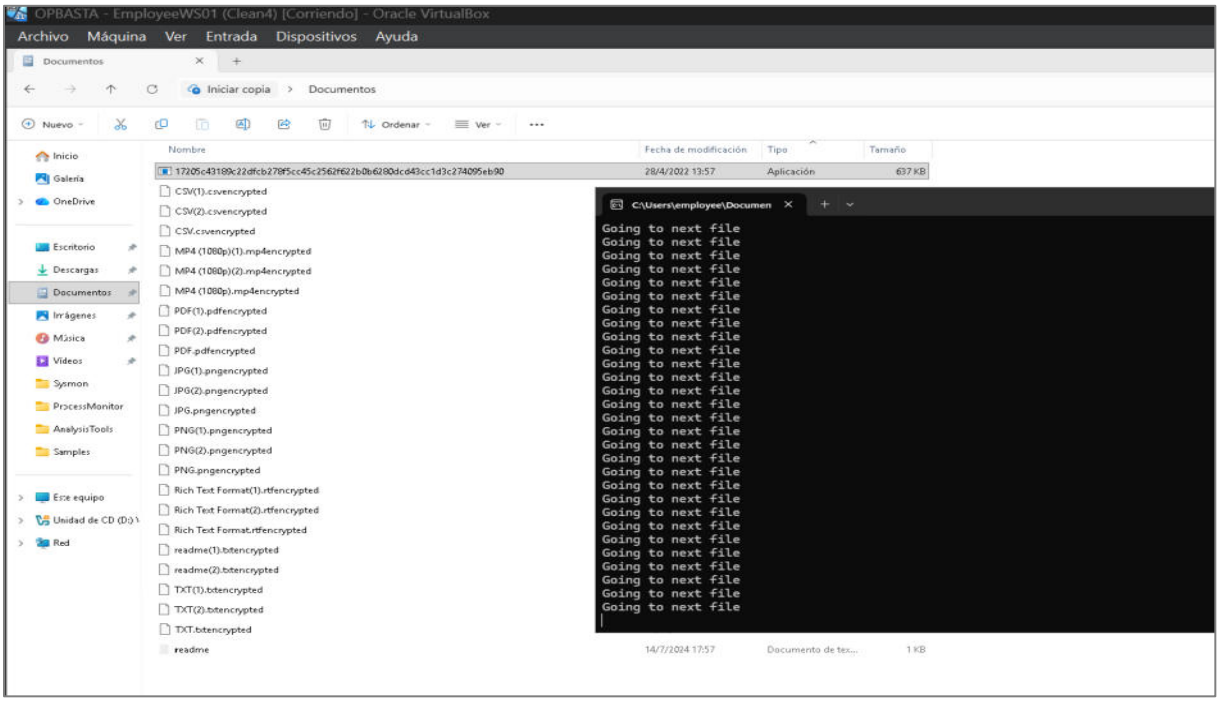
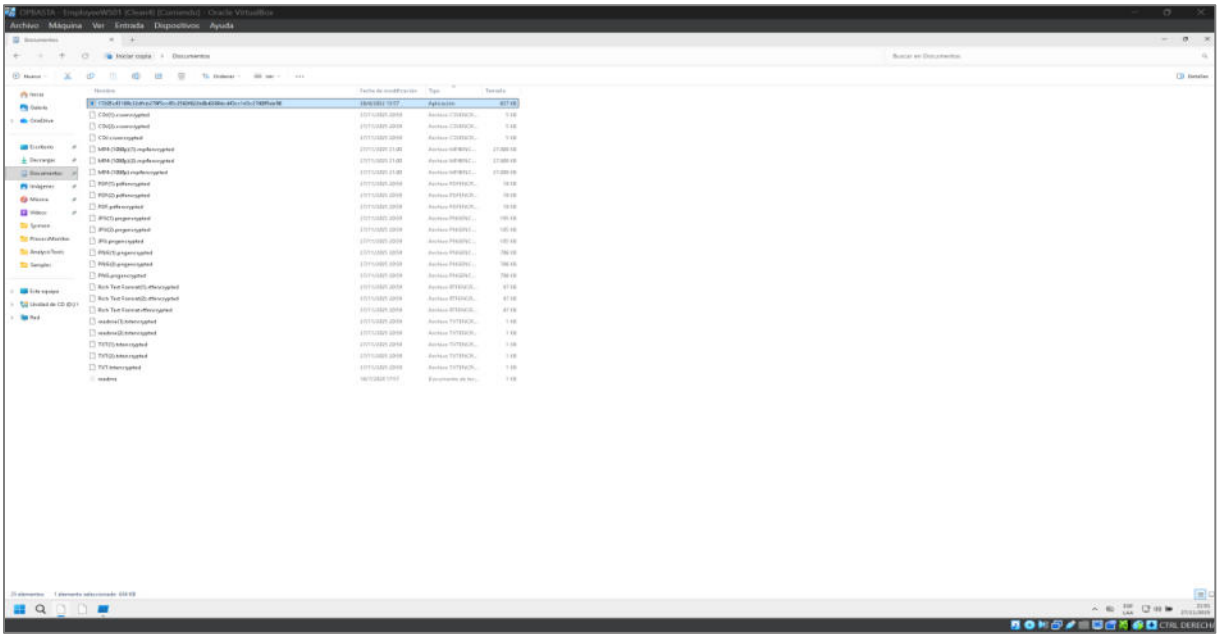


Figura 126

Archivos encriptados al finalizar la ejecución de Black Basta



- **Renombramiento de archivos encriptados:** Se evidenció cómo todos los archivos encriptados fueron reescritos y renombrados de acuerdo con la nomenclatura "encrypted" al final de cada nombre.

Figura 127

Archivos renombrados tras la encriptación de Black Basta

- **Modificación de apariencia del escritorio:** Adicionalmente, se verificó que el wallpaper del escritorio fue modificado asignando un fondo totalmente negro, este comportamiento fue previsto al analizar el código descompilado del ransomware, allí se observó cómo se hacía manipulación del wallpaper.

Figura 128

Cambio de fondo de pantalla registrado tras la ejecución de Black Basta

- **Creación de archivo readme.txt con contenido:** Se observó el archivo

"readme.txt" en el escritorio, el cual contiene exactamente el mismo mensaje que se leyó en una parte de las funciones del código que proporcionó Ghidra.

Figura 129

Creación de archivo readme.txt en el escritorio

Figura 130

Revisión del archivo readme.txt generado por Black Basta

- **Directorios excluidos de la encriptación:** En el análisis estático, se observó una sección del código en el cual se filtraban ciertos directorios para que no se aplique el cifrado. En el análisis dinámico, se corroboró cómo ningún archivo de los directorios “boot”, “Windows”, “RecycleBin” o el “readme.txt” fueron encriptados.

Figura 131

Revisión de los archivos que se encuentran en la papelería de reciclaje

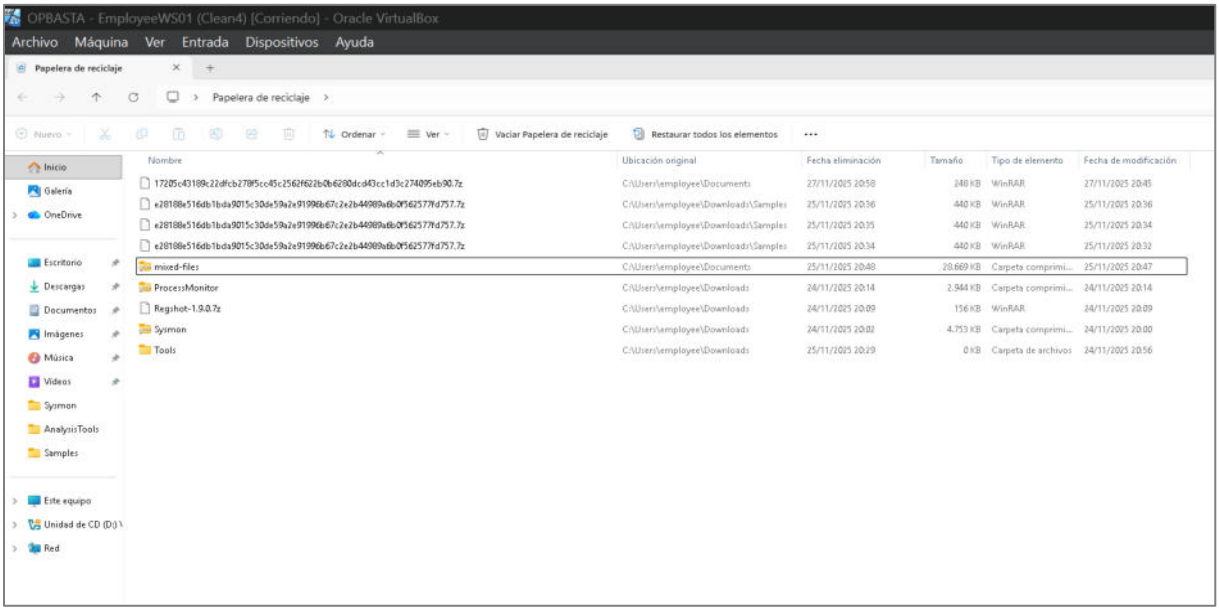


Figura 132

Revisión del estado de los archivos que se encuentran en las carpetas de Windows

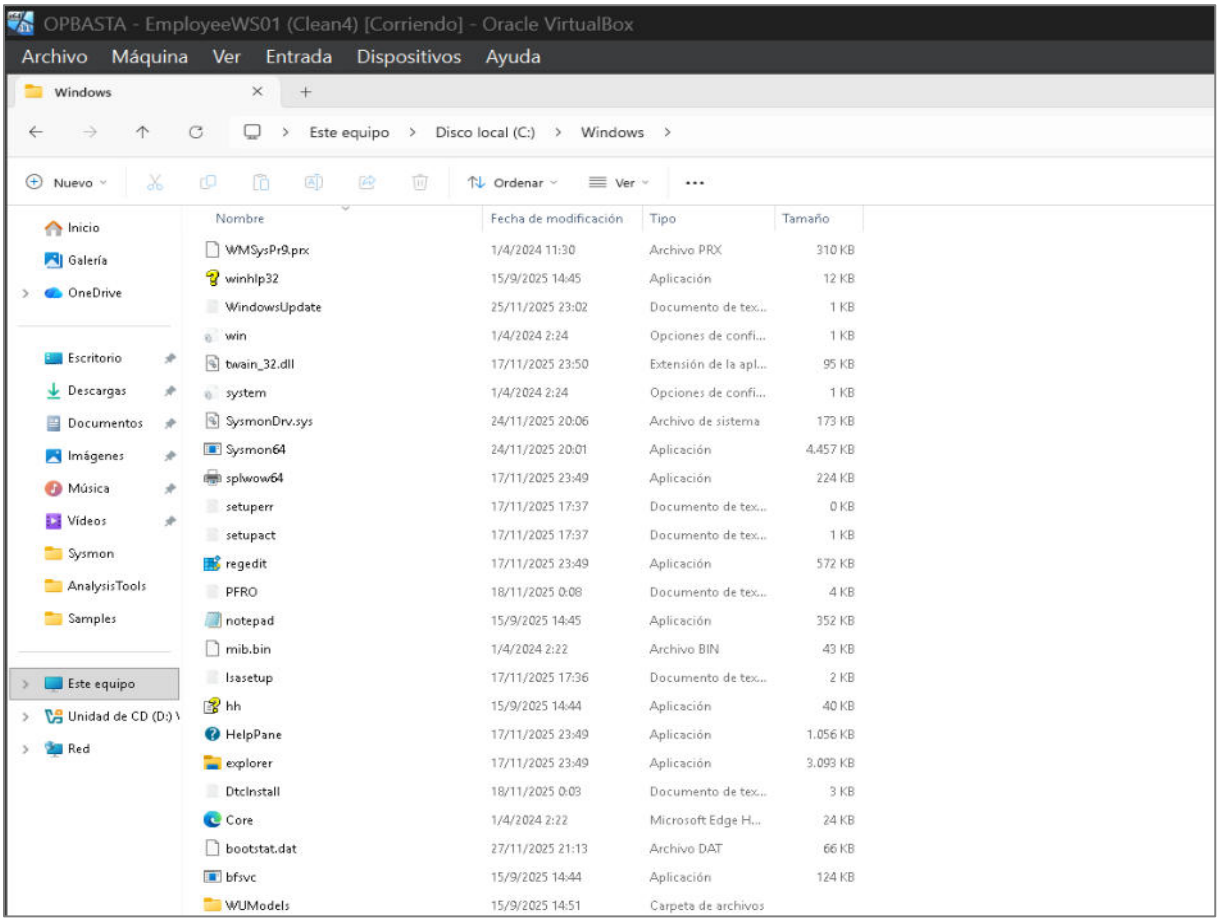
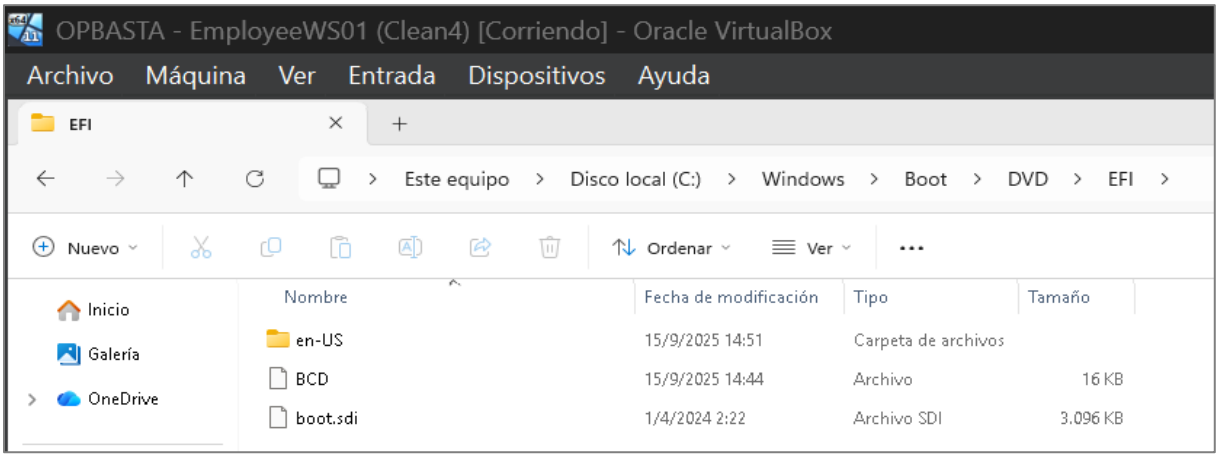


Figura 133

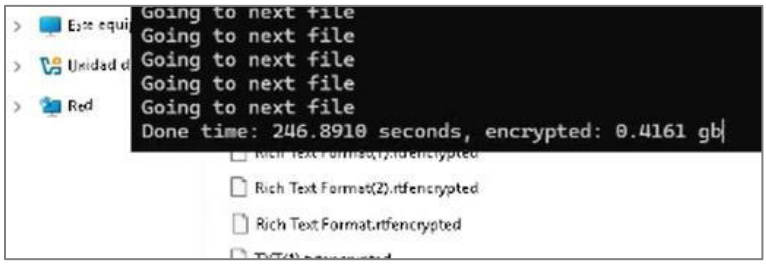
Revisión de los archivos que se encuentran en las subcarpetas de Windows



- **Información de tiempo de duración y tamaño de encriptado:** Al finalizar el encriptado, también se presentó en consola el mensaje con la cantidad de información en Gb que se cifró y el tiempo en el que lo hizo, lo cual concuerda con la función mostrada en el análisis estático en la que se hacía referencia a la misma cadena.

Figura 134

Mensaje de finalización de la ejecución de Black Basta con la cantidad de encriptados

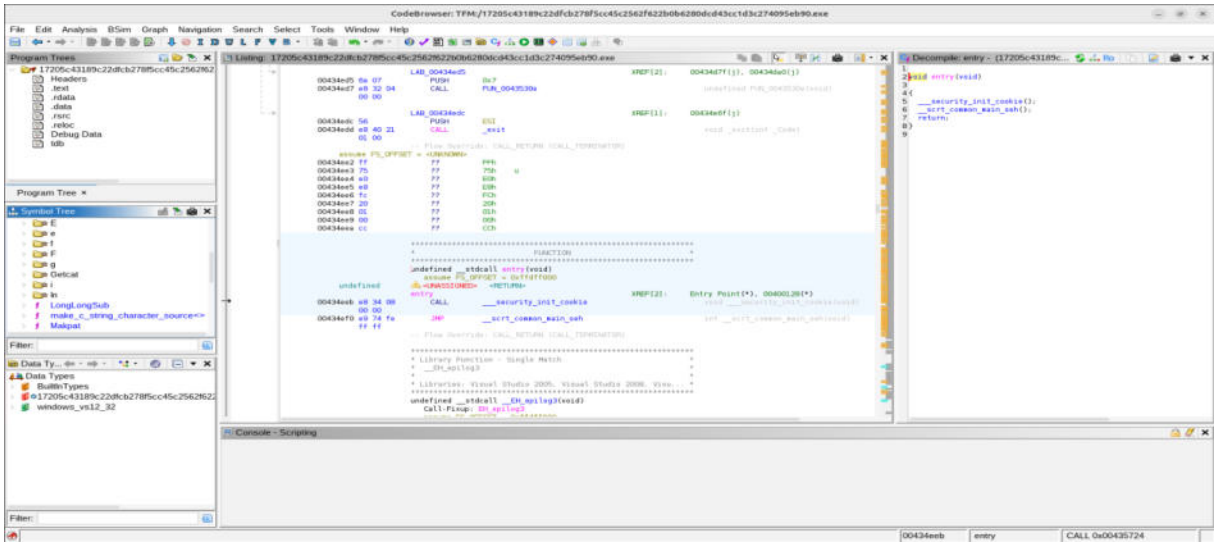


4.2. Interpretación de Resultados

4.2.1. Análisis Estático. La herramienta Ghidra se usó para realizar un análisis a profundidad de la muestra del ransomware Black Basta. A continuación, se redactarán los puntos clave del funcionamiento de Black Basta de acuerdo con el análisis del código descompilado.

Figura 135

Análisis del código de la muestra de Black Basta obtenidas de Ghidra



4.2.1.1. Arquitectura General de Ejecución. El ransomware sigue una arquitectura modular cuyas fases principales, reconstruidas a partir del código descompilado, son:

- Inicialización criptográfica y preparación de claves.
- Enumeración completa del sistema de archivos por volúmenes.
- Creación de múltiples hilos de trabajo para paralelizar el cifrado.
- Procesamiento por cada hilo: renombrado de archivo, generación de claves aleatorias por archivo, cifrado del material criptográfico con una clave pública interna (RSA-like), escritura de metadatos y blob cifrado en el archivo, cifrado del contenido.
- Alteraciones del entorno gráfico del usuario (Wallpaper).
- Finalización, limpieza de memoria y terminación controlada.

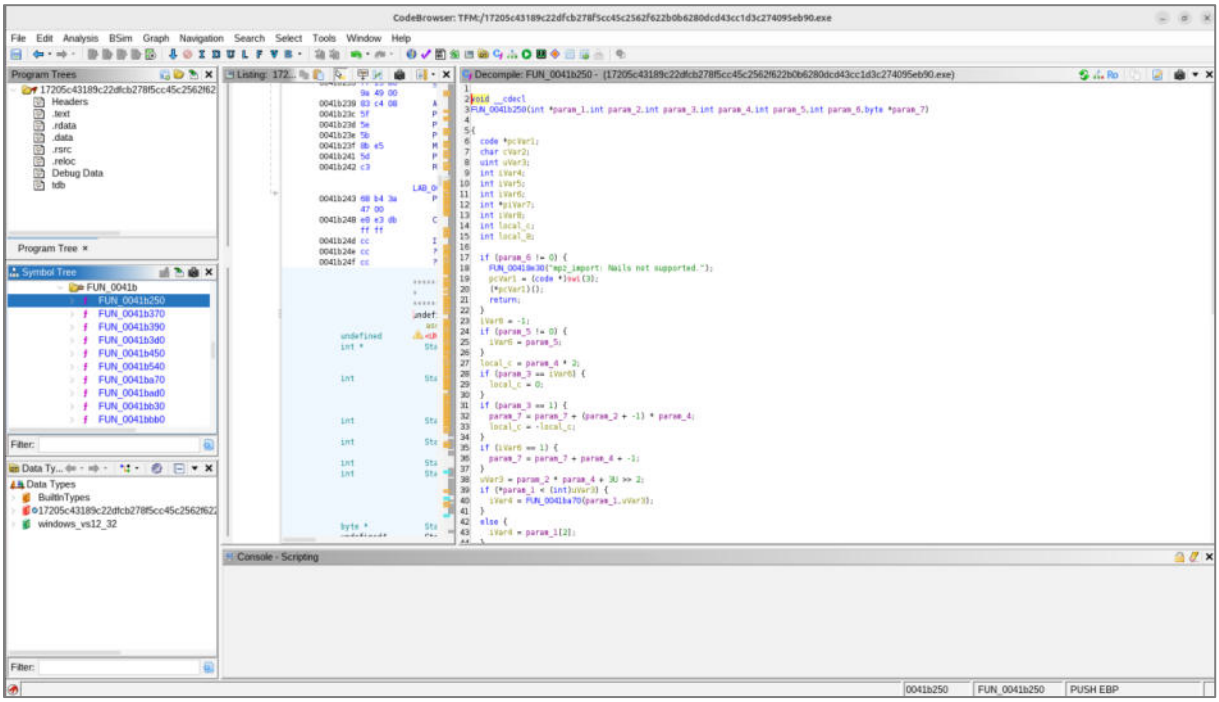
Este pipeline confirma que el atacante diseñó la muestra para ser lo más rápida posible, con mínima dependencia del entorno, y con técnicas que dificultan la recuperación de información.

4.2.1.2. Implementación Criptográfica con Enteros Grandes y RSA-Like. Uno de los hallazgos más importantes del análisis estático es la presencia de funciones conceptualmente idénticas a las primitivas de la biblioteca GMP, pero implementadas dentro del binario, estas son:

- **FUN_0041b250 > mpz_import.-** Convierte un buffer arbitrario de bytes en un entero multiprecisión (“limb array” de 32 bits). El código gestiona el orden de bytes, longitud variable, eliminación de ceros significativos, endianess, bloques de 4 bytes. Es una importación típica de RSA o de estructuras de clave pública.

Figura 136

Revisión de código en Decompile: FUN_0041b250 en Ghidra



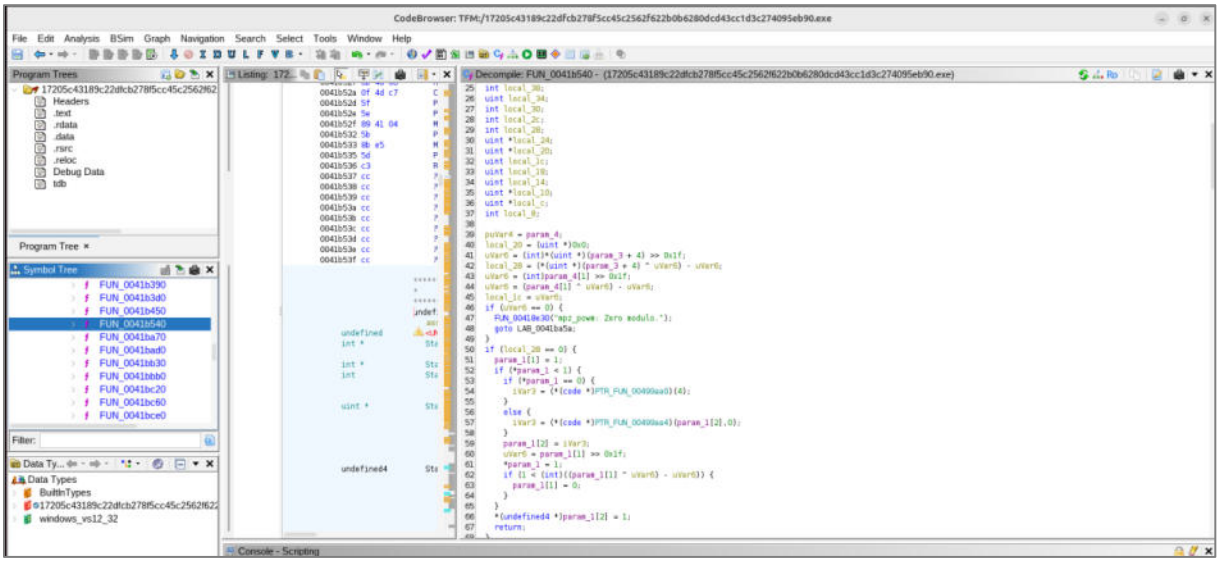
- **FUN_0041b540 > mpz_powm.-** Implementa exponenciación modular:

res = base^{exponente} mod modulo

El código incluye: manejo de exponentes negativos, verificación de módulo no nulo, algoritmo “square & multiply” sobre los bits del exponent, lógica para evitar overflows, control de signo basado en aritmética de 32 bits y normalización de límites intermedios. Los mensajes de error incrustados (“mpz_powm: Zero modulo”, “Negative exponent”) confirman la naturaleza de la función.

Figura 137

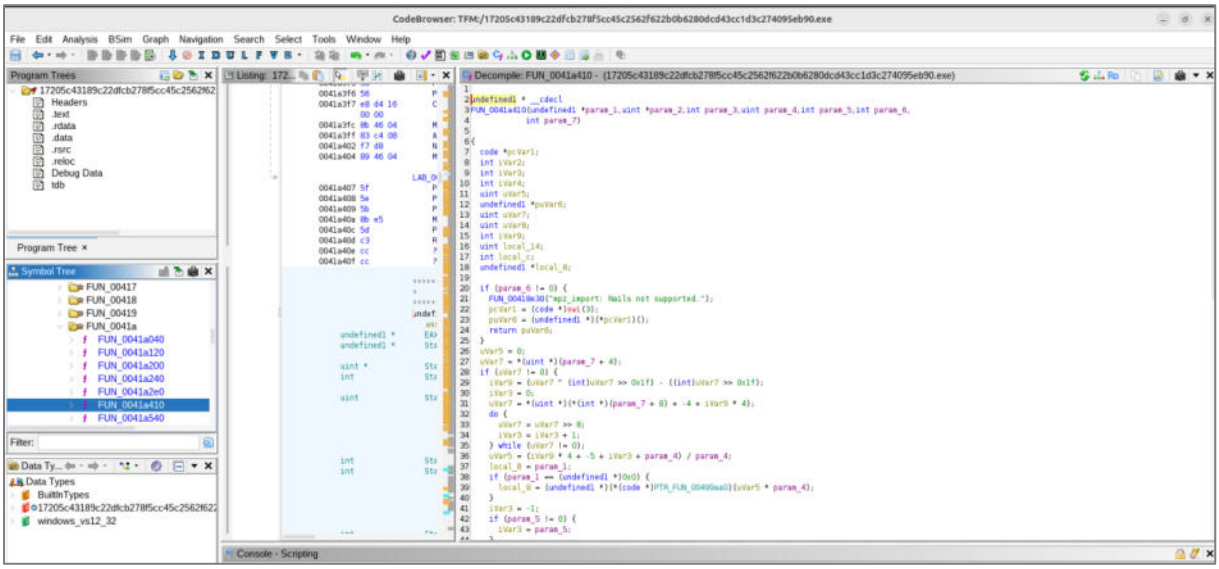
Revisión de código en Decompile: FUN_0041b540 en Ghidra



- **FUN_0041a410 > mpz_export.-** Este segmento de código extrae un entero multi-precisión a un buffer plano, devolviendo la cantidad de bytes generados y la secuencia representada en big- o little-endian según parámetros.

Figura 138

Revisión de código en Decompile: FUN_0041a410 en Ghidra



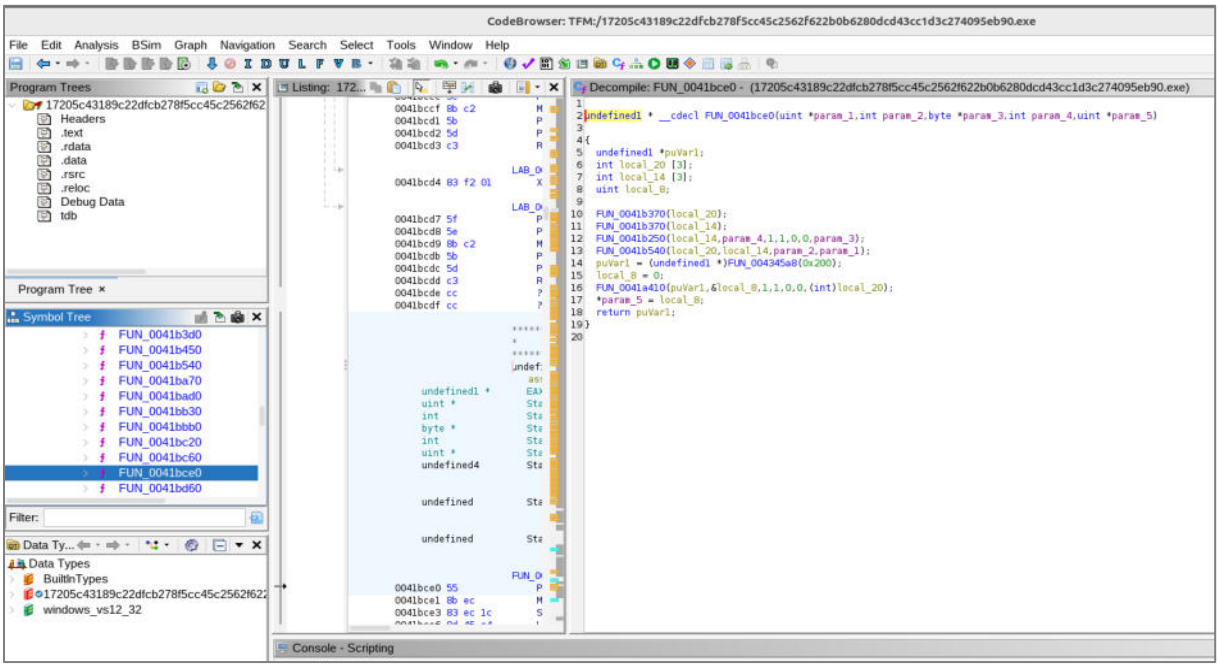
4.2.1.3. Integración Criptográfica y Cifrado Híbrido. El ransomware integra estas funciones a través de **FUN_0041bce0** y de otras funciones de nivel superior para implementar un esquema híbrido:

- Genera material aleatorio por archivo (ver X.5).
- Lo importa como entero grande.
- Lo eleva a un exponente grande dentro del módulo (típicamente RSA).
- Exporta el resultado como blob segur.
- Escribe ese blob en el archivo cifrado (cabecera/metadatos).

Este es exactamente el comportamiento de cifrado de una clave simétrica con una clave pública RSA. Es decir, la clave simétrica cifra los datos; la clave pública cifra la clave simétrica. Como resultado de esta combinación, si el atacante no libera la clave privada, no hay manera práctica de revertir el cifrado.

Figura 139

Revisión de código en Decompile: FUN_0041bce0 en Ghidra



4.2.1.4. Inicialización de Parámetros Criptográficos y Claves Internas. La función **FUN_0040d160** inicializa el componente criptográfico del ransomware:

- Desempaqueta cadenas embebidas codificadas, que contienen: módulo RSA (n), exponente público (e) y posibles parámetros extras.
- Utiliza las funciones mpz_import para cargarlas en memoria como enteros grandes.
- Prepara un contexto criptográfico global utilizado por todos los hilos.

Todas estas observaciones confirman que la clave del atacante está incorporada dentro del binario, ofuscada, pero completamente funcional durante la ejecución.

Figura 140

Revisión de código en Decompile: FUN_0040d160 en Ghidra

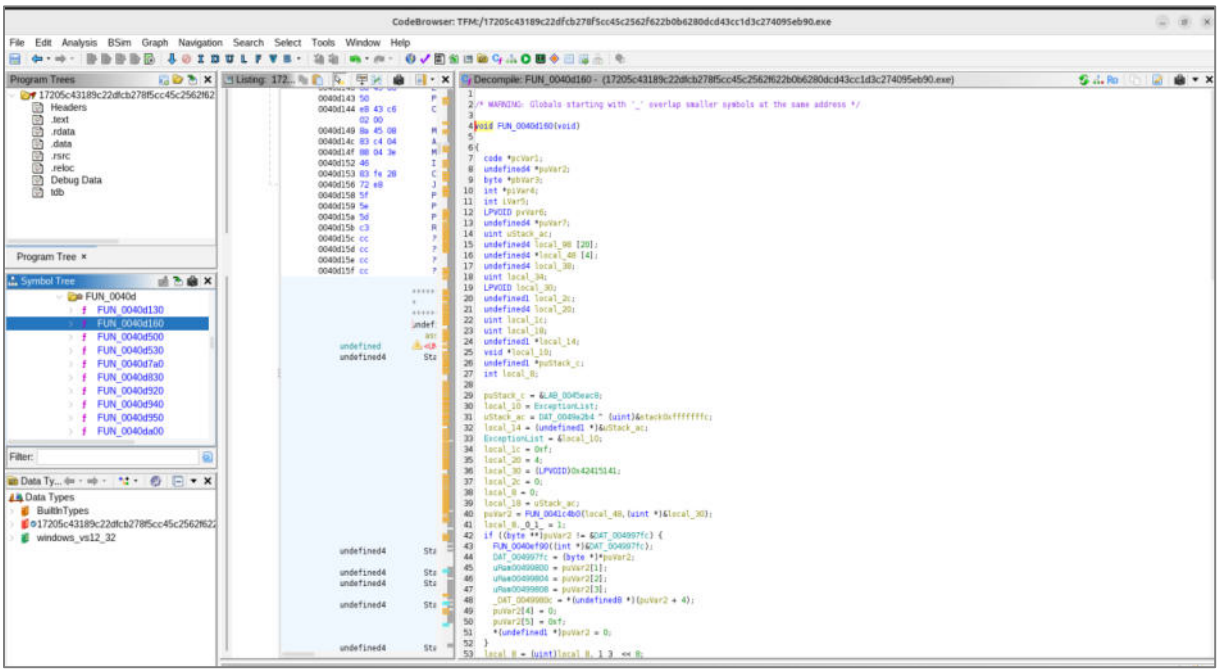


Figura 141

Análisis de código en Decompile: FUN_0040d160 en Ghidra

4.2.1.5. Enumeración de Volúmenes y Rutas en Windows. El ransomware implementa un sistema de descubrimiento completo del sistema de archivos usando APIs estándar:

- FindFirstVolumeW, FindNextVolumeW
- GetVolumePathNamesForVolumeNameW
- GetVolumeInformationW

La función **FUN_0040ca70** se encarga de enumerar todos los volúmenes (C:\, D:\, USBs, particiones ocultas, etc.), obtener rutas de montaje, filtrar volúmenes con atributos no compatibles y construir una lista dinámica de rutas raíz válidas. Esto le permite al ransomware abarcar todo el sistema y adaptarse a distintos entornos.

Figura 142

Revisión de código en Decompile: FUN_0040ca70 en Ghidra (parte 1)

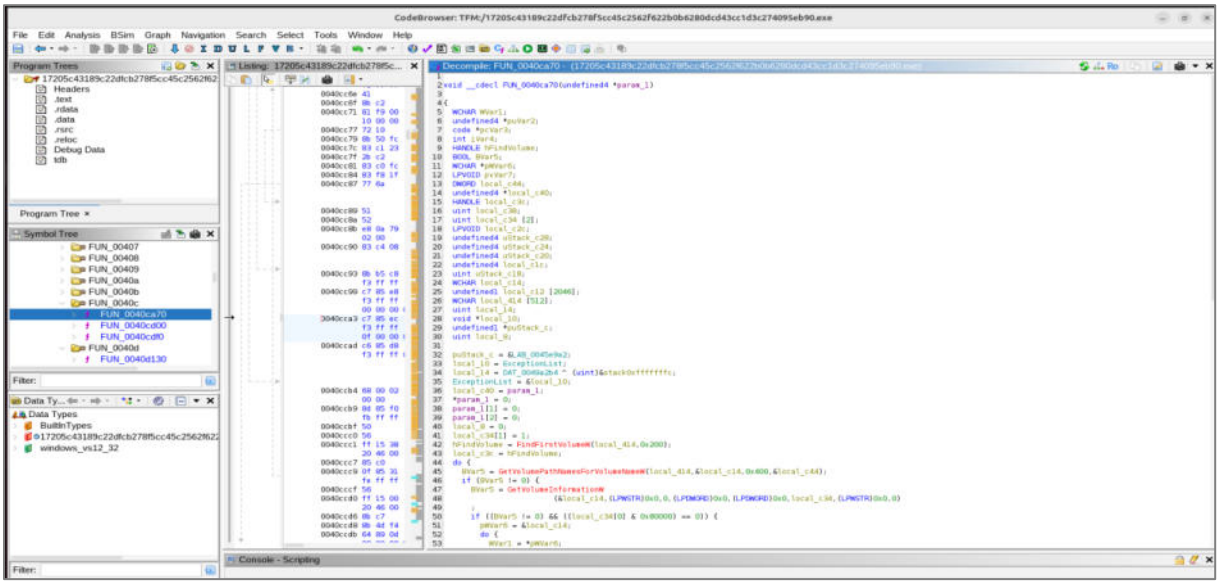


Figura 143

Revisión de código en Decompile: FUN_0040ca70 en Ghidra (parte 2)

```
42  hFindVolume = FindFirstVolumeW(local_414,0x200);
43  local_c3c = hFindVolume;
44  do {
45    BVar5 = GetVolumePathNamesForVolumeNameW(local_414,&local_c14,0x400,&local_c44);
46    if (BVar5 != 0) {
47      BVar5 = GetVolumeInformationW
48                        (&local_c14,(LPWSTR)0x0,0,(LPDWORD)0x0,(LPDWORD)0x0,local_c34,(LPWSTR)0x0,0)
49      ;
50      if ((BVar5 != 0) && ((local_c34[0] & 0x80000) == 0)) {
51        pWVar6 = &local_c14;
52        do {
53          WVar1 = *pWVar6;
54          pWVar6 = pWVar6 + 1;
55        } while (WVar1 != L'\0');
```

Figura 144

Revisión de código en Decompile: FUN_0040ca70 en Ghidra (parte 3)

```
83          local_8 = local_8 & 0xffffff00;
84          if (0xf < uStack_c18) {
85            pvVar7 = local_c2c;
86            if (0xfff < uStack_c18 + 1) {
87              pvVar7 = *(LPVOID *)((int)local_c2c + -4);
88              if (0x1f < (uint)((int)local_c2c + (-4 - (int)pvVar7))) {
89                FUN_004395bf();
90                pcVar3 = (code *)swi(3);
91                (*pcVar3)();
92                return;
93              }
94            }
95            FUN_0043459a(pvVar7);
96          }
97          local_c1c = 0;
98          uStack_c18 = 0xf;
99          local_c2c = (LPVOID)((uint)local_c2c & 0xffffff00);
100         hFindVolume = local_c3c;
101       }
102     }
103     BVar5 = FindNextVolumeW(hFindVolume,local_414,0x200);
104   } while (BVar5 != 0);
105   FindVolumeClose(hFindVolume);
106   ExceptionList = local_10;
107   FUN_00434a10(local_14 ^ (uint)&stack0xfffffffc);
108   return;
109 }
110
```

4.2.1.6. Planificación Multihilo y Cola de Trabajo. La función **FUN_0040b840** coordina el cifrado paralelo realizando las siguientes tareas:

- Calcula la cantidad de hilos óptima según el entorno (FUN_0041c859).
- Crea una lista de hilos que ejecutan FUN_0040bff0.
- Mantiene una cola de trabajo global con rutas pendientes.
- Utiliza mutexes para evitar condiciones de carrera.

- El proceso principal genera tareas nuevas (navegación de carpetas) mientras los hilos cifran.
- Filtra directorios como “Boot”, “RecycleBin”, “Windows” e incluso el mismo “readme.txt”, para evitar que el sistema se rompa o el archivo de texto con las instrucciones de la extorsión se corrompa con los demás archivos del sistema.

Este patrón descrito provoca uso masivo y simultáneo de CPU, fuerte presión sobre el disco, gran rapidez en el cifrado y evita dañar por completo el sistema. Esto coincide con el comportamiento observado de ransomware avanzado como Black Basta y LockBit.

Figura 145

Revisión de código en Decompile: FUN_0040b840 en Ghidra (parte 1)

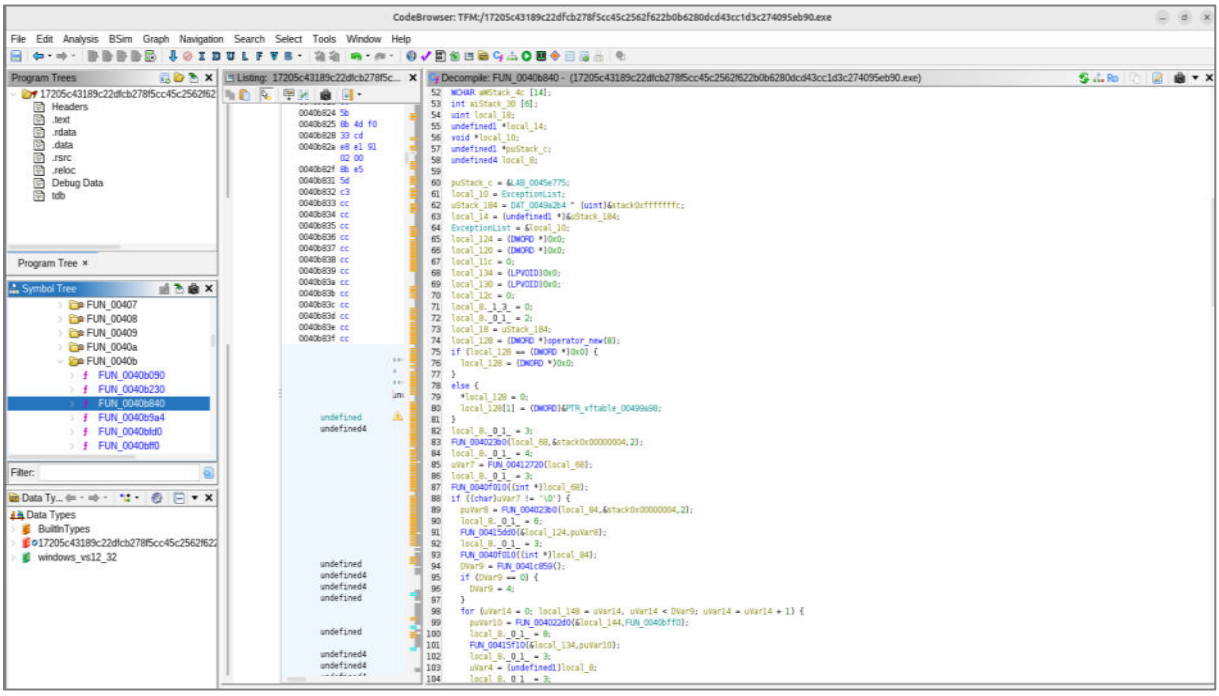


Figura 146

Revisión de código en Decompile: FUN_0040b840 en Ghidra (parte 2)

```
75    if (local_128 == (DWORD *)0x0) {
76      local_128 = (DWORD *)0x0;
77    }
78    else {
79      *local_128 = 0;
80      local_128[1] = (DWORD)&PTR_vftable_00499a98;
81    }
82    local_8._0_1_ = 3;
83    FUN_004023b0(local_68,&stack0x00000004,2);
84    local_8._0_1_ = 4;
85    uVar7 = FUN_00412720(local_68);
86    local_8._0_1_ = 3;
87    FUN_0040f010((int *)local_68);
88    if ((char)uVar7 != '\0') {
89      puVar8 = FUN_004023b0(local_84,&stack0x00000004,2);
90      local_8._0_1_ = 6;
91      FUN_00415dd0(&local_124,puVar8);
92      local_8._0_1_ = 3;
93      FUN_0040f010((int *)local_84);
94      DVar9 = FUN_0041c859();
```

Figura 147

Revisión de código en Decompile: FUN_0040b840 en Ghidra (parte 3)

```
130       psVar11 = (short *)FUN_00406140(aiStack_9c,(uint *)L"$Recycle.Bin");
131       local_8._0_1_ = 0x11;
132       uVar14 = FUN_00412aa0(aiStack_30,psVar11,0);
133       local_8._0_1_ = 0x10;
134       FUN_0040f010(aiStack_9c);
135       if (uVar14 == 0xffffffff) {
136         psVar11 = (short *)FUN_00406140(aiStack_b4,(uint *)L"Windows");
137         local_8._0_1_ = 0x13;
138         uVar14 = FUN_00412aa0(aiStack_30,psVar11,0);
139         local_8._0_1_ = 0x10;
140         FUN_0040f010(aiStack_b4);
141         if (uVar14 == 0xffffffff) {
142           psVar11 = (short *)FUN_00406140(aiStack_cc,(uint *)L"boot");
143           local_8._0_1_ = 0x15;
144           uVar14 = FUN_00412aa0(aiStack_30,psVar11,0);
145           local_8._0_1_ = 0x10;
146           FUN_0040f010(aiStack_cc);
147           if (uVar14 == 0xffffffff) {
148             psVar11 = (short *)FUN_00406140(aiStack_e4,(uint *)L"readme.txt");
149             local_8._0_1_ = 0x17;
150             uVar14 = FUN_00412aa0(aiStack_30,psVar11,0);
151             local_8._0_1_ = 0x10;
152             FUN_0040f010(aiStack_e4);
153             if (uVar14 == 0xffffffff) {
```

4.2.1.7. Rutina de Cifrado por Archivo. La función **FUN_0040bff0** es el "motor" real del cifrado. Analizando su código se observan las siguientes etapas:

- **Normalización de ruta:** Primero, construye rutas usando el prefijo \\?\ para manejar paths largos.

Figura 148

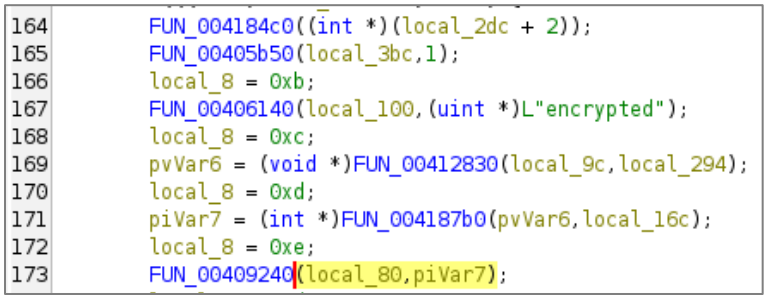
Revisión de código en Decompile: FUN_0040bff0 en Ghidra (parte 1)

```
138       FUN_004092d0(local_9c,puVar5);
139       pvVar6 = FUN_004187b0(local_9c,local_25c);
140       local_8 = 6;
141       puVar5 = FUN_00402c90(local_244,(uint *)L"\\\\?\\",pvVar6);
142       local_8 = 7;
143       piVar7 = FUN_00407c30(local_278,puVar5,2);
144       local_8 = 8;
145       FUN_00409280(local_9c,piVar7);
146       local_8 = 7;
147       FUN_0040f010(local_278);
148       local_8 = 6;
149       FUN_0040f010((int *)local_244);
150       local_8 = 5;
151       FUN_0040f010(local_25c);
152       local_2e0 = DAT_0049bf98;
153       FUN_00412660(&DAT_0049bf98,&local_2f4,DAT_0049bf98);
154       lpFileName = (LPCWSTR)FUN_004187b0(local_9c,local_184);
155       local_8 = 10;
156       if (7 < *(uint *)(lpFileName + 10)) {
157         lpFileName = *(LPCWSTR *)lpFileName;
158       }
```

- **Filtro para evitar recifrado:** La función obtiene el nombre de archivo, directorio y extensión. Si detecta la palabra "encrypted" en el nombre, evita cifrarlo de nuevo para no caer en loops.

Figura 149

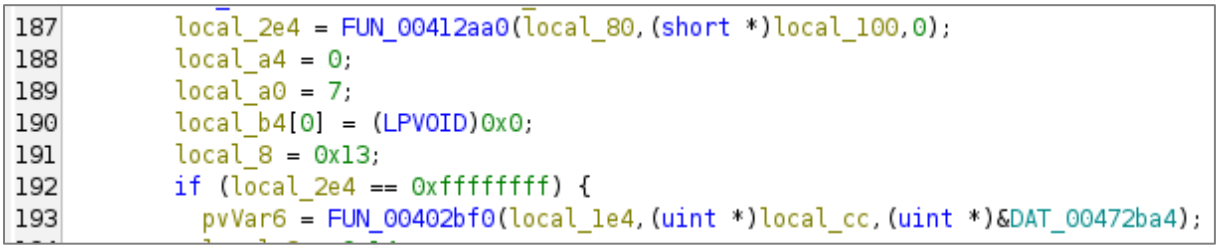
Revisión de código en Decompile: FUN_0040bff0 en Ghidra (parte 2)



```
164         FUN_004184c0((int *)(local_2dc + 2));
165         FUN_00405b50(local_3bc,1);
166         local_8 = 0xb;
167         FUN_00406140(local_100,(uint *)L"encrypted");
168         local_8 = 0xc;
169         pvVar6 = (void *)FUN_00412830(local_9c,local_294);
170         local_8 = 0xd;
171         piVar7 = (int *)FUN_004187b0(pvVar6,local_16c);
172         local_8 = 0xe;
173         FUN_00409240(local_80,piVar7);
```

Figura 150

Fragmento del código de la función FUN_0040bff0 donde se evaden loops de encriptación

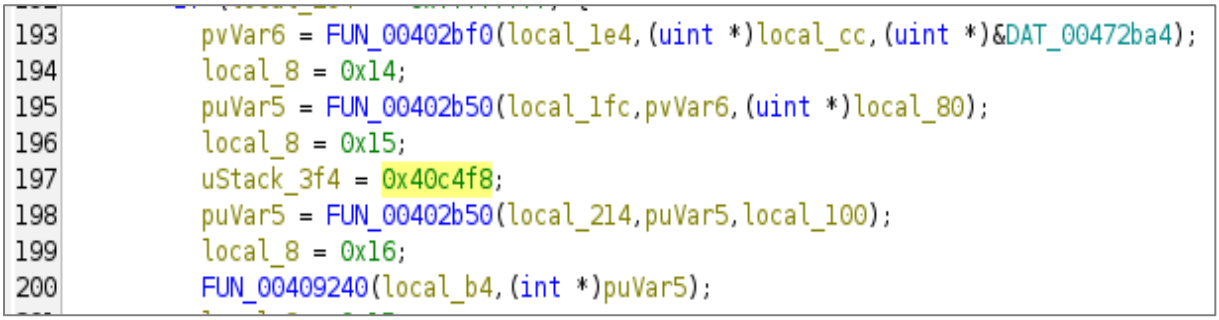


```
187         local_2e4 = FUN_00412aa0(local_80,(short *)local_100,0);
188         local_a4 = 0;
189         local_a0 = 7;
190         local_b4[0] = (LPVOID)0x0;
191         local_8 = 0x13;
192         if (local_2e4 == 0xffffffff) {
193           pvVar6 = FUN_00402bf0(local_1e4,(uint *)local_cc,(uint *)&DAT_00472ba4);
```

- **Construcción del nuevo nombre:** El ransomware genera una ruta destino con nueva extensión, por ejemplo, *archivo.doc* → *archivo.doc.encrypted*. Esta lógica está codificada explícitamente.

Figura 151

Fragmento del código de la función FUN_0040bff0 para renombrar archivos (parte 1)



```
193           pvVar6 = FUN_00402bf0(local_1e4,(uint *)local_cc,(uint *)&DAT_00472ba4);
194           local_8 = 0x14;
195           puVar5 = FUN_00402b50(local_1fc,pvVar6,(uint *)local_80);
196           local_8 = 0x15;
197           uStack_3f4 = 0x40c4f8;
198           puVar5 = FUN_00402b50(local_214,puVar5,local_100);
199           local_8 = 0x16;
200           FUN_00409240(local_b4,(int *)puVar5);
```

Figura 152

Fragmento del código de la función FUN_0040bff0 para renombrar archivos (parte 2)

```
207         pvVar6 = FUN_00402bf0(local_1b4,(uint *)local_cc,(uint *)&DAT_00472ba4);
208         local_8 = 0x17;
209         puVar5 = FUN_00402b50(local_1cc,pvVar6,(uint *)local_80);
210         local_8 = 0x18;
211         FUN_00407c30(local_138,puVar5,2);
212         local_8 = 0x19;
213         FUN_0040f0f0((undefined4 *)local_e8,local_138);
```

Figura 153

Fragmento del código de la función FUN_0040bff0 para renombrar archivos (parte 3)

```
220         FUN_00402440(local_154,(uint *)local_b4,2);
221         local_8 = 0x1f;
222         FUN_0040f0f0((undefined4 *)local_11c,local_154);
```

Adicionalmente, antes de cifrar valida que el origen existe:

Figura 154

Fragmento del código de la función FUN_0040bff0 para validación de origen

```
225         uVar8 = FUN_00412720(local_e8);
226         if ((char)uVar8 == '\0') {
227           FUN_004187b0(local_e8,local_19c);
228           local_8 = 0x23;
229           FUN_00418d70(0x473374);
230           local_8 = 0x21;
231           FUN_0040f010(local_1
232         }
233         bVar4 = FUN_0040d500(l
234         if (bVar4) {
235           FUN_004092d0(local_9
236           puVar5 = (uint *)FUN
237           local_8 = 0x24;
238           FUN_00415060(local_3
239           local_8 = 0x21;
240           FUN_0040f010(local_2
241           if (local_358 == 0)
242             FUN_0040d500(local
```

	Hex	Decimal
dword	473374h	4666228
sdword	473374h	4666228
char[]		00h,"G3t"
wchar16[]		u"Gbar"
Address		00473374
Symbol	*u_Path_1_not_exists:_%ls_00473374*	

- **Renombrado:** El siguiente fragmento de código demuestra el proceso que sigue el malware para el renombre de archivos.

Figura 155

Fragmento del código de la función FUN_0040bff0 para el renombre de archivos

```
233           bVar4 = FUN_0040d500(local_e8,local_11c);
234           if (bVar4) {
235             FUN_004092d0(local_9c,(uint *)local_11c);
236             puVar5 = (uint *)FUN_0040f0f0(local_2cc,(LPCWSTR)local_9c);
237             local_8 = 0x24;
238             FUN_00415060(local_3bc,puVar5,0x27);
239             local_8 = 0x21;
240             FUN_0040f010(local_2cc);
241             if (local_358 == 0) {
242               FUN_0040d500(local_11c,local_e8);
243             }
```

Se observó que **FUN_0040d500** ejecutó directamente la instrucción: *MoveFileW(old, new)*; lo que coincide con el comportamiento típico del malware (cambiar extensión tras cifrado).

Figura 156

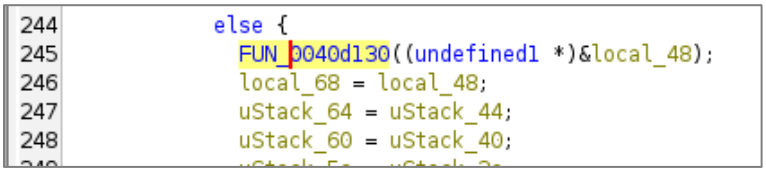
Fragmento del código de la función FUN_0040d500

```
1
2 bool __cdecl FUN_0040d500(LPCWSTR param_1,LPCWSTR param_2)
3
4 {
5   BOOL BVar1;
6
7   if (7 < *(uint *)(param_2 + 10)) {
8     param_2 = *(LPCWSTR *)param_2;
9   }
10  if (7 < *(uint *)(param_1 + 10)) {
11    param_1 = *(LPCWSTR *)param_1;
12  }
13  BVar1 = MoveFileW(param_1,param_2);
14  return BVar1 != 0;
15 }
16
```

- **Generación de Clave/Identificador por Archivo:** La función FUN_0040d130 genera un buffer de 40 bytes (0x28) donde el primer byte es “i”, los restantes 39 bytes son aleatorios generados con rand_s (CRNG de Windows). Este buffer actúa como clave de cifrado, identificador único de archivo y semilla para el cifrado híbrido.

Figura 157

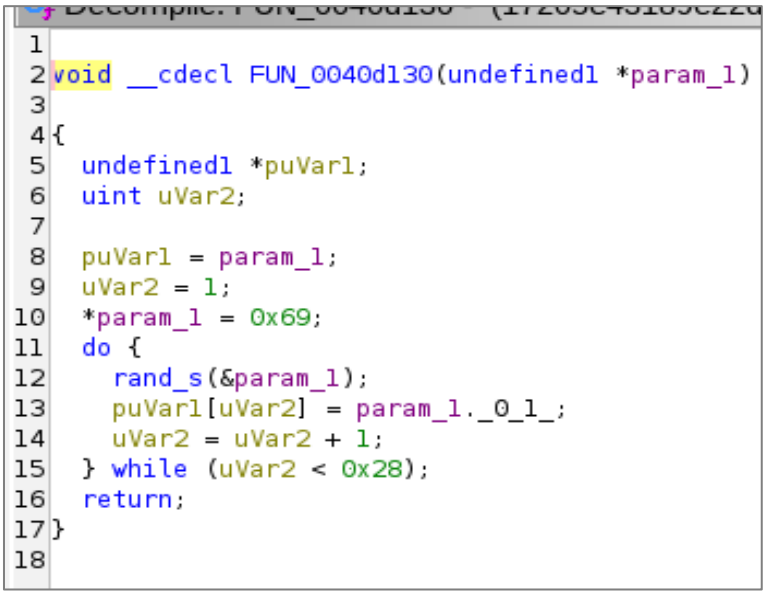
Fragmento del código de la función FUN_0040bff0 para la generación de clave



```
244            else {
245              FUN_0040d130((undefined1 *)&local_48);
246              local_68 = local_48;
247              uStack_64 = uStack_44;
248              uStack_60 = uStack_40;
```

Figura 158

Fragmento del código de la función FUN_0040d130



```
 1
 2 void __cdecl FUN_0040d130(undefined1 *param_1)
 3
 4 {
 5   undefined1 *puVar1;
 6   uint uVar2;
 7
 8   puVar1 = param_1;
 9   uVar2 = 1;
10   *param_1 = 0x69;
11   do {
12     rand_s(&param_1);
13     puVar1[uVar2] = param_1._0_1_;
14     uVar2 = uVar2 + 1;
15   } while (uVar2 < 0x28);
16   return;
17 }
18
```

- **Cifrado del material aleatorio (RSA):** Las acciones que realiza el módulo RSA internamente para cifrar el buffer son las siguientes: Primero, convierte bytes a un número gigante (multi-precisión); Segundo, aplica la operación matemática de RSA (exponenciación modular); Tercero, convierte el resultado de la operación a bytes nuevamente (blob cifrado); Finalmente, devuelve el blob para almacenarlo en el archivo.

- **Escritura de blob y metadatos en el archivo:** La FUN_00415060 abre el archivo renombrado y escribe la clave aleatoria original (“iXXXX…”), el tamaño del blob RSA

y el blob RSA resultante. Estos datos forman una cabecera característica en todos los archivos cifrados. Es una evidencia extremadamente útil para detección.

Figura 159

Fragmento del código de la función FUN_0040bff0 donde llama a la función FUN_00415060

```
233           bVar4 = FUN_0040d500(local_e8,local_11c);
234           if (bVar4) {
235             FUN_004092d0(local_9c,(uint *)local_11c);
236             puVar5 = (uint *)FUN_0040f0f0(local_2cc,(LPCWSTR)local_9c);
237             local_8 = 0x24;
238             FUN_00415060(local_3bc,puVar5,0x27);
239             local_8 = 0x21;
240             FUN_0040f010(local_2cc);
241             if (local_358 == 0) {
242               FUN_0040d500(local_11c,local_e8);
```

Figura 160

Fragmento del código de la función FUN_00415060

```
Decompile: FUN_00415060 - (17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe)
 1
 2 void __thiscall FUN_00415060(void *this,uint *param_1,uint param_2)
 3
 4 {
 5   int iVar1;
 6   code *pcVar2;
 7   wchar_t ***pppwVar3;
 8   basic_filebuf<> *pbVar4;
 9   uint uVar5;
10   undefined4 *puVar6;
11   char *pcVar7;
12   int local_4c [5];
13   undefined4 local_38 [2];
14   undefined4 local_30;
15   wchar_t **local_2c [5];
16   uint local_18;
17   uint local_14;
18   void *local_10;
19   undefined1 *puStack_c;
20   undefined4 local_8;
21
22   puStack_c = &LAB_0045f79e;
23   local_10 = ExceptionList;
24   local_14 = DAT_0049a2b4 ^ (uint)&stack0xfffffffc;
25   ExceptionList = &local_10;
26   local_30 = 0;
27   local_8 = 0;
28   FUN_00405fd0(local_2c,param_1);
29   pppwVar3 = local_2c;
30   if (7 < local_18) {
31     pppwVar3 = (wchar_t ***)local_2c[0];
32   }
33   local_30 = 1;
34   pbVar4 = FUN_00414f70((void *)((int)this + 0x18),(wchar_t *)pppwVar3,param_2,0x40);
35                     /* WARNING: Load size is inaccurate */
36   iVar1 = *(int *)(*this + 4);
37   if (pbVar4 == (basic_filebuf<> *)0x0) {
38     uVar5 = ((uint)(*(int *)(iVar1 + 0x38 + (int)this) == 0) * 4 + 2 |
39              *(uint *)(iVar1 + 0xc + (int)this)) & 0x17;
40     *(uint *)(iVar1 + 0xc + (int)this) = uVar5;
41     uVar5 = *(uint *)(iVar1 + 0x10 + (int)this) & uVar5;
42     if (uVar5 != 0) {
43       if ((uVar5 & 4) == 0) {
44         pcVar7 = "ios_base::failbit set";
45         if ((uVar5 & 2) == 0) {
46           pcVar7 = "ios_base::eofbit set";
47         }
48       }
49       else {
50         pcVar7 = "ios_base::badbit set";
51       }
52       puVar6 = (undefined4 *)FUN_00414980(local_38,1);
53       goto LAB_0041518a;
```

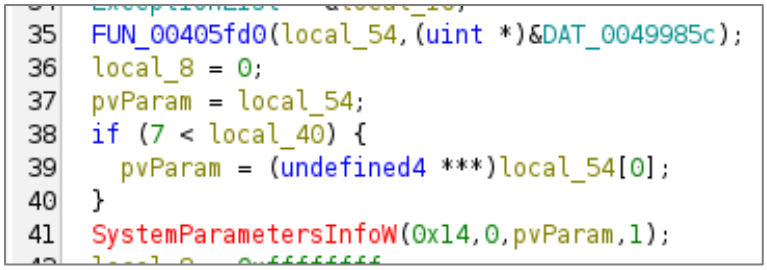
- **Cifrado del contenido del archivo.** Aunque la rutina concreta del cifrado simétrico no aparece completa en los fragmentos analizados, el flujo confirma que se realiza después de escribir los metadatos, como ocurre en ransomware real: AES o ChaCha20 para velocidad y RSA para proteger la clave AES.

4.2.1.8. Manipulación del Wallpaper de Fondo. En la función **FUN_0040ae10** se puede evidenciar el siguiente comportamiento de cambio de Wallpaper de Windows.

- Escritura de un archivo de imagen en el disco (identificado por las llamadas a CreateFileW, WriteFile).
- Uso de SystemParametersInfoW(SPI_SETDESKWALLPAPER) para aplicar la nueva imagen.

Figura 161

Fragmento del código de la función FUN_0040ae10



4.2.1.9. Manipulación de Servicios y Detección de Entornos Virtualizados. El código contiene una función (**FUN_00401180**) la cual construye listas de servicios de Windows:

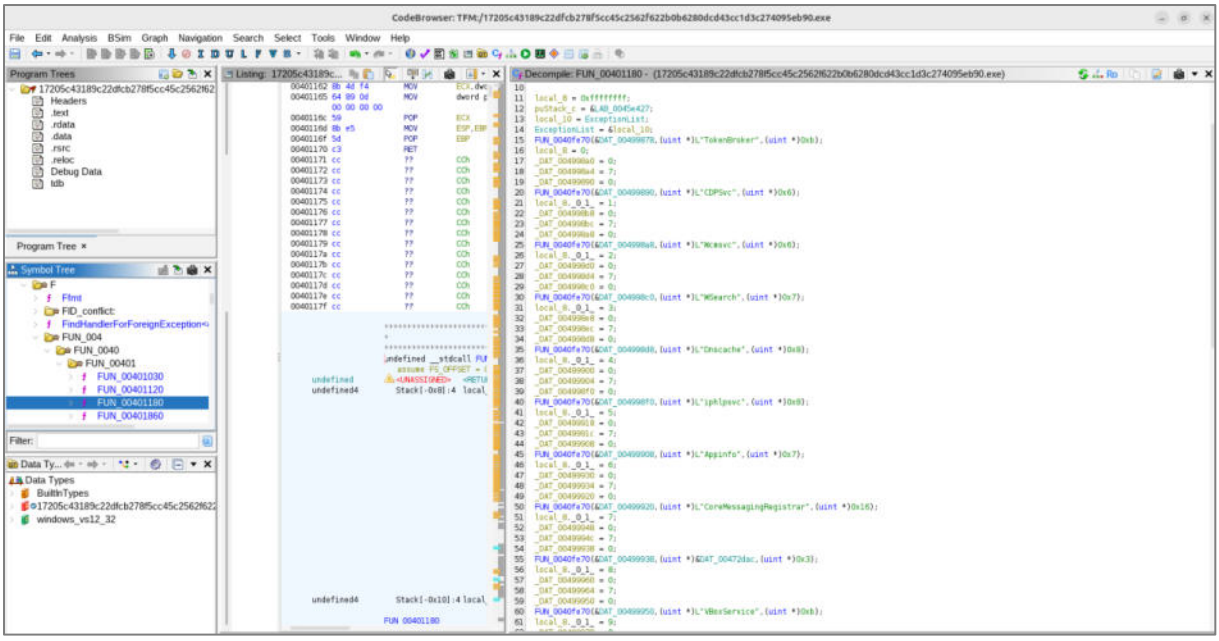
- TokenBroker,
- CDPSvc,
- WSearch,
- Dnscache,
- Appinfo,

- VBoxService,
- SamSs,
- Wuauserv,
- LanmanServer,
- Power, etc.

Aunque en los fragmentos analizados no se ve la ejecución directa sobre dichos servicios, su presencia indica: potencial detención de servicios que bloqueen archivos, búsqueda de entornos virtualizados (VBoxService) e incorporación de lógica anti-análisis. Esto refuerza el carácter profesional del ransomware.

Figura 162

Revisión de código en Decompile: FUN_00401180 en Ghidra



4.2.1.10. Creación de Readme.txt. En la función FUN_0040b090 se puede apreciar de forma bastante clara la creación del readme.txt, incluso mostrando de forma explícita su contenido. Del código resaltan los siguientes comportamientos:

- Llama a SHGetSpecialFolderPathW con csidl = 0. Eso corresponde al CSIDL_DESKTOP (virtual Desktop, y en muchos samples se usa para conseguir la ruta de escritorio del usuario).
- FUN_00406140 parece construir un std::wstring con "\\readme.txt".
- FUN_00402c90 concatena local_224 (ruta del escritorio) con "\readme.txt", se construye la ruta completa C:\Users\<user>\Desktop\readme.txt

Figura 163

Fragmento del código de la función FUN_0040b090

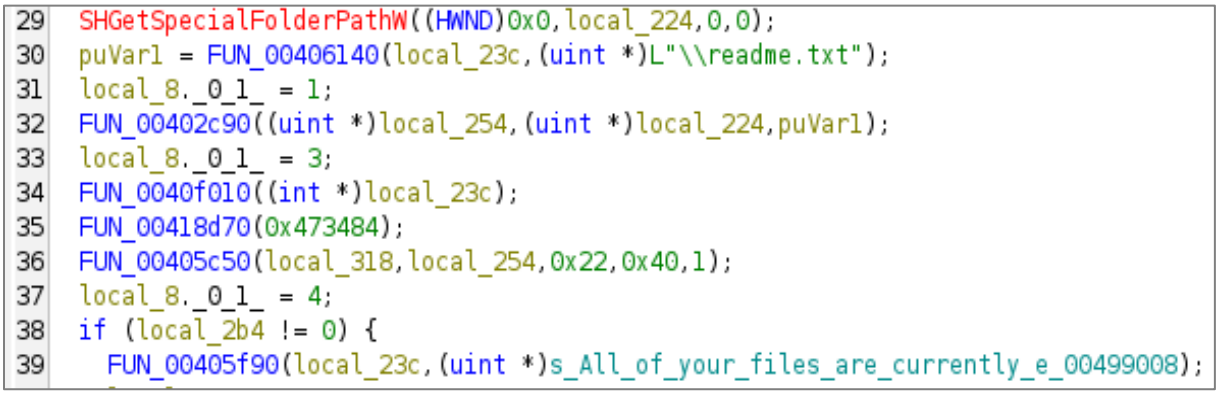
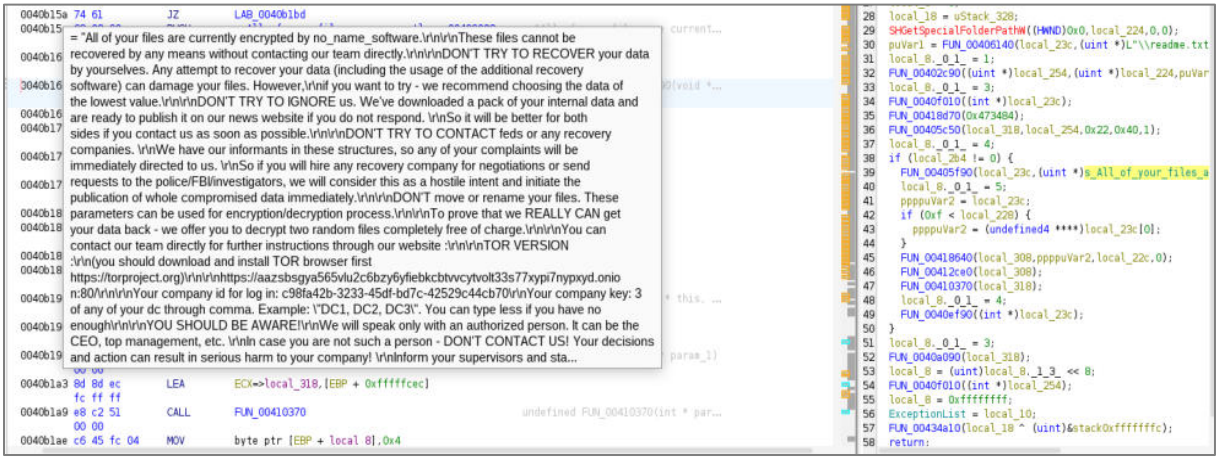


Figura 164

Detalle del código de la función FUN_0040b090



4.2.1.11. Métricas Finales, Tiempo y GB Cifrados. En la función FUN_0040ae10 también se puede visualizar como el malware muestra en consola la cantidad de memoria que se encriptó y el tiempo en el que lo hizo. Por lo tanto, se interpreta lo siguiente:

- uVar4 parece contener algún valor acumulado, podría ser tiempo total o bytes cifrados (o ambos empaquetados).
- FUN_00434c40 convierte un entero de 64 bits (DWORD64) a double.
- FUN_00418d10 (“Done time: %.4f seconds, encrypted: %.4f gb”) formatea y logea las estadísticas (tiempo total y GB cifrados). Puede ser a consola, debug log, fichero, etc.

Figura 165

Fragmento del código de la función FUN_0040ae10

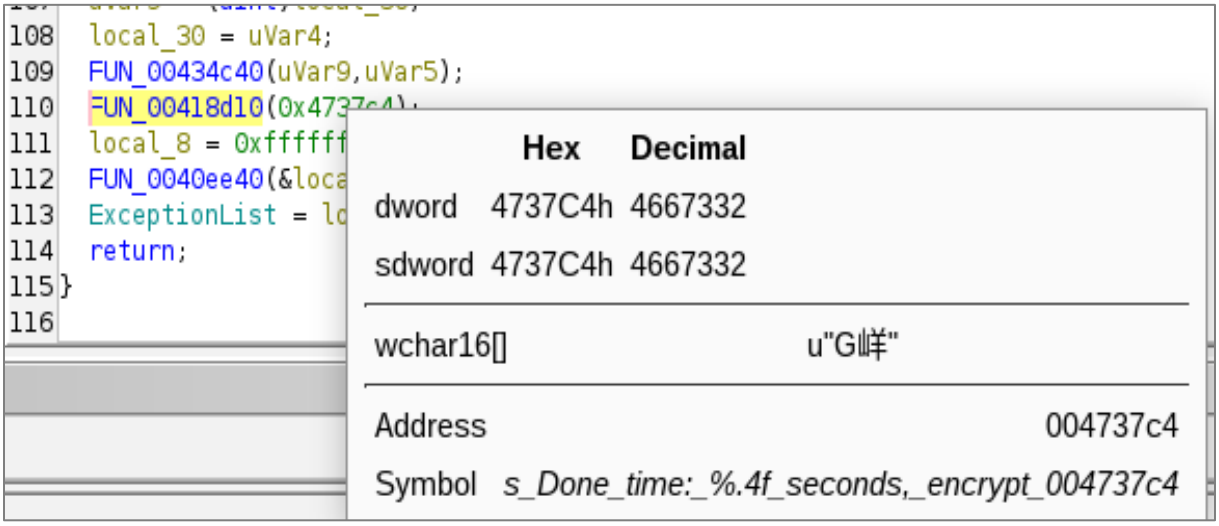


Figura 166

Fragmento del código de la función FUN_0040ae10 que muestra la cantidad de encriptados

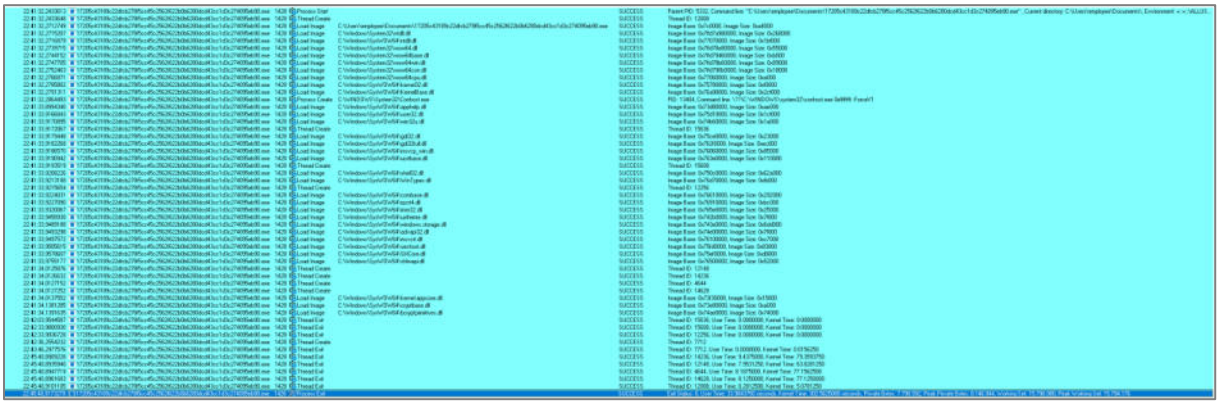
4.2.2. Análisis Dinámico. A continuación, se describirán los puntos clave del funcionamiento de Black Basta, de acuerdo con lo observado durante la ejecución del ransomware y la información recopilada de los artefactos forenses.

4.2.2.1. Carga de Imágenes o Librerías Criptográficas. En los registros de tipo "Process and Thread Activity" se puede ver carga continua de módulos (Load Image) por parte del ransomware, donde el proceso de la muestra específicamente carga las siguientes librerías:

- ntdll.dll
- kernel32.dll
- kernelbase.dll
- advapi32.dll
- bcrypt.dll
- cryptbase.dll
- rpcrt4.dll
- shell32.dll
- ole32.dll
- mpr.dll
- shlwapi.dll
- user32.dll
- gdi32.dll
- msvcrt.dll

Figura 167

Proceso de carga de imágenes y librerías criptográficas por parte de la muestra



Este patrón encaja perfectamente con lo previsto en el análisis estático, es decir, la muestra contiene implementaciones de funciones criptográficas propias.

Se observaron funciones relacionadas a GMP (big integers), modular exponentiation, importación/exportación de buffers, etc.

Este tipo de operaciones requiere cargar módulos relacionados con criptografía, matemáticas y manipulación de memoria, exactamente lo que aparece en Procmon. Esto conlleva una correlación directa con la rutina de cifrado.

A grandes rasgos se puede decir que el ransomware:

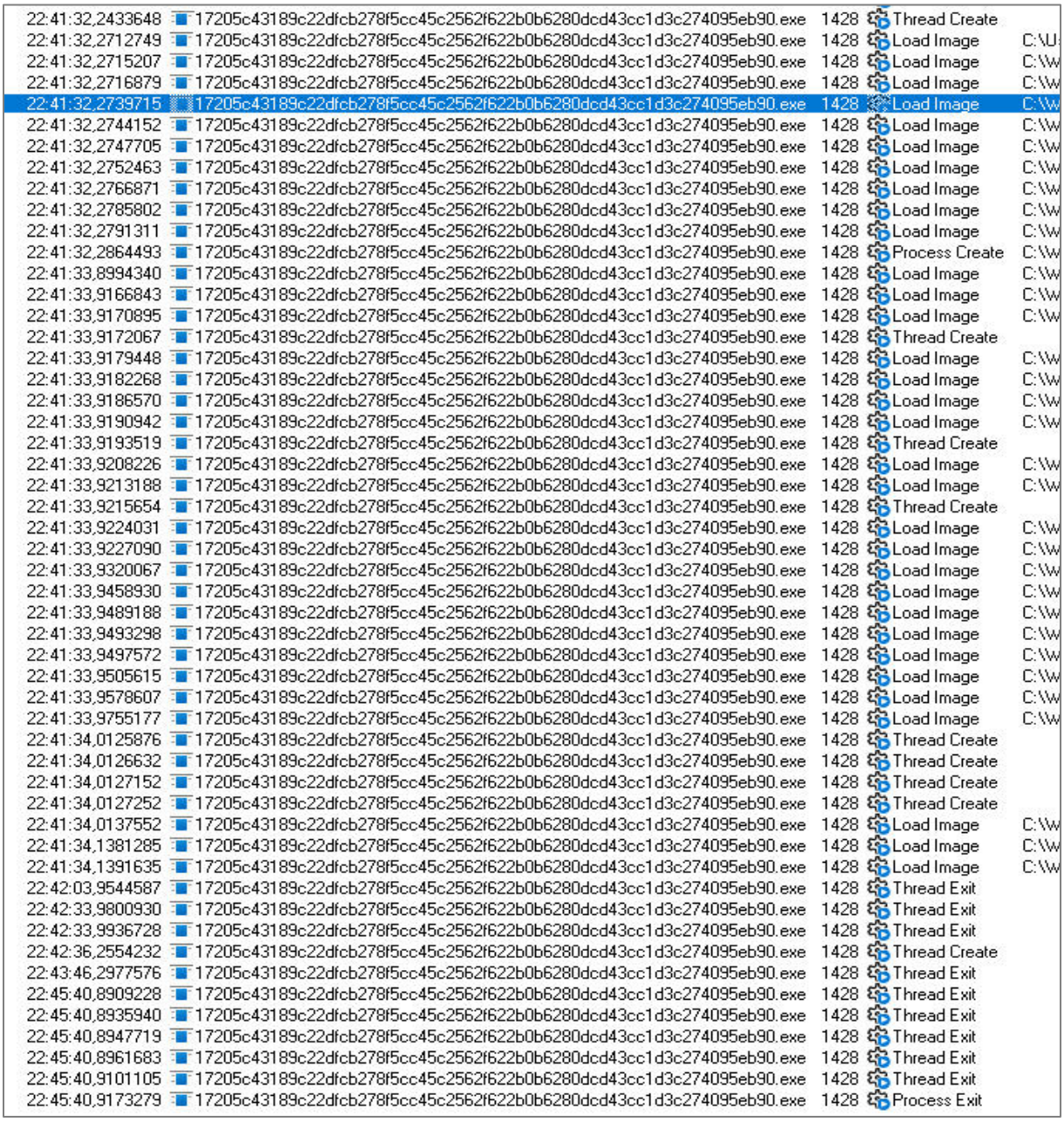
- Genera llaves.
- Prepara buffer.
- Cifra archivos.
- Escribe extensiones.
- Cambia nombre y metadatos.

Todo este pipeline exige una gran cantidad de DLLs cargadas dinámicamente, y eso es exactamente lo que se observa en el output de Procmon.

4.2.2.2. Creación Anómala de Threads. El ransomware crea varios threads, muchos en paralelo, y los cierra en poco tiempo; esto coincide con la estrategia de aceleración del cifrado (multithreading) y se correlaciona con el comportamiento de mpz_powm y funciones de cifrado en el binario visto con Ghidra.

Figura 168

Carga de imágenes y creación de threads



22:41:32,2433648	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:32,2712749	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\U
22:41:32,2715207	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2716879	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2739715	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2744152	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2747705	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2752463	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2766871	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2785802	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2791311	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:32,2864493	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Process Create	C:\W
22:41:33,8994340	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9166843	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9170895	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9172067	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:33,9179448	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9182268	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9186570	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9190942	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9193519	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:33,9208226	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9213188	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9215654	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:33,9224031	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9227090	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9320067	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9458930	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9489188	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9493298	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9497572	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9505615	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9578607	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:33,9755177	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:34,0125876	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:34,0126632	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:34,0127152	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:34,0127252	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:41:34,0137552	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:34,1381285	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:41:34,1391635	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Load Image	C:\W
22:42:03,9544587	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:42:33,9800930	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:42:33,9936728	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:42:36,2554232	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Create	
22:43:46,2977576	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:45:40,8909228	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:45:40,8935940	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:45:40,8947719	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:45:40,8961683	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:45:40,9101105	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Thread Exit	
22:45:40,9173279	17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe	1428	Process Exit	

4.2.2.3. Manipulación del Wallpaper de Fondo. También se verificó el proceso en el cual se hace un llamado al panel de control, específicamente al módulo de “Desktop\Wallpaper”, lo cual se asocia a la manipulación del Wallpaper o fondo de escritorio prevista tanto en el análisis estático con Ghidra, como confirmada en el análisis dinámico cuando se ejecutó el malware, ya que evidentemente el fondo de escritorio cambió. El ransomware sobrescribe el valor Wallpaper del usuario a una cadena vacía, por esta razón se muestra todo en negro.

Figura 169

Llamado al módulo de Desktop Wallpaper

Claves leídas:

- HKCU\Control Panel\Desktop\Wallpaper
- HKCU\Control Panel\Desktop
- HKCU\Software\Microsoft\Windows\CurrentVersion\Themes\Personalize

Y una escritura directa:

- RegSetValue
- Path: HKCU\Control Panel\Desktop\Wallpaper
- Type: REG_SZ, Length: 2, Data: “”

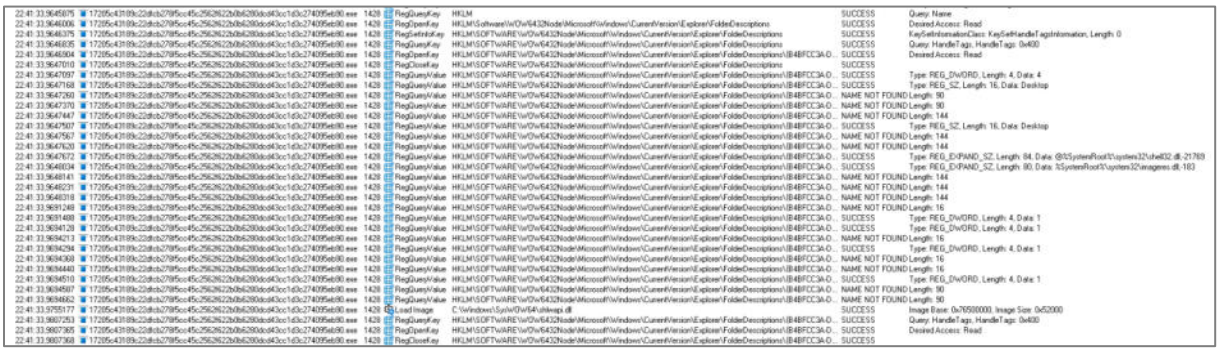
4.2.2.4. Descubrimiento del Entorno y Carpetas del Usuario. Se observa bastante actividad de lectura sobre claves de Explorer y del perfil:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{B4BFCC3A-...}
- HKCU\Control Panel\Desktop\UserPreferencesMask, etc.

Figura 170

Procesos de lectura sobre claves de Explorer y del perfil



Estas claves le permiten al malware resolver las rutas reales de Desktop, Documents, Downloads, etc., independientemente del idioma y de si el usuario ha movido carpetas.

En el análisis estático, se examinaron funciones que generan un listado de carpetas a cifrar y que excluyen Windows, boot, $Recycle.Bin y el propio readme.txt; ahora, mediante el análisis dinámico, se confirma, pues el proceso observado se apoya en la meta data de Explorer para saber dónde están los datos del usuario.

Cuando un proceso desconocido empieza a consultar “User Shell Folders” justo antes de hacer miles de operaciones de archivos, este patrón se convierte en un indicador muy bueno para detección comportamental.

4.2.2.5. Comprobaciones de Estado del Sistema SafeBoot, SmartLocker y Session Manager. En la captura de Procmon aparecen muchas consultas hacia claves del sistema bajo el proceso de la muestra del ransomware, estas son:

- HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SafeBoot

- HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SafeBoot\Option
- HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Configuration Manager

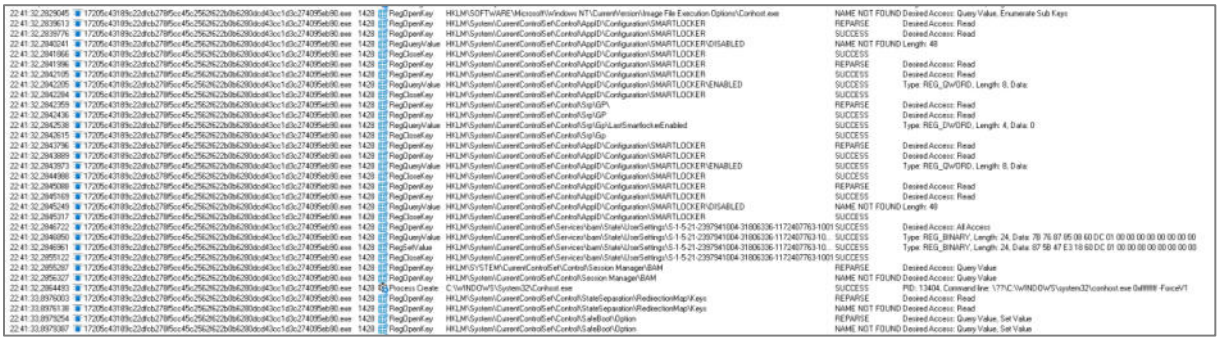
Adicionalmente, claves bajo HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Sdb y valores tipo sysmain.sdb

También aparecen otras consultas hacia claves relacionadas con un directorio llamado “SMARTLOCKER”, mismas que se enlistan a continuación:

- RegOpenKey HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Sdb\SMARTLOCKER
- RegQueryValue ...\SMARTLOCKER\ENABLED
- RegQueryValue ...\SMARTLOCKER\DISABLED

Figura 171

Consultas de claves del sistema por parte del ransomware



Lo anterior descrito sugiere que el binario realiza una comprobación para determinar si el sistema está en modo seguro / SafeBoot, y lee parámetros del Session Manager. Muchos ransomware ajustan su comportamiento de acuerdo con el modo de arranque del sistema, si es modo seguro o modo normal.

Inclusive, se observa la consulta hacia configuraciones de SmartLocker / AppLocker / AppCompat, probablemente, para saber si hay políticas que puedan impedir la ejecución o para adaptar algunas rutas.

Las lecturas concentradas sobre Session Manager\SafeBoot y claves de AppCompatFlags / SmartLocker por parte de un binario ejecutado desde el perfil del usuario, constituyen un patrón poco habitual en aplicaciones legítimas y encajan con la lógica de preparación del entorno observada en el análisis estático del ransomware.

4.2.2.6. Consultas a Image File Execution Options y Compatibilidad. En los registros de Procmon se distingen:

- RegOpenKey
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\17205c43189c22...

Figura 172

Comprobación de llamado a Image File Executions Options

Esto indica que el propio proceso comprueba si tiene asociada alguna política de Image File Execution Options, por ejemplo, si alguien ha configurado un debugger o un mitigador para ese ejecutable.

Desde el punto de vista de investigación, esto confirma el componente anti-análisis previsto durante el análisis estático, en referencias a funciones como IsDebuggerPresent y verificación de entorno en Ghidra.

4.2.2.7. Actividad Enorme de Sistema de Archivos. Usando Procmon se detectó que al ejecutar el binario se empiezan a capturar una cantidad enorme de eventos de “File System

Activity", encontrando millones de eventos de modificación, creación y renombramiento de archivos y carpetas.

Para conseguir los datos de los eventos que captura Procmon, se realizó una segunda ejecución del malware, partiendo de su estado inicial a través de la snapshot almacenada antes de la primera infección. Se empezó a capturar eventos con la herramienta Procmon, desde que inició hasta que culminó la ejecución del ransomware, obteniendo un total de 22.411.855 eventos. De este total, 21.968.037 eventos pertenecen a "File System Activity" y los restantes, es decir, 443.818 (1%) eventos pertenecen a otros tipos de actividades como "Network Activity", "Process and Threats activity" y "Registry Activity". Los resultados alcanzados son lógicos debido al comportamiento de un ransomware, y coinciden con la evaluación realizada al código en el análisis estático, ya que el malware realiza enumeración y barrido de todo el sistema de archivos para el cifrado, generando así gran cantidad de eventos.

Figura 173

Análisis realizado con la herramienta Procmon

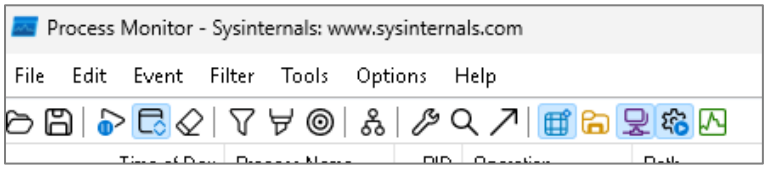
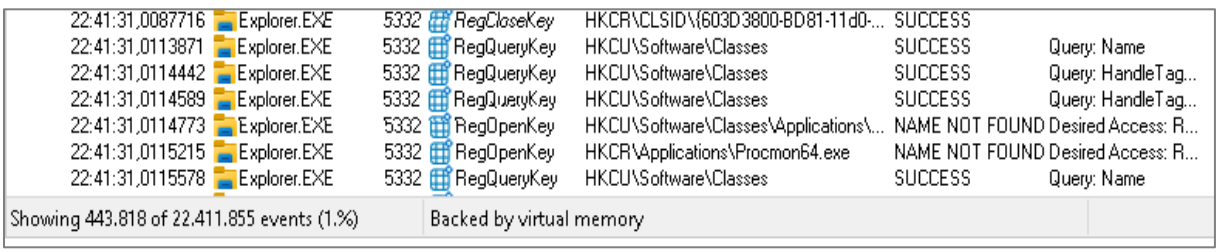


Figura 174

Total de eventos registrados por Procmon



4.2.2.8. Creación y Acceso a Archivos Temporales en AppData\Local\Temp. De estos eventos relacionados con la manipulación de archivos del sistema se presentan los siguientes ejemplos:

- C:\Users\employee\AppData\Local\Temp\dai-vvdiolxnjq.jpg
- C:\Users\employee\AppData\Local\Temp\dai-wqdioajhv.jpg

Figura 175

Registros de creación y acceso a archivos temporales (parte 1)

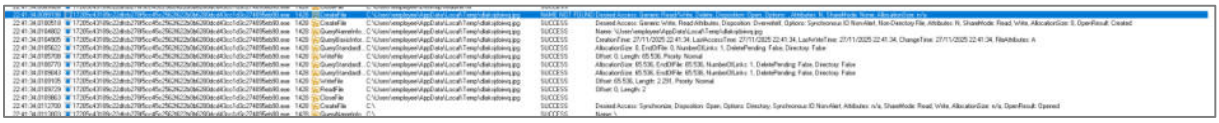
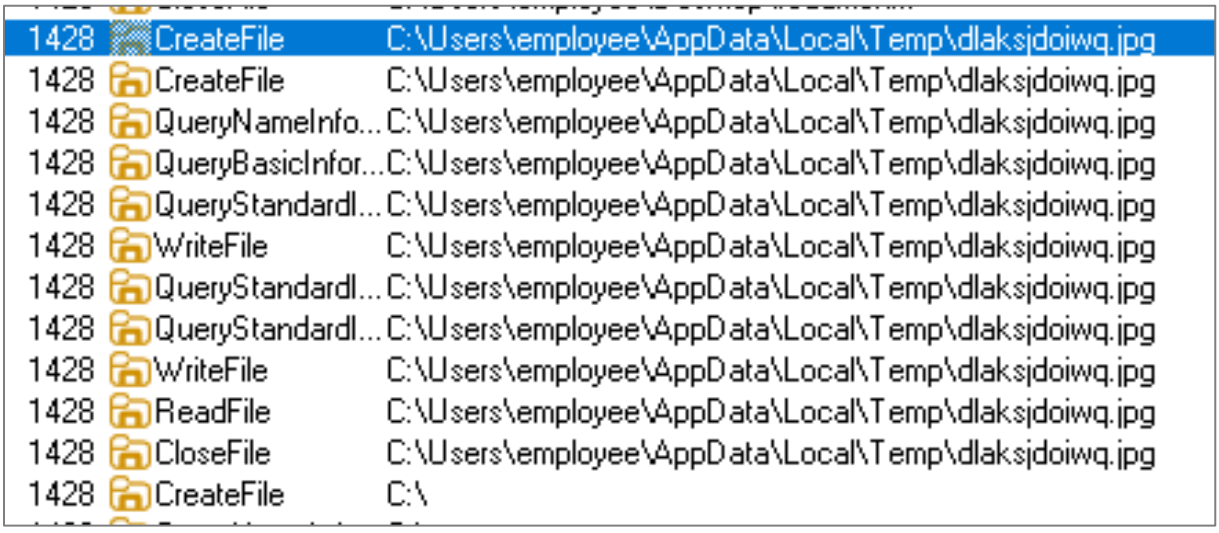


Figura 176

Registros de creación y acceso a archivos temporales (parte 2)



Estos eventos incluyen:

- CreateFile
- WriteFile
- CloseFile

Los archivos .jpg mencionados no contienen imágenes reales, suelen ser blobs de datos generados durante la inicialización del malware. Black Basta crea esto para generar buffers temporales.

Como se observó en el análisis estático, Black Basta utiliza buffers pseudoaleatorios generados con rand_s() (dentro de función FUN_0040d130). Estos archivos temporales concuerdan con este comportamiento.

Antes de cifrar, muchos ransomware generan claves parciales, estructuras intermedias, y segmentos para la protección de memoria.

No se observa persistencia ni intento de ocultamiento en esta fase, son archivos temporales de vida muy corta.

4.2.2.9. Enumeración del Sistema de Archivos. Se considera una fase pre-cifrado. Esta es una de las actividades más ricas e importantes para el análisis, ya que aparecen cientos de llamadas como:

- QueryDirectory
- CreateFile
- ReadFile
- CloseFile

Se observa actividad en directorios de usuario:

- C:\Users\employee\Documents
- C:\Users\employee\Desktop
- C:\Users\employee\Downloads

Se observa actividad en directorios del sistema:

- C:\Windows\System32

- C:\Program Files
- C:\ProgramData

Se observa actividad en directorios críticos del sistema de archivos:

- C:\Recovery
- C:\System Volume Information

Se observa actividad en Unidades completas, incluso aparecen: C:\, D:\, E:\ enumerándose recursivamente.

Figura 177

Creación y cierre de archivos en unidades de disco detectadas

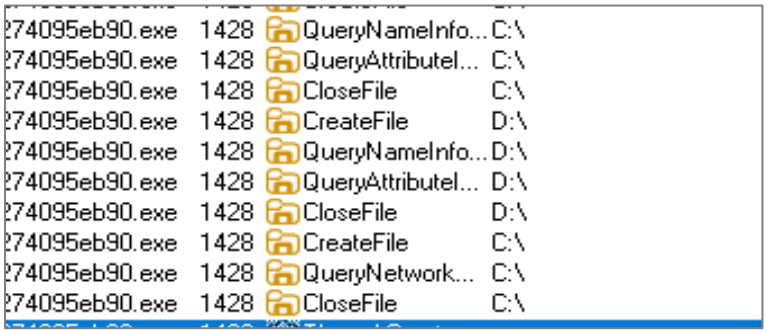
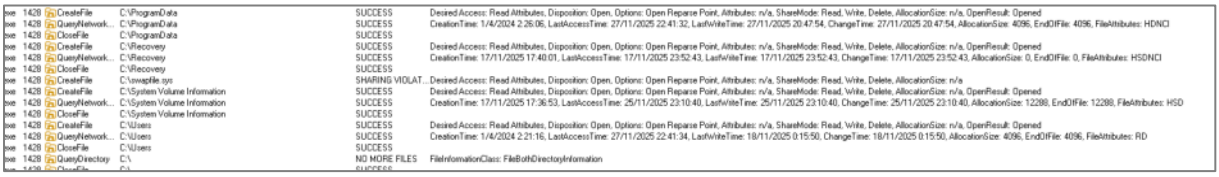


Figura 178

Registro de búsqueda del ransomware de archivos que cifrar



Esta sección de enumeración refleja la fase de descubrimiento en donde el ransomware recorre el sistema para localizar archivos que son candidatos para cifrar, por ejemplo, extensiones, tipos de documentos, bases de datos, etc. También, identifica rutas de exclusiones, por ejemplo, archivos del sistema que no debe dañar. De esta manera, construye su lista de objetivos antes de cifrar.

La forma masiva y secuencial en que aparecen estos eventos concuerda plenamente con la lógica interna que se logró ver con Ghidra. El sample evalúa y recorre directorios completos, pero omite cifrar algunos ubicados dentro de la propia carpeta del malware o archivos que están en uso.

4.2.2.10. Búsqueda de Archivos Sensibles. Archivos que se consideran sensibles son bases de datos, logs, configuraciones, entre otros. Se observaron accesos a:

- C:\Windows\AppPatch\MegSdb*
- C:\Windows\System32\LogFiles
- Archivos .db, .ini, .log

No se debe confundir esta búsqueda con el cifrado, pues todavía no lo es, sino que es parte de su proceso de identificación de archivos bloqueados. Los ransomware suelen evaluar si un archivo está en uso, bloqueado y protegido por permisos especiales.

Adicionalmente, identifica archivos grandes o de sistema, para evitar dañar la máquina y perder acceso de control.

4.2.2.11. Proceso de Encriptación de Archivos. El monitoreo de Procmon muestra claramente más de 20 millones de eventos de QueryInformationStandardFile, WriteFile, y ReadFile, en donde se visualiza como se renombran los archivos originales por el nuevo formato con la terminación “encrypted”, de acuerdo con lo previsto en el código descompilado del análisis estático.

Figura 179

Registros de procesos de encriptación de archivos

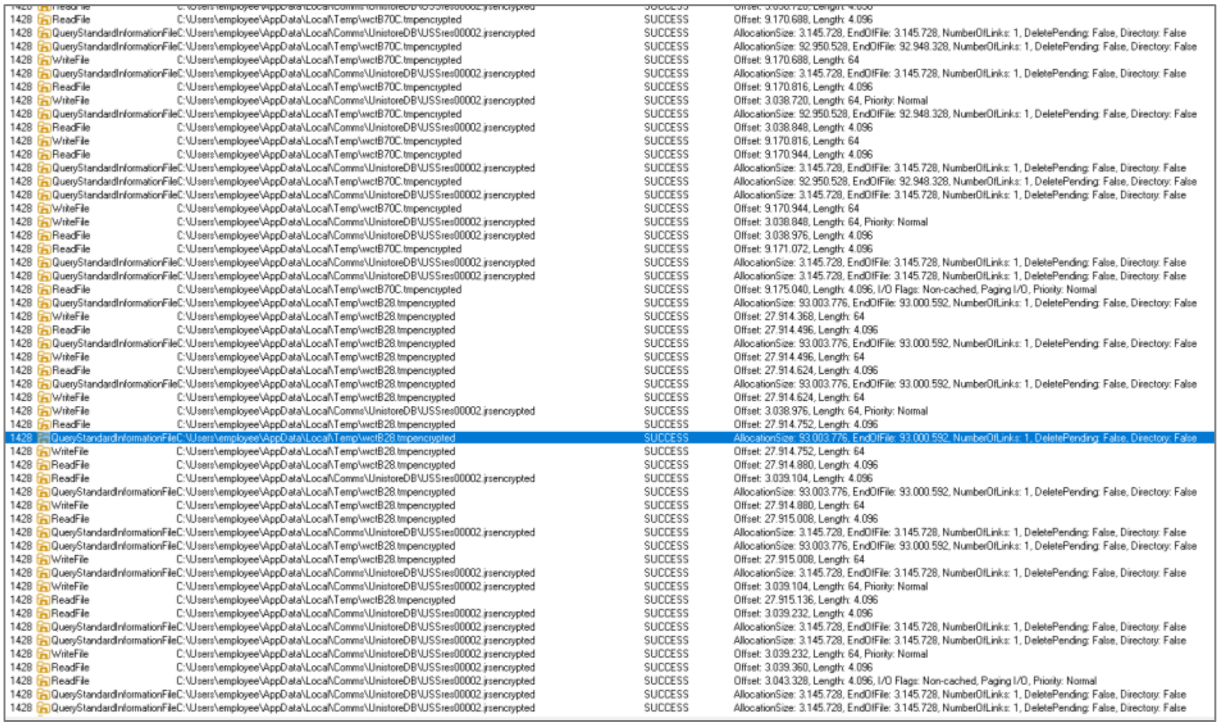
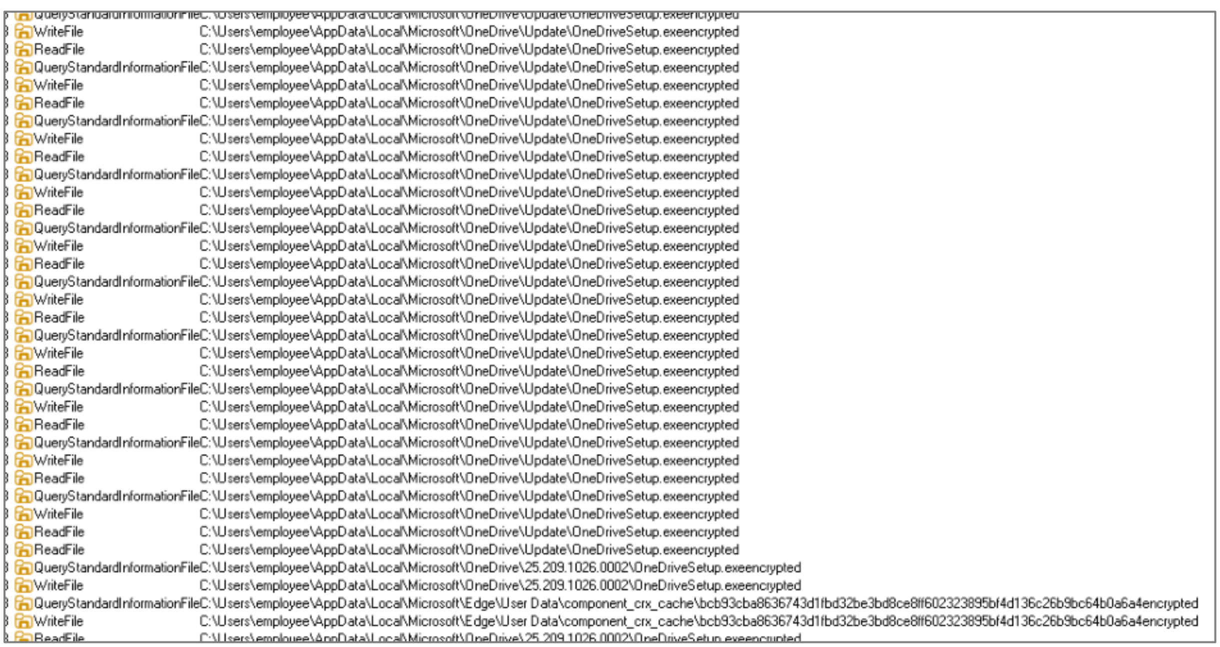


Figura 180

Registros de procesos de encriptación de archivos



4.2.3. Análisis de Artefactos .json. En esta sección se realiza una explicación de los artefactos encontrados como evidencia, resaltando sus hallazgos claves y herramienta que fue utilizada, preservando su integridad en el momento de la recolección.

4.2.3.1. Elastic.EventLogs.Sysmon.json. En la ejecución controlada, los eventos de Sysmon permiten reconstruir con precisión la secuencia completa que lleva a la activación del ransomware.

En primer lugar, un evento ID 1 muestra la ejecución de WinRAR por parte del usuario "employee", seguidamente, se registra un evento ID 11 (regla Downloads) donde WinRAR crea un archivo con nombre tipo hash en la carpeta Downloads. Instantes después, un segundo evento ID 1 evidencia la ejecución explícita del binario 17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe desde el directorio de descargas, con explorer.exe como proceso padre, no muestra información de producto o compañía y en el nombre se observa la coincidencia exacta con su hash SHA-256, características típicas de muestras de malware utilizadas en laboratorios o campañas delictivas.

Figura 181

Registros de eventos en Sysmon (parte 1)

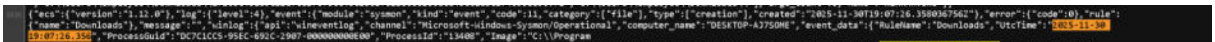


Figura 182

Registros de eventos en Sysmon (parte 2)

Figura 183

Registros de eventos en Sysmon por ejecución del Binario

Tras aproximadamente un minuto de actividad (evento ID 5, terminación de proceso), el sistema genera varios eventos ID 13 asociados tanto a las claves AppCompatFlags (Compatibility Assistant Store) como al inventario de aplicaciones (InventoryApplicationFile). En ellos se registra la ruta completa del ejecutable en minúsculas, una fecha de compilación vinculada al binario y la asociación explícita de dicho archivo al perfil del usuario.

Estos artefactos de compatibilidad e inventario resultan especialmente valiosos en el contexto forense, ya que permiten demostrar a posteriori que el sistema ejecutó un binario concreto desde el perfil del usuario, incluso aunque el archivo original haya sido eliminado.

Figura 184

Registros de eventos ID 13 en Sysmon por ejecución del Binario

Figura 185

Registro de evento ID 5 en Sysmon por terminación de proceso

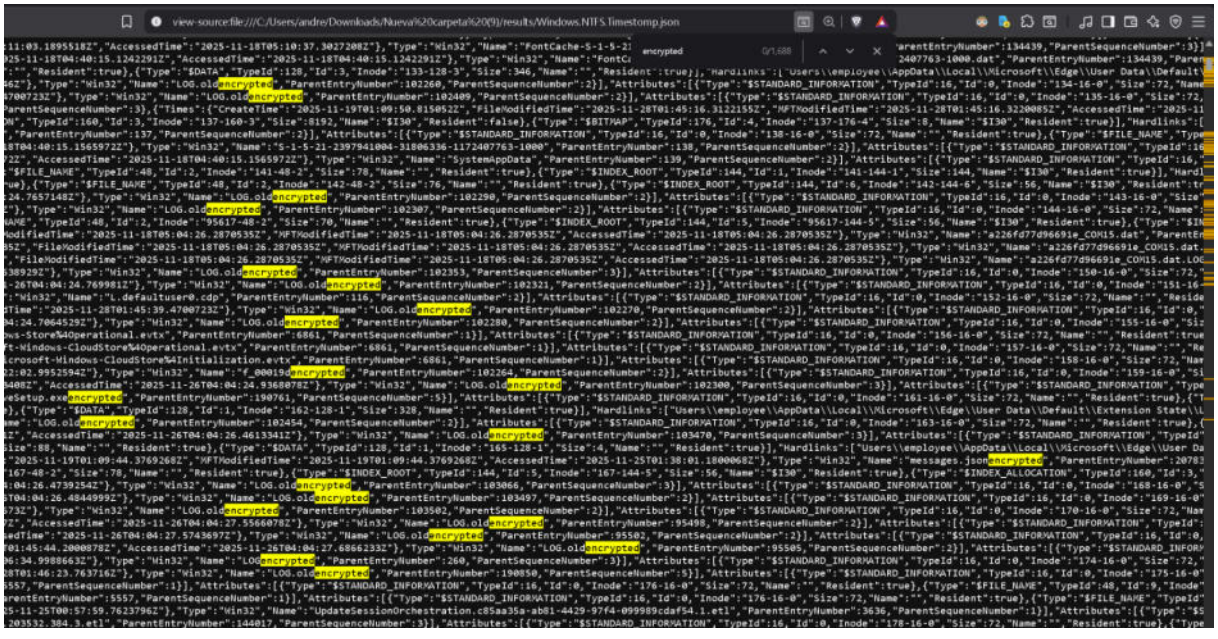
En paralelo a esta actividad se observan procesos legítimos del sistema operativo, como: taskhostw.exe, SecurityHealthHost.exe, dmclient.exe, Microsoft Edge y Windows Terminal, que constituyen ruido de fondo, propio de Windows y de la propia sesión de análisis.

Por tanto, el análisis del timeline con Sysmon no solo permite identificar de forma inequívoca el clic inicial del usuario sobre la muestra y la ventana temporal de ejecución del ransomware; sino también, separar esta actividad maliciosa de la actividad normal del sistema, aportando evidencia sólida para la definición de nuestro perfil mínimo de detección (PMD) basado en telemetría y artefactos nativos de Windows.

4.2.3.2. Windows.NTFS.Timestomp.json. El artefacto Windows.NTFS.Timestomp extraído por Velociraptor analiza metadatos NTFS a nivel de MFT, comparando las marcas temporales de los atributos $STANDARD_INFORMATION (SI) y $FILE_NAME (FN) de cada fichero. Su objetivo principal es identificar posibles técnicas de timestomping (manipulación de timestamps con fines anti-forenses), detectando discrepancias anómalas entre ambos conjuntos de tiempos.

Figura 186

Registro de eventos en artefacto Windows.NTFS.Timestomp extraído por Velociraptor



En la ejecución analizada del ransomware Black Basta, este artefacto permitió identificar múltiples archivos cuyo nombre fue modificado añadiendo el sufijo “encrypted”; por ejemplo, ficheros de bases de datos y logs de Microsoft Edge en C:\Users\employee\...

Además, el análisis de las marcas temporales mostró que:

- Los archivos existían y eran utilizados de forma legítima desde días previos al incidente (con CreateTime y FileModifiedTime antiguos en $FILE_NAME).
- Durante la ventana de ejecución del ransomware, el campo FileModifiedTime de $STANDARD_INFORMATION se actualizó a la fecha y hora del cifrado, mientras que los metadatos de nombre en $FILE_NAME conservaron tiempos anteriores.

Este patrón indica que la muestra de Black Basta realiza un cifrado “in situ”, es decir, sobrescribe el contenido de los archivos manteniendo la misma entrada de Master File Table (MFT), en lugar de crear nuevos ficheros.

Asimismo, se indica que en esta ejecución no se observaron evidencias de timestomping, ya que no se detectaron timestamps artificialmente retrodatados en $STANDARD_INFORMATION respecto a $FILE_NAME.

4.2.3.3. Windows.Packs.Persistence%2FStartup Items.json. Este artefacto enumera todos los elementos configurados para ejecutarse automáticamente al inicio del sistema; por ejemplo: clave Run, RunOnce, RunNotification, carpetas Startup y sus variantes por usuario y por máquina. Estos puntos de persistencia son indicadores típicos de malware, por lo que su revisión es esencial al analizar un ataque como Black Basta.

Figura 187

Registro de eventos en artefacto Windows.Packs.Persistence%2FStartup Items

En este caso, el artefacto devuelve 14 entradas, todas correspondientes a software legítimo del sistema, OneDrive, Microsoft Edge o VirtualBox. Es importante destacar que NO se observa ninguna persistencia del ransomware Black Basta. Esto es consistente con el comportamiento típico de Black Basta, en la mayoría de las muestras no intenta persistir, debido a que:

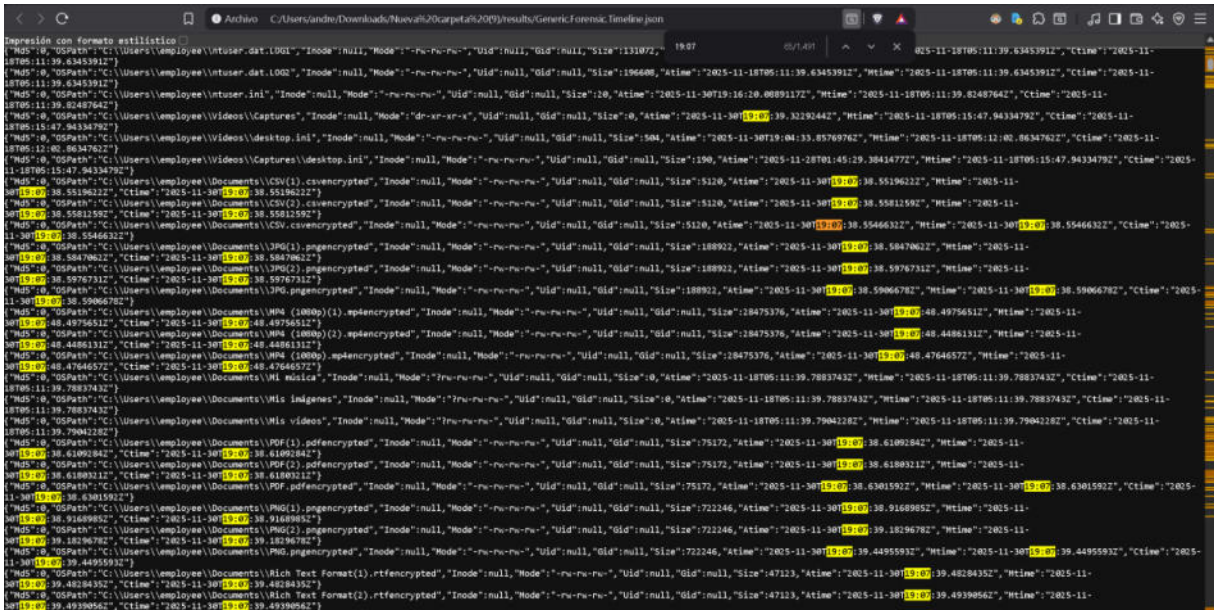
- Es un ransomware “smash-and-grab” orientado a cifrar lo más rápido posible.
- Frecuentemente es desplegado manualmente tras compromiso inicial, no necesita persistencia.
- Su operación termina en minutos y carece de mecanismos de autoinicio.

Por todo lo mencionado, se puede concluir que la ausencia de persistencia es un indicador del modus operandi de Black Basta. Su objetivo es una ejecución puntual, sin necesidad de mantenerse residente, lo cual se alinea con ataques manuales por operadores humanos.

4.2.3.4. Generic.Forensic.Timeline.json. Tras analizar el timeline de este artefacto, se evidencia cientos de entradas con archivos que fueron modificados exactamente en el mismo segundo, alrededor de: 2025-11-30 19:07:38 – 19:07:50 UTC.

Figura 188

Registro de eventos en artefacto Generic.Forensic.Timeline



Todos los tipos de archivos son renombrados agregando el sufijo “encrypted” y los timestamps; Mtime, Ctime y Atime, son idénticos. Esto significa que no hubo modificación paulatina, sino que la acción fue instantánea, típica del cifrado en bloque. Asimismo, la modificación la hizo un proceso automatizado, no el usuario, ya que el timeline muestra miles de archivos cifrados en una ventana de pocos segundos.

Lo mencionado en el párrafo anterior, evidencia un proceso automatizado que recorre directorios de usuario, cifra en paralelo, y que tiene un impacto transversal en tipos de archivos:

- .txt
- .pdf
- .png
- .jpg
- .mp4
- Bases de datos internas (.jtx, .jcp)

- Archivos de Edge, WinRAR, url shortcuts, LNK, etc.

A su vez, se observa que el ransomware cifra incluso accesos directos (.lnk).

Esto es una firma de comportamiento clara, pues los ransomware modernos cifran accesos directos para provocar pérdida de funcionalidad en los equipos, daños visibles al usuario e incremento del impacto psicológico.

Gracias a la información de este artefacto, se puede concluir que el proceso malicioso recorre directorios estándar del perfil de usuario y cifra todos los archivos independientemente del tipo, incluyendo archivos de sistema de usuario como accesos directos y bases de datos internas.

4.2.3.5. Windows.Forensics.Lnk.json. El análisis de los accesos directos recientes en Windows.Forensics.Lnk confirmó que la muestra de ransomware no generó accesos directos maliciosos, ni intentó establecer mecanismos de persistencia, lo cual es coherente con variantes modernas que ejecutan el cifrado de forma inmediata y finalizan sin dejar rastros duraderos.

Figura 189

Registro de eventos en artefacto Windows.Forensics.Lnk

4.2.4. Identificación de Evidencias Adicionales y Verificación de Integridad.

Recapitulando, en esta investigación se ejecutó una primera muestra de Black Basta en un entorno aislado de Windows, con el fin de recopilar diversos artefactos.

A partir de esta etapa se registran la información recopilada tras una nueva ejecución del malware, para la obtención de evidencias adicionales que aporten a la investigación del comportamiento y posterior creación del PMD (Perfil Mínimo de Detección). Entre las evidencias se incluyen archivos de registro de eventos (EVTX), capturas de actividades del sistema mediante Procmon (PML) y un volcado de memoria completo (RAW) obtenido con WinPmem.

Para asegurar la cadena de custodia y garantizar que los archivos no fueron afectados o modificados tras su extracción, se procedió a calcular el hash SHA-256 de cada uno.

A continuación, se muestra una tabla con los archivos EVTX analizados, y otras evidencias .pml, junto con el valor hash de cada uno.

Tabla 4

Lista de Hash SHA-256 de Archivos EVTX del primer directorio

N.º	Archivo	Directorio	SHA-256
1	Microsoft-Windows-Sysmon_Operational.evtx	evtx_20251114_150238	0e29c1def196bfd88db689ec2bca2b2402602949d8167ff290d9c5320bf6833a7d3b57437574e16da3b80457214ce05
2	Microsoft-Windows-PowerShell_Operational.evtx	evtx_20251114_150238	5a193a320372c57ecb19f40811421e225
3	Windows PowerShell.evtx	evtx_20251114_150238	5ced1290d59f396aad798726230f09bbb9651c196b5624b42fbef7a60a39b5f0

4	Security.evtx	evtx_20251114_150238	5fc0ece1ac2469b3bccd548d021c4eca454934ec458084a7ff0d286eebcdcb28
5	System.evtx	evtx_20251114_150238	67546bc73f7848bb2137db24531f0a8037d7af55cbd3cb5e5293d792e85b887c
6	Microsoft-Windows-Windows Firewall With Advanced Security_Firewall.evtx	evtx_20251114_150238	f3bb934218d975b77b3daeebec8b1461e0553f69025eabbb9c56479c09a85631
7	Microsoft-Windows-SmbClient_Security.evtx	evtx_20251114_150238	b790c872a389511da32f190adf66f301283c86a7acbbbfcfe154c70858fae70d
8 8	Microsoft-Windows-SMBServer_Security.evtx	evtx_20251114_150238	b790c872a389511da32f190adf66f301283c86a7acbbbfcfe154c70858fae70d

Nota. En la primera columna se indexa el número del archivo, en la segunda y tercera columna se indica el nombre del archivo y su correspondiente directorio, en la última columna se muestra el Hash SHA-256.

Tabla 5

Lista de Hash SHA-256 de Archivos EVTX del segundo directorio

N.º	Archivo	Directorio	SHA-256
1	Microsoft-Windows-Sysmon_Operational.evtx	evtx_20251114_154547	dd15aa581814191bbb0b9fa8de366cb568074b4704ccd081666e0d330773a088

2	Microsoft-Windows-PowerShell_Operational.evtx	evtx_20251114_154547	7025cb4bc9d57389eefdbe68359ff4937bb6ceb825e3925cb16051194e3ebf4d
3	Windows PowerShell.evtx	evtx_20251114_154547	09764b1332ff4e77e16a8ee987fdafbf69b68c268a112a282e81e12198c16de5
4	Security.evtx	evtx_20251114_154547	d576ce79722b7b8b8b9f77a7185d8db0836699e3d267a9158c9a08de13656578
5	System.evtx	evtx_20251114_154547	f9ba6bdf7f24d37210e329279bfb84415cd624612b23edbbf69f5609611afc3c
6	Microsoft-Windows-Windows Firewall With Advanced Security_Firewall.evtx	evtx_20251114_154547	f3bb934218d975b77b3daeebec8b1461e0553f69025eabbb9c56479c09a85631
7	Microsoft-Windows-SmbClient_Security.evtx	evtx_20251114_154547	4f930eafdc7b9ecfbf0b76bc054b3802826508d132869a6eb00dc1b49a466298
8	Microsoft-Windows-SMBServer_Security.evtx	evtx_20251114_154547	b790c872a389511da32f190adf66f301283c86a7acbbbfcfe154c70858fae70d

Nota. En la primera columna se indexa el número del archivo, en la segunda y tercera columna se indica el nombre del archivo y su correspondiente directorio, en la última columna se muestra el Hash SHA-256.

Tabla 6

Lista de Hash SHA-256 de Archivos PML de Otras evidencias recopiladas

N.º	Archivo	Tipo	SHA-256
1	procmon_20251114_121545-1.pml	Procmon Log	2cf136ca7e4ec8fa98ab9ee729de470662fa7b4690ec45368470d4763911636d
2	procmon_20251114_121545.pml	Procmon Log	527129616af9b5b0a848550095fe774a01609989b07082c554809e0fcd644315
3	winpmem_20251114_123536.raw	Volcado de Memoria	296cf640111a015b6ae57a3f5d60e67c9ec8a1e2390ad7c2086e36c3e7d4cf79

Nota. En la primera columna se indexa el número del archivo, en la segunda y tercera columna se indica el nombre del archivo y su tipo, en la última columna se muestra el Hash SHA-256.

4.2.5. Análisis de Archivos EVTX. Los archivos EVTX aportan información crítica que permite reconstruir la actividad del sistema durante y después de la ejecución de Black Basta. Su análisis tiene como objetivo principal:

- Determinar los procesos creados, modificados o finalizados.
- Detectar la ejecución de scripts o comandos (CMD / PowerShell).
- Identificar accesos a recursos del sistema.
- Verificar creación de servicios, modificaciones de firewall y cambios en políticas.
- Identificar actividad relacionada con Sysmon (si está presente).
- Correlacionar acciones típicas de Black Basta, como carga del payload, desactivación de mecanismos de seguridad, enumeración de sistema, movimientos laterales (si aplicara), ejecución del módulo de cifrado.

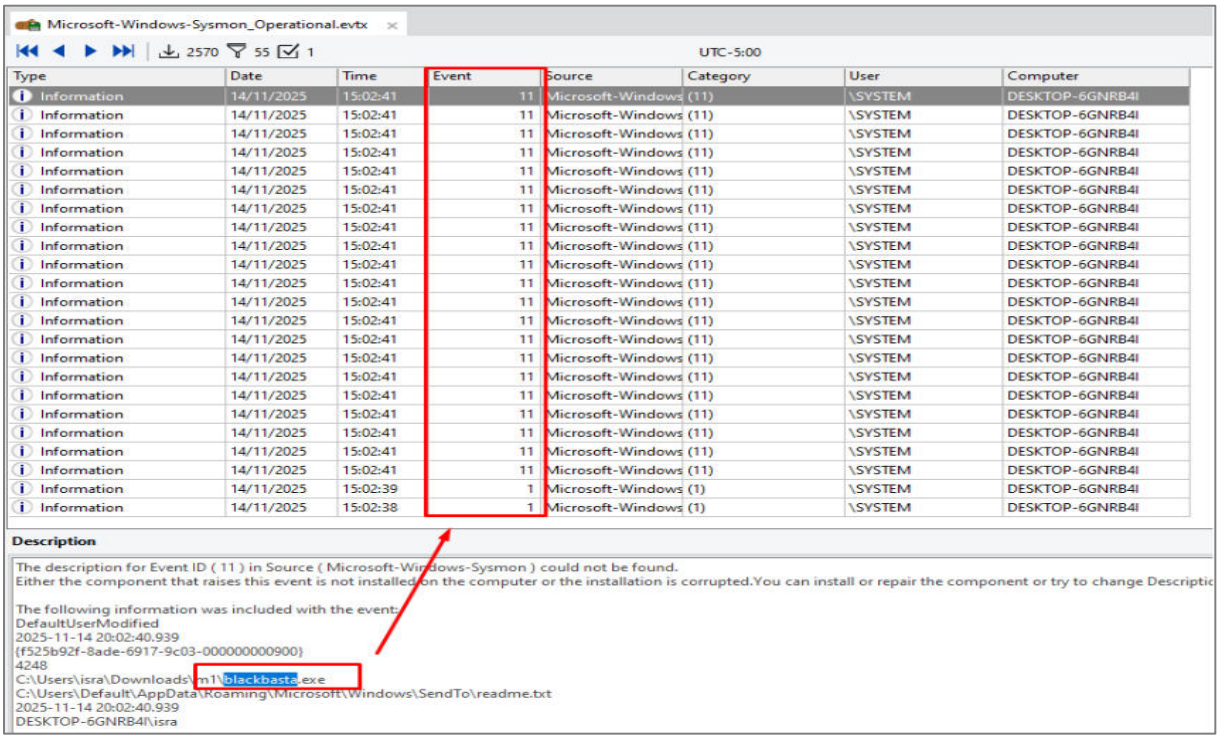
Esta etapa facilitará la elaboración de la línea de tiempo inicial y ayudará a establecer artefactos volátiles deben buscarse más adelante en el volcado de memoria.

4.2.5.1. Microsoft-Windows-Sysmon/Operational.evtx. Tras revisar el archivo Microsoft-Windows-Sysmon/Operational.evtx, se identificaron varios eventos relevantes típicos del comportamiento del ransomware Black Basta durante su fase inicial de ejecución y cifrado. Los eventos corresponden a la creación de procesos (Event ID 1) y creación de archivos (Event ID 11), por lo que, el presente análisis se limita a estos dos tipos de eventos.

En la **Figura 190**, se muestra información obtenida a raíz del análisis de Microsoft-Windows-Sysmon/Operational.evtx con hash: *dd15aa581814191bbb0b9fa8de366cb568074b4704ccd081666e0d330773a088,* pero filtrada buscando strings que contengan el proceso blackbasta.exe.

Figura 190

Eventos ID 1 y 11 capturados por herramienta Sysmon generados por Black Basta



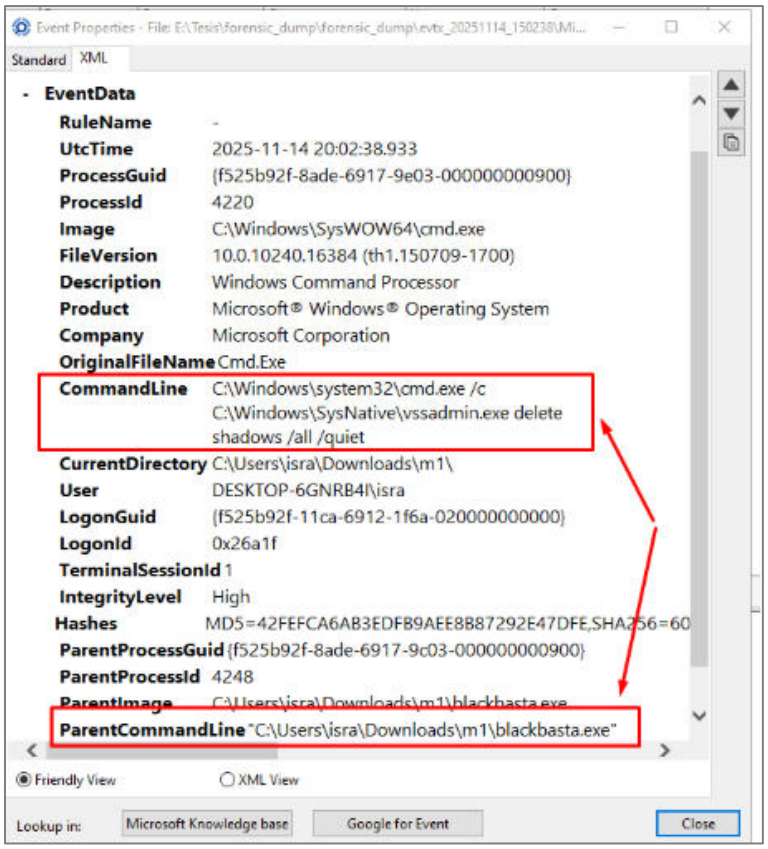
Sysmon registró la ejecución secuencial de los siguientes procesos:

- cmd.exe ejecutado desde C:\Windows\SysWOW64\cmd.exe
- Comando interno C:\Windows\SysNative\vssadmin.exe delete shadows /all /quiet
- Ejecución directa del binario malicioso en la ruta

C:\Users\isra\Downloads\m1\blackbasta.exe

Figura 191

Revisión del detalle de los eventos generados



Los registros incluyen la siguiente información:

- Ruta completa del ejecutable.
- Hashes MD5, SHA256 e IMPHASH del binario (que facilitan identificación y comparación).

- Usuario que realizó la ejecución.
- PID y ParentPID.
- Línea de comandos exacta.

Segundos después de la ejecución inicial del binario, a las 20:02:38 (UTC), se registraron dos eventos consecutivos de creación de procesos (Event ID 1). En ambos casos, el proceso padre blackbasta.exe invocó al intérprete de comandos “cmd.exe” para ejecutar la utilidad nativa “vssadmin.exe”. El comando ejecutado es el siguiente: `vssadmin.exe delete shadows /all /quiet`

Además, se logra identificar un comportamiento redundante diseñado para asegurar la ejecución en arquitecturas de 64 bits:

- **Primer intento:** Ejecución a través de C:\Windows\SysNative\vssadmin.exe
- **Segundo intento:** Ejecución a través de C:\Windows\SysWOW64\cmd.exe

Esta dualidad busca evadir la redirección del subsistema WOW64, asegurando que las *Volume Shadow Copies* (instantáneas de recuperación) sean eliminadas independientemente de si el entorno operativo interpreta la llamada como 32 o 64 bits.

Todo lo antes mencionado, evidencia la intención del malware de obstaculizar la recuperación forense simple del sistema.

A continuación, se muestra una tabla donde se enlista la relación de este comportamiento con la matriz MITRE ATT&CK.

Tabla 7

Relación del comportamiento observado (eliminación de Shadow copies) con MITRE ATT&CK

Técnica	Descripción
T1059.003 – Command Execution (cmd.exe)	El malware utiliza cmd.exe como proxy para ejecutar comandos.
T1490 – Inhibit System Recovery	vssadmin.exe delete shadows es una técnica directa para inutilizar mecanismos de recuperación.
T1204.002 – User Execution / Malicious File	El malware requiere ser ejecutado manualmente por el usuario.

Nota. En la primera columna se encuentra el nombre de la técnica, y en la segunda, una breve descripción de la relación encontrada.

La revisión de los registros de Sysmon revela entradas tales como:

- **blackbasta.exe** → C:\ProgramData\Microsoft\Windows\Start Menu\readme.txt
- **blackbasta.exe** → C:\Users\Default\readme.txt

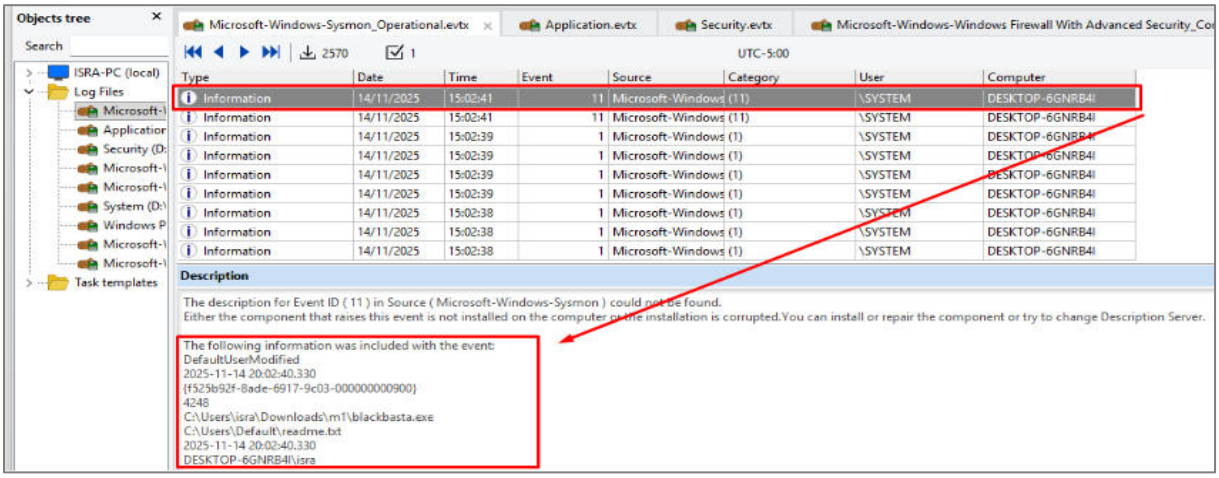
Los eventos reflejan los siguientes parámetros:

- Proceso responsable: blackbasta.exe
- Ruta donde se crean los archivos.
- Timestamp exacto.
- Usuario que ejecutó el proceso.

En la **Figura 192**, se observa que dentro de los eventos de Sysmon se registra la creación de notas de rescate y archivos asociados al cifrado bajo Event ID 11 – File Create, un comportamiento característico tras realizar el cifrado.

Figura 192

Registros de la creación de notas de rescate obtenidas de Sysmon



Las rutas encontradas son consistentes con las utilizadas por Black Basta:

- Directorios accesibles para todos los usuarios (ProgramData).
- Directorios de perfiles (C:\Users\Default\).

Esto indica que el malware busca garantizar que el mensaje sea visible en distintos contextos de usuario.

La presencia reiterada de eventos de creación del mismo archivo en distintos directorios, indica que el proceso estuvo replicando la nota de rescate a lo largo del sistema.

A continuación, se muestra una tabla donde se enlista la relación de este comportamiento con la matriz MITRE ATT&CK.

Tabla 8

Relación del comportamiento observado (creación de notas de rescate) con MITRE ATT&CK

Técnica	Descripción
T1486 – Data Encrypted for Impact	La creación de notas de rescate ocurre típicamente después del cifrado.

T1070 – Indicator Removal (parcial)	Algunos ransomware sobrescriben o eliminan archivos durante el proceso; aquí se observa solo la fase de escritura.

Nota. En la primera columna se encuentra el nombre de la técnica, y en la segunda, una breve descripción de la relación encontrada.

Para concluir, el análisis con Sysmon permitió evidenciar lo siguiente:

- La ejecución directa del binario malicioso.
- La eliminación de shadow copies mediante vssadmin.exe.
- La creación de notas de rescate en múltiples rutas del sistema.

Estos eventos muestran un flujo táctico típico del ransomware Black Basta: preparación, destrucción de mecanismos de recuperación, cifrado, e inserción del mensaje de extorsión.

Las evidencias recolectadas serán correlacionadas más adelante con los artefactos adicionales obtenidos (PML, VTX, RAW) para completar la trazabilidad del incidente.

Con respecto a los otros archivos .evtx, no se encontró información relacionada con Black Basta ni información relevante para este análisis.

4.2.6. Análisis de Archivos PML. Aunque todavía no se ha iniciado el análisis del volcado de memoria, estos componentes forman parte integral del proceso forense:

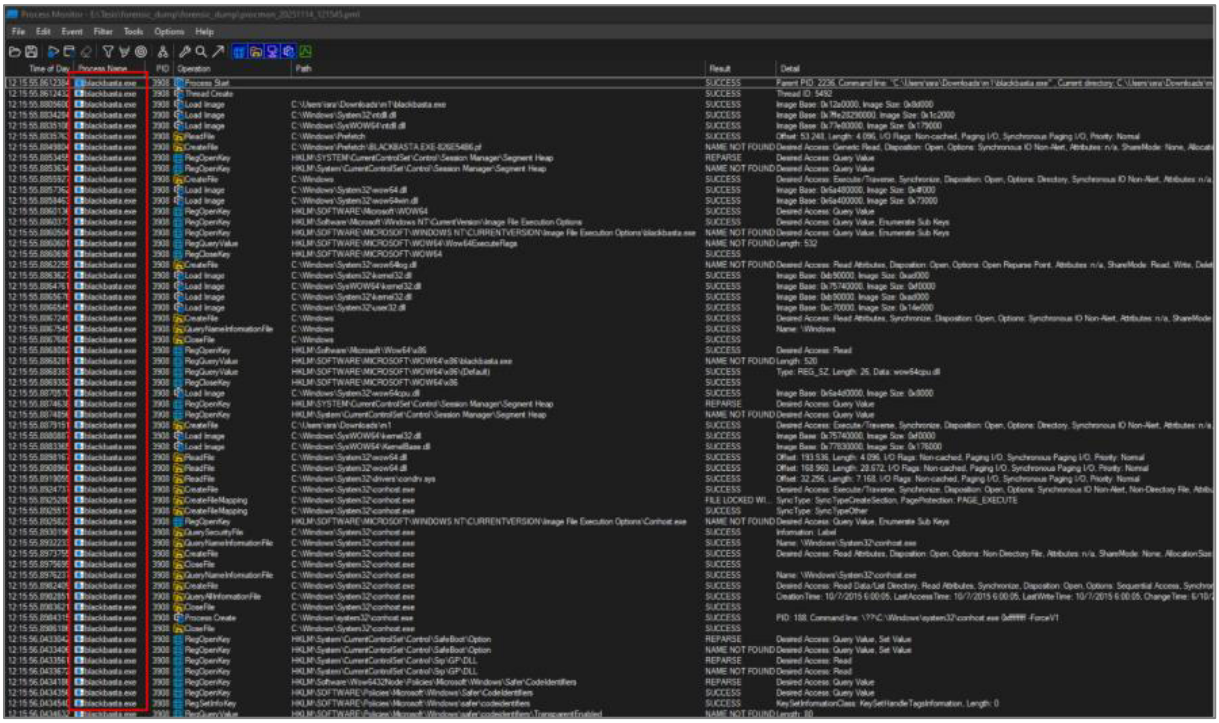
- Los archivos PML (2.59 GB y 3.85 GB) contienen el registro completo de llamadas del sistema durante la ejecución del malware.
- El archivo RAW obtenido con WinPmem (2 GB) representa el estado completo de la RAM, permitiendo investigar procesos activos, hilos, módulos cargados, claves de cifrado, conexiones de red, artefactos residuales de inyección y actividad anti-forense.

Estos elementos mencionados, se analizarán posteriormente para complementar la información obtenida desde los EVTX.

A continuación, en la **Figura 193**, se muestra una vista general del archivo procmon_20251114_121545.pml cargado en Procmon, aplicando filtro "*Process Name is blackbasta.exe"*.

Figura 193

Análisis de archivo procmon_20251114.pml filtrado por blackbasta.exe



4.2.6.1. Adquisición de Evidencia Volátil. Para adquirir evidencia volátil, se implementó una metodología de ejecución controlada con recolección automatizada, dada la naturaleza destructiva de la muestra de ransomware Black Basta y su capacidad para inutilizar registros del sistema en segundos.

El procedimiento consistió en la detonación manual del binario malicioso en el entorno aislado LAB-BLACK, descrito en esta sección a través de fases.

Inmediatamente después de la ejecución del malware, se activaron scripts de recolección en PowerShell diseñados para volcar y exfiltrar la memoria, y los registros de Process Monitor hacia el servidor seguro, justo antes de que el cifrado comprometiera la integridad del sistema operativo.

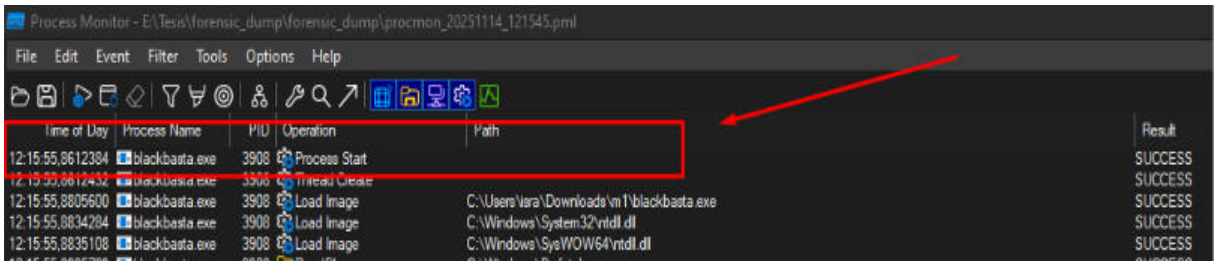
Esta estrategia de respuesta rápida justifica las marcas de tiempo observadas en el registro procmon_20251114_121545.pml analizado en Procmon, el cual captura los primeros **1.7 segundos críticos** de la infección (de 12:15:55 a 12:15:57), momento en el cual se detuvo la captura para garantizar su preservación.

- **Fase 1: Inicialización del Proceso (T+0s).-** El análisis forense inició con la identificación del proceso padre, aplicando el filtro en Process Monitor *"Operation is Process Start"*. Se aisló el evento inicial a las **12:15:55.861**.

Los detalles del evento confirman que la muestra se ejecutó desde el directorio de descargas (C:\Users\isra\Downloads\m1\blackbasta.exe), aprovechando las variables de entorno del sistema comprometido, lo que le facilita operar utilizando las credenciales del usuario en sesión.

Figura 194

Análisis de archivo procmon_20251114_121545.pml filtrado por Process Start



Como se muestra en la **Figura 194**, el inicio de ejecución de blackbasta.exe (Process Start) se encuentra en la línea 0 (Time: 12:15:55.8612384), con PID 3908.

- **Fase 2: Evasión de Defensa y Destrucción de Respaldos (T+0.2s).-** A 200 milisegundos de iniciarse la ejecución, el malware inició su rutina de evasión. Con el propósito de localizar esta actividad, se utilizó el filtro *"Detail contains vssadmin"* para búsqueda de palabras clave en los argumentos de comando. El log revela una técnica de redundancia diseñada para evadir las protecciones de arquitectura de Windows (WOW64).

También, se observaron dos llamadas consecutivas para eliminar las *Volume Shadow Copies*, la primera a las **12:15:56.088,** ejecución vía C:\Windows\SysNative\vssadmin.exe (Acceso directo a binarios de 64-bits); la segunda a las **12:15:56.462,** ejecución vía C:\Windows\System32\vssadmin.exe (Ruta estándar).

Esta doble verificación confirma que Black Basta no asume la arquitectura del sistema, sino que ataca ambas posibilidades para asegurar que el daño sea irreversible.

Figura 195

Análisis de archivo procmon_20251114_121545.pml filtrado por Detail contains vssadmin

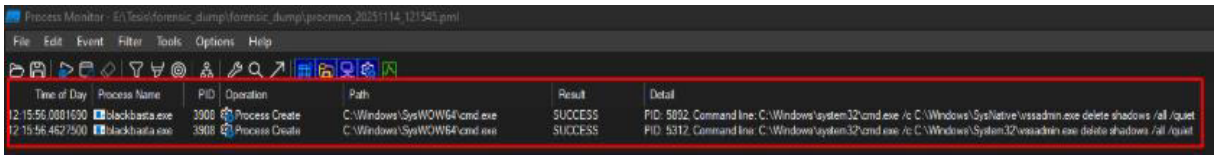
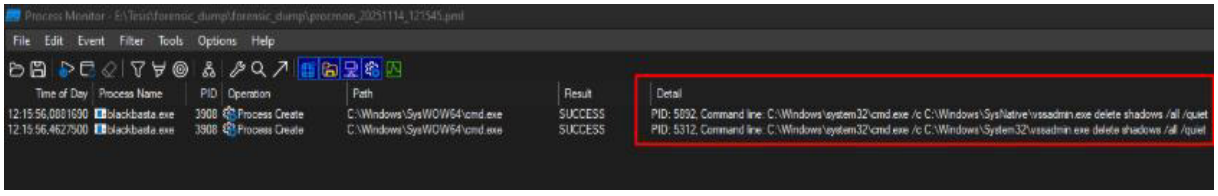


Figura 196

Análisis de archivo procmon_20251114_121545.pml llamadas para eliminar Volume shadow Copies



En la **Figura 196**, se evidencia la sofisticación del malware al usar rutas SysNative para saltar la redirección de archivos de 32 bits.

- **Fase 3: Preparación del Entorno (Staging).-** El malware prepara su escenario para la extorsión extrayendo recursos gráficos antes del cifrado. Para detectar esta actividad de escritura en directorios temporales se utilizó el siguiente filtro: *"Path contains Temp AND Operation is WriteFile"*.

Como resultado, se logró identificar la creación de dos artefactos esenciales; el primero a las **12:15:56.866** cuando crea la imagen de fondo de escritorio con la nota de rescate (dlaksjdoiwq.jpg); el segundo, a las **12:15:56.882**, cuando crea el icono personalizado para los archivos cifrados (fkdjsadasd.ico). El hallazgo de estos archivos en el directorio %TEMP% demuestra que el malware "prepara" el escenario del crimen antes de que la víctima perciba el cifrado de sus archivos.

Figura 197

Análisis de archivo procmon_20251114_121545.pml filtrado por Temp y por WriteFile

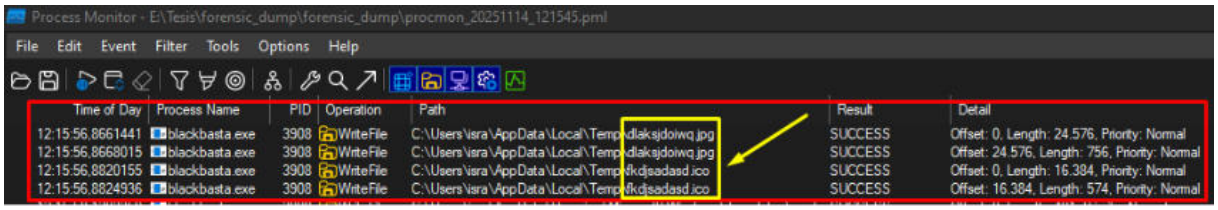
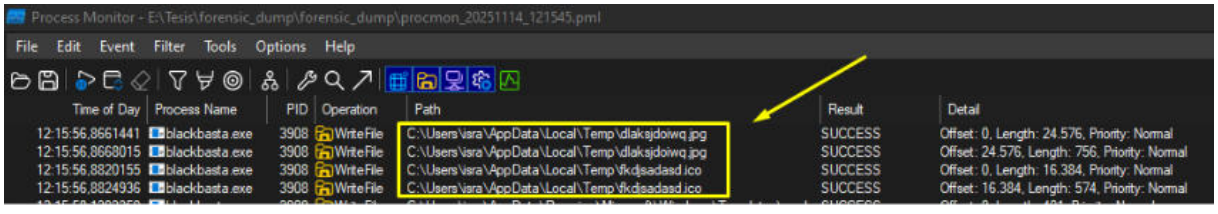


Figura 198

Análisis de archivo procmon_20251114_121545.pml filtrado por Temp



En la **Figura 198**, se ha señalado las líneas en Procmon donde el Path termina en .jpg o .ico. Se comprueba la existencia del "dropper" interno del malware.

- **Fase 4: Notificación Masiva (readme.txt).-** Cuando empezó la corrupción de datos, se identificó un comportamiento de propagación masiva y sostenida de la nota de rescate readme.txt. El análisis profundo de los registros de actividad confirmó que esta fase no fue un evento puntual, sino un proceso continuo paralelo a la rutina de cifrado.

El subproceso encargado del despliegue de notas operó de manera ininterrumpida durante 4 minutos y 30 segundos, abarcando la totalidad de la ventana de infección activa. **Tiempo Inicio:** 12:15:57.515; **Tiempo Fin:** 12:20:27.933; **Duración Total:** aprox. 270 segundos.

Esta duración extendida indica que Black Basta emplea una arquitectura multihilo (multithreading), donde uno o más hilos se dedican exclusivamente a recorrer la estructura del sistema de archivos (File System Traversal) para distribuir las instrucciones de extorsión, mientras que otros hilos independientes ejecutan las operaciones de I/O intensivas (Lectura/Cifrado/Escritura) sobre los archivos de datos.

Durante este periodo, el malware intentó escribir indiscriminadamente en la raíz del sistema (C:\) y en directorios de aplicaciones profundos (Program Files, AppData). El análisis de los resultados de la operación *"CreateFile"* revela un comportamiento de fuerza bruta.

Figura 199

Análisis de resultados del archivo procmon_20251114_121545.pml tras creación de readme.txt

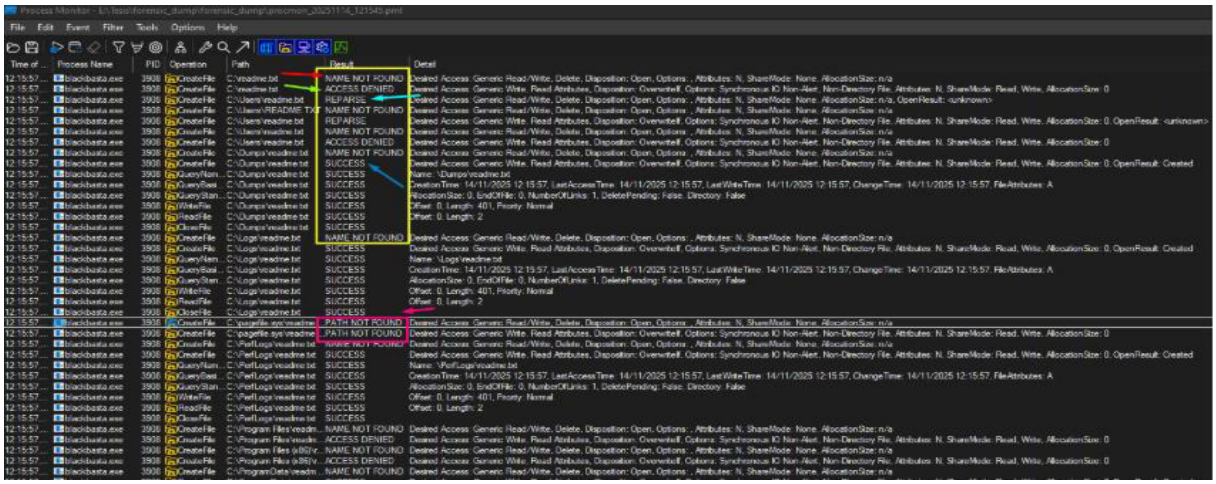
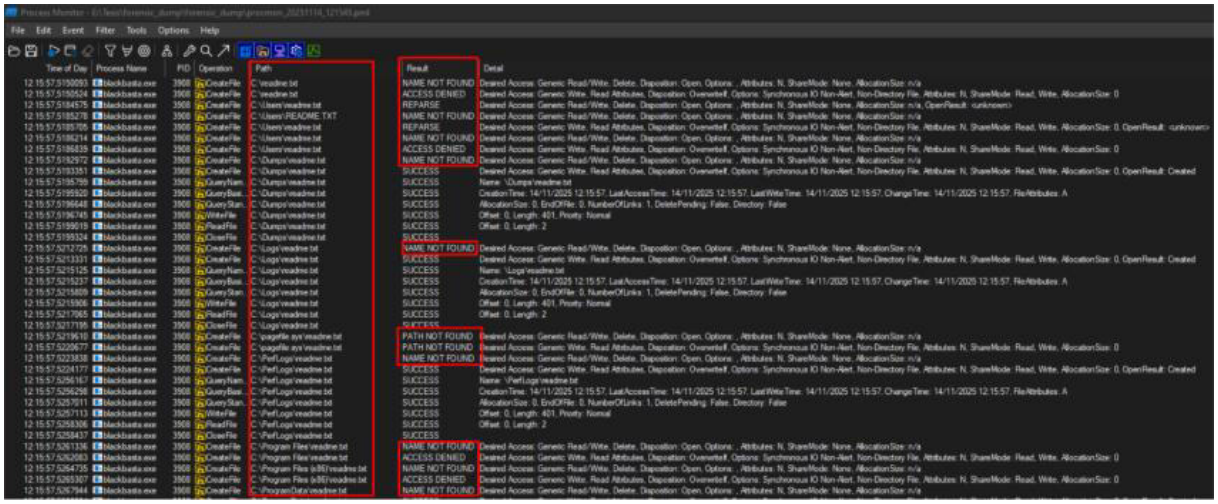


Figura 200

Actividad sostenida de creación de notas de rescate (readme.txt) paralela al proceso de cifrado



Como se visualiza en la **Figura 199** y **Figura 200**, en la columna Result de los procesos en lista se encuentran resultados tipo **“NAME NOT FOUND”, “REPARSE”, “ACCESS DENIED”, “SUCCESS”, “PATH NOT FOUND”;** a continuación, se explica brevemente lo que representa cada uno.

NAME NOT FOUND o Verificación de Existencia, es un resultado predominante antes de cada creación exitosa, e indica que el malware consulta primero si el archivo readme.txt ya existe en el directorio destino. Tras recibir la confirmación del sistema de que el nombre no se encuentra, procede a su creación. Esto confirma que el malware no sobrescribe sus propias notas, optimizando operaciones de I/O.

REPARSE, indica que el algoritmo de recorrido del malware interactúa con Puntos de Análisis (Reparse Points) como enlaces simbólicos o directorios especiales del sistema de archivos NTFS, por ejemplo, carpetas de compatibilidad como C:\Documents and Settings que redirigen internamente a C:\Users. El malware intenta acceder a estas rutas heredadas, lo que

demuestra un barrido exhaustivo de la estructura lógica del disco, sin discriminar entre directorios físicos y enlaces virtuales.

ACCESS DENIED, esto indica la resiliencia a permisos, es decir que se registraron múltiples bloqueos de seguridad al intentar escribir en directorios protegidos por el sistema, tales como C:\Program Files\MsEdgeCrashpad. El malware ignora estas excepciones y continúa su ejecución, confirmando un diseño de fuerza bruta orientado a cubrir la mayor superficie posible, sin detenerse por errores de privilegios.

SUCCESS, este resultado indica un compromiso exitoso en la escritura efectiva de la nota de rescate en directorios con permisos de escritura para el usuario comprometido (C:\Users, C:\ProgramData, C:\Temp).

PATH NOT FOUND a diferencia de NAME NOT FOUND evidencia un comportamiento de ataque basado en diccionarios o listas estáticas. Cuando el sistema devuelve como resultado “Ruta no encontrada” indica que el malware no verificó la existencia del directorio antes de atacar; sino que simplemente ejecutó una rutina de “disparar y olvidar” contra una lista de rutas de alto valor, con la intención de dejar la nota de rescate en carpetas que un administrador de sistemas revisaría durante la respuesta al incidente.

Con todo este análisis, se logra comprobar que el despliegue de notas es un proceso continuo y no solo una etapa inicial.

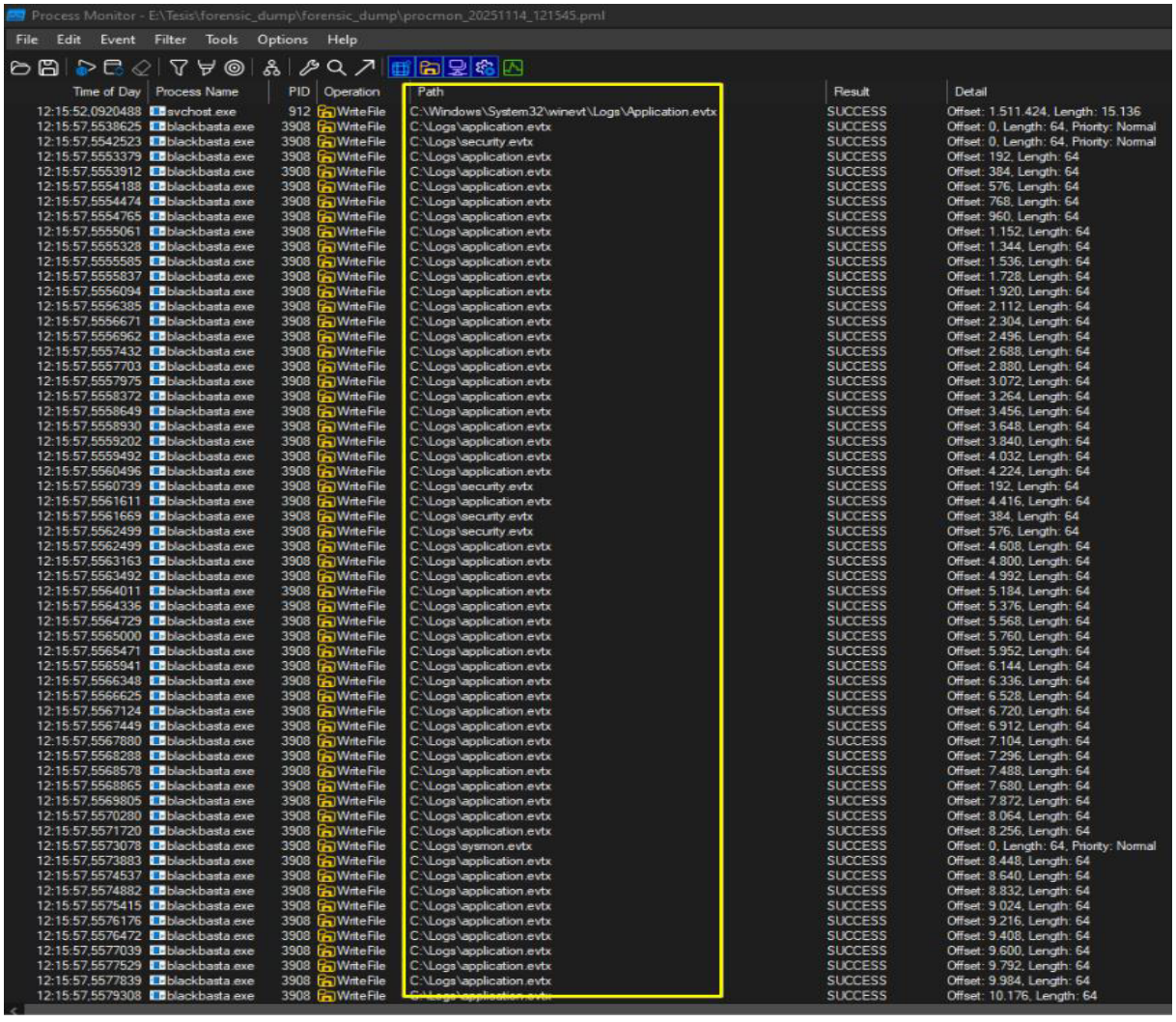
- **Fase 5: Inicio del Cifrado y Actividad Anti-Forense (T+1.6s).-** El punto clave de la investigación fue localizar el “Tiempo Cero” del cifrado. Se esperaba que el cifrado inicie con los documentos de usuario; sin embargo, el malware empezó atacando los registros de auditoría. Para esta fase, se usó el filtro *“Path contains .evtx AND Operation is WriteFile”*.

Como resultado, se obtuvo que a las **12:15:57.553** se registró la primera operación de escritura destructiva (Offset: 0, Length: 64) sobre el archivo C:\Logs\application.evtx.

Este hallazgo demuestra que Black Basta prioriza la corrupción de evidencias forenses (.evtx) siempre que se encuentren en directorios accesibles (C:\Logs\), antes de iniciar con el cifrado masivo de documentos. Este comportamiento refuerza la necesidad crítica de haber realizado la extracción de logs mediante scripts en tiempo real, ya que la evidencia local fue comprometida en menos de 2 segundos.

Figura 201

Análisis del archivo procmon_20251114_121545.pml por corrupción de evidencias



Al buscar en Procmon la línea correspondiente a la marca de tiempo **12:15:57.553,** se visualiza la operación WriteFile en application.evtx, lo cual indica actividad destructiva y anti-forense.

Para concluir, en esta sección se han expuesto hallazgos extraídos del artefacto procmon_20251114_121545.pml y como parte del protocolo de investigación también se ejecutaron capturas adicionales (procmon_20251114_121545-1.pml) en una ventana de tiempo posterior.

La comparación entre ambos registros confirmó que la muestra Black Basta sigue un comportamiento algorítmico determinista. Las rutinas de evasión, inyección y cifrado se replicaron de forma idéntica en ambas ejecuciones. Por este motivo, y para evitar redundancia técnica en la exposición de resultados, se ha optado por documentar exclusivamente ejecuciones y registros de evidencias sobresalientes, pero mencionando la realización de pruebas adicionales únicamente como constancia de validación del comportamiento del malware.

4.2.7. Correlación de Hallazgos. Gracias a la integración de los registros de granularidad fina de Procmon, se permitió profundizar y explicar técnicamente los eventos de alto nivel previamente identificados en el registro Sysmon. La correlación de evidencias permite establecer las siguientes conclusiones definitivas:

4.2.7.1. Identificación de Estrategia de Evasión. Sysmon identificó la ejecución de vssadmin.exe y Procmon reveló que esta ejecución no es lineal, sino una rutina redundante de doble invocación (SysNative y System32). Esto demuestra que el ransomware posee mecanismos de conciencia de arquitectura para asegurar la destrucción de respaldos tanto en entornos de 32 como de 64 bits.

4.2.7.2. Descubrimiento de la Fase de "Staging". Sysmon registró la actividad principal del proceso; pero Procmon mostró los milisegundos previos al ataque, descubriendo la extracción de artefactos visuales (.jpg, .ico) en %TEMP%. Esto confirma que la preparación del impacto psicológico ocurre en paralelo a la preparación técnica.

4.2.7.3. Objetivo de Cifrado. Sysmon advirtió sobre la creación de notas de rescate y Procmon estableció que el "Tiempo Cero" de cifrado no comenzó con documentos de usuario, sino con la corrupción intencional de archivos de auditoría (.evtx) ubicados en rutas locales (C:\Logs). De esta manera, se entiende que el malware identifica y destruye evidencias activamente antes de cifrar datos personales.

4.2.7.4. Fuerza Bruta en Notas de Rescate. La herramienta Sysmon identificó la aparición de readme.txt en múltiples rutas; mientras que, el uso de Procmon expuso un algoritmo que opera mediante fuerza bruta durante un tiempo aproximado 4.5 minutos. Al observar valores en Result como "REPARSE", "ACCESS DENIED" y "PATH NOT FOUND" demuestran que el malware recorre indiscriminadamente directorios especiales y ataca rutas predefinidas de software de seguridad, priorizando la saturación del sistema.

4.2.7.5. Reconstrucción del Flujo Táctico. Con la evidencia cruzada, se ratifica el flujo operativo de Black Basta en un entorno controlado, caracterizado por una alta velocidad y notable resiliencia frente a errores de permisos. Se enlista, a continuación, un resumen de sus etapas:

- **Ejecución e inicialización.-** Detonación del binario y herencia de entorno.
- **Preparación y evasión.-** Drop de recursos gráficos y destrucción redundante de Shadow Copies.

- **Destrucción Anti-Forense y cifrado.-** Corrupción prioritaria de logs .evtx seguida del cifrado de datos de usuario.
- **Saturación.-** Despliegue paralelo y sostenido de la nota de extorsión recorriendo todo el árbol de directorios.

4.2.7.6. Alineación con el Marco MITRE ATT&CK. Basado en la evidencia forense recolectada mediante Sysmon y Process Monitor durante la ejecución controlada en LAB-BLACK, se ha procedido a mapear el comportamiento del ransomware Black Basta con la matriz de conocimiento MITRE ATT&CK .

Esta correlación permite traducir los artefactos técnicos observados (Comandos, Offsets, Códigos de error) en técnicas estandarizadas.

Tabla 9

Alineación del comportamiento observado con el marco MITRE ATT&CK

Táctica (Objetivo)	ID Técnica	Técnica	Procedimiento Observado (Evidencia en Logs)
Execution	T1059.003	Command and Scripting Interpreter: Windows Command Shell	El proceso padre blackbasta.exe instanció cmd.exe con el argumento /c para ejecutar herramientas nativas del sistema, delegando la tarea de evasión al intérprete de comandos.
Defense Evasion	T1211	Exploitation for Defense Evasion	El malware explotó la ruta virtual C:\Windows\SysNative para saltar la redirección del sistema WOW64 y

			ejecutar comandos de 64 bits desde un proceso de 32 bits.
Defense Evasion	T1070.001	Indicator Removal: Clear Windows Event Logs	Se registró la sobrescritura directa (WriteFile) de archivos sysmon.evtx y application.evtx en C:\Logs. A diferencia del borrado estándar (wevtutil), Black Basta corrompe la estructura del archivo para inutilizar la auditoría.
Discovery	T1083	File and Directory Discovery	El algoritmo de despliegue de notas (readme.txt) realizó un barrido exhaustivo del sistema de archivos durante 4.5 minutos, interactuando con Junction Points (Result "REPARSE") para descubrir la estructura lógica del disco.
Impact	T1490	Inhibit System Recovery	Ejecución redundante de vssadmin.exe delete shadows /all /quiet. La doble invocación garantiza la destrucción de instantáneas de recuperación antes del cifrado.
Impact	T1486	Data Encrypted for Impact	Confirmado por el bucle de operaciones de I/O y la preparación

			visual mediante el drop de íconos (.ico) y fondo de pantalla (.jpg) en %TEMP% para notificar visualmente el impacto.

Nota. En la primera columna se encuentra el nombre de la táctica; en la segunda columna, se indica el identificador de la técnica; en la tercera, el nombre de la técnica y en la cuarta columna, una breve descripción del procedimiento observado en logs.

4.2.8. Análisis de la Cobertura Táctica. La muestra evaluada concentró la mayoría de sus capacidades en tácticas de impacto y evasión de defensa. Por lo cual, es relevante destacar la técnica T1070 (Indicator Removal), ya que la variante estudiada no se limitó a borrar huellas después del ataque, sino que, atacó activamente los mecanismos de registro durante la fase de inicialización (T+1.6s). Este comportamiento descrito sugiere una evolución en la sofisticación del ransomware, buscando dejar "ciego" al equipo de respuesta tanto como cifrar los datos.

4.2.9. Análisis del Volcado de Memoria. Una vez completada la adquisición de la imagen de memoria cruda (MEM_1200ms.raw) en el servidor de recolección, se procedió al traslado de la cadena de custodia hacia la estación de análisis forense.

La segmentación de entornos en este laboratorio se realizó para garantizar el uso de recursos de procesamiento dedicados y aprovechar la compatibilidad nativa del framework Volatility 3 con los archivos de símbolos derivados de los PDB públicos de Microsoft Windows, los cuales permiten interpretar estructuras internas del sistema operativo durante el análisis de memoria.

4.2.9.1. Identificación y Aislamiento del Proceso (Triage). El primer paso del análisis consistió en validar la presencia del artefacto malicioso dentro del volcado de memoria; para

lograr esto, se utilizó el plugin *windows.pslist*, el cual recorre la lista doblemente enlazada EPROCESS del kernel para enumerar los procesos que se encontraban activos en el instante exacto de la suspensión. Con el objetivo de reconocer el identificador de proceso (PID) asignado al ransomware, se ejecutó el siguiente comando:

```
python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw" windows.pslist
```

Figura 202

Análisis del volcado de memoria en Volatility3 mediante el uso del plugin Windows.pslist

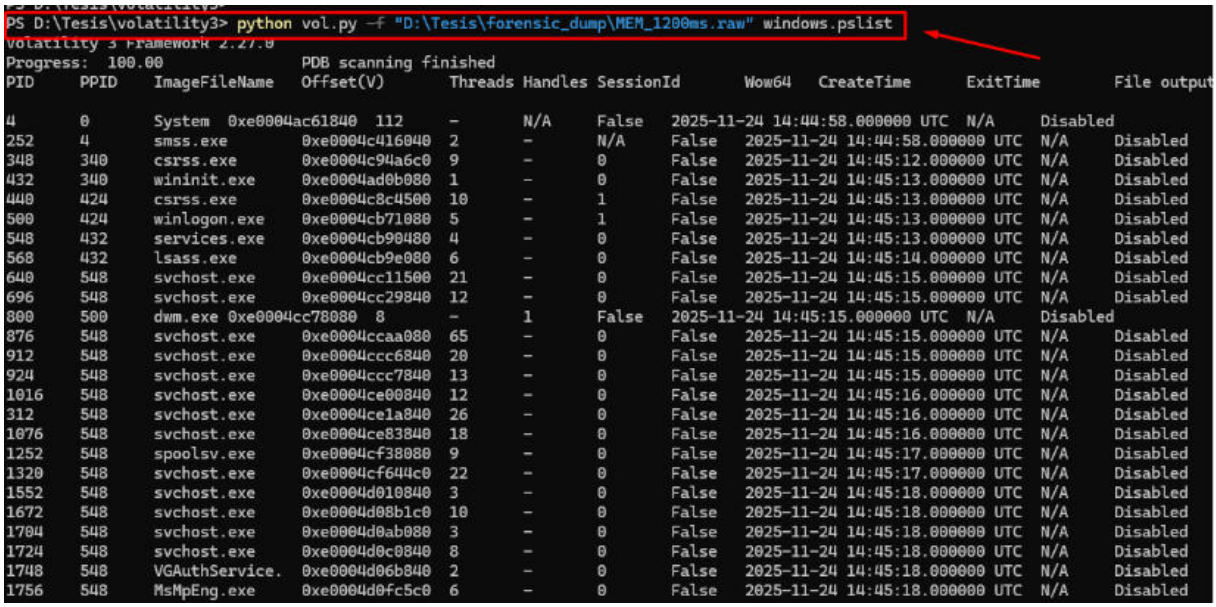
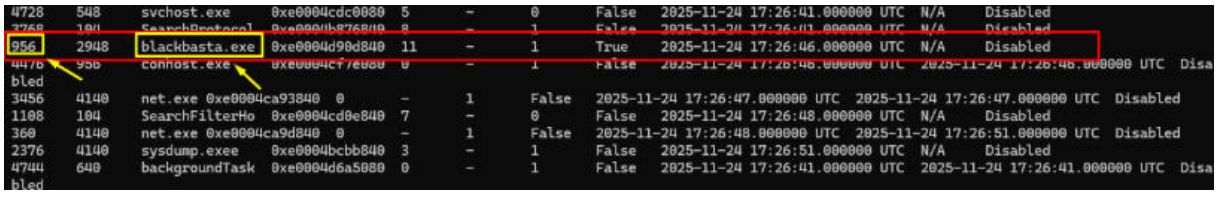


Figura 203

Identificación del PID 956 del proceso blackbasta.exe en el volcado de memoria



Como resultado de este primer paso, se identificó el proceso blackbasta.exe operando bajo el **PID 956**. La persistencia de este proceso en la lista de tareas activas confirma la efectividad de la técnica de "Suspensión de Hilos" (explicada en la sección anterior); esta técnica

mantuvo el binario cargado en RAM, sin permitirle finalizar su ejecución ni cifrar la herramienta de recolección.

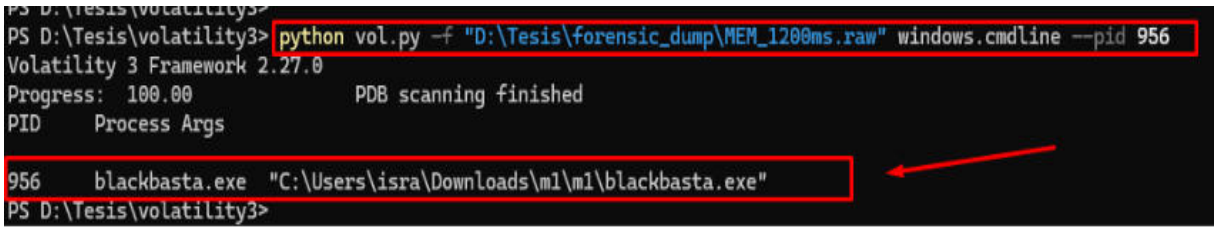
4.2.9.2. Ejecución de Comandos Para Descartar Falsos Positivos. Previo a realizar el análisis detallado, se ejecutaron plugin adicionales para corroborar que el PID 956 encontrado no se tratara de un falso positivo, por ejemplo, un proceso legítimo con inyección.

En primer lugar, se extrajeron los argumentos de ejecución del proceso para validar su origen, utilizando el comando:

```
python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw"
windows.cmdline --pid 956
```

Figura 204

Validación del origen del PID 956 en el volcado de memoria usando Volatility3



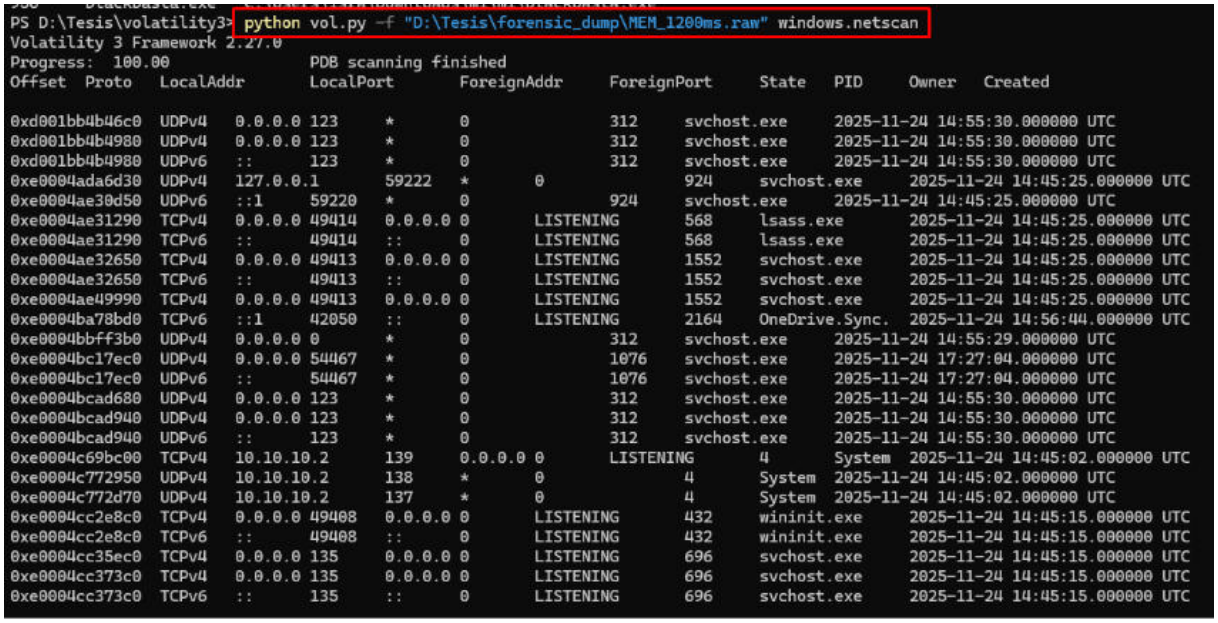
Como resultado, el análisis confirmó que el binario se ejecutó desde la ruta del usuario comprometido (C:\Users\isra\Downloads\...). Esto permite descartar técnicas de Masquerading, en las que el malware se disfraza con nombres de sistema como svchost.exe, pero se ejecuta desde rutas temporales.

En segundo lugar, se examinó la tabla de conexiones de red para identificar canales de Comando y Control (C2). El comando utilizado es el siguiente:

```
python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw"
windows.netscan
```

Figura 205

Verificación de persistencia a la red mediante el uso del plugin Windows.netscan



```
PS D:\Tesis\volatility3> python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw" windows.netscan
Volatility 3 Framework 2.27.0
Progress:  100.00               PDB scanning finished
Offset  Proto   LocalAddr       LocalPort       ForeignAddr     ForeignPort     State   PID     Owner   Created

0xd001bb4b46c0  UDPv4   0.0.0.0 123     *       0               312     svchost.exe     2025-11-24 14:55:30.000000 UTC
0xd001bb4b4980  UDPv4   0.0.0.0 123     *       0               312     svchost.exe     2025-11-24 14:55:30.000000 UTC
0xd001bb4b4980  UDPv6   ::      123     *       0               312     svchost.exe     2025-11-24 14:55:30.000000 UTC
0xe0004ada6d30  UDPv4   127.0.0.1       59222   *       0               924     svchost.exe     2025-11-24 14:45:25.000000 UTC
0xe0004ae30d50  UDPv6   ::1     59220   *       0               924     svchost.exe     2025-11-24 14:45:25.000000 UTC
0xe0004ae31290  TCPv4   0.0.0.0 49414   0.0.0.0 0       LISTENING       568     lsass.exe       2025-11-24 14:45:25.000000 UTC
0xe0004ae31290  TCPv6   ::      49414   ::      0       LISTENING       568     lsass.exe       2025-11-24 14:45:25.000000 UTC
0xe0004ae32650  TCPv4   0.0.0.0 49413   0.0.0.0 0       LISTENING       1552    svchost.exe     2025-11-24 14:45:25.000000 UTC
0xe0004ae32650  TCPv6   ::      49413   ::      0       LISTENING       1552    svchost.exe     2025-11-24 14:45:25.000000 UTC
0xe0004ae49990  TCPv4   0.0.0.0 49413   0.0.0.0 0       LISTENING       1552    svchost.exe     2025-11-24 14:45:25.000000 UTC
0xe0004ba78bd0  TCPv6   ::1     42050   ::      0       LISTENING       2164    OneDrive.Sync.  2025-11-24 14:56:44.000000 UTC
0xe0004bbff3b0  UDPv4   0.0.0.0 0       *       0               312     svchost.exe     2025-11-24 14:55:29.000000 UTC
0xe0004bc17ec0  UDPv4   0.0.0.0 54467   *       0               1076    svchost.exe     2025-11-24 17:27:04.000000 UTC
0xe0004bc17ec0  UDPv6   ::      54467   *       0               1076    svchost.exe     2025-11-24 17:27:04.000000 UTC
0xe0004bcad680  UDPv4   0.0.0.0 123     *       0               312     svchost.exe     2025-11-24 14:55:30.000000 UTC
0xe0004bcad940  UDPv4   0.0.0.0 123     *       0               312     svchost.exe     2025-11-24 14:55:30.000000 UTC
0xe0004bcad940  UDPv6   ::      123     *       0               312     svchost.exe     2025-11-24 14:55:30.000000 UTC
0xe0004c69bc00  TCPv4   10.10.10.2      139     0.0.0.0 0       LISTENING       4       System  2025-11-24 14:45:02.000000 UTC
0xe0004c772950  UDPv4   10.10.10.2      138     *       0               4       System  2025-11-24 14:45:02.000000 UTC
0xe0004c772d70  UDPv4   10.10.10.2      137     *       0               4       System  2025-11-24 14:45:02.000000 UTC
0xe0004cc2e8c0  TCPv4   0.0.0.0 49408   0.0.0.0 0       LISTENING       432     wininit.exe     2025-11-24 14:45:15.000000 UTC
0xe0004cc2e8c0  TCPv6   ::      49408   ::      0       LISTENING       432     wininit.exe     2025-11-24 14:45:15.000000 UTC
0xe0004cc35ec0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       696     svchost.exe     2025-11-24 14:45:15.000000 UTC
0xe0004cc373c0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       696     svchost.exe     2025-11-24 14:45:15.000000 UTC
0xe0004cc373c0  TCPv6   ::      135     ::      0       LISTENING       696     svchost.exe     2025-11-24 14:45:15.000000 UTC
```

Como resultado, el análisis no reveló conexiones de red activas, ni sockets a la escucha, iniciados directamente por el proceso blackbasta.exe; lo cual es coherente con el entorno de red aislado de LAB-BLACK. Al no tener salida a Internet en el diseño de este laboratorio, el malware no puede establecer comunicación con alguna infraestructura C2; así se restringe cualquier actividad de red del malware, únicamente se reflejan las exploraciones SMB locales.

4.2.9.3. Análisis de Inyección de Código. Con el objetivo de determinar si el malware utilizaba técnicas de evasión avanzada como *Process Hollowing* o *DLL Injection*, se analizó el mapa de memoria del proceso en busca de regiones con permisos de ejecución sospechosos (PAGE_EXECUTE_READWRITE); el comando utilizado es:

```
python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw"
windows.malfind --pid 956
```

Como resultado, se observa que el plugin no reportó hallazgos positivos, es decir se obtuvo una salida vacía; sin embargo, obtener una salida vacía constituye, en sí misma, una

forma de respuesta. Se puede interpretar que la falta de secciones de memoria inyectadas indica que, en la fase de 1.2 segundos, Black Basta opera como un ejecutable convencional y no depende de la inyección de código en procesos legítimos para funcionar.

4.2.9.4. Extracción Forense del Espacio de Memoria. Dado que el análisis preliminar de inyecciones de código mediante el plugin *windows.malfind* resultó negativo, indicando que Black Basta opera en esta fase como un ejecutable tradicional y no mediante inyección dispersa, se determinó que la evidencia crítica residía en el segmento de datos *Heap* del propio proceso.

Para analizar este contenido de forma aislada, se realizó un volcado de proceso o Process Dump. Se utilizó el plugin *windows.memmap* para extraer la totalidad del espacio de direcciones virtual asignado al PID 956 hacia un archivo estático en disco, a través del comando:

```
python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw" windows.memmap
--pid 956 --dump
```

Figura 206

Extracción del espacio de memoria mediante el plugin Windows.memmap

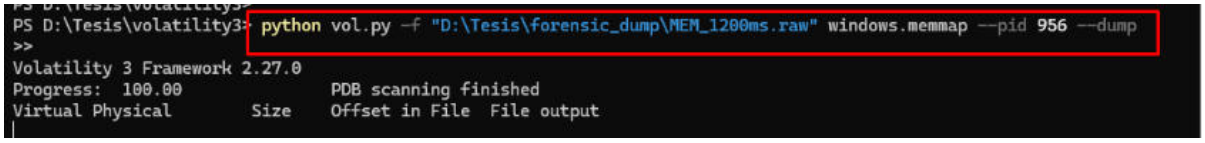
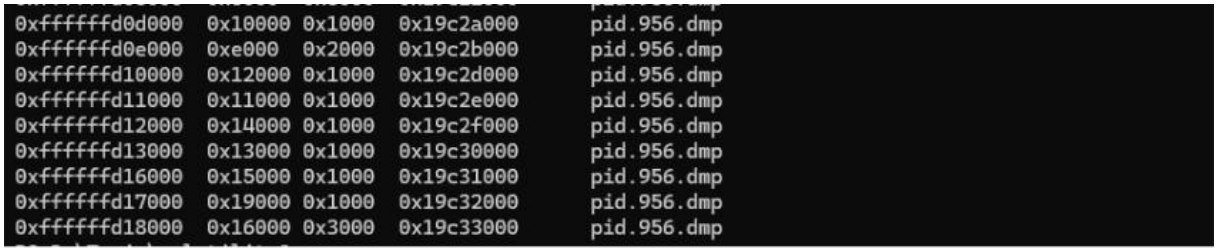


Figura 207

Generación del archivo pid.956.dmp



Como resultado, la operación generó exitosamente el archivo pid.956.dmp.

Con este procedimiento, se logra paralizar el estado de la memoria y extraer de ella las cadenas de texto, configuraciones internas ya desencriptadas, y funciones activas del ransomware. En contraste con el binario en disco, que permanece ofuscado, el volcado contiene todo en claro, lo cual permite un análisis de cadenas más preciso a continuación.

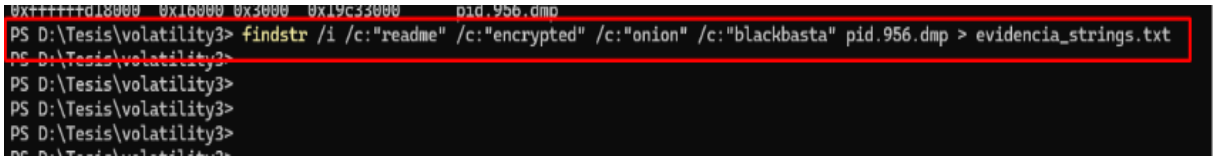
- **Análisis Estático de Contenido.-** Para interpretar la información contenida en el volcado de memoria cruda (pid.956.dmp), se empleó una metodología de "Strings Scavenging". Esta técnica permite identificar y extraer cadenas legibles (ASCII y Unicode) incrustadas entre las estructuras binarias del proceso, lo cual facilita la recuperación de configuraciones y datos de contexto que el ransomware mantiene en su espacio de *Heap*.

Se utilizó el comando findstr sobre el archivo de volcado, aplicando filtros para aislar cadenas de texto en formato *WideChar* (2 bytes por carácter); este formato es utilizado nativamente por el Kernel de Windows para el manejo de rutas y objetos del sistema.

El archivo resultante, nombrado como evidencia_strings.txt, fue depurado para eliminar segmentos de memoria no imprimibles, aislando los artefactos de interés.

Figura 208

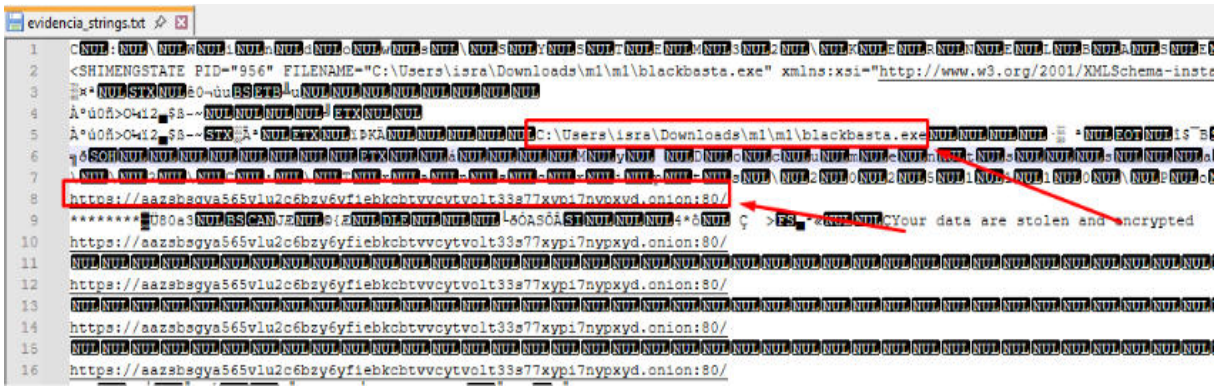
Procesamiento del archivo de volcado mediante el uso del comando findstr



Como primer hallazgo crítico, de la cadena recuperada (Unicode), se visualiza la ruta completa del ejecutable en su espacio de memoria, confirmando su origen exacto: "C:\Users\isra\Downloads\m1\m1\blackbasta.exe".

Figura 209

Hallazgo de la ruta del ejecutable dentro del archivo analizado



La recuperación de este artefacto proporciona confirmaciones técnicas trascendentes para el perfilado del incidente. Al confirmar que el código ejecutándose en la RAM corresponde al binario blackbasta.exe detonado desde el directorio /Downloads del usuario, se descarta la posibilidad de que el proceso sea un falso positivo o un servicio del sistema inyectado.

También, se demuestra que en el momento de la suspensión (al tiempo 1.2s), el *Process Environment Block* mantenía la referencia al archivo en disco, indicando que el ransomware no había realizado técnicas *Module Stomping* o borrado de referencias en esta etapa temprana.

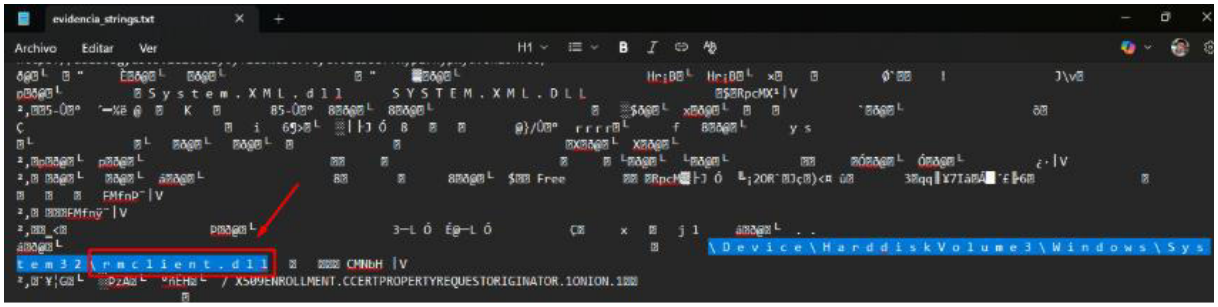
Finalmente, se logra correlación entre la evidencia física y volátil, es decir, entre el archivo hash en disco y el proceso en RAM, cerrando la cadena de custodia técnica del incidente.

Como segundo hallazgo crítico tras revisar las cadenas dentro del espacio de memoria, se encontró la presencia de referencias explícitas a la biblioteca del sistema rmclient.dll (Restart Manager Client), utilizando la nomenclatura de dispositivo nativa de Windows NT:

\Device\HarddiskVolume3\Windows\System32\rmclient.dll

Figura 210

Hallazgo de referencias explícitas a biblioteca rmclient.dll dentro del archivo analizado



Tal como se observa en la **Figura 210**, se detecta la ruta completa en formato de dispositivo, HarddiskVolume3, dentro del espacio de direcciones del proceso; esto confirma que el binario malicioso llegó a referenciar o mapear la librería **rmclient.dll**.

El *Restart Manager* es una API legítima considerada para instaladores, sin embargo, al encontrarla dentro de un ejecutable de usuario como blackbasta.exe se evidencia que el malware está aplicando la técnica de "File Unlocking". Esta técnica le permite al ransomware identificar y cerrar aplicaciones, por ejemplo, bases de datos o gestores de correo que mantienen archivos abiertos, liberando los candados del sistema para proceder a su cifrado exitoso sin errores de acceso. in

Se descartan falsos positivos, y se confirma que blackbasta.exe utilizó la librería *rmclient.dll;* puesto que, aunque no se haya analizado la imagen completa de la RAM sino solo el volcado del proceso PID 956, es técnicamente imposible que cadenas de texto o referencias pertenecientes a otros servicios del sistema aparezcan dentro del segmento de datos privados o "*Heap*" del ransomware. Windows aísla estrictamente el espacio de direcciones virtual de cada proceso.

4.2.10. Análisis de Otras Ejecuciones del Ransomware. Dados los resultados y el análisis obtenido hasta este punto, el objetivo de ejecutar otros análisis es realizar un ejercicio de contraste técnico, donde se refuerce la validación cruzada, que permite distinguir aquellos comportamientos de Black Basta que son circunstanciales, de aquellos que pueden ser considerados constantes.

Por lo tanto, se replicó estrictamente la misma metodología de adquisición y análisis detallada en los capítulos anteriores con la finalidad de garantizar la integridad y comparabilidad de los datos.

A continuación, se presentan los resultados de este análisis comparativo, destacando las similitudes tácticas y las variaciones en los Indicadores de Compromiso (IOCs).

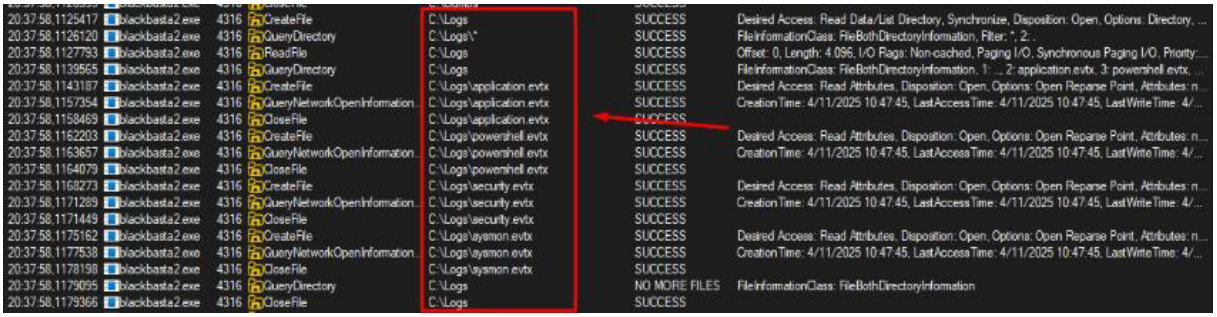
4.2.10.1 Análisis de archivo PML. Para validar la persistencia de las tácticas obtenidas, se analiza el registro de actividad procmon_20251127_203753.pml.

Se aísla el proceso de la muestra blackbasta2.exe (hash SHA-256 17205C43189C22DFCB278F5CC45C2562F622B0B6280DCD43CC1D3C274095EB90) con un PID: 4316, permitiendo una comparación directa con el comportamiento de la muestra original.

El análisis del registro PML confirma que el ransomware mantiene su prioridad sobre la destrucción de evidencias.

Figura 211

Identificación de accesos a directorio C:\Logs



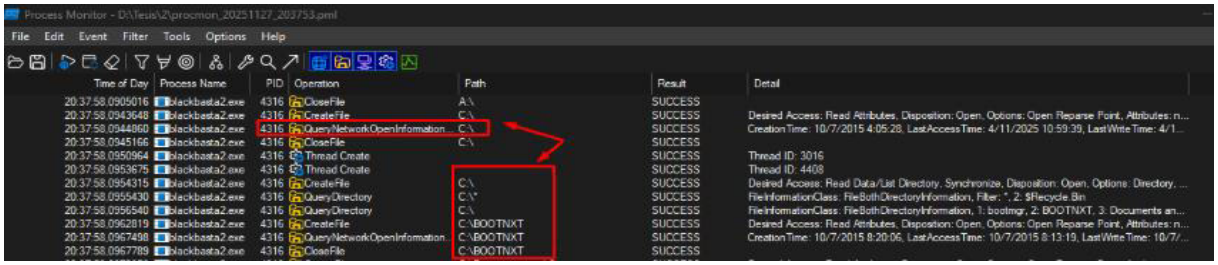
Unos milisegundos tras su ejecución, a las 20:37:58, se observa el acceso al directorio C:\Logs y la interacción con los archivos de registro del sistema: C:\Logs\application.evtx; C:\Logs\powershell.evtx; C:\Logs\security.evtx.

Esta conducta replica exactamente el patrón observado en análisis anteriores, validando que el ataque a la infraestructura de auditoría es un comportamiento invariante de la familia Black Basta.

4.2.10.2. Fingerprinting del Sistema de Archivos. Adicionalmente, se detectaron operaciones de consulta “QueryNetworkOpenInformationFile” sobre archivos de sistema críticos en la raíz, específicamente C:\BOOTNXT. Esto indica que el ransomware antes de desplegarse verifica la integridad o el tipo de arranque del sistema operativo anfitrión. Hasta este punto, no se ha observado algún comportamiento nuevo.

Figura 212

Identificación de operaciones de consulta sobre el directorio C:\BOOTNXT



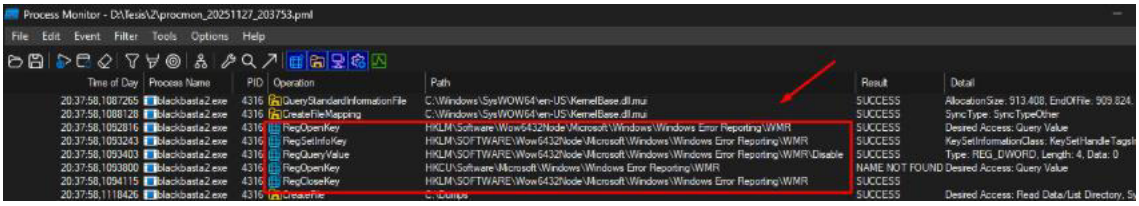
El proceso blackbasta2.exe PID: 4316, interactúa con la clave de registro HKLM\Software\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR.

Las operaciones registradas “RegSetInfoKey” sugieren un intento de deshabilitar o manipular el servicio de Reporte de Errores de Windows (WerFault).

Esta acción por parte del ransomware busca prevenir la generación de volcados de memoria o alertas visuales en caso de que el binario malicioso sufra una interrupción imprevista, con esto se dificulta el análisis forense post-mortem por parte de sistemas automatizados.

Figura 213

Identificación de interacciones con Windows Error Reporting



4.2.10.3. Mutación de IOCs. En las operaciones de renombrado se detecta que esta variante de análisis no utiliza la extensión .basta, en su lugar, aplica un algoritmo de renombrado que añade el sufijo “.evtxencrypted” o “encrypted” al nombre original del archivo.

Figura 214

Identificación de operaciones SetRenameInformationFile

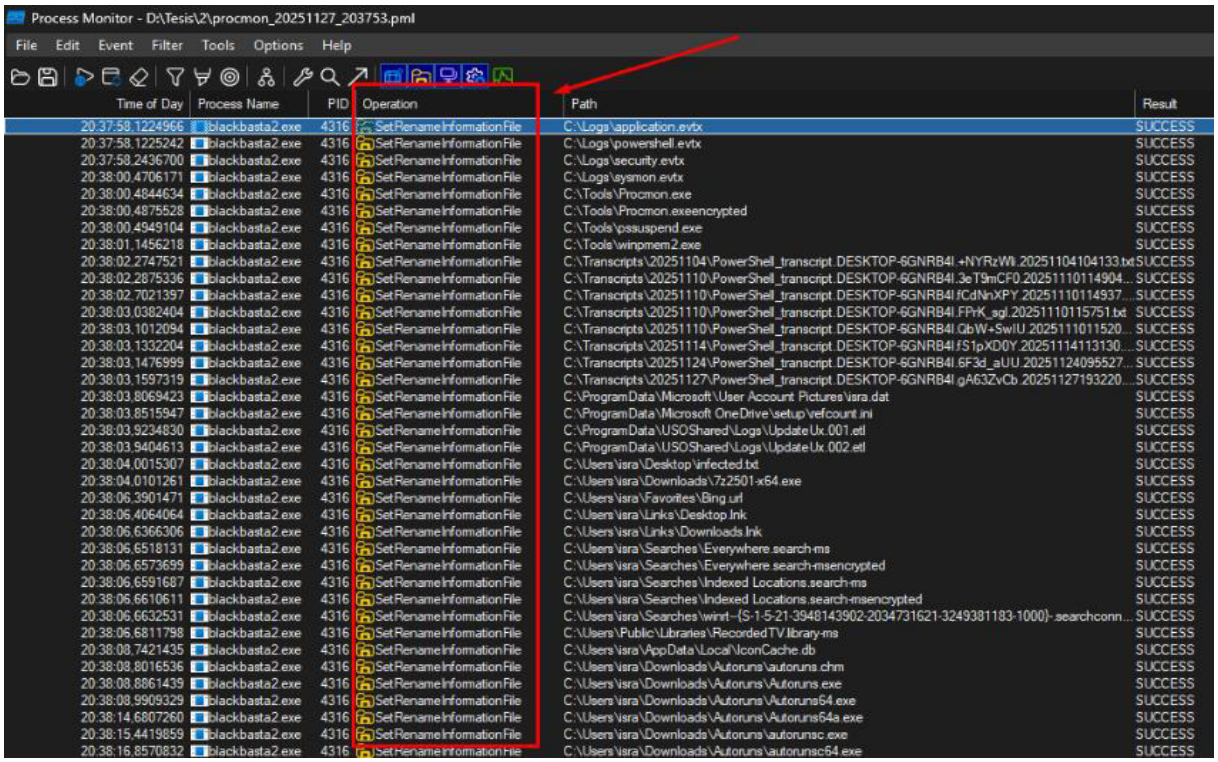
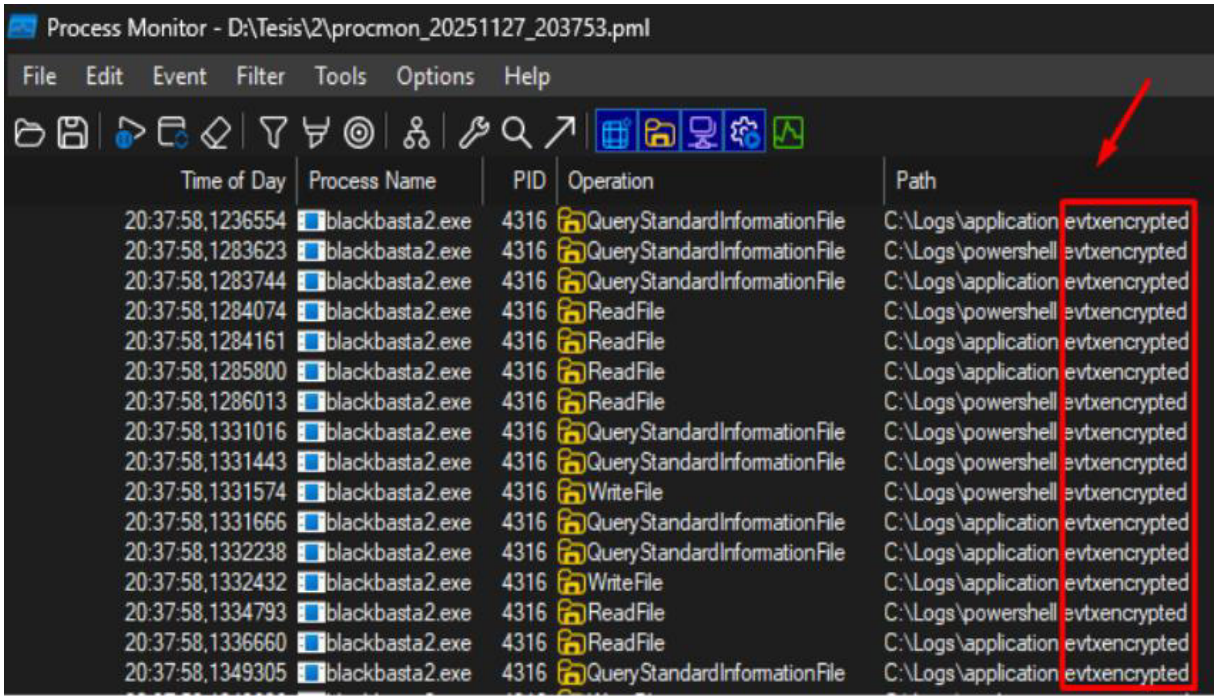


Figura 215

Identificación de operaciones QueryStandardInformationFile



Tal como se observa en la **Figura 214** y **Figura 215**, se identifican operaciones SetRenameInformationFile y QueryStandardInformationFile en las que el Path muestra el renombramiento o adición del sufijo (application.evtx $\rightarrow$ application.evtxencrypted); por lo tanto, se destaca la importancia de basar la detección en anomalías de comportamiento, en lugar de depender exclusivamente de listas estáticas de extensiones que son trivialmente modificables por cada campaña del ransomware.

4.2.10.4. Matriz de Diferencias y Similitudes entre Variantes de Black Basta.

Tabla 10

Matriz de Diferencias y Similitudes entre Variantes de Black Basta

Característica o Comportamiento	**Muestra Principal (blackbasta.exe)**	**Otras Muestras (blackbasta2.exe)**	**Interpretación Forense**
Indicador de Cifrado (Extensión)	.basta (Añadida al final)	.evtxencrypted (Sufijo compuesto)	Las muestras cambian marcadores estáticos para evadir firmas simples.
Objetivo Anti-Forense	Ataque inmediato a C:\Logs (.evtx).	Ataque inmediato a C:\Logs (.evtx).	La destrucción de logs es el distintivo de la familia.
Evasión de Defensa (Shadows)	Ejecución explícita de vssadmin (SysNative/System32).	Ausencia de llamadas a vssadmin.	Posible cambio en la técnica de borrado o ejecución diferida fuera del log capturado.
Persistencia Visual (Staging)	Creación de dlaksjdoiwq.jpg e .ico en %TEMP%.	No crea artefactos gráficos en %TEMP%.	La variante parece enfocarse en la destrucción silenciosa o rápida, omitiendo la preparación visual del escritorio.
Reconocimiento de Entorno	Genérico (Heredado del usuario).	Consulta a C:\BOOTNXT y	Mayor sofisticación en la fase de *Fingerprinting* para

		manipulación de Registro (WMR).	verificar si el sistema es viable antes de cifrar.
Gestión de Errores	No observada específicamente.	Intento de deshabilitar *Windows Error Reporting* (WMR).	Técnica de Sigilo Operativo para evitar alertas de Crash del sistema.

Nota. En la primera columna se encuentra la característica a destacar; en la segunda columna, se indica la respuesta de la muestra primaria; en la tercera, la respuesta de las muestras adicionales y en la cuarta columna, una breve interpretación tras el análisis forense.

4.2.10.5. Análisis Forense de Memoria. Al igual que en los primeros análisis, se aplicó la técnica de adquisición por suspensión, usando una ventana de 1.2 segundos, sobre la nueva muestra de la variante del ransomware, con el propósito de verificar si los indicadores en memoria (cadenas, extensiones y librerías cargadas) coincidían con la mutación observada en el análisis de disco o si presentaban divergencias adicionales.

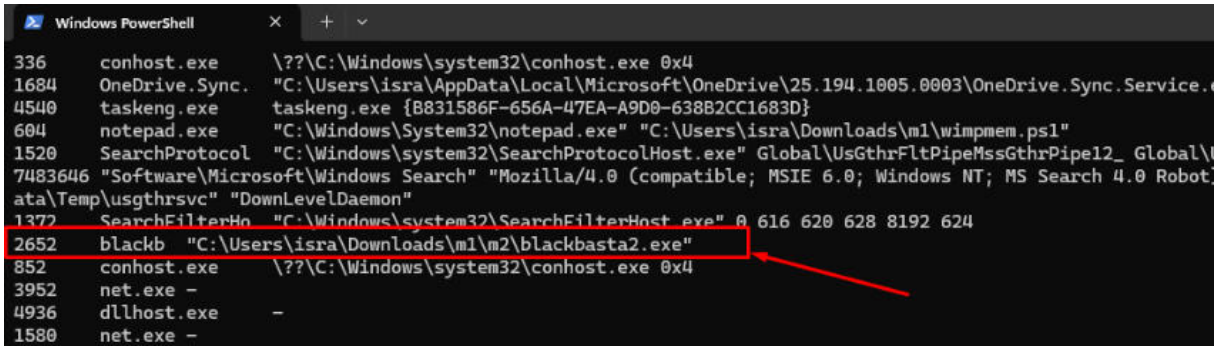
El volcado resultante MEM_1200ms_205208.raw fue procesado mediante Volatility 3 para la extracción de artefactos volátiles.

Mediante el plugin windows.pslist se aisló la ejecución del código malicioso. El comando utilizado fue:

```
python vol.py -f "D:\ Tesis\2\MEM_1200ms_205208.raw" windows.pslist
```

Figura 216

Identificación de procesos activos a través del plugin windows.pslist

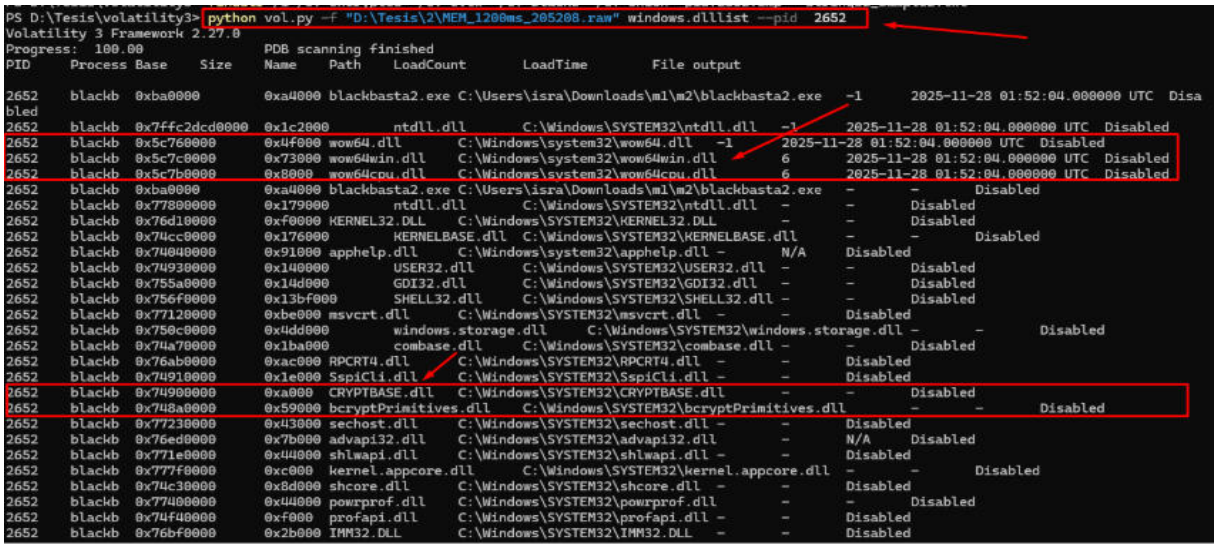


Como resultado, se identifica el proceso activo bajo el PID 2652. En contraste con los análisis anteriores, el nombre del proceso fue recuperado como “blackb”. Este comportamiento puede darse por la longitud del nombre del binario original “blackbasta2.exe”, o por la forma en que la estructura EPROCESS almacenó la cadena de imagen en esta ejecución específica.

Continuando con el análisis, la Tabla de Importación de Direcciones (IAT) fue examinada en memoria a fin de identificar las bibliotecas de enlace dinámico (DLL) utilizadas por el proceso PID 2652.

Figura 217

Análisis de la Tabla de Importación de Direcciones utilizando el plugin windows.dlllist



Utilizado el comando `python vol.py -f "D:\ Tesis\2\MEM_1200ms_205208.raw" windows.dlllist` se determinó la existencia de módulos como wow64.dll, wow64win.dll y wow64cpu.dll confirmando que blackbasta2.exe es un binario de 32 bits que se está ejecutando en un entorno de 64 bits. Debido a esta arquitectura, el malware se ve obligado a utilizar técnicas de redirección de sistema de archivos "SysNative" con el objetivo de acceder a herramientas administrativas de 64 bits, validando así los intentos de evasión observados durante la fase dinámica.

De manera adicional, se identifica la carga de bcryptPrimitives.dll y CRYPTBASE.dll, lo cual revela que el ransomware aprovecha APIs criptográficas nativas de Windows para generar números aleatorios o rutinas de cifrados complementarias, siendo una de las razones por las que el tamaño del binario disminuye al no incluir librerías criptográficas estáticas pesadas.

A diferencia de los indicios observados durante las ejecuciones de muestras anteriores, se puede mencionar que la capacidad de desbloqueo de archivos podría considerarse opcional, bajo demanda, o simplemente estar ausente en esta variante específica, debido a que no se detectó carga explícita de rmclient.dll (Restart Manager) en la lista de módulos activos a los 1.2 segundos.

4.2.10.6. Análisis de Contenido de Memoria. Para verificar la variabilidad de los indicadores estáticos, se extrajeron y filtraron las cadenas del proceso PID 2652 de blackbasta2.exe.

Este resultado es interesante y cambia ligeramente la perspectiva de lo observado; ya que a diferencia de los volcados anteriores donde se visualizaba código directo del ransomware, en esta ocasión se observan rastros de la base de datos de firmas del Antivirus Windows Defender, o de algún módulo de escaneo cargado en memoria. Microsoft Defender suele usar

identificadores internos para etiquetar amenazas, por ejemplo, cadenas como *!#TEL:Hacktool...* y *!#ALF:Trojan...*

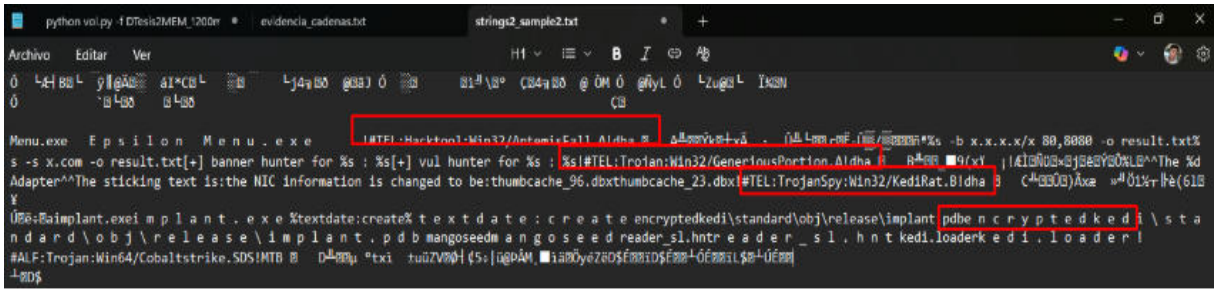
El volcado de memoria analizado incluye cadenas de texto que coinciden con firmas de amenazas de Microsoft; lo cual sugiere que el proceso de *Windows Defender "MsMpEng.exe"* escaneó estos componentes en memoria, aunque también existe la posibilidad de que el propio malware contiene estas cadenas para realizar comprobaciones internas con fines de evasión.

Algunos identificadores encontrados son:

- Hacktool:Win32/ArtemisFall.A!dha
- Trojan:Win64/Cobaltstrike.SDS!MTB
- Trojan:Win32/GeneriousPortion.A!dha
- TrojanSpy:Win32/KediRat.B!dha

Figura 218

Identificadores encontrados en strings



Como se observa en la **Figura 218**, se identificó el artefacto *encryptedkedi\standard\obj\release\implant.pdb* y presencia de símbolos de depuración relacionados con "Kedi RAT", esto sugiere que se utilizó este *spyware* como vector inicial de acceso, o probable persistencia, posiblemente con el objetivo de exfiltrar información antes de llevar a cabo el cifrado final con Black Basta.

Asimismo, se encontraron cadenas de formato “%s -b x.x.x.x/x 80,8080 -o result.txt%...” asociadas con herramientas de escaneo de red; lo cual sugiere que el sistema comprometido fue utilizado para escanear la red interna en busca de servicios web vulnerables en puertos 80 y 8080.

En conclusión, el análisis forense de memoria realizado sobre el volcado MEM_1200ms_205208.raw con PID 2652 demuestra que esta variante examinada de Black Basta ha evolucionado hacia un comportamiento más sofisticado y sigiloso, pero mantiene inmutable su misión táctico destructivo. A continuación, se destacan tres puntos clave para definir a esta variante:

- El cambio de la extensión en el cifrado empleado para el registro de .basta a .evtxencrypted en conjunto con la manipulación del Registro de Errores de Windows (WMR), evidencian un claro esfuerzo por evadir firmas estáticas.
- El malware emplea técnicas de redirección de sistema (SysNative) para afectar la integridad del sistema operativo anfitrión, ejecutándose como un proceso de 32 bits dentro de un entorno de 64 bits.
- Las firmas de detección encontradas en memoria, vinculadas a Kedi RAT y Cobalt Strike, indican que la muestra no actúa por si sola, sino que corresponde al payload final de un ataque complejo manejado por humanos.

4.3. Perfil Mínimo de Black Basta

4.3.1. Información General de Black Basta.

- **Tipo de Ransomware:** Black Basta
- **Sistema operativo objetivo:** Windows
- **Entorno de ejecución:** Laboratorio controlado sin conexión a Internet.
- **Tamaño del archivo:** 636.50 KiB
- **Lenguaje:** C

4.3.2. Comportamiento Observado.

4.3.2.1. Procesos. Se destacan los siguientes:

- **Proceso principal:** blackbasta.exe PID 956
- **Procesos Secundarios funcionando en simultaneo:** taskhostw.exe, SecurityHealthHost.exe, dmclient.exe, Microsoft Edge y Windows Terminal (No generados necesariamente por el ransomware).
- **Directorios comprometidos:** C:\Users, C:\ProgramData, C:\Temp, C:\Users\Default
- **Total de eventos generados:** 22.411.855
- **Eventos de File System Activity:** 21.968.037
- **Eventos de Network Activity, Process and Threats activity y Registry Activity:** 443.818.

4.3.2.2. Modificaciones del Sistema. Entre los archivos creados por el ransomware se tienen:

- **Notas de rescate:** Readme.txt
- **Imagen para Wallpaper:** dlaksjdoiwq.jpg

- **Ícono para archivos cifrados:** fkdjsadasd.ico
- **Archivos encriptados:** Archivos ubicados en accesos directos al ransomware.
- **Ruta desde donde se ejecutó:** C:\Users\isra\Downloads\blackbasta.exe
- **Actividades detectadas:** Carga continua de módulos Load Image. Llamadas a operaciones del tipo CreateFile, CloseFile, WriteFile, ReadFile, Query.
- **Preparación y Evasión:** Drop de recursos gráficos (%TEMP%) más destrucción redundante de Shadow Copies.
- **Saturación:** Corrupción de logs .evtx, despliegue de nota de extorsión y cifrado de datos.

4.3.2.3. Comportamiento del Cifrado. Se destacan los siguientes:

- **Extensión de cifrado observada:** .encrypted.
- **Tipo de archivos encriptados:** texto .txt, pdf, png, jpg, mp4, bases internas de datos jtx y jcp, archivos Edge, WinRAR, url, shortcuts, LNK, ejecutables .exe, e incluso accesos directos (.lnk).
- **Cambios en la estructura del archivo:** Nombre del archivo original sumado la extensión ".encrypted". La muestra contiene funciones criptográficas propias.

4.3.3. Funciones Utilizadas por Black Basta. Se detectó que el ransomware realiza la implementación de las siguientes funciones:

- **FUN_0041b250 → mpz_import :** Conversión de buffer de bytes a un entero multiprecisión.
- **FUN_0041b540 → mpz_powm:** Implementa la exponenciación modular.
- **FUN_0041a410 → mpz_export:** Extrae un entero multi-precisión a un buffer plano.

- **FUN_0041bce0:** Integra otras funciones para la implementación de un esquema híbrido. Cifrado de archivos.
- **FUN_0040d160:** Inicializa parámetros criptográficos y claves internas.
- **FUN_0040ca70:** Enumeración de volúmenes. Obtiene y construye listas de rutas.
- **FUN_0040b840:** Planificación multihilo y cola de trabajo.
- **FUN_0040bff0:** Construye el cifrado y renombre de archivos.
- **FUN_0040d130:** Identifica y genera claves.
- **FUN_00415060:** Escritura del blob RSA.
- **FUN_0040ae10:** Modificación del Wallpaper de fondo. Integra funciones destinadas a la definición de tiempo y GB cifrados.
- **FUN_00401180:** Manipulación de servicios y detección de entornos de virtualización.
- **FUN_0040b090:** Integra una función para la creación del archivo readme.txt

4.3.4. Actividades Registradas. Se utiliza la herramienta Procmon.

Lectura de claves:

- HKCU\Control Panel\Desktop\Wallpaper
- HKCU\Control Panel\Desktop
- HKCU\Software\Microsoft\Windows\CurrentVersion\Themes\Personalize
- HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SafeBoot
- Lectura sobre claves de Explorer y del perfil

Escritura directa:

- RegSetValue
- **Path:** HKCU\Control Panel\Desktop\Wallpaper

- **Type:** REG_SZ, Length: 2, Data: “”

Creación y acceso: Archivos temporales en **AppData\Local\Temp**

- C:\Users\employee\AppData\Local\Temp\dai-vvdiolxnjq.jpg
- C:\Users\employee\AppData\Local\Temp\dai-wqdioajhv.jpg

Llamados a funciones: Observamos funciones relacionadas a GMP(big integers).

- QueryDirectory
- CreateFile
- ReadFile
- CloseFile

Llamados a directorios de Usuario:

- C:\Users\employee\Documents
- C:\Users\employee\Desktop
- C:\Users\employee\Downloads

Llamados a directorios del sistema:

- C:\Windows\System32
- C:\Program Files
- C:\ProgramData

Llamados a directorios críticos del sistema de archivos:

- C:\Recovery
- C:\System Volume Information

Accesos a:

- C:\Windows\AppPatch\MegSdb*
- C:\Windows\System32\LogFiles

- Archivos .db, .ini, .log

4.3.5. Artefactos Forenses.

4.3.5.1. Lista de Artefactos Analizados.

- Microsoft-Windows-Sysmon/Operational
- procmon_20251114_121545.pml
- Mem_1200ms.raw
- Elastic.EventLogs.Sysmon.json
- Artefacto Windows.NTFS.Timestomp.json
- Artefacto Windows.Packs.Persistence%2FStartup Items.json
- Generic.Forensic.Timeline.json
- Windows.Forensics.Lnk

4.3.5.2. Artefactos Principales e Información Obtenida. Entre las evidencias principales se encuentran los archivos de registro de eventos (EVTX), capturas de actividades del sistema mediante Procmon (PML) y un volcado de memoria completo (RAW).

- **Actividad en archivos de registros EVTX.**

Herramienta utilizada: Sysmon

Artefacto: Microsoft-Windows-Sysmon-Operational

Hash: 0e29c1def196bfd88db689ec2bca2b2402602949d8167ff290d9c5320bf6833a

Eventos Generados:

Event ID 1: Creación de Procesos. Evasión de Defensa y destrucción de respaldos.

Ejecución de comandos:

C:\Windows\SYSTEM32\ cmd.exe/c

C:\Windows\SysNative\vssadmin.exe delete shadows /all /quiet

Event ID 11: Creación de archivos como notas de rescate.

Ejecución de comandos:

C:\ProgramData\Microsoft\Windows\Start Menu\readme.txt

C:\Users\Default\readme.txt

Event ID 5: Terminación de procesos.

Event ID 13: Claves e inventario de aplicaciones.

- **Actividad en archivos PML.**

Herramienta utilizada: Procmon

Artefacto: procmon_20251114_121545.pml

Hash: 2cf136ca7e4ec8fa98ab9ee729de470662fa7b4690ec45368470d4763911636d

Ruta de ejecución de la muestra: C:\Users\isra\Downloads\m1\

Peticiones de eliminar los Volume Shadow Copies:

C:\Windows\System32\vssadmin.exe

Usa la ruta SysNative para saltar la redirección de archivos de 32 bits.

- **Artefactos creados por Black Basta.** dlaksjdoiwq.jpg corresponde a la imagen de fondo de escritorio con la nota de rescate y fkdjsadasd.ico es el ícono personalizado para los archivos cifrados.

Tiempo de la ventana de infección: 4 min y 30 segundos.

Tiempo de Inicio: 12:15:57.515

Tiempo de finalización: 12:20:27.933

Arquitectura de Black Basta: Emplea una arquitectura multihilo (multithreading)

Primera operación: A las 12:15:57.553, se registró la primera operación de escritura destructiva (Offset: 0, Length: 64) sobre el archivo C:\Logs\application.evtx.

Hallazgo: Black Basta prioriza la corrupción de evidencias forenses (.evtx) siempre que se encuentren en directorios accesibles (C:\Logs\), antes de iniciar con el cifrado masivo de documentos.

- **Actividad en memoria.**

Herramienta utilizada: winpmem

Herramienta de análisis: Volatility 3

Artefacto: winpmem_20251114_123536.raw

Hash: 296cf640111a015b6ae57a3f5d60e67c9ec8a1e2390ad7c2086e36c3e7d4cf79

Proceso del ransomware: blackbasta.exe operando bajo el PID 956.

Ruta de ejecución: El archivo contenía la ruta completa del ejecutable en su espacio de memoria.

Conexiones de red: No se encontraron conexiones de red activas debido al aislamiento de la máquina.

4.3.6. Técnicas MITRE ATT&CK. Se destacan las siguientes:

T1059.003. El proceso padre blackbasta.exe instanció cmd.exe con el argumento /c para ejecutar herramientas nativas del sistema, delegando la tarea de evasión al intérprete de comandos.

T1211. Uso de SysNative. El malware explotó la ruta virtual C:\Windows\SysNative para saltar la redirección del sistema WOW64 y ejecutar comandos de 64 bits desde un proceso de 32 bits.

T1070.001. Corrupción de EVTX. Se registró la sobreescritura directa (WriteFile) de archivos sysmon.evtx y application.evtx en C:\Logs. Black Basta corrompe la estructura del archivo para inutilizar la auditoría.

T1083. Traversal Masivo. El algoritmo de despliegue de notas (readme.txt) realizó un barrido exhaustivo del sistema de archivos durante 4.5 minutos, interactuando con Junction Points (Result: REPARSE) para descubrir la estructura lógica del disco.

T1490. Borrado de Shadows. Ejecución redundante de vssadmin.exe delete shadows /all /quiet. La doble invocación garantiza la destrucción de instantáneas de recuperación antes del cifrado.

T1486. Cifrado de Datos. Confirmado por el bucle de operaciones de I/O y la preparación visual mediante el drop de íconos (.ico) y fondo de pantalla (.jpg) en %TEMP% para notificar visualmente el impacto.

4.3.7. Propuesta de Perfil de Detección (IOAs). Basado en la correlación forense de las muestras de Black Basta en el entorno controlado LAB-BLACK, se ha diseñado un perfil de detección unificado. Este perfil prioriza los Indicadores de Ataque (IOAs) conductuales sobre los Indicadores de Compromiso (IOCs) estáticos, garantizando la resiliencia de la defensa ante la mutación de extensiones o hashes.

A continuación, se detallan las reglas de detección estructuradas por vector de telemetría.

4.3.7.1. Detección de Actividad Anti-Forense (Alta Fidelidad). El rasgo persistente más crítico identificado en las variantes es el ataque a la infraestructura de auditoría. Esta conducta no es aleatoria, sino un requisito operativo del ransomware.

- **Lógica de Detección:** Alertar acerca de cualquier proceso ajeno al servicio de registro de eventos (svchost.exe - servicio EventLog) que solicite manejadores de escritura (GenericWrite / Delete) sobre archivos con extensión .evtx.
- **Regla propuesta (Formato SIGMA)(YAML):**

```
title: Manipulación No Autorizada de Archivos de Auditoría
status: experimental
```

```
logsource:
    product: windows
    category: file_event
detection:
    selection:
        TargetFilename|endswith: '.evtx'
    filter:
        Image|endswith: '\svchost.exe'
    condition: selection and not filter
level: critical
tags:
    - attack.defense_evasion
    - attack.t1070.001
```

- **Justificación:** En el análisis dinámico, se encontró que las muestras iniciaron la corrupción de C:\Logs*.evtx en los primeros 2 segundos de ejecución. Esta regla detecta las variantes instantáneamente.

4.3.7.2. Detección de Evasión y Reconocimiento. El segundo análisis reveló técnicas de *Fingerprinting* y manipulación de registro que complementan la destrucción de *Shadow Copies* observada en la primera.

Manipulación del Reporte de Errores (WMR).- Un indicador temprano de ejecución maliciosa es el intento de supresión de telemetría de fallos.

- **Vector:** Monitorización de cambios en el Registro de Windows.
- **Ruta Objetivo:**

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR

- **Acción:** Escritura de valores disable o modificación de claves por procesos no firmados por Microsoft.

Anomalías en la Invocación de Comandos (SysNative).- El malware se ve obligado a usar rutas virtuales para acceder a herramientas de 64 bits debido a su ejecución desde el subsistema WoW64.

- **Vector:** Creación de procesos (Event ID 1 / 4688).
- **Firma de Comportamiento:** Línea de comandos que contenga explícitamente \SysNative\ combinada con utilidades administrativas como vssadmin.exe, cmd.exe o powershell.exe.
- **Contexto:** Un administrador legítimo o un script de sistema rara vez invoca SysNative de forma directa; esta ruta es casi exclusiva de malware de 32 bits que trata de evadir la redirección de sistema.

Detección de Cifrado en Memoria (YARA).- Dado que la extensión de archivos en disco mutó en el renombramiento, la detección en memoria se convierte en el mecanismo de validación fiable. Se propone una regla YARA dirigida al *Heap* de los procesos, buscando la combinación de artefactos que aparecieron en ambos volcados.

- **Regla YARA propuesta:**

```
rule BlackBasta_Memory_Behavior {
    meta:
        description = "Detecta indicadores de Black Basta en memoria (Strings & DLLs)"
        author = "Investigación TFM"
        severity = "High"
    strings:
        // Infraestructura de Pago (Persistente en ambas muestras)
        $onion_url = ".onion" ascii wide
        $note_text = "Your data are stolen and encrypted" ascii wide
```

```
        // Artefactos de Evasión y Bloqueo
        $dll_ref = "rmclient.dll" ascii wide nocase
        $cmd_shadow = "delete shadows /all /quiet" ascii wide

        // Indicadores de Cifrado (Variables pero predictivos)
        $ext1 = ".basta" wide
        $ext2 = "encrypted" wide

    condition:
        // Detección de nota + (técnica de evasión O extensión
conocida)
        $note_text and ($onion_url or $dll_ref or $cmd_shadow or any of
($ext*))
}
```

- **Justificación:** Esta regla no depende solo de la extensión. Al buscar la nota de rescate y referencias a técnicas como rmclient (Restart Manager) o borrado de sombras, se asegura la detección incluso si el atacante utiliza otra extensión de sufijo de los archivos cifrados en futuras modificaciones del ransomware.

4.3.7.3. Detección de Artefactos de "Staging" (Persistencia Visual). El primer análisis reflejó una fase de preparación o *Staging*, donde antes de iniciar el cifrado el ransomware deposita recursos gráficos en el directorio temporal del usuario. Aunque este comportamiento puede variar entre versiones (como se evidenció en el análisis comparativo), la detección de estos artefactos específicos permite identificar variantes que empleen los mismos recursos.

- **Lógica de Detección:** Identificar la creación de archivos de imagen (.jpg) e íconos (.ico) con patrones de nomenclatura aleatoria o específica en directorios no

convencionales como %TEMP%, seguidos de forma inmedita por cambios en el registro de personalización de escritorio.

- **Regla de Detección de Logs (SIGMA):** Esta regla aprovecha el Event ID 11 de Sysmon (FileCreate) para alertar cuando un proceso sospechoso escribe los archivos identificados.

```
title: Detección de Artefactos Visuales de Black Basta (Staging)
id: blackbasta_staging_drops
status: experimental
description: Detecta la creación de archivos de imagen e icono
específicos en carpetas temporales, asociados a la nota de rescate visual.
logsource:
    product: windows
    category: file_event
detection:
    selection:
        TargetFilename|endswith:
            - '\AppData\Local\Temp\dlaksjdoiwq.jpg'
            - '\AppData\Local\Temp\fkdjsadasd.ico'
    condition: selection
level: high
tags:
    - attack.impact
    - attack.T1491.001
```

Nota Técnica: Los nombres de los artefactos dlaksjdoiwq.jpg y fkdjsadasd.ico parecen ser cadenas *hardcoded* en esta compilación específica del ransomware, esta regla se convierte en un detector de alta precisión para la variante analizada.

- **Regla de Detección en Disco/Memoria (YARA):** Dado que estos nombres de archivo deben existir como cadenas de texto dentro del binario para ser generados, es posible detectarlos antes de que se ejecuten.

```
rule BlackBasta_Dropper_Artifacts {
    meta:
        description = "Detecta nombres de archivos de staging
(Wallpaper/Icono) hardcoded en el binario o memoria"
        author = "Investigación TFM"
        severity = "Medium"
        verdict = "Malicious_Confidence_High"
    strings:
        // Cadenas específicas observadas en el análisis Procmon y
Memoria
        $s1 = "dlaksjdoiwq.jpg" ascii wide nocase
        $s2 = "fkdjsadasd.ico" ascii wide nocase
        $s3 = "t.txt.basta" ascii wide  // Fragmento de la rutina de
renombrado
    condition:
        uint16(0) == 0x5A4D and // Cabecera MZ (Ejecutable)
        any of ($s*)
}
```

- **Justificación de la Inclusión:** Si bien el segundo análisis demostró que estos artefactos pueden no estar presentes en todas las versiones, su inclusión en el perfil de detección obedece al principio de Defensa en Profundidad. Estas reglas son triviales de implementar y tienen una tasa de falsos positivos aproximada a cero. Es decir, es altamente improbable que un software legítimo cree un archivo llamado fkdjsadasd.ico en Temp.

4.3.8. Resumen del Perfil Defensivo. El análisis en laboratorio arroja que la defensa eficaz contra Black Basta en entornos Windows no debe basarse en listas negras de hashes, debido al polimorfismo del binario, por lo que debe realizarse sobre la monitorización de tres eventos:

- **Fase Temprana (T < 1s):** Acceso de lectura a archivos de arranque (BOOTNXT) y alertas por modificación de claves de Windows Error Reporting.
- **Fase de Impacto (T < 3s):** Bloqueo automático de procesos que intenten sobrescribir cabeceras de archivos .evtx o invocar vssadmin mediante rutas no estándar (SysNative).
- **Respuesta (Memoria):** Escaneo de memoria buscando coincidencias de cadenas de extorsión y referencias a librerías de desbloqueo de archivos (rmclient.dll), esto confirma la intencionalidad del proceso independientemente de su nombre o ubicación en disco.

Este perfil reduce el impacto y volumen de ataque, al centrarse en las tácticas invariantes del actor, ofreciendo una ventana de detección efectiva antes de que el cifrado masivo de datos de usuario sea completado.

4.4 Falsos Positivos

El proceso de análisis estático, dinámico y de artefactos permitió evaluar y determinar aquellos indicadores potenciales que debían ser considerados como parte de comportamientos válidos de Black Basta para la definición del perfil mínimo de detección. Así fue como se generaron algunos falsos positivos:

- **Técnicas de timestomping:** El análisis de Windows.NTFS.Timestomp.json evidenció que no existió manipulación antiforense ni de timestamps, la alta modificación de archivos y MFTentries corresponden al proceso de cifrado.

- **Técnicas de ejecución de sandbox:** Velociraptor detectó artefactos relacionados con adaptadores de red, sin embargo, esto correspondía a actividad propia del laboratorio y no a artefactos de sandbox en la muestra.
- **Baja actividad de red:** El aislamiento del laboratorio puede llevar a considerar que el ransomware tiene una actividad de red mínimo, lo cual es una conclusión incorrecta debido a las condiciones en las que se está ejecutando la muestra.
- **Generación masiva de eventos:** En Procmon es posible observar una alta cantidad de eventos, lo cual no es un comportamiento exclusivo del ransomware, sino que también pueden ser generados por un antivirus, sincronización de correos, actualizaciones, etc.
- **Relación con la librería rmclient.dll:** El análisis se realizó sobre un Volcado de Proceso (Process Dump) basado en su PID el cual contiene tanto la memoria ejecutada por el malware como las regiones privadas y mapeadas del proceso. En este contexto, la presencia de la librería rmclient.dll no puede atribuirse a un proceso genérico del sistema; por el contrario, su inclusión en las secciones mapeadas del dump confirma que Black Basta la cargó e invocó durante su ejecución, lo que evidencia su interacción directa con las funcionalidades proporcionadas por dicha librería.

Capítulo 5

5. Conclusiones y Recomendaciones

5.1. Conclusiones de Análisis Estático

El análisis estático realizado sobre la muestra, usando Detect It Easy y Ghidra, pone en evidencia un ransomware técnicamente robusto, cuidadosamente diseñado y altamente eficiente; ya que cuenta con más de 5 mil líneas de código descompilado y utiliza técnicas avanzadas de ofuscación; por este motivo, fue complicado identificar información relevante a la actividad que realiza el ejecutable. La alta entropía y la cantidad de código dificultan reconocer la lógica interna, lo cual es un resultado esperado para este tipo de ransomware.

Se observa el uso de criptografía híbrida con RSA y claves aleatorias por archivo, esto garantiza que el cifrado sea irreversible sin la clave privada del atacante. La arquitectura multihilo le otorga una velocidad de cifrado muy elevada, mientras que la modificación del Wallpaper, los íconos y las extensiones busca causar un impacto psicológico inmediato.

Adicionalmente, el uso de funciones personalizadas de operaciones matemáticas de precisión múltiple demuestra que la muestra analizada no depende de librerías externas o del sistema, sino que incorpora internamente todo lo necesario para realizar operaciones criptográficas avanzadas, inicialización de claves, enumeración de volúmenes y rutas, identificación de archivos, entre otras funciones de fuerza bruta, mostrando un nivel claro de profesionalismo en su implementación.

En resumen, todo lo observado confirma que se trata de un ransomware moderno, sofisticado y alineado con las técnicas utilizadas por familias activas de alto impacto. El análisis estático proporciona una base sólida para comprender su comportamiento y sustentar el desarrollo del Perfil Mínimo de Detección.

5.2. Conclusiones de Análisis Dinámico

Al realizarse la ejecución del ransomware, se observó que el tiempo que le toma en encriptar archivos, ejecutar comandos, crear procesos, en general, alterar el sistema, es alrededor de 4 minutos y 30 segundos, en donde, en los primeros minutos con la ayuda de la herramienta Procmon se visualiza una alta cantidad de operaciones de Thread Create, Load Image y Process Create, un patrón que refuerza lo detectado en el análisis estático y que a simple vista es notorio con la modificación del fondo de pantalla

También se comprobó la generación de archivos temporales en AppData, en algunos casos con extensiones .jpg los cuales actúan como buffers criptográficos, lo que coincide con la actividad observada en memoria asociada a operaciones RSA e hilos de cifrado.

Por otro lado, las herramientas como Sysmon, Procmon, Regshot y Velociraptor detectaron patrones que revelan gran consistencia, incluyendo cargas de DLL criptográficas, cambios de registro asociados al escritorio, lectura de claves y creación masiva de archivos. Esto confirma que el comportamiento dinámico es suficientemente llamativo como para generar múltiples puntos de telemetría confiables.

Este análisis revela las primeras instancias de ejecución del malware, iniciando por la exploración y enumeración de todas las unidades, usuarios, carpetas y directorios del sistema. Seguido la muestra accede, crea y elimina archivos temporales en AppDta, carga librerías criptográficas del sistema e identifica archivos accesibles. Esta actividad demuestra que el ransomware prepara una lista completa de archivos objetivo antes de iniciar el cifrado, evaluando permisos, extensiones y rutas excluidas. No se observa persistencia, elevación de privilegios ni modificación de configuraciones del sistema en esta fase inicial, lo que confirma que se trata de un payload de cifrado directo (“hit-and-run”).

5.3. Análisis de Artefactos

El análisis forense realizado mediante Velociraptor permitió reconstruir de manera precisa y verificable toda la cadena de ejecución asociada a la muestra de ransomware Black Basta, identificando no solo el punto de inicio y el vector de ejecución, sino también los artefactos críticos que deja en el sistema antes, durante y después del proceso de cifrado.

Entre los principales artefactos encontrados están los archivos de registros .evtx, archivos .PML y el volcado de memoria .raw obtenidos post ejecución del malware. Estos artefactos brindan una mayor información del comportamiento de Black Basta. Empezando por los registros .evtx, se identificaron eventos como Event ID 1 (evasión de defensa, destrucción de respaldos y creación de procesos) y el Event ID 11 (creación de notas de rescate), mientras que los archivos .PML aportaron con información sobre artefactos creados para la modificación del Wallpaper y una línea de tiempo de actividad del ransomware, y por último el volcado de memoria, el cual fue el artefacto más enriquecedor, debido a que brinda el sustento para corroborar características detectadas anteriormente como polimorfismo, arquitectura hibrida, técnicas de evasión, entre otras.

El análisis de Windows.NTFS.Timestomp muestra que ciertos archivos sí presentan cambios “in situ” en atributos temporales, lo cual permite establecer un criterio más dentro del PMD: modificación anómala de timestamps en archivos recientemente creados o cifrados. Los artefactos de AppCompatFlags revelan la ejecución manual del archivo malicioso tras descomprimirlo con WinRAR, demostrando una ejecución directa por el usuario, clave para el análisis del comportamiento inicial del ransomware.

Velociraptor permitió reconstruir con precisión todo el flujo de ejecución del ransomware Black Basta, desde la extracción del archivo, la ejecución explícita del binario malicioso por

parte del usuario, las evidencias persistentes en AppCompatFlags e Inventory, la ventana exacta de actividad maliciosa, hasta el análisis de artefactos secundarios como timestamps, prefetch y startup ítems. Este conjunto de evidencias permitió identificar indicadores mínimos, repetibles y verificables que conforman un Perfil Mínimo de Detección (PMD) aplicable a entornos Windows sin soluciones EDR, demostrando que es posible detectar comportamientos de ransomware únicamente con artefactos forenses y telemetría nativa del sistema.

5.4. Perfil de Detección Mínimo Black Basta

El PMD revela que Black Basta no emplea mecanismos de persistencia, lo cual confirma un enfoque puramente destructivo y orientado al impacto. Este tipo de amenazas obliga a que las detecciones se basen en actividades previas al cifrado, no en rastros posteriores como claves Run, servicios o tareas programadas.

El PMD permite identificar fases claramente separadas del ataque (preparación, evasión, cifrado, impacto visual), lo que facilita construir detecciones por etapas. Incluso si una fase fuera modificada en variantes futuras, otras fases permanecerían constantes, ofreciendo resiliencia ante cambios del malware. También, demuestra que gran parte de las defensas clásicas basadas en firmas o en la detección de extensiones cifradas llegan demasiado tarde. Las acciones críticas ocurren en los primeros 100 - 500 ms después de la ejecución, por lo que la detección debe centrarse en actividad precursora, no en el impacto final.

5.5. Recomendaciones Generales

Al ejecutar el malware se encriptan gran cantidad de directorios, incluyendo la ruta ProgramFiles, por lo cual es importante que se guarden todas las herramientas y capturas en System32 o en cualquier otra ruta excluida por las muestras analizadas, esto con el objetivo de que el Ransomware no encripte los hallazgos y se pueda verificar y analizar la información obtenida sin inconvenientes; en System32 el malware no realiza cambios porque, como se confirmó en el análisis estático, la muestra lo excluye del proceso de encriptación y eso debido a que si se modifican archivos sensibles de Windows el sistema se rompería por completo y el ciberdelincuente no podrá cumplir con su objetivo final.

Para facilitar el análisis de los archivos que cambian en el sistema después de ejecutarse la muestra de ransomware es de gran ayuda contar con una herramienta que haga comparación, así como RegShot, pero a nivel de carpetas, es decir, que mapee todas las carpetas o directorios de Windows en una primera captura, y luego de ejecutar el malware realizar una segunda captura y que se muestre de forma visual todo lo que se modificó, eliminó, creó, etc.

Automatizar la recolección de artefactos con herramientas como Velociraptor, que demostraron ser fundamentales para capturar evidencia en memoria, MFT, Prefetch y USN Journal. Con el objetivo de reducir grandemente el riesgo de perder información crítica por apagados inesperados, reinicios o cifrado adicional.

Dado que Black Basta carga sus funciones criptográficas personalizadas directamente en RAM, se recomienda: hacer dumps de memoria del proceso mientras está activo, utilizar YARA sobre imágenes de memoria para detectar cadenas internas únicas y monitorear funciones matemáticas internas asociadas al cifrado RSA con el fin de identificar trazas del ransomware aun cuando no deja artefactos persistentes en disco.

Finalmente, se aconseja documentar y guardar de forma centralizada todas las evidencias conseguidas en el laboratorio, reglas YARA, eventos característicos y firmas de comportamiento. Esto con el objetivo de crear una base de conocimiento reutilizable para futuras investigaciones, mejorar la capacidad de respuesta ante incidentes y que en el futuro sirva como guía para construir nuevos perfiles mínimos de detección.

Referencias

AUTOPSY. (2025). Obtenido de AUTOPSY: https://www.autopsy.com/

Brandt A, M. P. (2020). *Maze Attackers Adopt Ragnar Locker Virtual Machine Technique.*

Brinkmann, M. (6 de Febrero de 2016). *g hacks.net*. Obtenido de https://www.ghacks.net/2016/02/06/detect-it-easy/

Brumm, C. (4 de Agosto de 2021). *My learnings on Microsoft Defender for Endpoint and Exclusions.* Obtenido de Medium: https://medium.com/codex/my-learnings-on-microsoft-defender-for-endpoint-and-exclusions-ddacf2fdd047

CISA, C. a. (08 de Noviembre de 2024). *#StopRansomware: Black Basta*. Obtenido de https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-131a

Command and Scripting Interpreter: Windows Command Shell (T1059.003). MITRE ATT&CK. (2024). Obtenido de MITRE Corporation: https://attack.mitre.org/techniques/T1059/003/

Crawford, D. (28 de Diciembre de 2023). *What is ChaCha20?* Obtenido de https://protonvpn.com/blog/chacha20?srsltid=AfmBOooB-pz3LxQ-SiMpY1bXkwV2Er7GktCNq2WEg7uRxVzWxurSElez

Cybersecurity & Infrastructure Security Agency (CISA), F. B.-S.-I. (2024). *#StopRansomware: Black Basta.* Obtenido de https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-131a

Finkel. (2007). *Advanced Windows Debugging.* Addison-Wesley.

Fox, N. (6 de Abril de 2023). *Varonis*. Obtenido de https://www.varonis.com/blog/how-to-use-ghidra

Franco, D. (30 de Noviembre de 2022). *Digital Forensic Analysis of Ransomwares for Identification and Binary Extraction of Cryptographic Keys*. Obtenido de

https://www.heraldopenaccess.us/openaccess/digital-forensic-analysis-of-ransomwares-for-identification-and-binary-extraction-of-cryptographic-keys

g0tmi1k. (18 de Junio de 2025). *KALI*. Obtenido de KALI: https://www.kali.org/docs/introduction/what-is-kali-linux/

Gobierno de Canarias. (s.f.). Obtenido de Gobierno de Canarias: https://www3.gobiernodecanarias.org/medusa/ecoescuela/seguridad/ciudadania-y-seguridad-tic/principios-legales/software-libre/ubuntu-linux/

Hendrikx, M. (20 de Octubre de 2014). *How To Geek*. Obtenido de How To Geek: https://www.howtogeek.com/198679/how-to-use-regshot-to-monitor-your-registry/

Ibáñez, J. T. (2025). *Iníciate en Análisis Forense Digital.*

INCIBE. (Junio de 2025). Obtenido de INCIBE: https://www.incibe.es/ciudadania/herramientas/wireshark

INCIBE. (20 de Marzo de 2025). *Claves forenses en Windows: artefactos esenciales para la investigación digital*. Obtenido de https://www.incibe.es/incibe-cert/blog/claves-forenses-en-windows-artefactos-esenciales-para-la-investigacion-digital

Jamil Ispahany, M. R. (2024). iCNN-LSTM: A batch-based incremental ransomware detection system using Sysmon. *Cryptography and Security (cs.CR)*.

Jim Hodsworth, M. K. (5 de septiembre de 2024). *¿Qué es el ransomware como servicio (RaaS)?* Obtenido de https://www.ibm.com/mx-es/think/topics/ransomware-as-a-service

KaliTools. (30 de Agosto de 2021). Obtenido de https://en.kali.tools/?p=1644

Kosinski, M. (4 de junio de 2024). *¿Qué es el ransomware?* Obtenido de https://www.ibm.com/es-es/think/topics/ransomware

Learn Microsoft. (07 de Julio de 2025). Obtenido de Learn Microsoft: https://learn.microsoft.com/es-es/powershell/scripting/overview?view=powershell-7.5

Paliwal, A. (6 de Mayo de 2025). *What Is Black Basta Ransomware and How to Mitigate Attack*. Obtenido de https://blog.qualys.com/vulnerabilities-threat-research/2024/09/19/black-basta-ransomware-what-you-need-to-know

Russinovich, & Solomon. (2017). *Windows Internals.* Microsoft Press.

Russinovich, M. (6 de Febrero de 2024). *Learn Microsoft*. Obtenido de Learn Microsoft: https://learn.microsoft.com/es-es/sysinternals/downloads/autoruns

Russinovich, M. (20 de junio de 2024). *Learn Microsoft*. Obtenido de Learn Microsoft: https://learn.microsoft.com/es-es/sysinternals/downloads/procmon

Russinovich, M., & Garnier, T. (23 de Julio de 2024). *Learn Microsoft*. Obtenido de Learn Microsoft: https://learn.microsoft.com/es-es/sysinternals/downloads/sysmon

Shklar, L., & Rich, R. (s.f.). *Web Application Architecture: Principles, Protocols and Practices.*

softonic. (s.f.). Obtenido de softonic: https://virtualbox.softonic.com/

SOTHIS. (2025). Obtenido de SOTHIS: https://www.sothis.tech/adquisicion-de-memoria-ram-2/

Velociraptor. (2025). Obtenido de Velociraptor: https://docs.velociraptor.app/docs/overview/

VOLATILITY. (s.f.). Obtenido de VOLATILITY: https://volatilityfoundation.org/

YARA. (2022). Obtenido de YARA: https://yara.readthedocs.io/en/latest/

Apéndices

Información del laboratorio OPBASTA-LAB:

https://drive.google.com/drive/folders/16hvzXp-WBmYF14Kf-ic4rI0WjCPWcZ_r?usp=sharing