Maestría en

CIBERSEGURIDAD

Trabajo previo a la obtención de título de
Magister en Ciberseguridad

AUTORES:

BRANDT RAMIREZ CARL EMIL

BRICEÑO ALVAREZ JONATHAN FABRICIO

GAIBOR VELASQUEZ JEFFERSON ANDRES

VITERI ZAMBRANO DENNIS ADRIAN

TUTORES:

Iván Reyes Chacón

Juan Fernando López

TEMA

EVALUACIÓN DE LA DETECTABILIDAD DE TÉCNICAS ANTIFORENSES EN SISTEMAS WINDOWS MEDIANTE ANÁLISIS FORENSE DIGITAL ESTÁNDAR

Quito - Ecuador

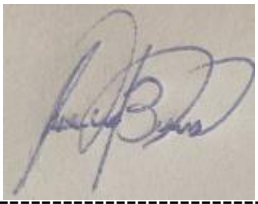
Junio – 20206

REINVENTEMOS EL FUTURO

Certificación de autoría

Nosotros, **Carl Emil Brandt Ramírez, Jonathan Fabricio Briceño Álvarez, Jefferson Andrés Gaibor Velásquez, Dennis Adrián Viteri Zambrano,** 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
Carl Emil Brandt Ramírez



Firma
Jonathan Fabricio Briceño Álvarez

Firma
Jefferson Andrés Gaibor Velásquez

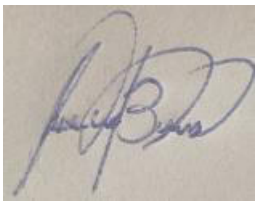
Firma
Dennis Adrián Viteri Zambrano

Autorización de Derechos de Propiedad Intelectual

Nosotros, **Carl Emil Brandt Ramírez, Jonathan Fabricio Briceño Álvarez, Jefferson Andrés Gaibor Velásquez, Dennis Adrián Viteri Zambrano**, en calidad de autores del trabajo de investigación titulado **Evaluación De La Detectabilidad De Técnicas Antiforenses En Sistemas Windows Mediante Análisis Forense Digital Estándar**, 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, (mayo 2026)

Firma
Carl Emil Brandt Ramírez



Firma
Jonathan Fabricio Briceño Álvarez

Firma
Jefferson Andrés Gaibor Velásquez

Firma
Dennis Adrián Viteri Zambrano

APROBACIÓN DE DIRECCIÓN Y COORDINACIÓN DEL PROGRAMA

Nosotros, Alejandro Cortés López e Iván Galo Reyes Chacón, declaramos que: Carl Emil Brandt Ramírez, Jonathan Fabricio Briceño Álvarez, Jefferson Andrés Gaibor Velásquez, Dennis Adrián Viteri Zambrano, son los autores exclusivos de la presente investigación y que esta es original, auténtica y personal de ellos.

Alejandro Cortés
Director/a de la
Maestría en Ciberseguridad

Iván Reyes
Coordinador/a de la
Maestría en Ciberseguridad

DEDICATORIA

Dedicamos este trabajo en primer lugar a nuestras familias, por su apoyo, comprensión y constante motivación en cada etapa de nuestra formación académica. Su confianza y acompañamiento fueron fundamentales para alcanzar esta meta y superar los desafíos que se presentaron a lo largo de este camino.

Asimismo, expresamos nuestra gratitud a nuestra universidad y docentes, quienes mediante su guía, conocimiento y experiencia contribuyeron significativamente a nuestro desarrollo profesional y al fortalecimiento de nuestras capacidades investigativas.

Finalmente, dedicamos este logro a nuestro equipo de trabajo, cuya dedicación, esfuerzo y compromiso hicieron posible la realización de esta investigación. La colaboración, responsabilidad y perseverancia demostrada por cada integrante fueron esenciales para culminar con éxito este proyecto.

AGRADECIMIENTOS

Queremos agradecer a la **Universidad Internacional del Ecuador** por brindarnos la oportunidad de formar parte de la Maestría en Ciberseguridad. Este programa nos permitió fortalecer nuestros conocimientos y adquirir nuevas herramientas que nos ayudaran a desarrollar proyectos para mejorar la ciberseguridad en el país y que de igual manera fueron fundamentales para el desarrollo de esta investigación.

A nuestros docentes, por compartir su experiencia y conocimientos a lo largo de la maestría. Cada asignatura aportó de alguna manera al desarrollo de este trabajo y nos permitió ampliar nuestra perspectiva sobre la ciberseguridad y la informática forense.

A nuestro director y coordinador de tesis, **Alejandro Cortés López** y **Juan Fernando López Tito**, por el tiempo dedicado y las observaciones realizadas durante el desarrollo de esta investigación. Sus recomendaciones fueron importantes para mejorar el contenido y la calidad del trabajo presentado.

También queremos agradecer a nuestras familias por el apoyo que nos brindaron durante esta etapa. Sabemos que el desarrollo de esta investigación requirió tiempo y dedicación, por lo que su respaldo fue fundamental para poder culminar este proyecto.

Finalmente, agradecemos a todas las personas que, de una u otra manera, nos acompañaron durante este proceso académico y aportaron con sus consejos, experiencias o palabras de apoyo para alcanzar esta meta.

ANÁLISIS DE TÉCNICAS ANTIFORENSES

RESUMEN

El presente estudio tuvo como objetivo evaluar la detectabilidad de técnicas antiforenses aplicadas en sistemas operativos Windows mediante un enfoque experimental en un entorno controlado. Estas técnicas, diseñadas para ocultar, alterar o destruir evidencia digital, representan un desafío significativo para los procesos de análisis forense, ya que afectan la integridad y trazabilidad de la información.

La investigación se desarrolló en un laboratorio virtualizado que permite implementar de forma reproducible diversas técnicas antiforenses clasificadas en: manipulación temporal, alteración de artefactos del sistema, ocultación en el sistema de archivos, ocultación avanzada de contenido y destrucción de evidencia. Entre las técnicas analizadas se incluyen el *timestomping*, la eliminación de logs de eventos, la eliminación de archivos Prefetch, el uso de Alternate Data Streams (ADS), la esteganografía y el file wiping.

El estudio utilizó métricas estructuradas como la recuperación de evidencia, la persistencia del rastro digital, la dificultad de detección y el tiempo de análisis. Con base en estas variables, se construyó una matriz de detectabilidad que permite clasificar las técnicas según su impacto y nivel de evasión.

Los resultados permitirán identificar la efectividad de las técnicas antiforenses frente a métodos de análisis forense digital y fortalecer las capacidades de detección en entornos reales.

Palabras Claves: informática forense, técnicas antiforenses, evidencia digital, detectabilidad, Windows

ABSTRACT

This study aims to evaluate the detectability of anti-forensic techniques applied in Windows operating systems through an experimental approach in a controlled environment. These techniques, designed to conceal, alter, or destroy digital evidence, represent a significant challenge for forensic analysis processes, as they affect the integrity and traceability of information.

The research is conducted in a virtualized laboratory that enables the reproducible implementation of various anti-forensic techniques, classified into temporal manipulation, system artifact alteration, file system-level concealment, advanced content hiding, and evidence destruction. The techniques analyzed include timestomping, event log deletion, Prefetch file removal, the use of Alternate Data Streams (ADS), steganography, and file wiping.

The study uses structured metrics such as evidence recovery, persistence of the digital trace, detection difficulty, and analysis time. Based on these variables, a detectability matrix is constructed to classify techniques according to their impact and level of evasion.

The results will help determine the effectiveness of anti-forensic techniques against digital forensic analysis methods and strengthen detection capabilities in real-world environments.

Keywords: digital forensics, anti-forensic techniques, digital evidence, detectability, Windows

TABLA DE CONTENIDOS (Índice)

Contenido

LISTA DE TABLAS (Índice de tablas)

LISTA DE FIGURAS (Índice de figuras)

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Definición del proyecto

El presente trabajo de investigación tiene como finalidad evaluar la detectabilidad de diversas técnicas antiforenses aplicadas en sistemas operativos Windows, analizando su impacto sobre la integridad, disponibilidad y trazabilidad de la evidencia digital en el contexto de un análisis forense.

A pesar de la consolidación de marcos forenses digitales estandarizados, persiste una laguna crítica en la investigación debido a la creciente asimetría entre la rápida automatización de las capacidades antiforenses, como la ejecución sin archivos, el borrado automatizado de registros, la manipulación de marcas de tiempo, y la naturaleza relativamente estática de los protocolos de triaje tradicionales. En los entornos corporativos y adversarios modernos, los ciberdelincuentes y las amenazas persistentes avanzadas (APT) explotan activamente estos mecanismos para socavar la recopilación de datos. Esto introduce una grave vulnerabilidad en la investigación: los examinadores forenses corren el riesgo de basarse en cadenas de pruebas comprometidas, alteradas o incompletas sin identificar que se han desplegado contramedidas antiforenses. En consecuencia, este estudio está motivado por la urgente necesidad de medir empíricamente los límites de las herramientas de análisis estándar del sector, estableciendo con precisión dónde falla el triaje tradicional y definiendo los umbrales de visibilidad necesarios para mitigar la evasión de defensas sofisticadas.

Para abordar este problema de manera estructurada, el estudio implementará un conjunto representativo de técnicas antiforenses dentro de un entorno virtualizado controlado. La taxonomía utilizada para clasificar estas técnicas no responde a criterios arbitrarios, sino que se sustenta firmemente en la literatura académica clásica y en marcos técnicos reconocidos globalmente, alineándose de forma directa con la matriz de tácticas de Evasión de Defensas (Defense Evasion) de MITRE Corporation (2026), así como con los modelos taxonómicos fundamentales propuestos por Rogers (2006, citado en Garfinkel,2007) y Garfinkel (2007). Las técnicas se agrupan en cinco categorías formales: Esta clasificación permite abordar el fenómeno antiforense desde una perspectiva estructurada, facilitando el análisis comparativo entre distintos tipos de manipulación. (Garfinkel, 2007).

- Manipulación Temporal (Timestomping): Alteración maliciosa de los metadatos de tiempo en el sistema de archivos (atributos $STANDARD_INFORMATION y $FILE_NAME dentro de la Tabla Maestra de Archivos o MFT) con el fin de desarticular el análisis de líneas de tiempo (timeline) se llevará a cabo con base en los modelos de detección de inconsistencias temporales propuestos por Minnaard (2014), citado en Alij y Choudali (2019), y en los modelos de identificación automatizada desarrollados por Alij y Choudali (2019).
- Alteración de Artefactos del Sistema: Modificación, deshabilitación o purga selectiva de las estructuras de registro y rastreo nativas de Windows (tales como Event Logs, archivos Prefetch, tablas Shimcache y colmenas de Amcache) diseñadas para registrar el histórico de ejecución de procesos. (MITRE Corporation, 2026).

- Ocultación de Información a Nivel de Sistema de Archivos: Almacenamiento de datos u herramientas en capas ocultas o no convencionales del sistema de archivos, orientadas específicamente al uso de Flujos Alternos de Datos (Alternate Data Streams - ADS), espacio no asignado (unallocated space) y copias de sombra de volumen (VSS), integrando las dimensiones de ocultación descritas por Huebner et al. (2006, citado en Garfinkel, 2007).
- Ocultación Avanzada de Contenido: Uso de técnicas de empaquetado personalizado (packing), codificación y cifrado criptográfico de cargas útiles, junto con técnicas de inyección en memoria (fileless execution), diseñadas para evadir las firmas estáticas y el filtrado por palabras clave durante la respuesta inicial del analista (Garfinkel, 2007; MITRE Corporation, 2026).
- Destrucción de Evidencia: Ejecución de algoritmos de borrado seguro (wiping) y eliminación definitiva de registros específicos del sistema o archivos de usuario, con el propósito de fragmentar y romper de manera irreversible el nexo causal de la prueba digital. (Garfinkel, 2007).

Para evaluar la detectabilidad de las manipulaciones anteriores se utilizará un conjunto de herramientas forenses estándar, cuya selección responde a capacidades técnicas específicas y necesarias para contrarrestar cada técnica antiforense:

- FTK Imager: Se justifica su uso como el estándar de la industria para la adquisición de la imagen forense bit a bit de los discos y la captura de la memoria volátil. Su arquitectura garantiza que el proceso de clonación no altere los metadatos subyacentes del sistema y asegura la integridad absoluta de los datos mediante la verificación de hashes criptográficos (MD5/SHA-1).

- Autopsy: Esta plataforma de análisis digital de código abierto se ha seleccionado por su robustez y el despliegue de módulos de ingesta automatizados capaces de examinar estructuras de bajo nivel. Aporta capacidades fundamentales para desarticular el antiforense, tales como el parseo automático de registros del sistema, el indexado profundo de cadenas de texto cifradas/codificadas, el mapeo de datos eliminados de la MFT y la detección nativa de flujos alternos (ADS). Esto permite al examinador cruzar referencias e identificar inconsistencias (como discrepancias entre los registros temporales de la MFT o saltos anómalos en los logs) que exponen la manipulación.

El núcleo integrador de este proyecto es el diseño de una Matriz de Detectabilidad. Con la finalidad de erradicar la subjetividad en la fase de análisis y establecer variables científicamente medibles, se definen operacionalmente las siguientes cuatro métricas conceptuales:

- Recuperación de Evidencia: Mide cuantitativamente el volumen y cualitativamente la integridad del dato original que el analista es capaz de reconstruir tras el ataque antiforense. Evalúa si el payload o el metadato afectado se recupera de forma íntegra, parcial o si ha quedado permanentemente inaccesible.
- Persistencia del Rastro Digital: Evalúa la tasa de supervivencia de la telemetría secundaria o adyacente del sistema de archivos. Determina si la técnica antiforense borró con éxito todas sus huellas o si dejó "migajas digitales" indirectas (por ejemplo, vacíos temporales en registros cronológicos, inconsistencias de tamaño o firmas de ejecución en la memoria).

- Dificultad de Detección: Cuantifica la complejidad técnica del flujo de investigación requerido por el perito para descubrir e identificar que el sistema fue manipulado. Se escala desde un nivel Bajo (identificable mediante búsquedas automatizadas de palabras clave o firmas) hasta un nivel Alto (requiriendo ingeniería inversa, análisis forense de memoria complejo o edición manual en hexadecimal de estructuras MFT).
- Tiempo de Análisis: Mide el costo operativo y el recurso temporal exacto que un analista forense debe invertir para aislar, auditar y validar la veracidad de la evidencia afectada utilizando los flujos de trabajo estándar.

Para los fines de esta investigación, el término análisis forense digital estándar se refiere al conjunto de procedimientos metodológicos empleados para la identificación, preservación, adquisición, examen, análisis e interpretación de evidencia digital, siguiendo las buenas prácticas reconocidas por el National Institute of Standards and Technology (NIST). De acuerdo con los modelos de investigación digital revisados por NIST, estas actividades permiten garantizar la integridad de la evidencia y la reproducibilidad de los resultados obtenidos durante el proceso pericial (Lyle et al., 2022). En este contexto, la evaluación se realizará mediante el análisis de imágenes forenses de sistemas Windows con sistema de archivos NTFS, aplicando procedimientos orientados a la recuperación de artefactos, análisis de metadatos, construcción de líneas de tiempo, correlación de eventos y generación de hallazgos forenses. Este procedimiento constituirá la línea base de evaluación frente a la cual se medirán los efectos de las técnicas antiforenses estudiadas.

De este modo, la definición del presente proyecto establece un marco de evaluación empírico y sistemático para auditar los límites del análisis forense digital estándar frente a las tácticas modernas de evasión. En el contexto de este estudio, dicho análisis estándar se define operacionalmente como el conjunto de procedimientos estructurados y fases secuenciales adquisición, preservación, análisis y documentación que se rigen por los principios de repetibilidad y rigurosidad metodológica establecidos en los estándares internacionales ISO/IEC 27037 e ISO/IEC 27042. Esta base de evaluación no se limita a la ejecución automatizada de herramientas comerciales o de código abierto, sino que constituye el marco de referencia formal que guiará la búsqueda e interpretación sistemática de artefactos lógicos residuales. Con ello, se busca transformar la evaluación de la persistencia y recuperación de evidencia en un proceso medible y reproducible, contribuyendo directamente a la robustez, fiabilidad y validez pericial de los peritajes informáticos ante incidentes complejos en sistemas operativos Windows.

1.2. Justificación e importancia del trabajo de investigación

Las técnicas antiforenses representan uno de los principales desafíos en el ámbito de la informática forense, ya que están diseñadas para ocultar, alterar o destruir evidencia digital con el objetivo de evadir la detección y dificultar la reconstrucción de eventos. Diversas investigaciones han señalado que estas técnicas buscan interferir en los procesos de adquisición, análisis y preservación de evidencia digital, reduciendo la capacidad de los investigadores para reconstruir de manera precisa los eventos

asociados a un incidente informático (González Arias et al., 2024; Yaacoub et al., 2021).

A diferencia de los mecanismos tradicionales de ocultamiento, las técnicas antiforenses modernas actúan directamente sobre los artefactos del sistema, comprometiendo la integridad, autenticidad y trazabilidad de la información. Esta situación representa un reto significativo para la informática forense, ya que puede afectar la confiabilidad de los hallazgos obtenidos durante una investigación y dificultar la validación de la evidencia digital en entornos corporativos y judiciales, donde la preservación de la integridad de los datos constituye un requisito fundamental para garantizar la validez de los procesos periciales (González Arias et al., 2024).

Un ejemplo representativo es la técnica de Manipulación Temporal (Timestomping), mediante la cual un atacante altera los atributos de fecha y hora de archivos en sistemas NTFS para modificar la secuencia cronológica de eventos registrada por el sistema. Como consecuencia, la reconstrucción de líneas de tiempo forenses puede verse afectada, dificultando la identificación de actividades maliciosas y la correcta interpretación de la evidencia digital durante una investigación (Alji & Chougdali, 2019).

En entornos corporativos y judiciales, la confiabilidad de un análisis forense depende de la correcta identificación, preservación y correlación de artefactos digitales. La evidencia digital constituye un elemento fundamental para la reconstrucción de incidentes de seguridad, investigaciones internas y procesos judiciales relacionados

con delitos informáticos. No obstante, la aplicación de técnicas antiforenses como la manipulación temporal, la alteración de registros del sistema, la ocultación de información en estructuras del sistema de archivos, el uso de métodos avanzados de ocultación de contenido y la destrucción de evidencia introduce un nivel significativo de complejidad que puede afectar la validez de los hallazgos periciales.

La efectividad de estas técnicas ha sido documentada en diversos estudios. Por ejemplo, Joo, Lee y Jeong (2023) evaluaron diez herramientas de borrado seguro de archivos en sistemas Windows y demostraron que estas pueden eliminar permanentemente evidencia digital, dificultando su recuperación mediante procedimientos forenses convencionales. Asimismo, los autores identificaron rastros residuales asociados a la ejecución de dichas herramientas en distintos artefactos del sistema operativo, evidenciando que, aunque las técnicas antiforenses pueden reducir significativamente la disponibilidad de evidencia, también pueden generar indicadores útiles para su detección durante una investigación forense.

La relevancia de las técnicas antiforenses ha sido ampliamente documentada en la literatura científica reciente. Yaacoub et al. (2021) realizaron una revisión de las principales técnicas antiforenses utilizadas para dificultar las investigaciones digitales, identificando mecanismos orientados a la destrucción, ocultación y manipulación de evidencia que afectan procesos como la recuperación de información, el análisis de artefactos y la reconstrucción cronológica de eventos. Por su parte, González Arias et al. (2024), mediante una revisión sistemática de la literatura, concluyeron que técnicas

como la manipulación de metadatos, el borrado seguro de archivos y la alteración de registros del sistema continúan representando desafíos significativos para los investigadores forenses debido a su capacidad para comprometer la integridad, autenticidad y trazabilidad de la evidencia digital. Estos hallazgos respaldan la necesidad de evaluar de forma sistemática el impacto de las técnicas antiforenses sobre los artefactos de sistemas Windows, objetivo central del presente proyecto.

Desde una perspectiva académica, el estudio de las técnicas antiforenses requiere ir más allá de las aproximaciones teóricas tradicionales. La constante evolución de los mecanismos utilizados para ocultar, modificar o eliminar evidencia digital genera nuevos desafíos para la informática forense, lo que hace necesario desarrollar investigaciones que permitan comprender mejor su comportamiento e impacto sobre los procesos de análisis digital. En este contexto, resulta fundamental generar evidencia empírica que contribuya a ampliar el conocimiento existente sobre las características, efectos y limitaciones de estas técnicas, aportando información útil para futuras investigaciones y para el fortalecimiento de las metodologías de análisis forense (González Arias et al., 2024; Yaacoub et al., 2021).

Asimismo, la incorporación de métricas estructuradas —como el nivel de recuperación de evidencia, la persistencia del rastro digital, la dificultad de detección y el tiempo de análisis requerido— permite reducir la subjetividad en la evaluación y aportar mayor rigor científico al estudio. La construcción de una matriz de detectabilidad basada en

estas métricas constituye un aporte metodológico relevante, ya que proporciona un marco sistemático para clasificar y comparar el impacto de las técnicas antiforenses.

Desde el punto de vista práctico, el enfoque del proyecto resulta pertinente al centrarse en los sistemas operativos Windows 10 y Windows 11, debido a su amplia utilización en entornos corporativos, académicos y domésticos. Datos recientes de StatCounter indican que Windows 10 y Windows 11 concentran la gran mayoría de las instalaciones de sistemas operativos Windows a nivel mundial, manteniéndose como las plataformas de escritorio predominantes tanto en usuarios finales como en organizaciones. Durante 2025, ambos sistemas representaron conjuntamente más del 90 % de las instalaciones de Windows en equipos de escritorio, consolidando su relevancia para estudios relacionados con la seguridad informática y el análisis forense digital.

Estos sistemas operativos generan una gran cantidad de artefactos digitales relevantes para las investigaciones forenses, incluyendo registros del sistema, eventos de seguridad, metadatos, historiales de actividad y archivos de configuración que pueden ser analizados para reconstruir eventos y detectar posibles alteraciones. Asimismo, la amplia adopción de ambas plataformas en organizaciones públicas y privadas permite que los resultados obtenidos en esta investigación tengan aplicabilidad en escenarios reales de análisis forense digital. En este contexto, la evaluación de técnicas antiforenses sobre sistemas Windows proporciona una base adecuada para analizar su impacto sobre la evidencia digital y valorar la efectividad de los mecanismos de

detección empleados durante una investigación forense (Casey, 2024; González Arias et al., 2024).

En síntesis, este trabajo no solo contribuye a la comprensión del comportamiento de las técnicas antiforenses, sino que también aporta herramientas metodológicas para mejorar su identificación, fortaleciendo así la capacidad de respuesta ante escenarios donde la evidencia digital ha sido manipulada de forma deliberada.

1.3. Alcance

El presente estudio se delimita al análisis experimental de técnicas antiforenses aplicadas en sistemas operativos Windows, con el objetivo de evaluar su impacto sobre la evidencia digital y su detectabilidad mediante herramientas forenses estándar.

1.3.1. Entorno tecnológico

El desarrollo del proyecto se realizará en un laboratorio virtualizado bajo sistemas operativos Microsoft Windows, empleando diferentes configuraciones de máquinas virtuales utilizadas por los integrantes del equipo. Las características generales del entorno experimental fueron las siguientes:

- **Sistema Operativo:** Microsoft Windows 10 Home 22H2 y Microsoft Windows 11 (Home y Pro).
- **Sistema de archivos:** NTFS.
- **Entorno de ejecución:** Máquinas virtuales implementadas mediante VMware Workstation y Oracle VirtualBox. La utilización de ambas plataformas responde a la necesidad de ejecutar los escenarios experimentales en entornos de virtualización

ampliamente utilizados, lo que permite validar la ejecución de técnicas antiforenses en diferentes plataformas de virtualización.

- **Procesamiento:** Dos procesadores virtuales asignados por máquina.
- **Memoria RAM:** Entre 4 GB y 8 GB.
- **Almacenamiento virtual:** Entre 60 GB y 80 GB.
- **Herramientas utilizadas:** VMware Workstation, Oracle VirtualBox, Windows PowerShell, CMD, Event Viewer, wevtutil, FTK Imager, Autopsy Forensic Browser, WinPrefetchView, SDelete, Eraser, Recuva y otras utilidades nativas de Windows empleadas para la ejecución de las técnicas antiforenses y el análisis de la evidencia generada durante las pruebas.
- **Número de pruebas:** Seis escenarios experimentales independientes, correspondientes a las seis técnicas antiforenses analizadas.
- **Tipo de análisis:** Análisis forense de disco (post-mortem).

El uso de un entorno virtualizado permite controlar las variables del experimento, garantizar la reproducibilidad de los resultados y evitar afectar sistemas reales.

1.3.2. Técnicas antiforenses analizadas

El estudio contempla la implementación y análisis de seis técnicas antiforenses, agrupadas en cinco categorías:

a. Manipulación temporal

- Timestomping

b. Alteración de artefactos del sistema

- Eliminación de logs de eventos
- Eliminación de archivos Prefetch

c. Ocultación de información a nivel de sistema de archivos

- Alternate Data Streams (ADS)

d. Ocultación avanzada de contenido

- Esteganografía

e. Destrucción de evidencia

- File wiping

Cada técnica será aplicada de forma independiente con el fin de aislar sus efectos y facilitar su evaluación dentro del proceso de análisis forense.

1.3.3. Delimitaciones del estudio

El presente trabajo no contempla:

- Técnicas antiforenses aplicadas a sistemas Linux, macOS o dispositivos móviles,
- Análisis forense de memoria RAM (live forensics)
- Técnicas antiforenses basadas en malware, rootkits o bootkits.
- Evaluaciones en infraestructura empresariales o entornos productivos reales.
- Análisis de tráfico de red o evidencia obtenida mediante monitoreo de red.
- Comparaciones exhaustivas entre herramientas forenses comerciales y de código abierto.

El estudio se centra exclusivamente en la generación, alteración, ocultación y destrucción de evidencia digital almacenada en disco dentro de un entorno controlado de laboratorio.

En conjunto, estas condiciones permiten evaluar de manera controlada el impacto de distintas técnicas antiforenses sobre la evidencia digital y las capacidades de detección de las herramientas forenses utilizadas durante la investigación.

1.4.**Objetivos**

1.4.1. Objetivo general

Evaluar la detectabilidad de técnicas antiforenses aplicadas en sistemas Windows mediante pruebas realizadas en un ambiente controlado, analizando el impacto que generan sobre la evidencia digital y la capacidad de los métodos forenses para detectarlas, con el fin de construir una matriz de evaluación que permita compararlas de acuerdo con su nivel de detectabilidad.

1.4.2. Objetivo especifico

1. Seleccionar técnicas antiforenses representativas a partir de la literatura especializada para definir el conjunto de técnicas que será implementado en el laboratorio.
2. Implementar técnicas antiforenses seleccionadas en un ambiente controlado y documentar los cambios que producen sobre la evidencia digital y los artefactos del sistema.
3. Analizar los artefactos generados en el sistema para identificar rastros, inconsistencias y evidencias que permitan reconocer la aplicación de cada técnica antiforense.
4. Evaluar la capacidad de las herramientas utilizadas durante la investigación para detectar los cambios producidos por las técnicas antiforenses implementadas.
5. Determinar el nivel de detectabilidad de cada técnica utilizando las métricas definidas en la investigación, considerando la recuperación de evidencia, la persistencia del rastro digital, la dificultad de detección y el tiempo de análisis.

6. Construir una matriz de evaluación con los resultados obtenidos para comparar las técnicas antiforenses de acuerdo con su impacto sobre la evidencia digital y su nivel de detectabilidad.

CAPÍTULO 2

2. REVISIÓN DE LITERATURA

2.1. Estado del Arte

El análisis forense digital constituye una disciplina fundamental para la identificación, preservación, análisis y presentación de evidencia digital en investigaciones relacionadas con incidentes de seguridad y delitos informáticos. Sin embargo, la creciente sofisticación de las técnicas utilizadas para ocultar o manipular información ha generado nuevos desafíos para los investigadores forenses, especialmente en sistemas operativos ampliamente utilizados como Windows (Yaacoub et al., 2021; González Arias et al., 2024).

Diversos estudios han señalado que las técnicas antiforenses representan una de las principales amenazas para la efectividad de los procesos de análisis forense, debido a que están diseñadas para dificultar la adquisición, recuperación y correlación de evidencia digital. Estas técnicas pueden actuar sobre diferentes artefactos del sistema mediante la modificación de marcas temporales, la alteración de registros, la ocultación de archivos o la eliminación de información relevante para la investigación (Yaacoub et al., 2021).

La literatura reciente ha propuesto diferentes clasificaciones de las técnicas antiforenses con el objetivo de comprender su funcionamiento y facilitar su análisis. Entre las categorías más estudiadas se encuentran la manipulación temporal, la alteración de artefactos del sistema, la ocultación de información y la destrucción de evidencia digital. Estas clasificaciones han permitido establecer marcos conceptuales para analizar el impacto de las técnicas antiforenses sobre los procesos de investigación digital (González Arias et al., 2024).

Asimismo, diversas investigaciones han evaluado herramientas y metodologías orientadas a la detección de rastros antiforenses, evidenciando que muchas de estas técnicas no eliminan completamente la evidencia digital, sino que generan artefactos residuales que pueden ser identificados mediante procedimientos especializados de análisis forense. No obstante, los resultados reportados varían considerablemente según la técnica utilizada, el tipo de artefacto afectado y las herramientas empleadas durante la investigación (Yaacoub et al., 2021).

En el contexto de los sistemas Windows, la evidencia digital se encuentra distribuida en múltiples artefactos, incluyendo registros del sistema, metadatos de archivos, eventos de seguridad y configuraciones internas del sistema operativo. Debido a la relevancia de estos artefactos dentro de las investigaciones forenses, diversos trabajos han destacado la necesidad de analizar cómo las técnicas antiforenses afectan su integridad y qué mecanismos permiten detectar alteraciones realizadas de forma deliberada (Casey, 2024).

A pesar de los avances reportados en la literatura, se observa una limitada disponibilidad de estudios experimentales que comparen de manera sistemática la detectabilidad de diferentes técnicas antiforenses bajo criterios cuantificables y reproducibles. La mayoría de las investigaciones se enfocan en describir las técnicas o analizar casos particulares, pero existe una menor atención hacia modelos que permitan evaluar y comparar objetivamente su nivel de detección. Esta brecha de conocimiento justifica el desarrollo de una matriz de detectabilidad basada en métricas de evaluación, orientada a determinar la persistencia de artefactos, la recuperación de evidencia y la dificultad de detección asociada a cada técnica antiforense aplicada en sistemas Windows.

2.2. Marco Teórico

2.2.1. Informática Forense

La informática forense es una disciplina especializada dentro del ámbito de la ciberseguridad que se encarga de identificar, preservar, adquirir, analizar y presentar evidencia digital obtenida de sistemas informáticos, con el propósito de reconstruir eventos y apoyar procesos de investigación (Casey, 2024; NIST, 2024). Su principal objetivo es determinar qué ocurrió en un sistema, cómo ocurrió y quién pudo haber participado en las acciones analizadas, garantizando en todo momento la integridad y validez de la evidencia digital recolectada.

El crecimiento de incidentes relacionados con ataques informáticos, accesos no autorizados, fraude digital y manipulación de información ha incrementado la importancia de la informática forense en entornos corporativos, gubernamentales y judiciales. En este contexto, las investigaciones forenses permiten identificar actividades maliciosas, reconstruir secuencias de eventos y generar evidencia técnica que pueda ser utilizada en procesos legales o administrativos (Casey, 2024).

El proceso de análisis forense digital generalmente comprende varias etapas fundamentales: identificación, preservación, adquisición, análisis, documentación y presentación de resultados. La etapa de identificación consiste en localizar posibles fuentes de evidencia digital; la preservación busca proteger la integridad de la información para evitar alteraciones; la adquisición implica la obtención de copias forenses de los sistemas analizados; el análisis se enfoca en el examen y correlación de artefactos digitales; y la documentación y presentación permiten registrar los hallazgos obtenidos y comunicar los resultados de la investigación. Estas etapas garantizan que la evidencia digital mantenga su integridad durante todo el proceso forense (Carrier, 2005; NIST, 2024).

Dentro de un sistema operativo, la evidencia digital puede encontrarse en diferentes artefactos, tales como registros de eventos, metadatos de archivos, historial de ejecución de aplicaciones, estructuras del sistema de archivos y otros elementos generados por el sistema o el usuario. El análisis de estos artefactos permite

reconstruir actividades realizadas en un equipo y establecer relaciones entre eventos relevantes para una investigación (Carrier, 2005).

Sin embargo, el proceso de análisis forense puede verse afectado por la aplicación de técnicas antiforenses, cuyo propósito es ocultar, alterar o eliminar evidencia digital para dificultar la investigación. Estas técnicas representan un desafío significativo para los investigadores, ya que pueden comprometer la confiabilidad de los resultados obtenidos y dificultar la reconstrucción precisa de los eventos analizados (Yaacoub et al., 2021; González Arias et al., 2024).

En sistemas Windows, el análisis forense adquiere especial relevancia debido a la gran cantidad de artefactos digitales generados por el sistema operativo, incluyendo registros de eventos, metadatos de archivos, configuraciones internas y rastros de actividad de los usuarios. Para la adquisición y análisis de evidencia digital se emplean herramientas como FTK Imager y Autopsy. FTK Imager permite realizar imágenes forenses y obtener copias de dispositivos de almacenamiento preservando la integridad de la información original, mientras que Autopsy facilita el análisis de artefactos digitales, la recuperación de evidencia y la correlación de información obtenida durante la investigación.

No obstante, la efectividad de estas herramientas puede verse limitada frente a determinadas técnicas antiforenses que modifican, ocultan o eliminan artefactos digitales. Aunque permiten identificar rastros residuales y posibles inconsistencias en la evidencia, la capacidad de detección puede variar dependiendo de la técnica aplicada y del tipo de

artefacto afectado. Esta situación evidencia la necesidad de evaluar de forma sistemática el impacto de las técnicas antiforenses sobre la evidencia digital y determinar el nivel de detectabilidad que presentan en entornos Windows.

En este sentido, la informática forense no solo cumple un rol técnico dentro de las investigaciones digitales, sino que también constituye un elemento fundamental para garantizar la confiabilidad, integridad y trazabilidad de la evidencia digital en escenarios donde pueden existir intentos deliberados de evasión o manipulación de la información.

2.2.2. Evidencia Digital

La evidencia digital se refiere a cualquier información almacenada, transmitida o generada por dispositivos electrónicos que puede ser utilizada para demostrar la ocurrencia de un hecho o apoyar un proceso de investigación. En el ámbito de la informática forense, la evidencia digital constituye el elemento central del análisis, ya que permite reconstruir eventos, identificar actividades realizadas dentro de un sistema y establecer relaciones entre usuarios, archivos y acciones ejecutadas (Casey, 2024).

La evidencia digital puede encontrarse en múltiples fuentes, incluyendo discos duros, sistemas operativos, registros de eventos, archivos temporales, metadatos, historial de navegación, memoria del sistema y dispositivos de almacenamiento externo. En sistemas Windows, artefactos como los logs de eventos, archivos Prefetch, estructuras NTFS y atributos temporales de archivos representan fuentes relevantes de información para los procesos de análisis forense.

A diferencia de la evidencia física tradicional, la evidencia digital posee características particulares que requieren procedimientos especializados para su preservación y análisis. De acuerdo con Casey (2024) y Nelson, Phillips y Steuart (2019), entre las principales características se encuentran:

- Volatilidad: Cierta información puede perderse rápidamente debido a cambios en el sistema o a la interrupción de su funcionamiento.
- Facilidad de alteración: Los datos digitales pueden ser modificados, ocultados o eliminados mediante acciones intencionales o accidentales.
- Duplicabilidad: La información puede copiarse sin afectar el contenido original, permitiendo generar réplicas exactas para análisis forense.
- Persistencia: Determinados artefactos pueden permanecer almacenados en el sistema incluso después de la eliminación de archivos o aplicaciones.

Debido a estas características, uno de los principios fundamentales de la informática forense es garantizar la integridad de la evidencia digital durante todo el proceso de investigación. Esto implica que la información recolectada debe preservarse de manera que los que los resultados obtenidos sean confiables, verificables y reproducibles.

Para preservar la integridad de la evidencia, es común utilizar procedimientos de adquisición forense que permiten generar copias exactas de los sistemas analizados. Estas copias, conocidas como imágenes forenses, permiten realizar el análisis sin modificar la información original y mantienen la validez de los hallazgos obtenidos (NIST, 2014).

Dentro de un proceso forense, la evidencia digital debe cumplir criterios fundamentales ampliamente aceptados en la literatura especializada (Casey, 2024; NIST, 2014):

- Integridad: La información no debe ser modificada durante las etapas de adquisición, almacenamiento y análisis.
- Autenticidad: La evidencia debe corresponder al origen real del sistema o dispositivo analizado.
- Trazabilidad: Todas las acciones realizadas sobre la evidencia deben encontrarse documentadas y ser verificables.
- Disponibilidad: La información debe mantenerse accesible para su análisis, validación y revisión posterior.

Las técnicas antiforenses representan una amenaza directa para estos principios, ya que buscan alterar, ocultar o destruir artefactos digitales con el propósito de dificultar su recuperación y análisis. Técnicas como el timestomping, la eliminación de registros de eventos, la supresión de archivos Prefetch, el uso de Alternate Data Streams (ADS), la esteganografía y el borrado seguro de archivos pueden afectar significativamente la integridad, disponibilidad y trazabilidad de la evidencia digital.

En el contexto de esta investigación, la evidencia digital constituye el principal objeto de análisis y el medio mediante el cual se evaluará el impacto de las técnicas antiforenses estudiadas. La detectabilidad de dichas técnicas será determinada mediante la identificación de artefactos residuales, inconsistencias en metadatos, registros recuperables y otros indicadores forenses obtenidos a través de herramientas de análisis. De esta manera, la evidencia digital no solo representa la fuente de información examinada, sino también el criterio fundamental para medir la efectividad de las técnicas antiforenses y la capacidad de las herramientas forenses para detectar su aplicación.

2.2.3. Técnicas Antiforenses

Las técnicas antiforenses corresponden a un conjunto de métodos utilizados para ocultar, modificar o eliminar evidencia digital con el propósito de dificultar su identificación durante un proceso de análisis forense. Su principal objetivo es reducir la capacidad de los investigadores para obtener información confiable sobre las actividades realizadas en un sistema, afectando la reconstrucción de eventos y la interpretación de la evidencia disponible. (Harris, 2006)

De acuerdo con Harris (2006), la antiforensia comprende aquellas acciones destinadas a comprometer la disponibilidad o el valor probatorio de la evidencia digital. En la práctica, estas técnicas pueden aplicarse sobre diferentes componentes del sistema operativo, modificando información que normalmente es utilizada por los analistas para determinar qué ocurrió, cuándo ocurrió y cómo se ejecutaron determinadas acciones dentro de un equipo.

En sistemas Windows, gran parte del análisis forense se basa en la revisión de artefactos digitales generados por el propio sistema operativo. Estos artefactos contienen información relacionada con la ejecución de programas, modificaciones de archivos, actividades de los usuarios y eventos registrados durante el funcionamiento del sistema. Debido a su importancia dentro de una investigación, suelen convertirse en uno de los principales objetivos de las técnicas antiforenses.

Entre los artefactos más relevantes se encuentran los registros de eventos de Windows (Event Logs), los atributos temporales de archivos y directorios, los archivos Prefetch, los flujos alternos de datos del sistema NTFS (Alternate Data Streams - ADS) y el espacio no asignado del disco. La manipulación de estos elementos puede afectar la

integridad y trazabilidad de la evidencia digital, dificultando la correlación entre los diferentes hallazgos obtenidos durante el análisis. (Carrier, 2005; Casey, 2024)

Por ejemplo, la modificación de timestamps mediante técnicas de timestomping puede alterar la secuencia cronológica de los eventos registrados en el sistema. De forma similar, la eliminación de logs puede ocultar actividades previamente registradas, mientras que la eliminación de archivos Prefetch puede dificultar la identificación de aplicaciones ejecutadas por el usuario. Por otra parte, los Alternate Data Streams permiten almacenar información de forma oculta dentro del sistema de archivos NTFS, y técnicas como la esteganografía posibilitan ocultar contenido dentro de archivos aparentemente legítimos. Finalmente, herramientas de file wiping buscan eliminar evidencia mediante procesos de sobrescritura que dificultan o impiden su recuperación posterior. (Casey, 2024; Yaacoub et al., 2021)

A pesar de que estas técnicas buscan ocultar rastros de actividad, en muchos casos continúan existiendo artefactos residuales o inconsistencias que pueden ser identificadas mediante herramientas de análisis forense. Por esta razón, la detección de actividades antiforenses suele requerir la correlación de múltiples fuentes de evidencia y la búsqueda de anomalías en los registros, metadatos y estructuras del sistema de archivos. (Casey, 2024; González Arias et al., 2024)

Tabla 1.

Principales artefactos analizados en esta investigación

Artefacto de Windows	**Técnica asociada**	**Posible afectación**	**Método de detección**
Timestamps de archivos NTFS	Timestomping	Alteración de fechas de creación, modificación y acceso	Análisis de la MFT y correlación de metadatos
Event Logs	Eliminación de logs	Pérdida de registros de actividad	Identificación de eventos faltantes e inconsistencias
Archivos Prefetch	Eliminación de Prefetch	Ocultamiento de evidencia de ejecución	Correlación con otros artefactos del sistema
Alternate Data Streams (ADS)	Ocultación de información	Almacenamiento oculto de datos	Identificación de flujos alternos NTFS
Archivos multimedia	Esteganografía	Ocultación de contenido	Herramientas de análisis y detección de esteganografía
Espacio no asignado del disco	File Wiping	Destrucción o sobrescritura de evidencia	Identificación de patrones de sobrescritura

Nota. Elaboración propia con base en Harris (2006), Carrier (2005), Casey (2024), Yaacoub et al. (2021) y González Arias et al. (2024).

2.2.4. Clasificación de técnicas antiforenses

Las técnicas antiforenses pueden clasificarse de distintas formas dependiendo del tipo de manipulación que realizan sobre la evidencia digital. A lo largo de los años, diversos autores han propuesto diferentes enfoques para agrupar estas técnicas, considerando aspectos como el tipo de información afectada, el mecanismo utilizado para ocultar evidencia o el impacto generado sobre el proceso de análisis forense (Harris, 2006).

Aunque no existe una clasificación única aceptada de manera universal, la mayoría de los estudios coinciden en que las técnicas antiforenses tienen como finalidad dificultar la obtención, análisis o interpretación de la evidencia digital mediante la alteración, ocultación o eliminación de información relevante para una investigación (Casey, 2024; Yaacoub et al., 2021).

Para esta investigación se adoptó una clasificación basada en el tipo de manipulación realizada sobre los artefactos y la evidencia digital presentes en sistemas Windows. Esta clasificación permite organizar de mejor manera las técnicas analizadas y facilita la comparación de sus efectos durante el desarrollo de las pruebas realizadas en el laboratorio.

Las técnicas seleccionadas fueron agrupadas en cinco categorías: manipulación temporal, alteración de artefactos del sistema, ocultación de información a nivel de sistema de archivos, ocultación avanzada de contenido y destrucción de evidencia. Cada una de

estas categorías representa una forma diferente de afectar la información que normalmente es utilizada durante un proceso de análisis forense.

La categoría de manipulación temporal incluye técnicas orientadas a modificar fechas y horas asociadas a archivos y directorios. La alteración de artefactos del sistema comprende acciones dirigidas a eliminar o modificar información generada por el sistema operativo, como registros de eventos o archivos Prefetch. Por su parte, la ocultación de información a nivel de sistema de archivos aprovecha características propias del sistema NTFS para almacenar datos de forma menos visible para el usuario. La ocultación avanzada de contenido utiliza mecanismos que permiten esconder información dentro de otros archivos aparentemente legítimos, mientras que la destrucción de evidencia agrupa técnicas cuyo objetivo es eliminar información de forma que su recuperación resulte difícil o imposible mediante herramientas forenses convencionales.

La selección de estas cinco categorías se realizó considerando tanto la literatura especializada como las características de las técnicas implementadas en el presente estudio. Esta organización permite agrupar de forma coherente las técnicas de Timestomping, eliminación de logs de eventos, eliminación de archivos Prefetch, Alternate Data Streams (ADS), esteganografía y file wiping, facilitando el análisis de su impacto sobre la evidencia digital y su posterior evaluación en términos de detectabilidad.

Figura 1

Clasificación de técnicas antiforenses

Nota. La figura muestra las principales categorías de técnicas antiforenses empleadas para dificultar la adquisición, preservación y análisis de evidencias digitales. Elaboración propia.

2.2.5. Manipulación Temporal

La manipulación temporal comprende aquellas técnicas antiforenses orientadas a modificar la información de fecha y hora asociada a archivos, directorios o eventos del sistema. El objetivo de estas técnicas es alterar la secuencia cronológica de las actividades registradas, dificultando la reconstrucción de eventos durante un proceso de análisis forense. Debido a que gran parte de las investigaciones digitales se apoyan en líneas de tiempo para determinar qué ocurrió y en qué momento ocurrió, la alteración de

información temporal puede afectar significativamente la interpretación de la evidencia digital (Casey, 2024).

Dentro de esta categoría, una de las técnicas más conocidas es el Timestomping, la cual consiste en modificar los atributos temporales de un archivo con el fin de ocultar su verdadera fecha de creación, modificación o acceso. Esta técnica es utilizada frecuentemente para hacer que un archivo parezca más antiguo o más reciente de lo que realmente es, generando inconsistencias en los registros temporales utilizados durante una investigación forense (Harris, 2006).

En sistemas de archivos NTFS, cada archivo almacena información temporal relacionada con su creación, modificación y último acceso. Estos valores son conocidos comúnmente como atributos MAC (Modified, Accessed y Created). Durante un análisis forense, estos atributos permiten establecer relaciones entre archivos, actividades de usuarios y eventos registrados por el sistema operativo. Por esta razón, suelen ser uno de los principales objetivos de las técnicas de manipulación temporal (Carrier, 2005).

Aunque el Timestomping puede modificar los atributos temporales visibles de un archivo, existen otros artefactos del sistema que pueden conservar información útil para identificar posibles alteraciones. Uno de los más importantes es la Master File Table (MFT), estructura central del sistema de archivos NTFS que almacena metadatos asociados a cada archivo y directorio. Dentro de la MFT se encuentran atributos como $STANDARD_INFORMATION y $FILE_NAME, los cuales pueden contener valores temporales diferentes cuando se ha realizado una modificación deliberada de los timestamps (Carrier, 2005).

La correlación entre estas fuentes permite detectar discrepancias temporales que podrían indicar la aplicación de técnicas antiforenses. Por ejemplo, puede existir una diferencia entre la fecha de creación registrada en un archivo y la información almacenada en otros artefactos que evidencien una actividad más reciente.

Desde el punto de vista forense, la detección de Timestomping no se basa únicamente en la revisión de los atributos temporales visibles, sino en la comparación de múltiples fuentes de evidencia. La presencia de inconsistencias entre timestamps, registros de actividad y metadatos del sistema constituye uno de los principales indicadores utilizados para identificar manipulaciones temporales (Casey, 2024).

Debido a su capacidad para alterar la línea temporal de los eventos sin modificar necesariamente el contenido de los archivos, el Timestomping representa una técnica antiforense relevante dentro de los sistemas Windows. Su estudio permite comprender las limitaciones de los análisis basados exclusivamente en fechas de archivos y resalta la importancia de correlacionar diferentes artefactos durante una investigación forense digital.

Tabla 2.

Tecnica antiforense Timestomping

Artefacto	Información relevante	Utilidad en la detección
$STANDARD_INFORMATION	Fechas MAC	Comparación de atributos temporales y detección de inconsistencias
$FILE_NAME	Fechas asociadas al archivo	Comparación con $STANDARD_INFORMATION para identificar discrepancias temporales

Master File Table (MFT)	Registros del sistema de archivos NTFS	Correlación de metadatos e identificación de patrones de manipulación

Nota. Elaboración propia con base en Carrier (2005) y Casey (2024).

2.2.6. Alteración de artefactos del sistema

Esta categoría agrupa las técnicas dirigidas a modificar, deshabilitar o eliminar los artefactos generados automáticamente por el sistema operativo Windows, los cuales son utilizados comúnmente por los investigadores durante las fases de reconstrucción cronológica y análisis de ejecución en investigaciones forenses. Los adversarios emplean estas tácticas de evasión de defensas con el objetivo de cegar las capacidades de auditoría del analista; sin embargo, los procedimientos de análisis forense estándar cuentan con mecanismos específicos para detectar estas actividades de manipulación.

Entre las técnicas consideradas en este estudio se encuentran:

- Eliminación de logs de eventos: Modificación o purga completa de los registros de auditoría almacenados en los archivos .evtx. En un análisis forense estándar, la eliminación de estos logs no pasa inadvertida. Herramientas de análisis automatizado y plataformas como Autopsy identifican esta manipulación mediante la detección del Evento ID 1102 (en el log de Security), el cual registra explícitamente el momento en que un usuario administrador borra el historial de eventos, o el Evento ID 104 (en el log de System). Asimismo, la correlación temporal revela anomalías estructurales indiciarias, tales como "vacíos de información" (log gaps) o discrepancias numéricas en los índices secuenciales de los registros (MITRE Corporation, 2026).
- Eliminación de archivos Prefetch: Borrado de los archivos con extensión .pf ubicados en el directorio %SystemRoot%\Prefetch, los cuales guardan metadatos críticos de los

últimos 128 ejecutables procesados. A través de un análisis forense estándar, la eliminación deliberada de estos archivos se detecta mediante el examen de inconsistencias en los contadores globales y el análisis de estructuras del sistema de archivos de bajo nivel. Específicamente, herramientas como Autopsy procesan la Tabla Maestra de Archivos (MFT) identificando entradas de archivos marcadas como eliminadas pero cuyos metadatos e índices residuales permanecen recuperables en el espacio no asignado (unallocated space). Adicionalmente, se audita el registro de transacciones $LogFile y el diario $UsnJrnl, los cuales retienen las operaciones físicas de borrado en el volumen NTFS, exponiendo de forma inequívoca el patrón de manipulación temporal de los ejecutables (MITRE Corporation, 2026).

2.2.7. Ocultación de información a nivel de sistema de archivos

Las técnicas de esta categoría aprovechan las características estructurales de bajo nivel del sistema de archivos para ocultar información maliciosa o exfiltrada sin modificar la apariencia visible ni el tamaño reportado de los archivos principales en la interfaz de usuario estándar.

En sistemas NTFS, la técnica fundamental corresponde al uso de Flujos Alternos de Datos (Alternate Data Streams - ADS), un mecanismo diseñado originalmente para la compatibilidad con sistemas de archivos jerárquicos (HFS) que permite asociar múltiples bifurcaciones de datos a un solo descriptor de archivo sin que estas sean visibles en el Explorador de Windows o mediante comandos básicos tradicionales como dir o type.

Para resolver las demandas experimentales de este proyecto, la presencia y el contenido de los ADS se comprobarán rigurosamente mediante tres capas metodológicas de verificación forense estándar:

1. Detección e inspección mediante comandos nativos (Triage Inicial): La verificación primaria en la consola de comandos de Windows se realizará utilizando el parámetro de enumeración /R dentro del comando de listado clásico (dir /R). Este procedimiento permite visualizar de forma directa el descriptor del flujo alterno anexado al archivo base bajo la sintaxis archivo.txt:oculto.exe:$DATA. Asimismo, se implementará el cmdlet nativo de PowerShell Get-Item -Stream *, diseñado específicamente para mapear e identificar el nombre y tamaño exacto de cualquier flujo de datos no estándar adosado a la estructura lógica del archivo (Microsoft Corporation, 2025).
2. Análisis forense de bajo nivel automatizado (Autopsy): En la fase de post-adquisición de la imagen bit a bit, la plataforma Autopsy procesará el volumen NTFS analizando de forma directa las estructuras de la Tabla Maestra de Archivos (MFT). El módulo de ingesta automatizado de Autopsy examinará los atributos $DATA de los registros de archivo de la MFT. Mientras que un archivo legítimo posee un único atributo $DATA sin nombre, Autopsy identificará y alertará la presencia de atributos $DATA adicionales que posean un nombre asignado (los cuales constituyen técnicamente el ADS), extrayendo el contenido oculto de forma automática independientemente de si se trata de texto plano o binarios ejecutables.
3. Correlación con el diario de cambios (Journal de NTFS): Dado que la creación, modificación o destrucción de un archivo deja rastros transaccionales inmediatos en la estructura del sistema de archivos, se comprobará la actividad histórica del borrado

seguro parseando el archivo $UsnJrnl (Update Sequence Number Journal). Las herramientas de análisis estándar extraerán las operaciones físicas registradas bajo las razones de control FILE_DELETE, STREAM_CHANGE y DATA_EXTEND. Este análisis transaccional permite validar temporalmente el momento exacto de la manipulación, aislando las inconsistencias estructurales entre las entradas eliminadas de la MFT y los metadatos históricos del diario de cambios de Windows (Microsoft Corporation, 2025; MITRE Corporation, 2026).

4. Eliminación de Pruebas (File Wiping): Esta metodología implica la destrucción de información para evitar su recuperación a través de técnicas forenses. Con el fin de analizar su posibilidad de detección en un sistema Windows, el experimento se enfocará en diferenciar rigurosamente las condiciones técnicas del proceso de borrado. Mientras que una eliminación lógica estándar simplemente desvincula el puntero del archivo en la MFT, dejando los clústeres sin tocar en el espacio no asignado para que puedan ser recuperados fácilmente, el borrado seguro (wiping) elimina físicamente la información al sobrescribir bloques lógicos con patrones de datos aleatorios o ceros. Sin embargo, la capacidad de recuperación y la duración de estos rastros dependen del hardware involucrado: en las unidades de estado sólido (SSD), el uso automático del comando TRIM y los procesos de recolección de basura (Garbage Collection) limpian de manera física e irreversible las celdas de memoria NAND poco después de que se realiza una eliminación lógica, acelerando la desaparición de la evidencia, sin depender de los algoritmos de borrado seguro comunes (NIST, 2015).

2.2.8. Ocultación avanzada de contenido

La ocultación avanzada de contenido abarca técnicas diseñadas de forma específica para incrustar información confidencial, herramientas de explotación o datos exfiltrados dentro de portadores aparentemente legítimos, manteniendo intacta la estructura externa del archivo contenedor ante una inspección convencional.

La técnica principal analizada en esta categoría corresponde a la esteganografía digital, la cual consiste en insertar datos ocultos dentro de archivos multimedia (como imágenes de formato .png o .jpg y archivos de audio) sin generar distorsiones perceptibles para el usuario.

La inclusión de la esteganografía dentro del diseño experimental de este proyecto se justifica debido a su creciente adopción por parte de actores de amenazas para eludir sistemas de prevención de pérdida de datos (DLP) y evadir la detección en el endpoint. Al no alterar la funcionalidad ni la extensión del archivo portador, un análisis forense basado únicamente en el Explorador de Windows o en el Administrador de Tareas resulta completamente ineficaz. Por lo tanto, su evaluación dentro del entorno Windows se comprobará y auditará mediante las siguientes tres metodologías de análisis forense estándar:

1. Auditoría de Inconsistencias Estructurales y Tamaño de Archivos: Aunque visualmente el archivo contenedor parece idéntico al original, el proceso experimental evaluará las anomalías en sus metadatos. El análisis estándar mediante Autopsy inspeccionará las discrepancias de tamaño físico en el disco frente al tamaño lógico del archivo, detectando inyecciones de datos que ocurren frecuentemente al final del archivo (técnicas de Append tras los marcadores de fin de archivo o End-of-File).

2. Análisis Estadístico de Firmas y Hash Criptográfico: Se comprobará la presencia de contenido oculto calculando y comparando las firmas hash (MD5 y SHA-256) de los archivos sospechosos frente a una base de datos de archivos de origen conocidos o imágenes de referencia limpias. El triaje estándar identificará alteraciones en la entropía del archivo y firmas de herramientas esteganográficas comerciales automatizadas que rompen los patrones lógicos de compresión de los contenedores multimedia (Kaur et al., 2026).
3. Correlación con Tácticas de Evasión de Defensas: El flujo experimental mapeará el comportamiento de las utilidades empleadas para la ocultación con los vectores de ejecución documentados en el sector. Esto permite evaluar si la inyección del payload multimedia generó artefactos secundarios de persistencia u operaciones de lectura y escritura sospechosas registradas en las colmenas del Registro de Windows o en los diarios transaccionales NTFS (MITRE Corporation, 2026).

2.2.9. Destrucción de evidencia

La destrucción de evidencia comprende técnicas orientadas a eliminar información de forma irreversible, impidiendo su recuperación mediante procedimientos forenses tradicionales. Dentro de esta categoría se encuentra el file wiping, técnica basada en la sobrescritura segura de archivos o espacios del disco mediante patrones específicos de escritura.

A diferencia de una eliminación convencional, donde parte de la información puede permanecer almacenada y ser recuperada mediante herramientas forenses, la sobrescritura segura busca reducir significativamente la posibilidad de reconstruir los

datos originales mediante la sustitución del contenido almacenado por nuevos patrones de información.

En el contexto de esta investigación, la técnica de file wiping será implementada mediante la herramienta Eraser, la cual realiza la sobrescritura segura de archivos antes de su eliminación. Esta técnica será aplicada sobre archivos previamente generados en el entorno de pruebas con el propósito de afectar tanto el contenido del archivo como los posibles fragmentos almacenados en el espacio no asignado del sistema de archivos.

El efecto esperado sobre la evidencia digital consiste en la reducción significativa de la capacidad de recuperación de los archivos eliminados mediante herramientas forenses convencionales, dificultando la reconstrucción de su contenido original y limitando la disponibilidad de información relevante para el proceso de investigación. De esta manera, el file wiping constituye una de las técnicas antiforenses con mayor impacto potencial sobre la preservación y recuperación de evidencia digital.

CAPÍTULO 3

3. DESARROLLO

3.1. Diseño del laboratorio experimental

Con el propósito de que todas las técnicas antiforenses fueran evaluadas bajo las mismas condiciones, antes de iniciar las pruebas se definió un protocolo experimental común para todo el laboratorio. Este protocolo estableció la configuración del entorno de trabajo, los recursos asignados a la máquina virtual, las herramientas utilizadas y las métricas que se aplicarían durante el análisis de cada técnica.

Mantener estas condiciones permitió que los resultados obtenidos fueran comparables entre sí, ya que todas las pruebas se realizaron sobre el mismo sistema operativo y utilizando el mismo procedimiento de evaluación. De esta manera, cualquier diferencia observada durante el análisis corresponde al comportamiento propio de la técnica antiforense implementada y no a cambios en el entorno de ejecución.

En las siguientes tablas se presenta el protocolo experimental utilizado y las métricas definidas para evaluar cada una de las técnicas desarrolladas durante la investigación.

Tabla 3.

Protocolo experimental común para las técnicas antiforenses evaluadas

Elemento del protocolo	**Timestomping**	**Eliminación de Logs**	**Prefetch**	**ADS**	**Esteganografía**	**File Wiping**
1. Sistema operativo	Windows 11	Windows 10 Home	Windows 10 Home	Windows 11	Windows 11	Windows 10 Home
3. Hipervisor / disco virtual	VMware /Workstation	VMware /Workstation	VMware /Workstation	VMware /Workstation	VMware /Workstation	VMware /Workstation
4. Recursos asignados	2 vCPU / 4-8 GB RAM / 60-80 GB disco	2 vCPU / 4-8 GB RAM / 60-80 GB disco	2 vCPU / 4-8 GB RAM / 60-80 GB disco	2 vCPU / 4-8 GB RAM / 60-80 GB disco	2 vCPU / 4-8 GB RAM / 60-80 GB disco	2 vCPU / 4-8 GB RAM / 60-80 GB disco
6. Método de adquisición	FTK Imager + KAPE	FTK Imager	FTK Imager	FTK Imager	FTK Imager	FTK Imager
7. Herramientas de análisis	Autopsy 4.23.1, MFTECmd y PowerShell	Autopsy 4.23.1 Event Viewer	WinPrefetchView	Autopsy 4.23.1 PowerShell	Autopsy 4.23.1 PowerShell	Autopsy 4.23.1 Recuva
8. Número de repeticiones	2 repeticiones del procedimiento experimental y un análisis forense mediante correlación de metadatos de la MFT	2 repeticiones	2 repeticiones	2 repeticiones	2 repeticiones	2 repeticiones 1 (validado en 3 niveles: lógico, físico, Recuva)

Nota. La tabla presenta el protocolo experimental estandarizado para las seis técnicas antiforenses analizadas, incluyendo sistema operativo, hipervisor, recursos asignados, método de adquisición, herramientas de análisis y número de repeticiones. Las

variaciones respecto al estándar se identifican y justifican de forma individual en cada subsección correspondiente. Fuente: Elaboración propia.

Tabla 4.

Definición operacional de las métricas comunes de evaluación

Métrica	**Definición operacional**	**Escala de valoración**
Detectabilidad	¿Qué tan fácil fue identificar la aplicación de la técnica mediante el análisis forense?	1 (muy difícil de detectar) – 10 (muy fácil de detectar)
Trazabilidad	¿En qué medida permanecieron artefactos o rastros digitales que permitieran reconstruir la acción realizada?	1 (sin rastros identificables) – 10 (rastros claramente identificables)
Recuperabilidad	¿Qué nivel de recuperación de la evidencia fue posible después de aplicar la técnica?	1 (evidencia no recuperable) – 10 (evidencia recuperada en su totalidad)
Complejidad de análisis	¿Qué esfuerzo requirió identificar los rastros y analizar la evidencia afectada por la técnica?	1 (requiere mucho tiempo de análisis) – 10 (requiere poco tiempo de análisis)

Nota. La tabla presenta la definición operacional y la escala de valoración (1-5) de las cuatro métricas comunes aplicadas de manera uniforme en la evaluación de las seis técnicas antiforenses: detectabilidad, trazabilidad, recuperabilidad e integridad verificable. Fuente: Elaboración propia.

3.1. Implementación de Técnica Antiforense:

3.1.1. Timestomping

El timestomping es una técnica antiforense utilizada para modificar las marcas temporales asociadas a un archivo dentro del sistema de archivos. Estas marcas, conocidas comúnmente como timestamps, registran información relacionada con la creación, modificación y último acceso de un archivo, y constituyen una fuente importante de evidencia durante una investigación forense digital.

El objetivo principal del timestomping es alterar la cronología real de los eventos para dificultar el análisis de actividades realizadas en un sistema. Un atacante puede modificar estas fechas para hacer que un archivo parezca más antiguo, más reciente o incluso para que coincida con los atributos temporales de otros archivos legítimos, reduciendo así las probabilidades de que resulte sospechoso durante una revisión inicial.

En esta práctica se ejecutó la técnica de Timestomping sobre un conjunto de archivos de prueba almacenados en un sistema Windows 11 virtualizado, mediante la modificación deliberada de sus atributos temporales de creación, modificación y acceso utilizando PowerShell. Posteriormente, la evidencia fue adquirida mediante FTK Imager, complementándose con la extracción de la Master File Table (MFT) utilizando KAPE. Finalmente, el análisis forense se realizó mediante Autopsy y MFTECmd, permitiendo correlacionar los metadatos de los archivos y detectar indicadores de manipulación a partir de la comparación de los atributos $STANDARD_INFORMATION y $FILE_NAME.

3.1.1.1. Herramientas utilizadas

Para la ejecución del laboratorio se utilizaron diversas herramientas destinadas a la implementación de la técnica de Timestomping, la adquisición de la evidencia digital y el análisis forense de los metadatos generados. La combinación de estas herramientas permitió no solo ejecutar la técnica antiforense, sino también evaluar sus efectos mediante el análisis de artefactos del sistema de archivos NTFS.

- VMware Workstation

Se utilizó VMware Workstation como plataforma de virtualización para la construcción del laboratorio. Esta herramienta permitió desplegar una máquina virtual con Windows 11, proporcionando un entorno controlado para la ejecución de la técnica de Timestomping y la posterior adquisición de la evidencia digital.

- Windows 11

Windows 11 fue utilizado como sistema operativo objetivo del laboratorio. Sobre este entorno se crearon los archivos de prueba y se ejecutaron las modificaciones de las marcas temporales mediante PowerShell, simulando un escenario de manipulación de metadatos en un sistema de archivos NTFS.

- PowerShell

PowerShell fue empleado para manipular los atributos temporales de los archivos analizados. Mediante esta herramienta fue posible modificar y clonar los valores de creación, modificación y acceso entre diferentes archivos, simulando escenarios comunes de timestomping utilizados para ocultar actividades dentro de un sistema.

- FTK Imager

FTK Imager se utilizó para la adquisición inicial de la evidencia digital y la verificación de

los archivos generados durante el laboratorio. La herramienta permitió examinar los metadatos de los archivos, validar su integridad mediante funciones hash y realizar una primera revisión de los atributos temporales presentes en el sistema de archivos.

- Autopsy

Autopsy se utilizó como plataforma de análisis forense para examinar la evidencia digital obtenida durante el laboratorio. La herramienta permitió inspeccionar la estructura del sistema de archivos, revisar los metadatos de los archivos analizados y contrastar los resultados obtenidos con el análisis de la Master File Table, facilitando la correlación entre las evidencias obtenidas mediante diferentes herramientas forenses.

- KAPE (Kroll Artifact Parser and Extractor)

Con el propósito de acceder a artefactos protegidos del sistema de archivos NTFS, se utilizó KAPE para extraer el archivo Master File Table ($MFT) desde el volumen del sistema. Esta herramienta permitió obtener una copia del artefacto preservando su contenido para su posterior procesamiento, evitando las restricciones de acceso propias del sistema operativo

- MFTECmd

MFTECmd fue utilizado para procesar el archivo $MFT extraído mediante KAPE. La herramienta permitió interpretar los registros de la Master File Table y generar un archivo CSV con la información de los metadatos almacenados por NTFS. A partir de estos resultados se compararon los atributos $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30) de los archivos utilizados en el laboratorio, identificando discrepancias temporales que fueron analizadas como parte del proceso de detección de la técnica de Timestomping.

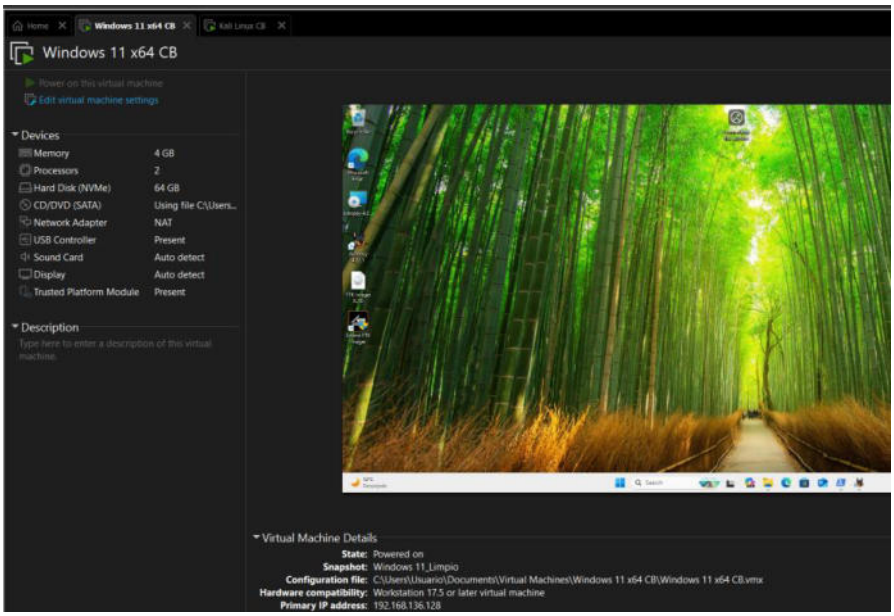
3.1.1.2. Procedimiento realizado

La práctica se desarrolló en un entorno virtualizado implementado sobre VMware Workstation, utilizando una máquina virtual con Windows 11 como sistema objetivo. Este entorno permitió realizar las pruebas de manera controlada, sin afectar sistemas de producción y garantizando la repetibilidad de los resultados obtenidos.

Inicialmente se creó una carpeta denominada Forense dentro del sistema Windows 11, destinada a almacenar los archivos utilizados durante las diferentes pruebas. Como punto de partida se generó un archivo de referencia denominado archivo_legitimo.txt, cuyos atributos temporales fueron utilizados posteriormente como base para los escenarios de manipulación.

Figura 2

Entorno de laboratorio utilizado para la práctica de timestomping en Windows 11

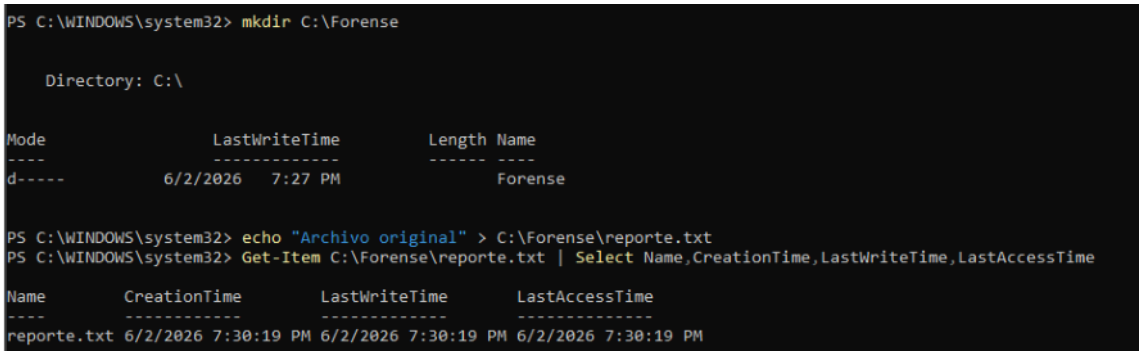


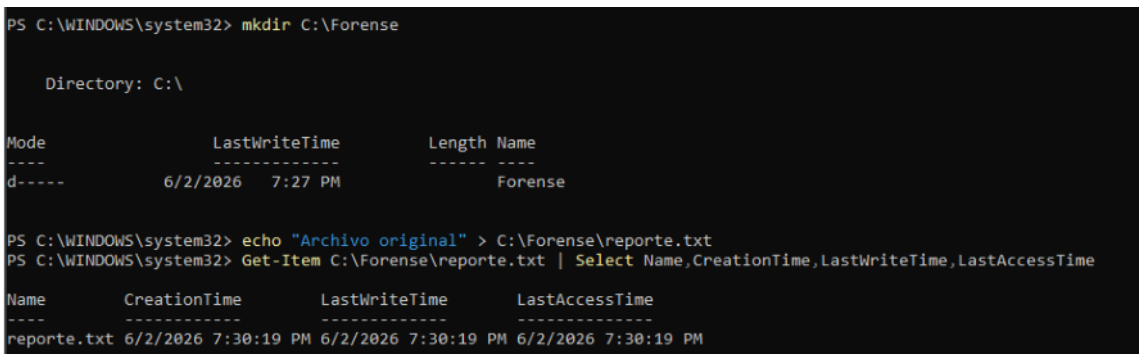
Nota. Elaboración propia. La captura muestra la máquina virtual Windows 11 configurada en VMware Workstation, utilizada como entorno de pruebas para la ejecución de la técnica de timestomping.

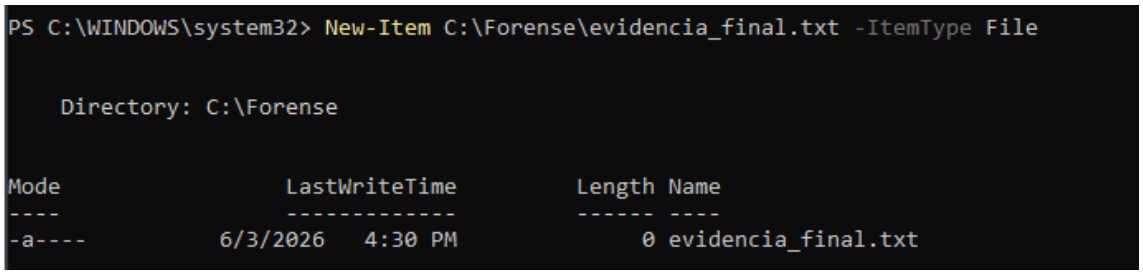
Posteriormente se verificaron los atributos temporales de los archivos mediante PowerShell, registrando los valores correspondientes a fecha de creación, modificación y último acceso. Esta información permitió establecer una línea base que serviría como referencia para comparar los cambios realizados durante la ejecución de la técnica antiforense.

Figura 3

Consulta inicial de timestamps mediante PowerShell.




PS C:\WINDOWS\system32> mkdir C:\Forense
Directory: C:\
Mode LastWriteTime Length Name
d----- 6/2/2026 7:27 PM Forense
PS C:\WINDOWS\system32> echo "Archivo original" > C:\Forense\reporte.txt
PS C:\WINDOWS\system32> Get-Item C:\Forense\reporte.txt | Select Name,CreationTime,LastWriteTime,LastAccessTime
Name CreationTime LastWriteTime LastAccessTime
reporte.txt 6/2/2026 7:30:19 PM 6/2/2026 7:30:19 PM 6/2/2026 7:30:19 PM

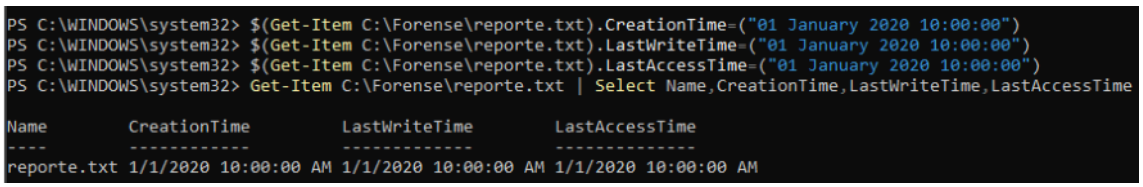


Nota. Elaboración propia. La captura evidencia la creación del archivo de prueba y la consulta inicial de sus atributos temporales (Creation Time, LastWriteTime y LastAccessTime) antes de aplicar la técnica de timestomping.

En el primer escenario se creó un archivo denominado reporte.txt y se modificaron manualmente sus atributos temporales utilizando PowerShell. Para ello se alteraron los campos Creation Time, Last Write Time y Last Access Time, asignando valores históricos distintos a la fecha real de creación del archivo. Esta prueba permitió comprobar que las marcas temporales almacenadas por el sistema de archivos NTFS pueden ser modificadas de forma directa.

Figura 4

Modificación manual de timestamps mediante PowerShell.



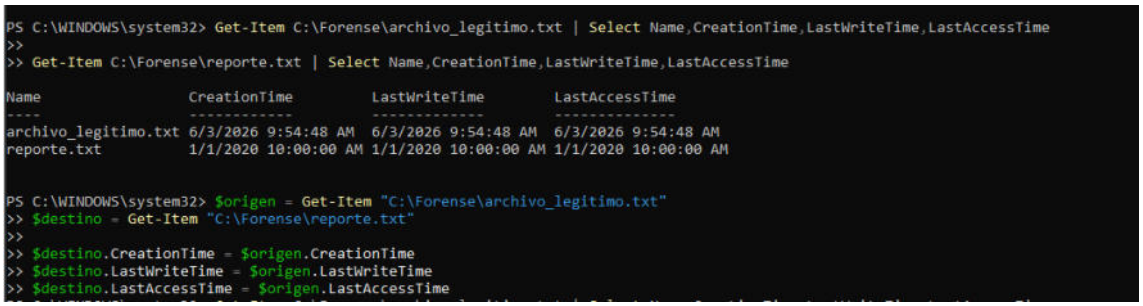
Nota. Elaboración propia. La captura muestra la modificación deliberada de los atributos temporales del archivo reporte.txt utilizando comandos de PowerShell, asignándole una fecha distinta a la original.

Como segundo escenario se implementó una técnica de clonación de timestamps. En este caso, los atributos temporales del archivo archivo_legitimo.txt fueron copiados al

archivo reporte.txt, logrando que ambos presentaran los mismos valores de creación, modificación y acceso. Este procedimiento reproduce un escenario más cercano a situaciones reales, en las que un atacante busca ocultar la verdadera cronología de un archivo haciéndolo coincidir con artefactos legítimos del sistema.

Figura 5

Clonación de timestamps entre archivos mediante PowerShell.

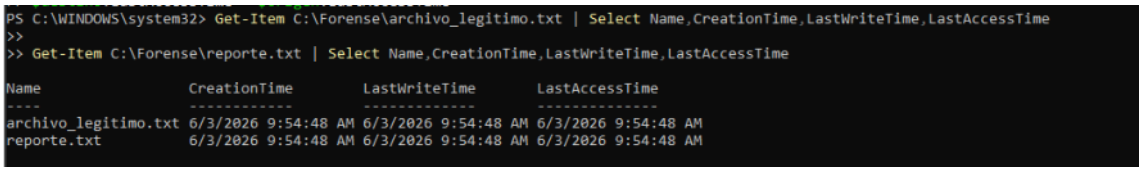


Nota. Elaboración propia. La captura muestra la modificación deliberada de los atributos temporales del archivo reporte.txt utilizando comandos de PowerShell, asignándole una fecha distinta a la original.

Una vez realizada la modificación, se verificó nuevamente la información temporal de los archivos involucrados, confirmando que los cambios habían sido aplicados correctamente y que los archivos presentaban atributos temporales coincidentes.

Figura 6

Verificación de timestamps después de la manipulación.

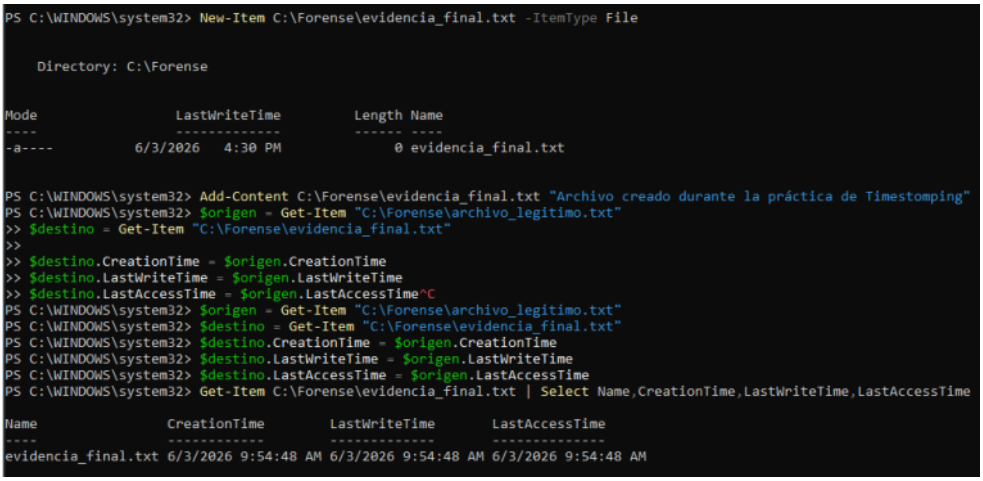


Nota. Elaboración propia. La captura muestra la comparación de los metadatos temporales de archivo_legitimo.txt y reporte.txt después de la clonación, evidenciando que ambos presentan valores idénticos en sus marcas de tiempo.

Como parte de una tercera prueba se creó el archivo evidencia_final.txt, incorporando contenido generado durante la ejecución de la práctica. Posteriormente se replicaron los atributos temporales del archivo legítimo sobre este nuevo archivo. De esta manera, se obtuvo un artefacto con contenido reciente que aparentaba haber sido creado en el mismo momento que el archivo de referencia, simulando una situación de ocultamiento temporal más compleja.

Figura 7

Creación de un archivo con contenido reciente y timestamps falsificados.

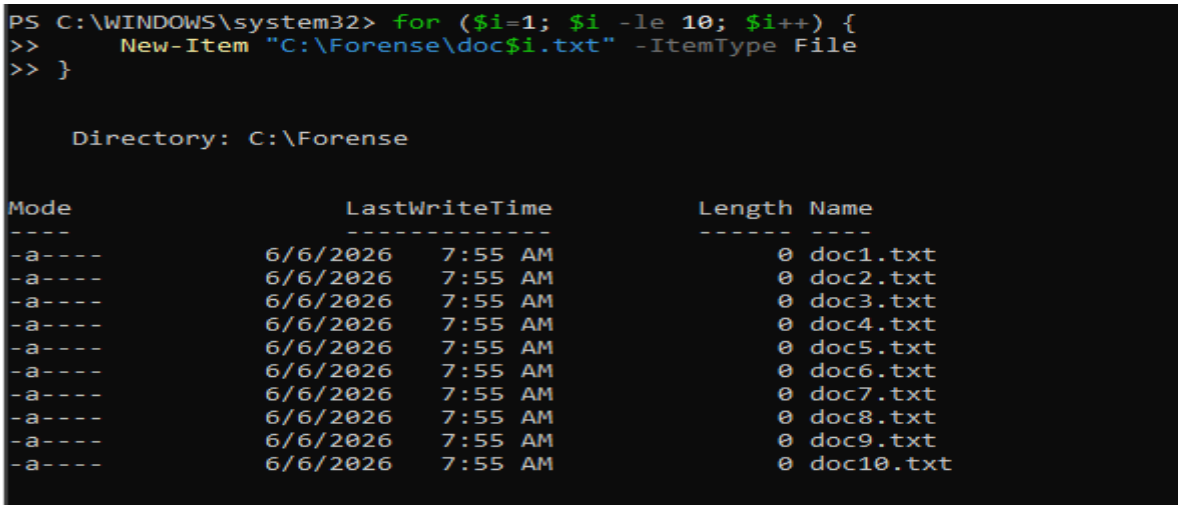


Nota. Elaboración propia. La captura muestra la creación del archivo evidencia_final.txt, la incorporación de contenido durante la práctica y la posterior modificación de sus atributos temporales mediante PowerShell, utilizando como referencia las marcas de tiempo del archivo archivo_legitimo.txt.

Con el propósito de ampliar el alcance de la evaluación y obtener una muestra más representativa para el análisis forense, se implementó un escenario adicional de manipulación masiva de timestamps. Para ello se generaron diez archivos de prueba denominados doc1.txt a doc10.txt, los cuales fueron almacenados dentro de la carpeta de trabajo utilizada durante la práctica.

Figura 8

Creación de múltiples archivos de prueba para la ejecución masiva de timestomping.



Nota. Elaboración propia. La captura muestra la generación automática de diez archivos de prueba (doc1.txt a doc10.txt) mediante un script de PowerShell, los cuales fueron utilizados para evaluar la aplicación masiva de la técnica de timestomping y el posterior análisis forense de sus atributos temporales.

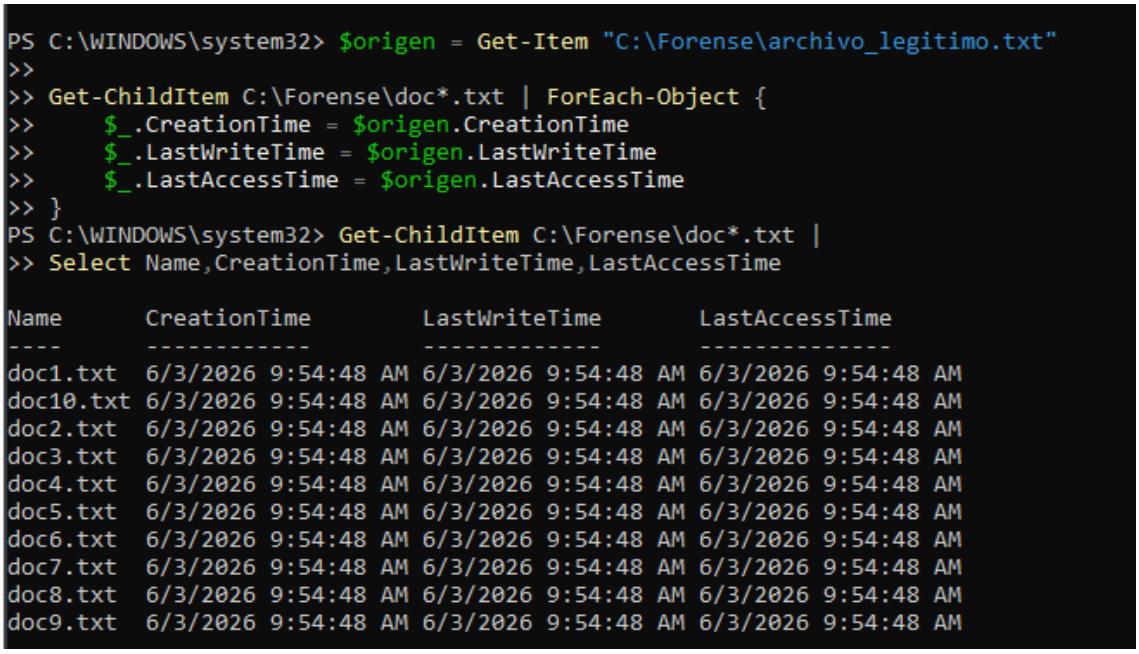
Posteriormente, mediante PowerShell, se copiaron los atributos temporales del archivo legítimo de referencia hacia cada uno de los archivos creados. Como resultado, los diez artefactos presentaron los mismos valores de creación, modificación y acceso que el archivo original, reproduciendo un patrón de clonación masiva de timestamps

similar al que podría observarse en actividades antiforenses orientadas a ocultar la verdadera cronología de múltiples archivos.

Una vez completada la modificación de los atributos temporales, se verificó que los doce archivos manipulados presentaban los mismos valores de creación, modificación y acceso que el archivo archivo_legitimo.txt, utilizado como referencia. Posteriormente, la evidencia fue adquirida y analizada para comprobar que la clonación de los metadatos podía identificarse mediante la correlación de los atributos temporales y el análisis de la Master File Table (MFT).

Figura 9

Verificación de la clonación masiva de timestamps mediante PowerShell.

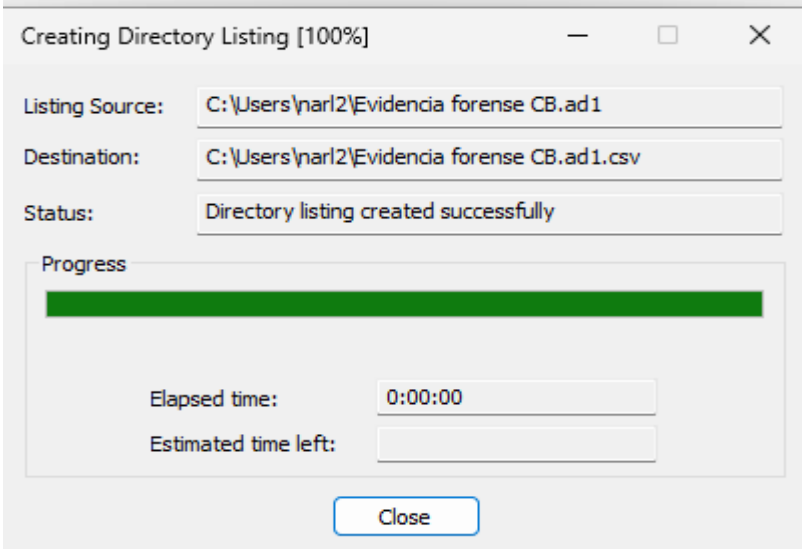


Nota. Elaboración propia. La captura muestra la clonación de los atributos temporales del archivo archivo_legitimo.txt hacia los archivos doc1.txt a doc10.txt, así como la verificación posterior mediante PowerShell, evidenciando que todos presentan las mismas fechas de creación, modificación y acceso.

Finalizada la manipulación de los atributos temporales, se realizó la adquisición forense de la evidencia mediante FTK Imager. La herramienta permitió generar una imagen lógica del conjunto de archivos involucrados en la práctica y verificar su integridad mediante el cálculo de funciones hash MD5 y SHA1. Este procedimiento garantizó que la evidencia utilizada durante el análisis conservara las mismas características que la fuente original y no presentara alteraciones derivadas del proceso de adquisición.

Figura 10

Adquisición de evidencia mediante FTK Imager.



Nota. Elaboración propia. La captura muestra la finalización del proceso de adquisición lógica realizado con FTK Imager, correspondiente al conjunto de archivos utilizado durante la práctica de Timestomping.

Figura 11

Verificación de integridad de la evidencia mediante funciones hash.

Nota. Elaboración propia. La captura muestra los resultados de la verificación de integridad realizada en FTK Imager, donde los valores hash MD5 y SHA-1 calculados coinciden con los registrados durante la adquisición de la evidencia, confirmando que el archivo analizado no sufrió alteraciones y preserva su autenticidad para el proceso forense.

Posteriormente, se verificó la integridad de la evidencia adquirida mediante la comparación de los valores hash MD5 y SHA1 generados por FTK Imager. La coincidencia entre los valores calculados durante la adquisición y los obtenidos en la verificación confirmó que la evidencia permaneció íntegra durante todo el procedimiento, garantizando la confiabilidad de los análisis posteriores.

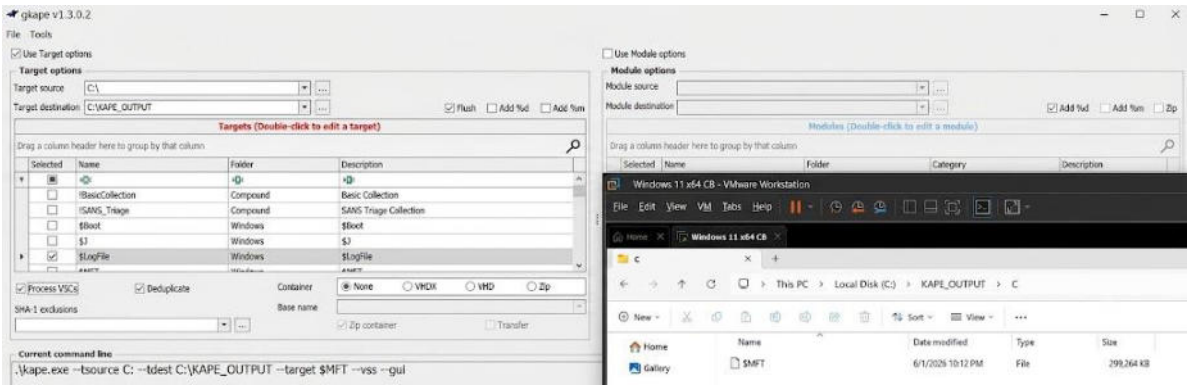
Con el propósito de obtener una visión más completa de los metadatos del sistema de archivos NTFS, se realizó la extracción del archivo Master File Table ($MFT) mediante KAPE. Posteriormente, dicho artefacto fue procesado con MFTECmd, obteniéndose un registro estructurado de los metadatos asociados a los archivos analizados. Esta información permitió comparar los atributos $STANDARD_INFORMATION y $FILE_NAME, identificar discrepancias temporales y correlacionar los registros correspondientes a los archivos manipulados durante la práctica.

Con el propósito de complementar el análisis de los metadatos obtenidos durante la adquisición forense, se realizó la extracción del archivo Master File Table ($MFT)

mediante KAPE. Este procedimiento permitió obtener una copia del principal artefacto del sistema de archivos NTFS, utilizada posteriormente para el procesamiento de los registros mediante MFTECmd y la correlación de los atributos temporales asociados a los archivos manipulados.

Figura 12

Extracción del archivo $MFT mediante KAPE.

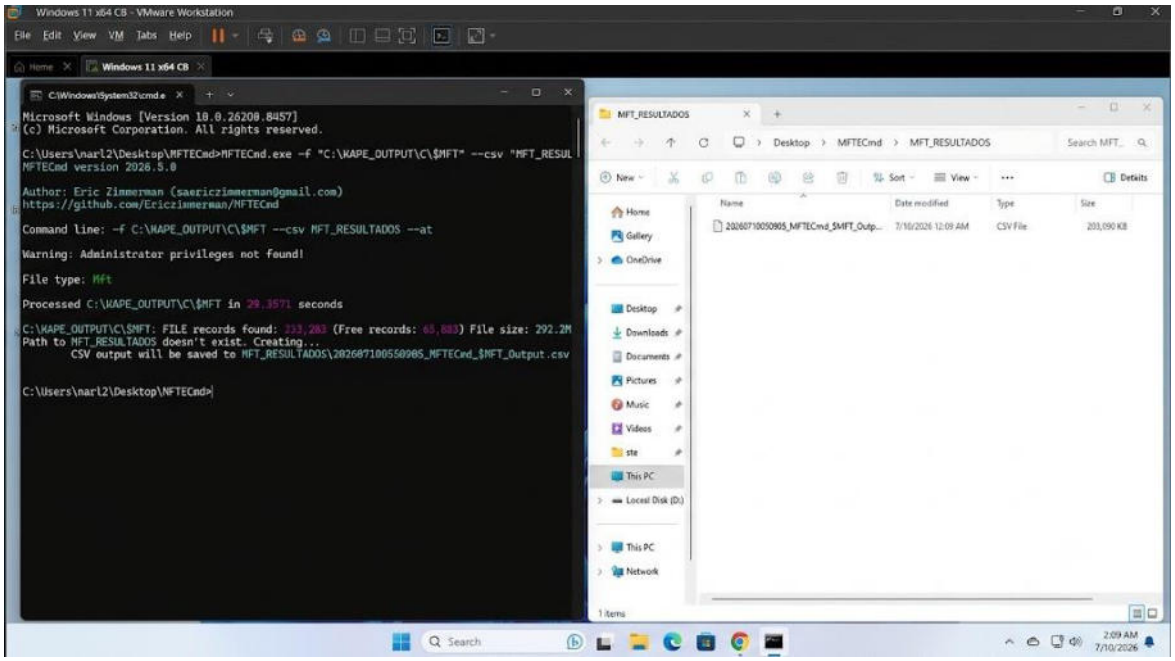


Nota. Elaboración propia. La figura muestra la configuración de KAPE para la extracción del archivo Master File Table ($MFT) desde el volumen analizado y el resultado obtenido en el directorio de salida (KAPE_OUTPUT), donde se generó el archivo utilizado posteriormente en el análisis forense.

Una vez obtenido el archivo $MFT, este fue procesado mediante la herramienta MFTECmd, generándose un archivo en formato CSV con los registros del sistema de archivos NTFS. Dicho procesamiento permitió examinar los atributos $STANDARD_INFORMATION y $FILE_NAME, así como correlacionar los metadatos asociados a los archivos manipulados durante la ejecución de la técnica de Timestomping.

Figura 13

Procesamiento del archivo $MFT mediante MFTECmd.

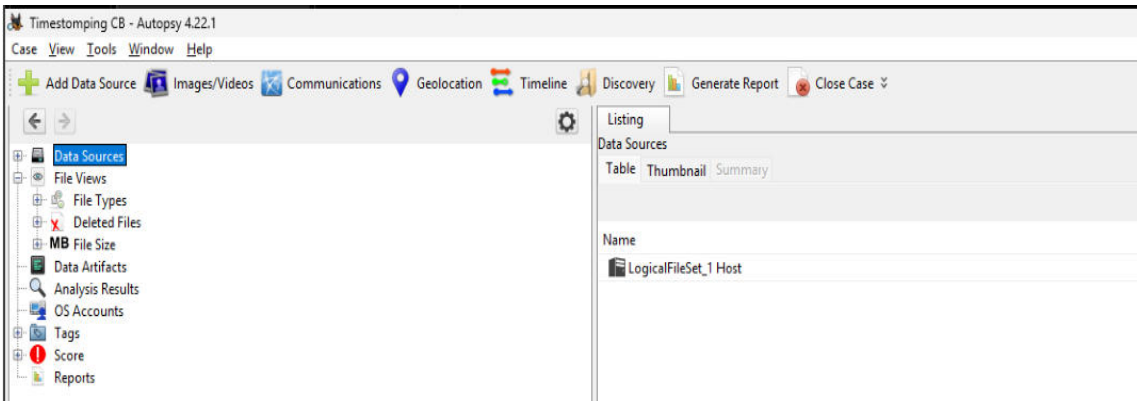


Nota. Elaboración propia La figura presenta el procesamiento del archivo $MFT mediante MFTECmd y la generación del archivo en formato CSV, el cual contiene los registros de la Master File Table utilizados para analizar los metadatos de los archivos y detectar indicadores de manipulación temporal.

Finalmente, la evidencia lógica adquirida mediante FTK Imager fue incorporada a Autopsy para complementar el análisis forense. La información obtenida fue contrastada con los resultados del procesamiento de la Master File Table, permitiendo correlacionar los metadatos de los archivos manipulados e identificar patrones compatibles con la aplicación de la técnica de Timestomping.

Figura 14

Incorporación de la evidencia al caso forense en Autopsy.



Nota. Elaboración propia La captura muestra la incorporación de la evidencia lógica al caso forense en Autopsy, etapa previa al análisis de los metadatos de los archivos manipulados y a la correlación de los resultados con la información obtenida del procesamiento de la Master File Table (MFT).

Finalmente, la evidencia adquirida fue analizada en Autopsy, permitiendo examinar los metadatos de los archivos manipulados y correlacionarlos con la información obtenida del procesamiento de la Master File Table (MFT) mediante MFTECmd. La integración de ambas fuentes de evidencia permitió identificar patrones temporales consistentes, verificar la clonación de los atributos de creación, modificación y acceso, así como detectar discrepancias entre los atributos $STANDARD_INFORMATION y $FILE_NAME, constituyendo indicadores de la aplicación de la técnica de Timestomping.

3.1.1.3. Artefactos afectados

La técnica de Timestomping afecta principalmente los metadatos temporales almacenados por el sistema de archivos NTFS. Estos metadatos constituyen una fuente fundamental de evidencia en las investigaciones forenses, ya que permiten reconstruir la secuencia

cronológica de eventos relacionados con la creación, modificación, acceso y actualización de los archivos presentes en un sistema.

Durante la ejecución de la práctica se manipuló un conjunto de doce archivos, conformado por reporte.txt, evidencia_final.txt y diez archivos de prueba denominados doc1.txt a doc10.txt. Sobre estos artefactos se aplicó la modificación deliberada de sus atributos temporales utilizando como referencia el archivo archivo_legitimo.txt, con el propósito de simular un escenario de ocultamiento de actividad mediante la alteración de metadatos.

Los artefactos afectados corresponden a los atributos temporales conocidos como MAC Times (Modified, Accessed y Created), utilizados habitualmente para la construcción de líneas de tiempo forenses. En sistemas NTFS, estos valores se almacenan principalmente en los atributos $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30) de la Master File Table (MFT). La comparación entre ambos atributos constituye una práctica habitual durante el análisis forense, ya que pueden existir discrepancias cuando los metadatos han sido modificados o cuando determinadas operaciones del sistema actualizan únicamente uno de ellos.

Los atributos modificados durante la práctica fueron los siguientes:

- **Creation Time (Fecha de creación)**: registra el momento en que el archivo fue creado dentro del sistema de archivos
- **Last Write Time (Fecha de modificación)**: registra la última ocasión en que el contenido del archivo fue modificado.
- **Last Access Time (Fecha de acceso**): registra la última vez que el archivo fue abierto o consultado.

Como resultado de la ejecución de la técnica, se modificaron treinta y seis atributos temporales, correspondientes a tres marcas de tiempo en cada uno de los doce archivos manipulados. Esta alteración produjo un patrón de coincidencia temporal entre múltiples archivos, dificultando la reconstrucción cronológica de los eventos cuando el análisis se limita a la observación superficial de los metadatos.

Desde la perspectiva forense, la manipulación de estos atributos puede afectar los procesos de reconstrucción de líneas de tiempo, correlación de eventos y atribución de actividades realizadas por un usuario o un posible atacante. No obstante, las inconsistencias generadas entre los atributos temporales almacenados en $STANDARD_INFORMATION y $FILE_NAME, así como la existencia de múltiples archivos con marcas temporales idénticas, pueden constituir indicadores útiles para la detección de posibles manipulaciones.

En esta práctica, la evaluación de los artefactos afectados se realizó mediante un proceso de análisis en dos niveles. Inicialmente, Autopsy permitió verificar visualmente las coincidencias temporales entre los archivos manipulados y el archivo legítimo de referencia. Posteriormente, la extracción del archivo $MFT mediante KAPE y su procesamiento con MFTECmd hicieron posible comparar los atributos $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30) de los archivos del laboratorio. Este análisis permitió identificar discrepancias entre ambos conjuntos de metadatos, proporcionando evidencia adicional para evaluar el impacto de la técnica de Timestomping sobre los artefactos temporales del sistema de archivos NTFS.

3.1.1.4. Método de detección aplicado

Con el propósito de identificar evidencias asociadas a la manipulación de metadatos mediante la técnica de Timestomping, se aplicó un procedimiento de análisis forense basado en la correlación de diferentes fuentes de evidencia del sistema de archivos NTFS. El proceso se desarrolló en un entorno controlado utilizando una máquina virtual con Windows 11, donde previamente se ejecutó la técnica antiforense sobre los archivos definidos para el laboratorio.

Inicialmente, se realizó la revisión de los metadatos visibles mediante FTK Imager y Autopsy, verificando los atributos temporales de los archivos antes y después de la ejecución del Timestomping. Este análisis permitió identificar modificaciones en las fechas de creación, modificación y acceso, constituyendo una primera aproximación para detectar alteraciones sobre los metadatos.

Posteriormente, con el fin de profundizar el análisis, se efectuó la adquisición del archivo Master File Table ($MFT) mediante la herramienta KAPE, la cual permite copiar artefactos protegidos del sistema de archivos NTFS preservando su integridad. El archivo obtenido fue procesado utilizando MFTECmd, herramienta especializada en el análisis de registros de la MFT, generando un conjunto estructurado de metadatos correspondiente a 233 283 registros del sistema de archivos.

A partir de la información obtenida se analizaron los atributos temporales contenidos en $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30) para los archivos utilizados durante el laboratorio. La comparación entre ambos atributos permitió identificar discrepancias en los tiempos de creación, modificación, acceso y cambio del registro, las cuales fueron interpretadas en conjunto con el procedimiento

experimental ejecutado. Este análisis no se basó únicamente en la observación de diferencias temporales, sino en la correlación de dichas diferencias con la secuencia de acciones realizadas durante la implementación de la técnica antiforense.

Como criterio de interpretación, se consideró que la presencia de discrepancias entre los atributos temporales de $STANDARD_INFORMATION y $FILE_NAME constituye un indicador forense de posible manipulación de metadatos; sin embargo, dichas discrepancias no fueron interpretadas de forma aislada como evidencia concluyente de Timestomping. Su valoración se realizó conjuntamente con las evidencias obtenidas mediante FTK Imager, Autopsy y la documentación del procedimiento experimental, reduciendo la posibilidad de atribuir las diferencias observadas a modificaciones legítimas del sistema operativo.

Finalmente, aunque la literatura especializada señala que la correlación con artefactos como $UsnJrnl y $LogFile puede fortalecer el proceso de detección, estos no fueron analizados dentro del alcance experimental de esta investigación. Por tanto, el estudio se centró en el análisis de los atributos temporales de la MFT y su comparación con los metadatos observados mediante las herramientas forenses empleadas.

3.1.1.5. Resultados obtenidos

La ejecución de los escenarios experimentales permitió comprobar que la técnica de Timestomping puede aplicarse de forma efectiva sobre múltiples archivos de un sistema de archivos NTFS, alterando sus atributos temporales sin modificar el contenido de los archivos. Como resultado, los artefactos manipulados adquirieron marcas de tiempo compatibles con las de un archivo legítimo previamente

seleccionado como referencia, dificultando la reconstrucción cronológica de los eventos mediante una inspección superficial de los metadatos.

Durante la práctica se manipuló un conjunto de doce archivos mediante la modificación de sus atributos de creación, modificación y acceso. Inicialmente se realizaron pruebas individuales sobre los archivos reporte.txt y evidencia_final.txt, verificando la posibilidad de modificar sus marcas temporales utilizando comandos de PowerShell. Posteriormente se ejecutó un escenario de clonación masiva sobre diez archivos adicionales (doc1.txt a doc10.txt), cuyos atributos temporales fueron igualados a los registrados por el archivo archivo_legitimo.txt.

La verificación inicial realizada mediante 162 que los archivos manipulados presentaban coincidencias en los atributos temporales visibles, evidenciando que la técnica había sido aplicada correctamente desde la perspectiva de las herramientas forenses convencionales.

Con el propósito de profundizar el análisis, se extrajo el archivo Master File Table ($MFT) mediante la herramienta KAPE, obteniéndose una copia íntegra del artefacto para su posterior procesamiento. El archivo fue analizado utilizando MFTECmd, herramienta que permitió interpretar los registros de la MFT y generar un conjunto estructurado de información correspondiente a 233 283 registros del sistema de archivos NTFS

El análisis de los registros asociados a los archivos utilizados durante el laboratorio permitió comparar los atributos temporales almacenados en $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30). Los resultados evidenciaron discrepancias entre ambos atributos para los archivos sometidos al procedimiento experimental, observándose diferencias en las fechas de creación, modificación, acceso y cambio

del registro. Asimismo, se identificó un patrón repetitivo en los archivos manipulados, caracterizado por la presencia de valores temporales idénticos entre múltiples artefactos y diferencias entre los atributos temporales almacenados por NTFS.

Los resultados obtenidos demuestran que la manipulación de los metadatos puede dificultar la reconstrucción cronológica de los eventos cuando el análisis se limita a la observación directa de los timestamps. Sin embargo, la comparación de los atributos $STANDARD_INFORMATION y $FILE_NAME, complementada con la revisión de los metadatos mediante herramientas forenses, permitió identificar indicadores compatibles con la alteración de metadatos. Estos hallazgos evidencian la importancia de correlacionar diferentes fuentes de evidencia durante una investigación forense y de no basar las conclusiones únicamente en los atributos temporales visibles del sistema operativo.

3.1.2. Eliminación de logs de eventos

3.1.2.1. Descripción de la Técnica

La eliminación de logs de eventos constituye una técnica antiforense utilizada para ocultar o dificultar la reconstrucción de actividades realizadas dentro de un sistema informático. En sistemas operativos Windows, los registros de eventos almacenan información relevante relacionada con accesos al sistema, ejecución de aplicaciones, autenticaciones, errores, actividades administrativas y eventos de seguridad. Debido a su importancia dentro de una investigación forense digital, estos registros representan una de las principales fuentes de evidencia para identificar actividades sospechosas o incidentes de seguridad.

La técnica antiforense analizada en el presente laboratorio consistió en la eliminación deliberada de registros del log de seguridad (Security.evtx) utilizando herramientas nativas del sistema operativo Windows. El propósito de esta acción fue evaluar el nivel de detectabilidad que mantienen este tipo de manipulaciones cuando son analizadas posteriormente mediante herramientas forenses digitales estándar.

Durante el desarrollo experimental se aplicaron distintos métodos de eliminación de logs, incluyendo la limpieza manual desde el Visor de Eventos y el uso de comandos administrativos mediante PowerShell y la utilidad wevtutil. Posteriormente, se realizó la adquisición y análisis de evidencia digital residual utilizando FTK Imager y Autopsy.

El objetivo principal del laboratorio fue determinar si la eliminación de registros logra eliminar completamente la trazabilidad digital del sistema o si persisten artefactos residuales detectables mediante técnicas forenses convencionales.

3.1.2.2. Herramientas Utilizadas

Para la implementación del laboratorio se utilizaron diversas herramientas de virtualización, administración del sistema y análisis forense digital. Cada una cumplió una función específica dentro del proceso experimental.

Tabla 5.

Herramientas utilizadas durante el laboratorio experimental

Herramienta	Versión	Función dentro del laboratorio

VMware Fusion	13.6.4	Virtualización del entorno experimental Windows
Windows	Windows 10 Home	Sistema operativo utilizado para las pruebas antiforenses
Event Viewer	Windows integrado	Visualización y administración de logs del sistema
PowerShell	Windows integrado	Ejecución de comandos administrativos
wevtutil	Windows integrado	Herramienta nativa para administración y eliminación de logs
FTK Imager	4.7.3.81	Adquisición y exportación de evidencia digital
Autopsy	4.23.1	Plataforma de análisis forense digital
Prefetch de Windows	Windows integrado	Artefactos utilizados para detección de ejecución de programas

Nota. La tabla presenta las herramientas utilizadas durante el desarrollo experimental de la investigación. Fuente: Elaboración propia.

3.1.2.3. Procedimiento Realizado

El procedimiento se desarrolló evaluando tres métodos distintos de eliminación del log de Seguridad: Event Viewer, PowerShell y wevtutil. Es importante señalar que la generación de actividad descrita en el apartado siguiente no se ejecutó una única vez, sino que se repitió antes de la aplicación de cada uno de los tres métodos evaluados. De esta manera, previo a la eliminación mediante Event Viewer, PowerShell y wevtutil respectivamente, se cerraron las aplicaciones abiertas y se generó nuevamente actividad en el sistema (apertura de carpetas, creación de documentos, ejecución de comandos), garantizando que el registro de seguridad contuviera eventos recientes antes de cada prueba. Esta regeneración de actividad se evidencia en que, tras la aplicación de cada uno de los tres métodos, el sistema generó consistentemente el Event ID 1102, confirmando que en cada caso existía contenido real en el log al momento de su eliminación.

3.1.2.3.1. Generación de actividad del sistema

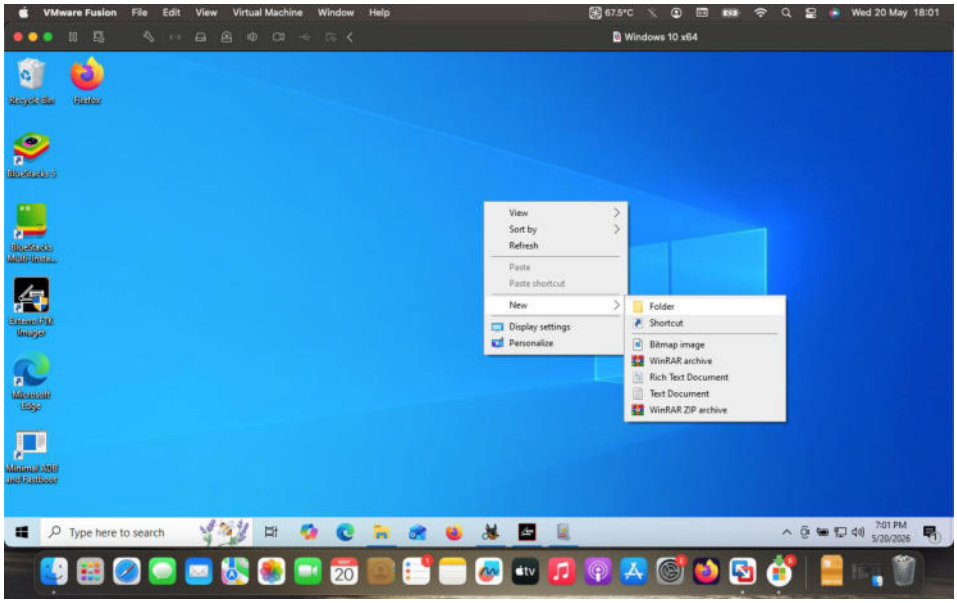
Antes de aplicar las técnicas antiforenses, se generó actividad dentro del sistema operativo con el objetivo de producir registros de eventos que posteriormente pudieran ser eliminados y analizados.

Para ello se realizaron las siguientes acciones:

- Apertura de aplicaciones del sistema.
- Ejecución de comandos administrativos.
- Uso de PowerShell.
- Acceso al Visor de Eventos.
- Navegación entre herramientas del sistema.

Figura 15

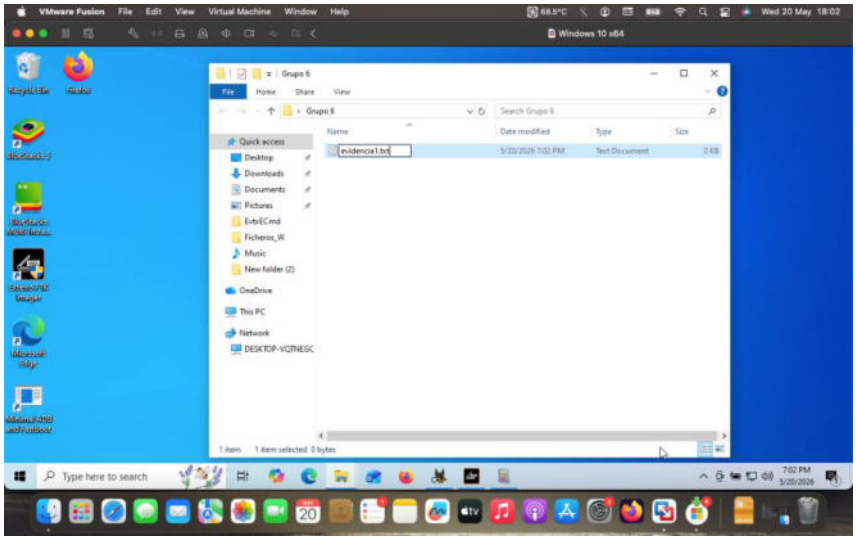
Carpeta de prueba.



Nota. La figura presenta la forma en crear una carpeta en el escritorio de Windows. Fuente: Elaboración propia.

Figura 16

Archivo .txt.

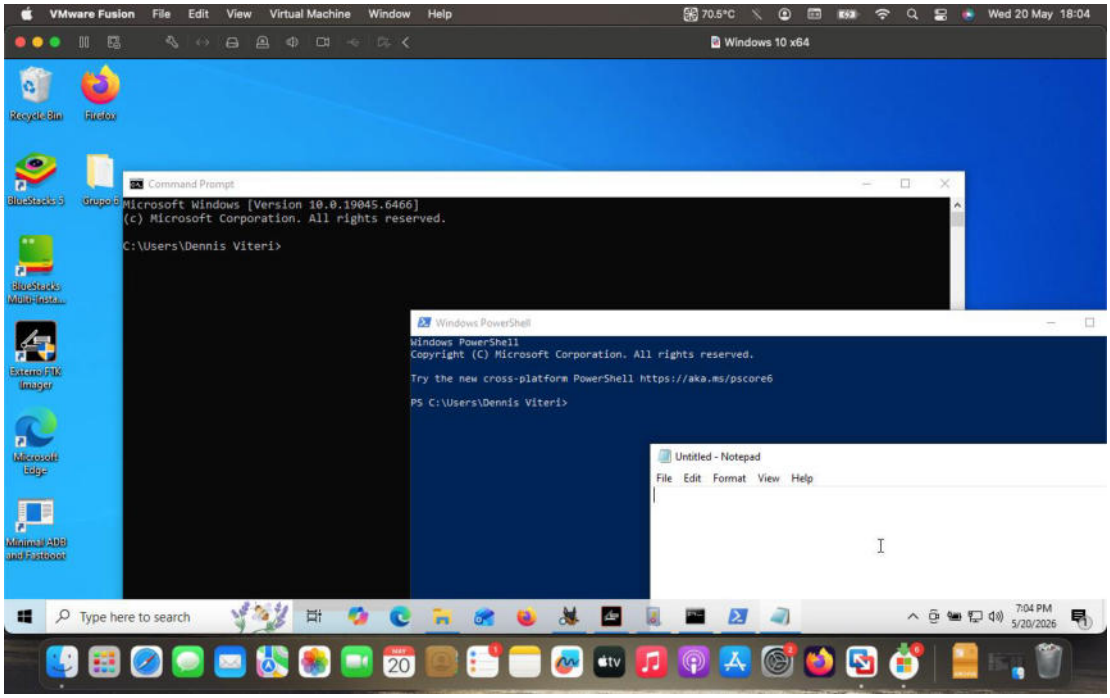


Nota. La figura presenta el archivo .txt creado dentro de una carpeta.

Fuente: Elaboración propia.

Figura 17

Apertura de aplicaciones del sistema.

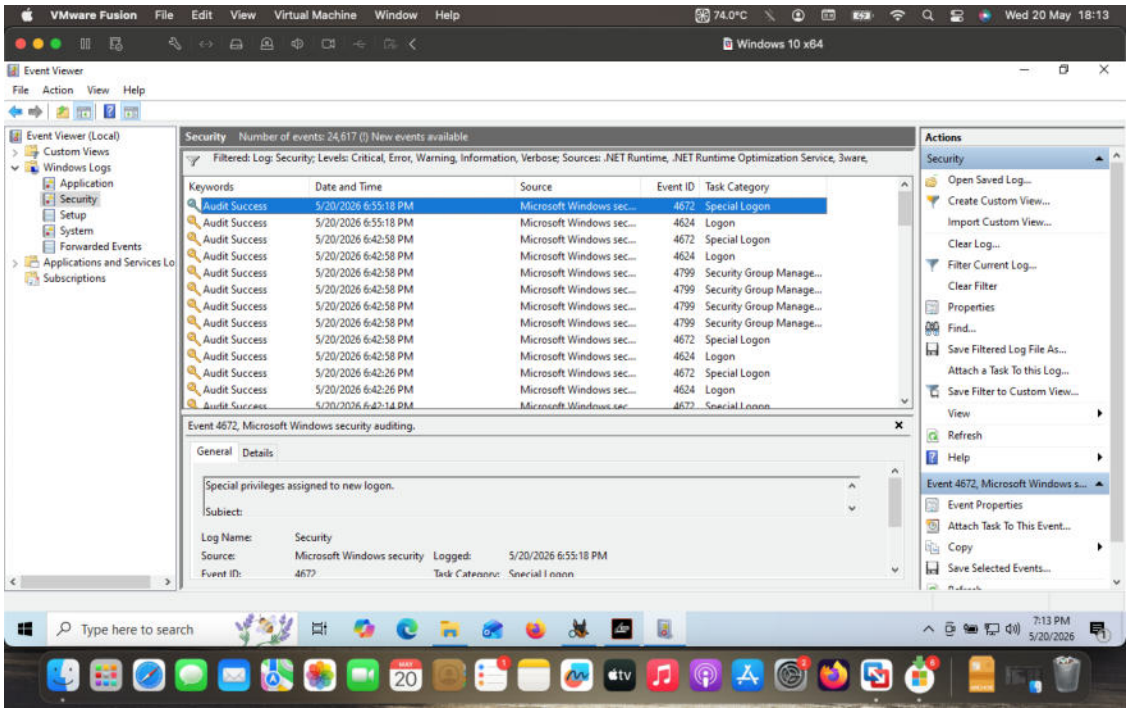


Nota. La figura presenta CMD, POWERSHELL y el Notepad abiertos en el escritorio.

Fuente: Elaboración propia.

Figura 18

Event Viewer.



Nota. La figura presenta Event Viewer donde se podrá obersarvar los Logs de Windows.

Fuente: Elaboración propia.

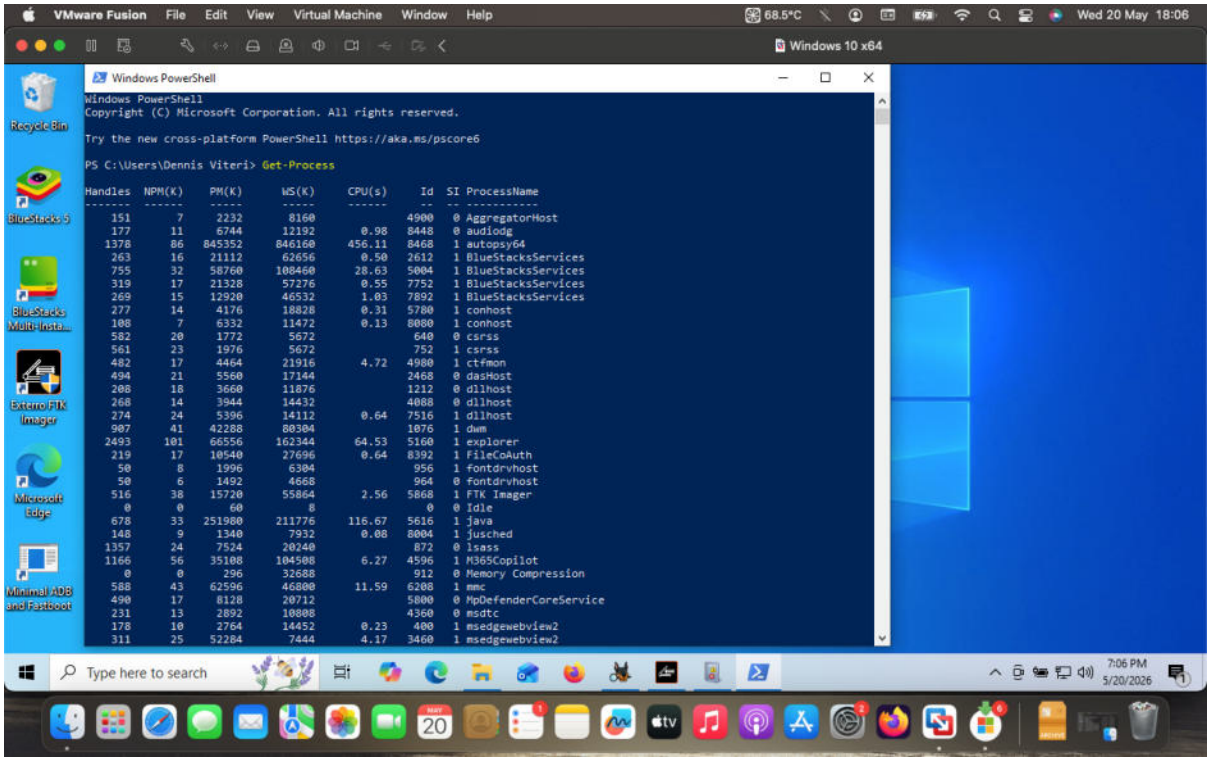
Además, se ejecutaron comandos en PowerShell como:

PS C:\Users\Get-Date

PS C:\Users\Get-Process

Figura 19

Comandos en Powershell.



Nota. La figura presenta Powershell realizando la ejecución del código. Fuente: Elaboración propia.

Estas acciones permitieron generar actividad visible dentro de los registros de eventos y artefactos del sistema operativo.

Posteriormente, se verificó la existencia de eventos dentro del registro de seguridad accediendo al Visor de Eventos mediante:

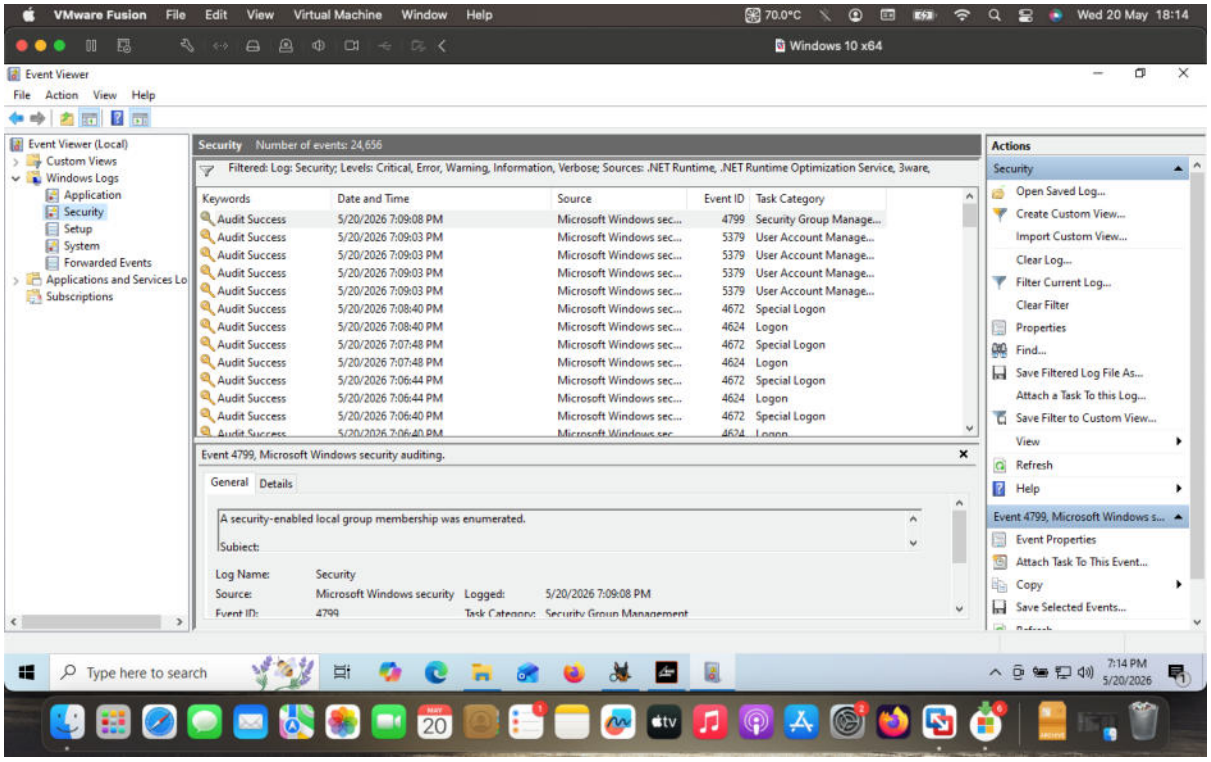
eventvwr.msc

y navegando hacia:

Windows Logs > Security

Figura 20

Logs en Event Viewer.



Nota. La figura presenta los Logs que se han generado desde que inicio Windows. Fuente: Elaboración propia.

3.1.2.3.2. Aplicación de la técnica antiforense

Una vez generada la actividad inicial del sistema, se procedió a aplicar la técnica antiforense de eliminación de logs.

3.1.2.3.3. Eliminación manual mediante Event Viewer

Como primera prueba, se realizó la eliminación manual del log de seguridad utilizando el Visor de Eventos de Windows.

El procedimiento consistió en:

1. Acceder al registro Security.

2. Seleccionar la opción:

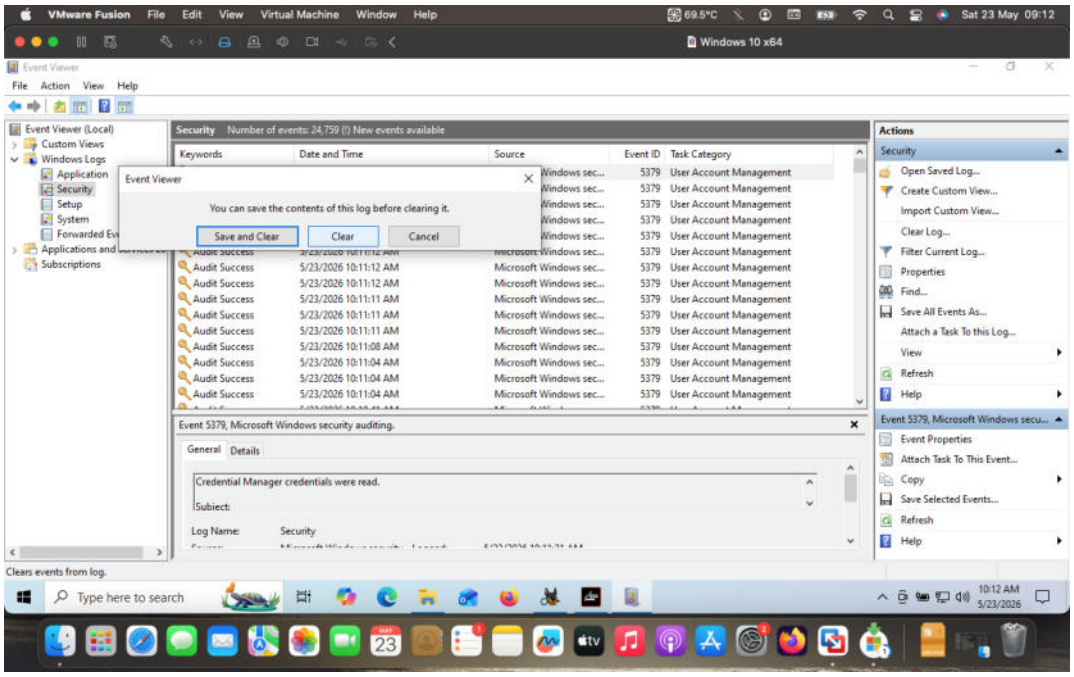
Clear Log

3. Confirmar la eliminación sin guardar respaldo.

Esta acción eliminó los eventos almacenados en el registro de seguridad del sistema.

Figura 21

Eliminación de los Logs en Event Viewer

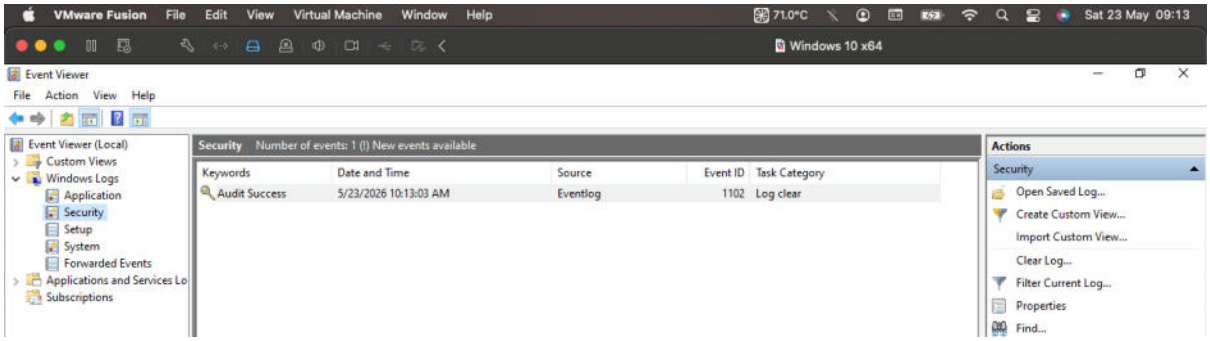


Nota. La figura presenta la forma de eliminar los Logs dando en la opción de Clear Log en Event Viewer. Fuente: Elaboración propia.

Después de ejecutar el comando, se verificó la eliminación del contenido del log Security mediante el Visor de Eventos.

Figura 22

Eliminación de Logs en Event Viewer.



Nota. La figura presenta la eliminación de los Logs en Event Viewer. Fuente: Elaboración propia.

3.1.2.3.4. Eliminación mediante PowerShell

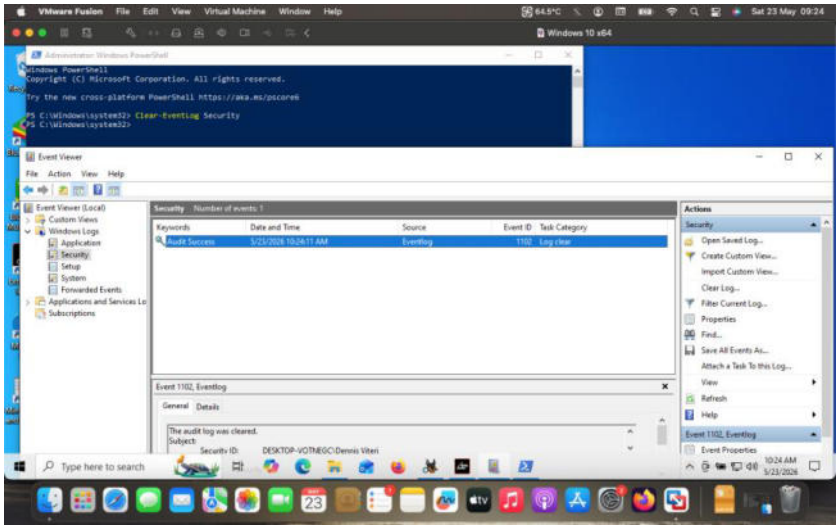
Posteriormente, se realizaron pruebas utilizando PowerShell con privilegios administrativos.

Se utilizó el comando:

```
PS C:\Windows\system32> Clear-EventLog Security
```

Figura 23

Eliminación de Logs mediante Powershell.



Nota. La figura presenta la eliminación de los Logs mediante Powershell, también se observa

el resultado en Event Viewer. Fuente: Elaboración propia.

3.1.2.3.5. Eliminación mediante wevtutil

Finalmente, se utilizó la utilidad nativa wevtutil, ampliamente utilizada en administración de eventos de Windows y comúnmente relacionada con técnicas antiforenses orientadas a manipulación de registros.

El comando ejecutado fue:

```
PS C:\Windows\system32> wevtutil cl Security
```

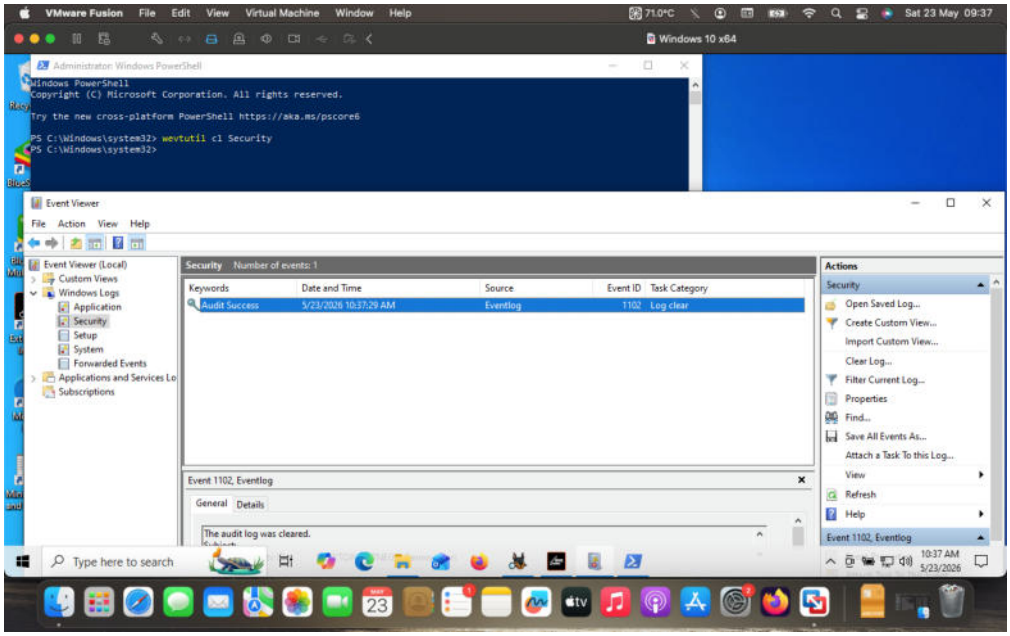
El parámetro:

```
cl
```

corresponde a la acción “clear log”, utilizada para limpiar completamente el contenido del registro seleccionado.

Figura 24

Eliminación de Logs mediante wevtutil.



Nota. La figura presenta la eliminación de los Logs mediante el comando wevtutil, también se observa el resultado en Event Viewer. Fuente: Elaboración propia.

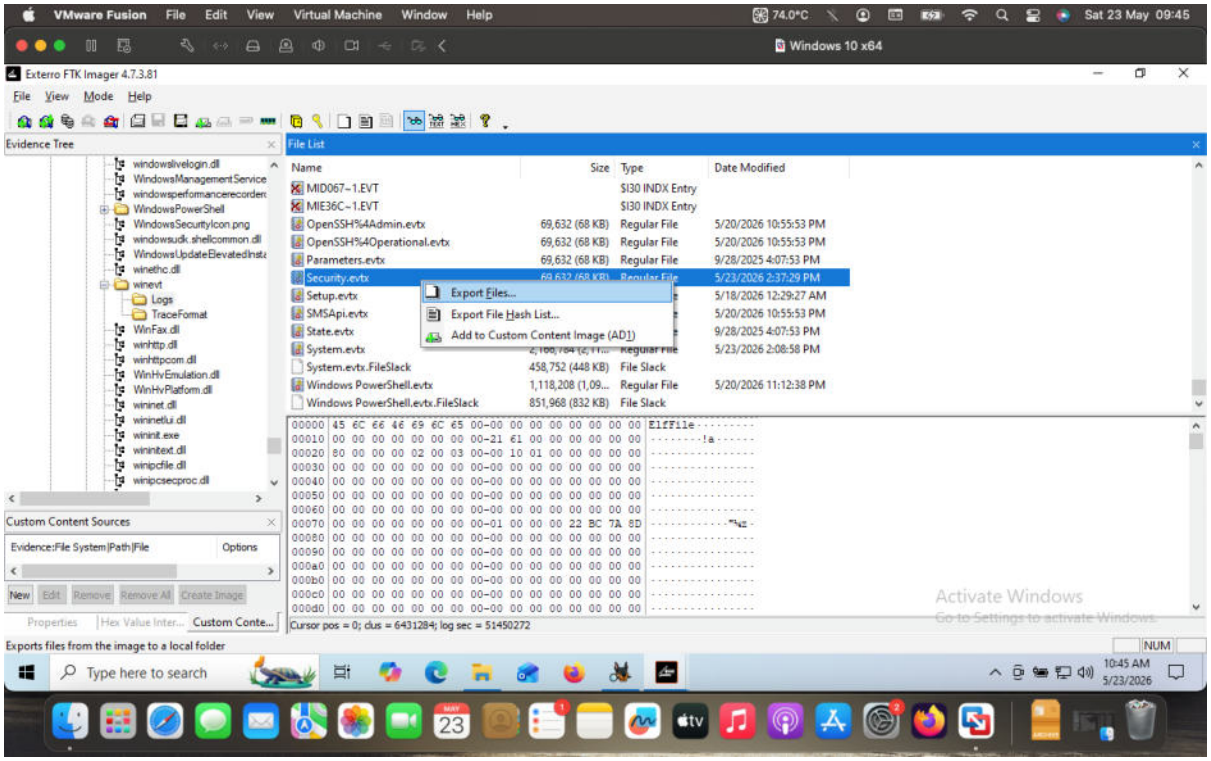
Después de ejecutar el comando, se verificó la eliminación del contenido del log Security mediante el Visor de Eventos.

3.1.2.3.6. Adquisición de evidencia digital

Una vez aplicada la técnica antiforense, se realizó la adquisición de evidencia digital utilizando FTK Imager.

Figura 25

Exportando archivo en FTK Imager.



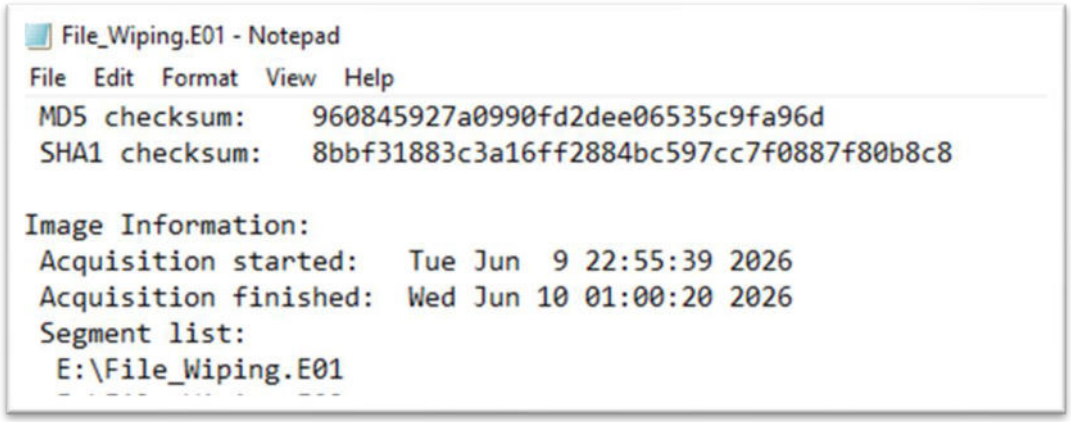
Nota. La figura presenta la forma en exportar los archivos necesarios para verificar los Logs eliminados. Fuente: Elaboración propia.

La herramienta permitió acceder al sistema de archivos y exportar artefactos relevantes para el análisis preliminar. Los archivos fueron exportados hacia una carpeta de evidencia para su posterior análisis forense. Es importante precisar que este procedimiento corresponde a una exportación lógica de archivos, dado que no involucra la generación de una imagen completa del medio de almacenamiento ni el cálculo de funciones hash sobre los archivos individuales en esa etapa. Con el objetivo de respaldar el análisis con una adquisición forense completa, se generó posteriormente una imagen física del disco mediante FTK Imager (versión 4.7.3.81), correspondiente al mismo entorno virtual en el que se ejecutó la técnica de eliminación de logs. La imagen fue creada en formato E01, segmentada en 21 volúmenes, con un tamaño total de origen de 61,440 MB. El proceso de adquisición se documentó de la siguiente manera:

```
Caso: 01
Evidencia: 001
Examinador: Grupo6
Inicio de adquisición: 09 de junio de 2026, 22:55:39
Fin de adquisición: 10 de junio de 2026, 01:00:20
Verificación iniciada: 10 de junio de 2026, 01:00:21
Verificación finalizada: 10 de junio de 2026, 02:19:23
```

Figura 26

Archivo de reporte por FTK Imager.



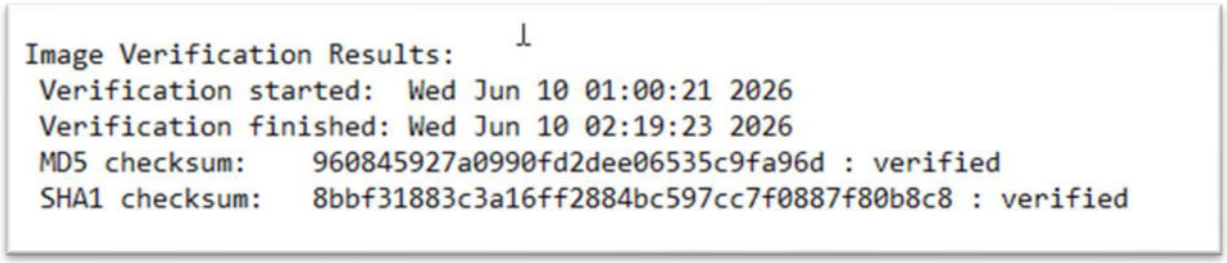
Nota. La figura presenta el reporte generado por FTK Imager al momento de crear la imagen del disco local C. Fuente: Elaboración propia.

La integridad de la imagen fue verificada mediante el cálculo y comprobación de funciones hash MD5 y SHA1, ambas con resultado "verified":

```
MD5: 960845927a0990fd2dee06535c9fa96d
SHA1: 8bbf31883c3a16ff2884bc597cc7f0887f80b8c8
```

Figura 27

Archivo de verificación por FTK Imager.



Nota. La figura presenta la verificación de resultados generado por FTK Imager al

momento de crear la imagen del disco local C. Fuente: Elaboración propia.

Esta imagen física constituye la evidencia forense primaria utilizada para el análisis del archivo Security.evtx y de los artefactos asociados a la eliminación de logs de eventos, garantizando la trazabilidad y la integridad de la evidencia conforme a los principios de la cadena de custodia digital.

Los principales archivos extraídos fueron:

Tabla 6.

Archivos extraídos

Archivo	Descripción
Security.evtx	Registro de eventos de seguridad
POWERSHELL.EXE.pf	Artefacto Prefetch asociado a PowerShell
WEVTUTIL.EXE.pf	Artefacto Prefetch asociado a wevtutil
CMD.EXE.pf	Artefacto Prefetch asociado al intérprete de comandos

Nota. La tabla presenta los principales artefactos digitales adquiridos mediante FTK Imager para su posterior análisis forense en Autopsy. Fuente: Elaboración propia.

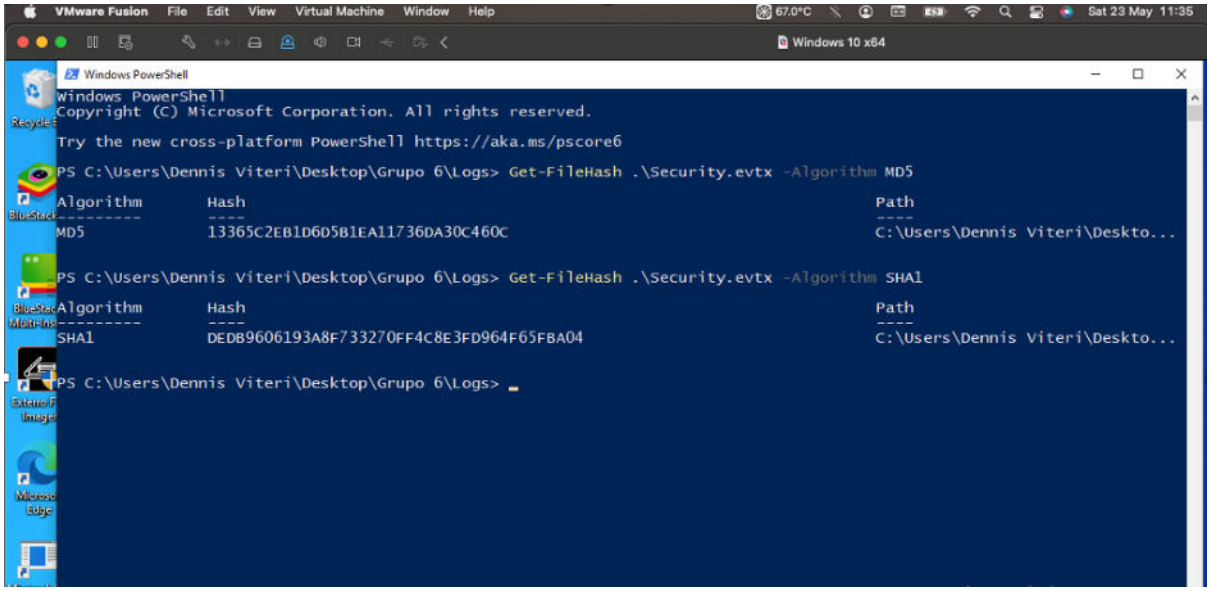
Los archivos fueron exportados hacia una carpeta de evidencia para su posterior análisis forense. Con el fin de garantizar la integridad del archivo security.evtx exportado, se calcularon sus respectivas funciones hash mediante PowerShell:

```
MD5: 13365C2EB1D6D5B1EA11736DA30C460C

SHA1: DEDB9606193A8F733270FF4C8E3FD964F65FBA04
```

Estos valores permiten verificar que el contenido del archivo no fue alterado durante su exportación y manipulación posterior dentro del proceso de análisis forense.

Figura 28

Cálculo de hash MD5 y SHA1 del archivo Security.evtx.



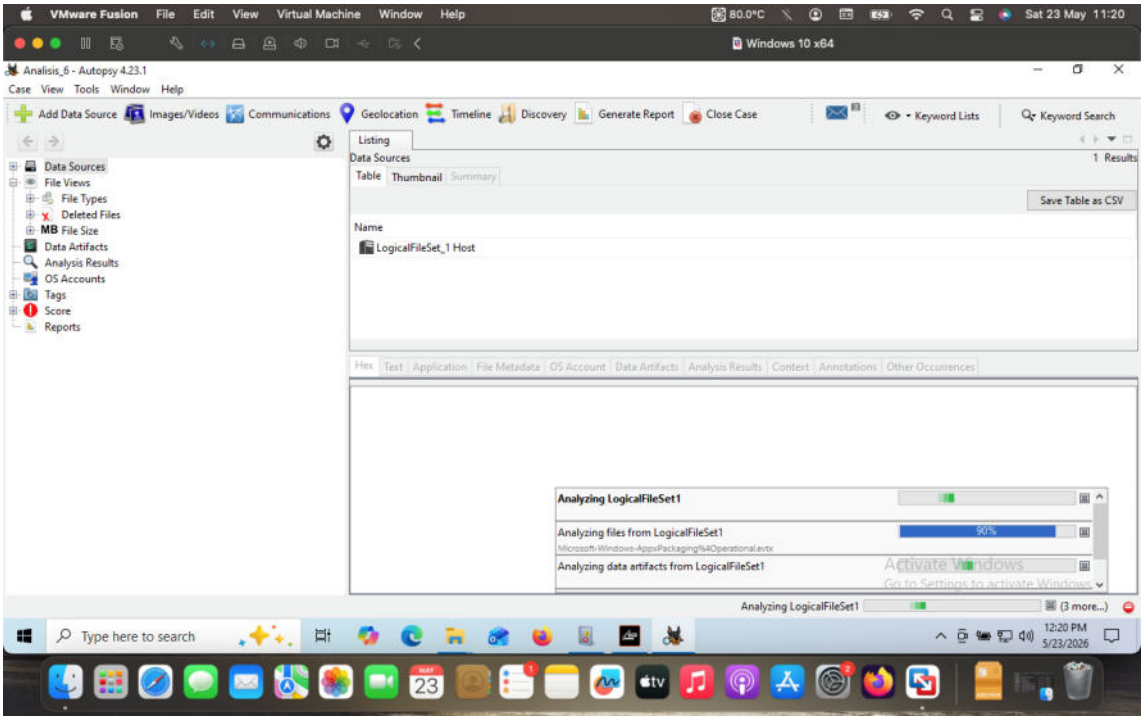
Nota. La figura presenta el cálculo de las funciones hash MD5 y SHA1 del archivo Security.evtx exportado, mediante el cmdlet Get-FileHash de PowerShell, con el objetivo de verificar su integridad. Fuente: Elaboración propia.

3.1.2.3.7. Análisis forense mediante Autopsy

El análisis forense se realizó en dos etapas, utilizando dos fuentes de evidencia complementarias. En una primera etapa, se analizaron mediante Autopsy los archivos obtenidos a través de la exportación lógica descrita en el apartado anterior (modalidad Logical Files). Durante este proceso se creó un caso forense y se importaron los archivos extraídos previamente mediante FTK Imager.

Figura 29

Análisis mediante Autopsy por archivo exportado.



Nota. La figura presenta la forma en analizar los archivos exportados de FTK Imager para verificar los Logs eliminados previamente. Fuente: Elaboración propia.

El análisis permitió identificar:

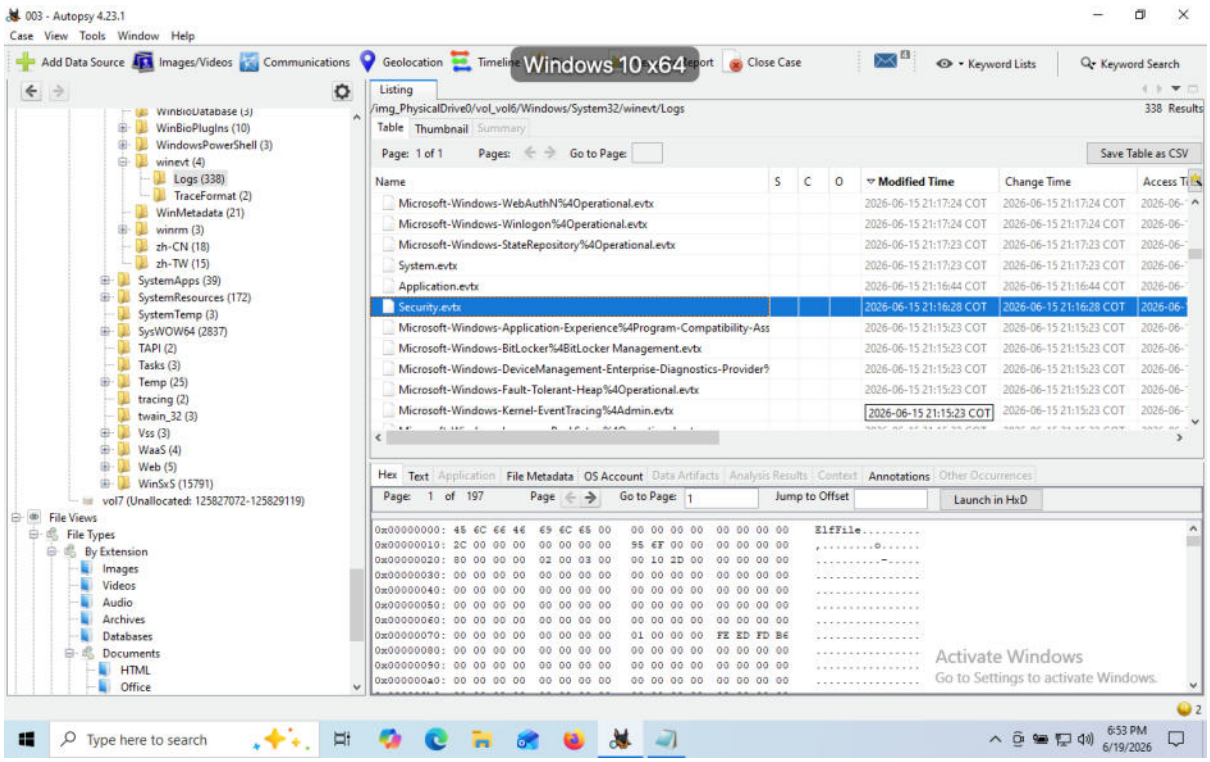
- Archivos Prefetch asociados a herramientas administrativas.
- Persistencia de artefactos relacionados con PowerShell.
- Existencia del archivo Security.evtx aun después de la eliminación de eventos.
- Evidencia residual relacionada con manipulación del sistema.

En una segunda etapa, con el objetivo de validar los hallazgos sobre una fuente de evidencia forense completa, se cargó en Autopsy la imagen física E01 descrita en el apartado 3.1.2.3.6, mediante la modalidad Disk Image. Navegando hacia la ruta original del

sistema de archivos (Windows/System32/winevt/Logs/), se confirmó la existencia del archivo Security.evtx dentro de la imagen forense, validando que el artefacto identificado en la exportación lógica corresponde efectivamente al archivo presente en el disco adquirido.

Figura 30

Análisis mediante Autopsy por imagen de disco local.



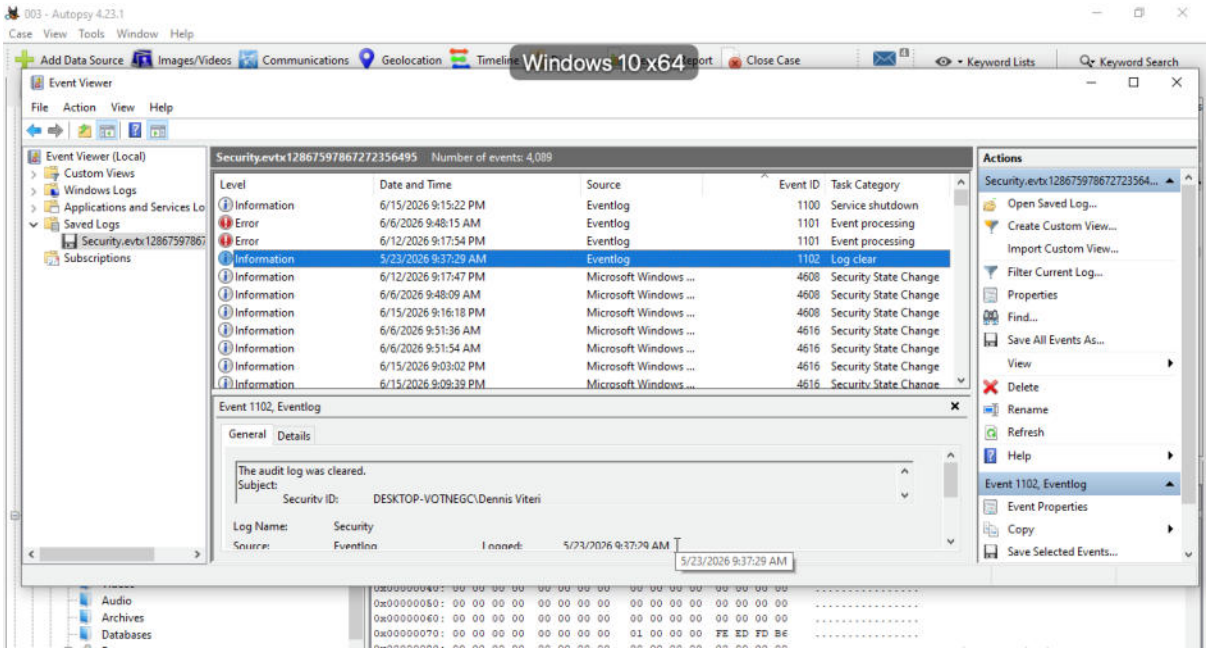
Nota. La figura presenta la forma en analizar la imagen de disco local C que se creó con FTK Imager. Fuente: Elaboración propia.

Con el fin de validar la integridad del hallazgo, dicho archivo fue extraído desde la imagen física y abierto externamente mediante el Visor de Eventos de Windows, confirmándose nuevamente la presencia del Event ID 1102. Este resultado coincide

con lo identificado previamente sobre la copia obtenida mediante exportación lógica, validando que el artefacto recuperado desde la adquisición forense completa corresponde efectivamente al mismo evento de eliminación detectado en la primera etapa del análisis. Adicionalmente, se realizaron búsquedas por palabras clave y análisis de metadata con el objetivo de correlacionar la actividad antiforense realizada.

Figura 31

Verificación de archivo con Event Viewer.

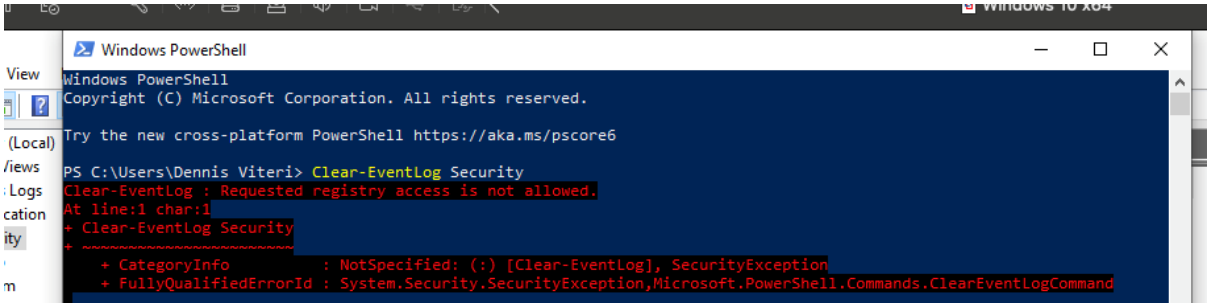


Nota. La figura presenta la aplicación externa de autopsy para verificar el archivo security.evtx. Fuente: Elaboración propia.

3.1.2.4. Evidencia visual

Figura 32

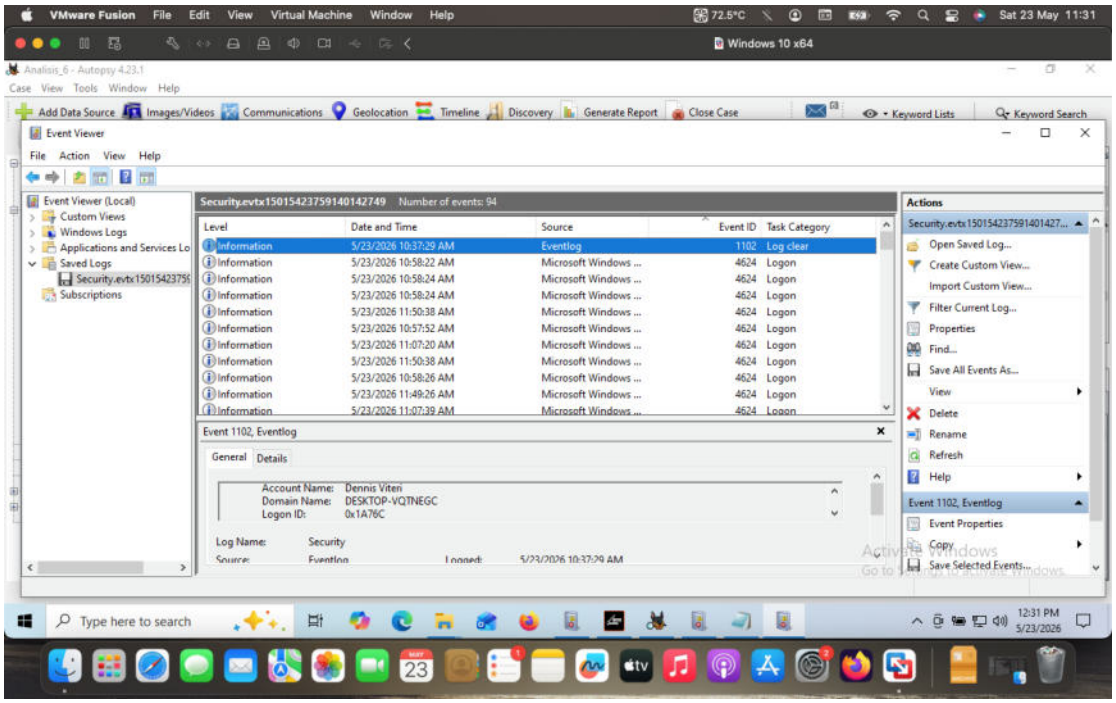
Comando Clear-Eventlog.



Nota. La figura presenta la implementación del comando en PowerShell sin permisos de administrador lo cual dio conflicto por la falta de permisos. Fuente: Elaboración propia.

Figura 33

Security.evtx en Event Viewer.



Nota. La figura presenta como desde Autopsy se abre el archivo security.evtx se abre en

Event Viewer y se verifica el Event ID 1102. Fuente: Elaboración propia.

3.1.2.5. Artefactos afectados

La técnica antiforense aplicada afectó diversos artefactos digitales del sistema operativo Windows.

Tabla 7.

Archivos afectados

Artefacto	Descripción
Security.evtx	Registro principal de eventos de seguridad
Prefetch (.pf)	Rastros de ejecución de aplicaciones
PowerShell activity	Evidencia residual de comandos ejecutados
Timestamps	Marcas temporales asociadas a modificaciones
Logs de eventos	Información relacionada con actividad del sistema

Fuente:Elaboración propio

A pesar de la eliminación de registros, varios artefactos permanecieron disponibles para análisis forense posterior.

3.1.2.6. Método de detección aplicado

La detección de la actividad antiforense se realizó mediante técnicas de análisis forense digital estándar utilizando FTK Imager y Autopsy.

El procedimiento de detección consistió en:

- Exportación de artefactos relevantes del sistema.
- Identificación de archivos Prefetch.
- Análisis de metadata.
- Correlación temporal de actividad.
- Verificación de evidencia residual.

Especial atención se dio a los archivos Prefetch asociados a PowerShell y herramientas administrativas, debido a que estos permiten identificar la ejecución previa de programas incluso cuando los logs principales han sido eliminados.

3.1.2.7. Resultados obtenidos

Los resultados obtenidos durante el laboratorio demostraron que la eliminación de logs de eventos en Windows no elimina completamente la trazabilidad digital del sistema. Aunque los registros originales fueron eliminados exitosamente, el análisis forense permitió identificar múltiples artefactos residuales relacionados con la actividad antiforense realizada.

Entre los principales hallazgos destacan:

- Persistencia de archivos Prefetch relacionados con PowerShell y herramientas administrativas.
- Existencia residual del archivo Security.evtx.
- Evidencia de ejecución de comandos administrativos.
- Persistencia de metadata asociada a los archivos analizados.

Los resultados evidencian que las técnicas antiforenses orientadas únicamente a la eliminación de logs presentan limitaciones significativas frente a procesos de análisis forense digital.

3.1.3. Eliminación de archivos prefetch

3.1.3.1. Descripción de la Técnica

La eliminación de archivos Prefetch es considerada una técnica antiforense utilizada para dificultar el análisis de evidencias en sistemas operativos Windows. Estos archivos,

con extensión .pf, son creados automáticamente por el sistema para optimizar la carga y ejecución de aplicaciones utilizadas con frecuencia.

Desde el ámbito forense, los archivos Prefetch son de gran importancia, ya que permiten obtener información relacionada con programas ejecutados en el equipo, fechas de última ejecución e incluso la frecuencia de uso de determinadas aplicaciones. Gracias a ello, los analistas pueden reconstruir actividades realizadas dentro de un sistema comprometido.

Debido a esto, un atacante puede optar por eliminar los archivos Prefetch con el fin de ocultar rastros de ejecución de herramientas o programas utilizados durante un incidente. Esta acción busca reducir la cantidad de evidencia disponible y complicar el proceso de investigación forense y análisis de eventos en el sistema afectado.

En esta investigación, además de evaluar el impacto de la eliminación de los archivos Prefetch, se incorpora una fase experimental de recuperación del artefacto mediante herramientas forenses. Esta evaluación permite determinar si los archivos eliminados pueden recuperarse y si la información contenida en ellos conserva la integridad necesaria para apoyar un análisis forense, fortaleciendo así la valoración del criterio de recuperabilidad dentro de la matriz de evaluación propuesta.

3.1.3.2. Herramientas utilizadas

La siguiente tabla resume las principales herramientas utilizadas durante el desarrollo de los laboratorios y la función que cumplió cada una dentro del proceso de generación, manipulación, recuperación y análisis forense de los archivos Prefetch. Cada herramienta permitió cubrir una fase específica del procedimiento experimental, desde

la preparación del entorno hasta la evaluación de la integridad de la evidencia recuperada.

En conjunto, estas herramientas facilitaron la generación de los artefactos Prefetch, la aplicación de la técnica antiforense mediante su eliminación, la recuperación de los archivos utilizando una herramienta especializada y el análisis comparativo de la información forense contenida en los artefactos originales y recuperados. Este proceso permitió evaluar no solo el impacto de la técnica sobre la evidencia digital, sino también la recuperabilidad e integridad de los artefactos obtenidos durante el laboratorio.

Tabla 8.

Herramientas utilizadas en la implementación de la técnica de eliminación de archivos Prefetch

Herramienta	**Función**
VMware Workstation	Virtualización del entorno de laboratorio
Windows 10 Pro	Sistema operativo objetivo
Windows 11 Home	Sistema operativo físico
CMD / PowerShell	Eliminación de archivos Prefetch
WinPrefetchView	Visualización de archivos Prefetch
Recuva	Recuperación de archivos Prefetch eliminados y evaluación de su estado de recuperabilidad.

Nota. La tabla presenta las herramientas y componentes utilizados durante el desarrollo del laboratorio forense.

3.1.3.3. Procedimiento realizado

3.1.3.3.1. Preparación del entorno

Para la realización de los laboratorios se implementó un entorno virtualizado mediante VMware Workstation sobre un equipo host con sistema operativo Windows 11. Dentro de este entorno se desplegó una máquina virtual con Windows 10 Pro, utilizada como sistema operativo objetivo para la generación, eliminación, recuperación y análisis de evidencias digitales relacionadas con los archivos Prefetch.

Como parte de la preparación del laboratorio, se verificó la existencia de la carpeta Prefetch dentro del sistema operativo, ubicada en la siguiente ruta:

C:\Windows\Prefetch

Este directorio almacena los archivos con extensión **.pf**, generados automáticamente por Windows a partir de la ejecución de aplicaciones y procesos del sistema. Debido a que se trata de una carpeta protegida del sistema operativo, fue necesario acceder con privilegios de administrador para visualizar, copiar, eliminar y recuperar los artefactos utilizados durante el desarrollo de la práctica.

Con el entorno preparado, se ejecutaron las aplicaciones seleccionadas para generar nuevos archivos Prefetch, los cuales fueron verificados mediante herramientas de análisis antes de aplicar la técnica antiforense. Posteriormente, estos artefactos fueron utilizados para evaluar tanto el efecto de su eliminación como su recuperabilidad e integridad mediante herramientas especializadas de recuperación y análisis forense.

3.1.3.3.2. Generación de evidencia

Con el propósito de generar artefactos Prefetch válidos para el análisis forense, se ejecutaron distintas aplicaciones nativas del sistema operativo Windows, entre ellas Símbolo del sistema (CMD)**,** PowerShell**,** Paint **y** WordPad. La ejecución controlada de estas aplicaciones permitió provocar la creación automática de nuevos archivos Prefetch, los cuales constituyeron la evidencia digital utilizada durante el desarrollo de los laboratorios.

Una vez ejecutadas las aplicaciones, el sistema operativo generó automáticamente los correspondientes archivos con extensión **.pf** dentro de la carpeta Prefetch, registrando información asociada a la ejecución de cada programa. Estos artefactos fueron verificados mediante PowerShell **y** WinPrefetchView, confirmando su correcta generación y disponibilidad para las fases posteriores de eliminación, recuperación y análisis forense.

3.1.3.3.3. Aplicación de la técnica antiforense

Una vez verificada la generación de los archivos Prefetch, se accedió al directorio **C:\Windows\Prefetch** mediante PowerShell con el propósito de identificar los artefactos creados durante la ejecución de las aplicaciones utilizadas en el laboratorio. Para facilitar su localización, se empleó el siguiente comando:

```
Get-ChildItem C:\Windows\Prefetch\*.pf | Sort-Object LastWriteTime
```

Este procedimiento permitió ordenar los archivos **.pf** según su fecha de modificación, facilitando la identificación de los artefactos generados recientemente. De manera complementaria, se utilizó la herramienta WinPrefetchView, la cual proporcionó una

visualización detallada de la información contenida en cada archivo Prefetch, incluyendo la aplicación ejecutada, la fecha de última ejecución y otros metadatos relevantes para el análisis forense.

Una vez identificados los archivos correspondientes a las aplicaciones ejecutadas durante la práctica, se procedió a eliminarlos del directorio C:\Windows\Prefetch, simulando la aplicación de la técnica antiforense. Esta acción tuvo como finalidad suprimir los artefactos de ejecución almacenados por el sistema operativo, reduciendo la disponibilidad de evidencia potencialmente útil durante una investigación forense.

Finalmente, se verificó la ausencia de los archivos eliminados mediante PowerShell y WinPrefetchView, confirmando que los artefactos Prefetch ya no se encontraban disponibles para su análisis dentro del sistema operativo.

3.1.3.3.4. Recolección de evidencia digital

Una vez identificados los archivos Prefetch generados durante la ejecución de las aplicaciones seleccionadas, se realizó la recolección lógica de la evidencia digital mediante la copia del artefacto correspondiente desde la ruta **C:\Windows\Prefetch** hacia un directorio de trabajo previamente definido. Esta actividad tuvo como objetivo preservar una copia íntegra del archivo original antes de aplicar la técnica antiforense.

La evidencia recolectada se utilizó como muestra de referencia durante el laboratorio de recuperación, permitiendo comparar el artefacto original con el archivo recuperado y evaluar la integridad de la información forense contenida en ambos. De esta manera, fue posible analizar el impacto de la eliminación sobre la recuperabilidad del artefacto

y determinar si la información relevante para una investigación forense se conservó tras el proceso de recuperación.

3.1.3.3.5. Análisis de artefactos Prefetch

Los archivos Prefetch obtenidos durante el desarrollo de los laboratorios fueron analizados mediante la herramienta **WinPrefetchView**, con el propósito de evaluar la información forense contenida en los artefactos antes y después de la aplicación de la técnica antiforense. Para ello, se examinó el archivo original previamente recolectado y el archivo recuperado posteriormente, permitiendo realizar un análisis comparativo de la información disponible en ambos.

Durante el análisis se verificó la presencia de metadatos relevantes, como el nombre de la aplicación ejecutada, la ruta del proceso, la fecha de última ejecución, el contador de ejecuciones y otros registros asociados al funcionamiento del sistema operativo. Asimismo, se evaluó si la recuperación del archivo permitía conservar la información necesaria para apoyar una investigación forense o si existían pérdidas de integridad que limitaran su utilidad como evidencia digital.

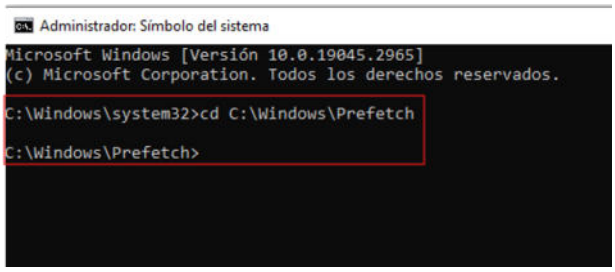
Los resultados obtenidos permitieron determinar el impacto de la eliminación de los archivos Prefetch sobre la recuperabilidad del artefacto y establecer el grado de conservación de la información forense después del proceso de recuperación.

Evidencia visual

Generación y eliminación de archivos prefetch

Figura 34

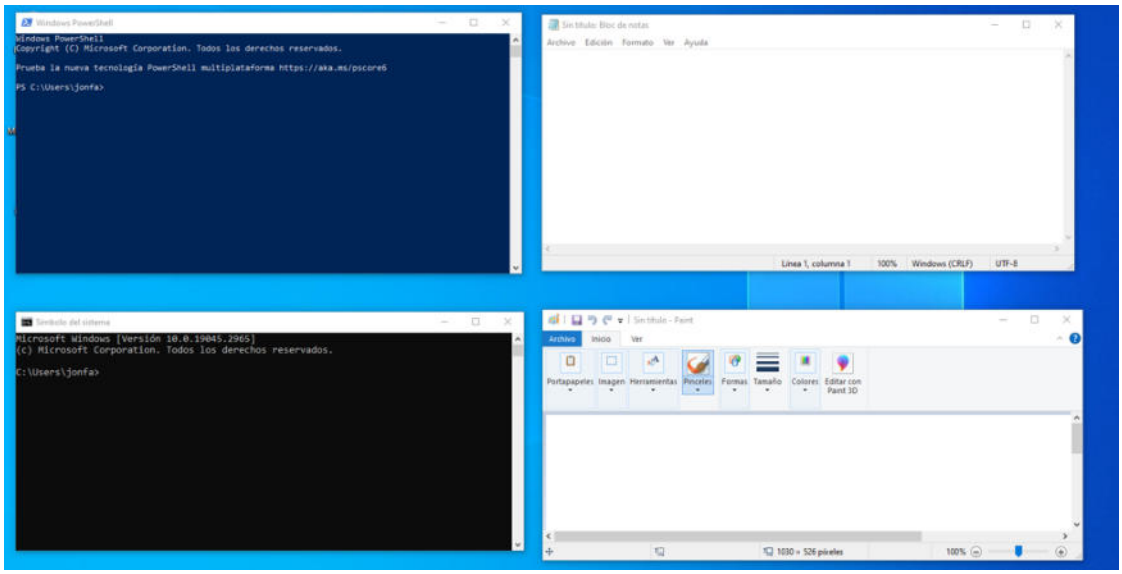
Verificación inicial de la carpeta Prefetch.



Nota. Acceso al directorio C:\Windows\Prefetch mediante CMD para verificar la existencia de artefactos Prefetch.

Figura 35

Ejecución de aplicaciones Windows.

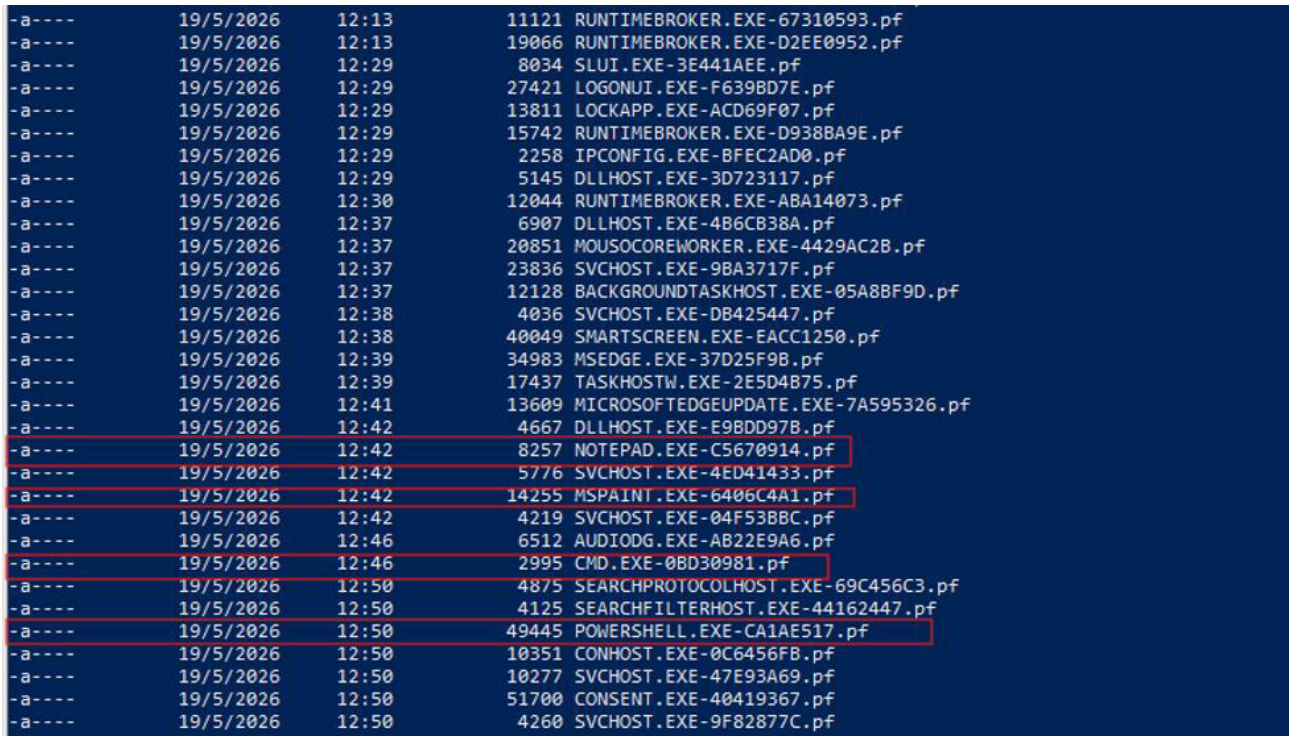


Nota. En esta imagen se evidencia la ejecución de varias aplicaciones del sistema operativo Windows, entre ellas PowerShell, Bloc de notas, Símbolo del sistema y Paint. La apertura de estos programas tiene como finalidad generar actividad en el sistema para

que Windows cree automáticamente archivos Prefetch asociados a cada aplicación ejecutada.

Figura 36

Contenido de carpeta Prefetch.

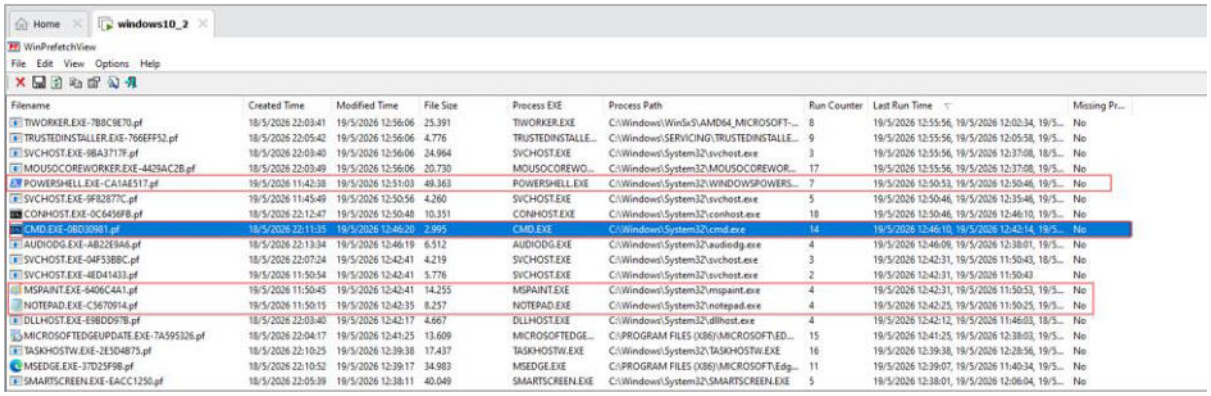


-a----	19/5/2026	12:13	11121	RUNTIMEBROKER.EXE-67310593.pf
-a----	19/5/2026	12:13	19066	RUNTIMEBROKER.EXE-D2EE0952.pf
-a----	19/5/2026	12:29	8034	SLUI.EXE-3E441AEE.pf
-a----	19/5/2026	12:29	27421	LOGONUI.EXE-F639BD7E.pf
-a----	19/5/2026	12:29	13811	LOCKAPP.EXE-ACD69F07.pf
-a----	19/5/2026	12:29	15742	RUNTIMEBROKER.EXE-D938BA9E.pf
-a----	19/5/2026	12:29	2258	IPCONFIG.EXE-BFEC2AD0.pf
-a----	19/5/2026	12:29	5145	DLLHOST.EXE-3D723117.pf
-a----	19/5/2026	12:30	12044	RUNTIMEBROKER.EXE-ABA14073.pf
-a----	19/5/2026	12:37	6907	DLLHOST.EXE-4B6CB38A.pf
-a----	19/5/2026	12:37	20851	MOUSOCOREWORKER.EXE-4429AC2B.pf
-a----	19/5/2026	12:37	23836	SVCHOST.EXE-9BA3717F.pf
-a----	19/5/2026	12:37	12128	BACKGROUNDTASKHOST.EXE-05A8BF9D.pf
-a----	19/5/2026	12:38	4036	SVCHOST.EXE-DB425447.pf
-a----	19/5/2026	12:38	40049	SMARTSCREEN.EXE-EACC1250.pf
-a----	19/5/2026	12:39	34983	MSEDGE.EXE-37D25F9B.pf
-a----	19/5/2026	12:39	17437	TASKHOSTW.EXE-2E5D4B75.pf
-a----	19/5/2026	12:41	13609	MICROSOFTEDGEUPDATE.EXE-7A595326.pf
-a----	19/5/2026	12:42	4667	DLLHOST.EXE-E9BDD97B.pf
-a----	19/5/2026	12:42	8257	NOTEPAD.EXE-C5670914.pf
-a----	19/5/2026	12:42	5776	SVCHOST.EXE-4ED41433.pf
-a----	19/5/2026	12:42	14255	MSPAINT.EXE-6406C4A1.pf
-a----	19/5/2026	12:42	4219	SVCHOST.EXE-04F53BBC.pf
-a----	19/5/2026	12:46	6512	AUDIODG.EXE-AB22E9A6.pf
-a----	19/5/2026	12:46	2995	CMD.EXE-0BD30981.pf
-a----	19/5/2026	12:50	4875	SEARCHPROTOCOLHOST.EXE-69C456C3.pf
-a----	19/5/2026	12:50	4125	SEARCHFILTERHOST.EXE-44162447.pf
-a----	19/5/2026	12:50	49445	POWERSHELL.EXE-CA1AE517.pf
-a----	19/5/2026	12:50	10351	CONHOST.EXE-0C6456FB.pf
-a----	19/5/2026	12:50	10277	SVCHOST.EXE-47E93A69.pf
-a----	19/5/2026	12:50	51700	CONSENT.EXE-40419367.pf
-a----	19/5/2026	12:50	4260	SVCHOST.EXE-9F82877C.pf

Nota. En la imagen se visualiza el contenido de la carpeta Prefetch después de la ejecución de varias aplicaciones en el sistema.

Figura 37

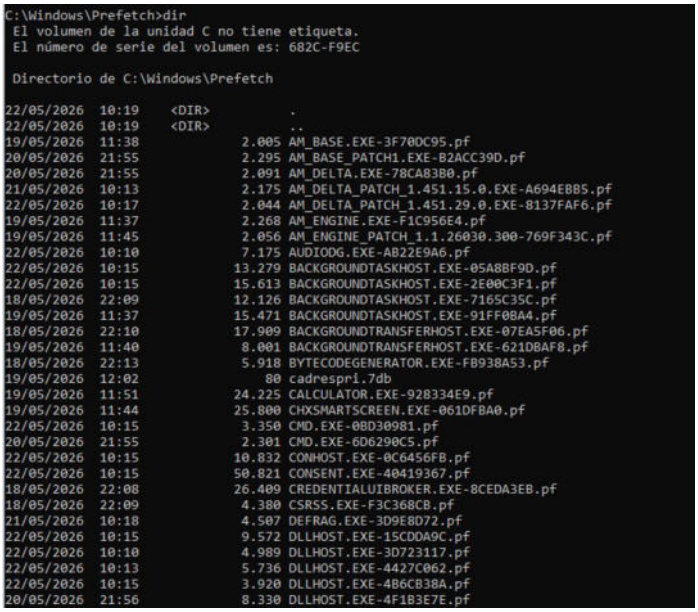
Análisis de archivos Prefetch con WinPrefetchView



Nota. En la imagen se observa el análisis de los archivos Prefetch mediante la herramienta WinPrefetchView. Se destacan los registros asociados a aplicaciones ejecutadas previamente, como POWERSHELL.EXE, CMD.EXE, MSPAINT.EXE y NOTEPAD.EXE.

Figura 38

Archivos .pf generados tras la ejecución de aplicaciones.

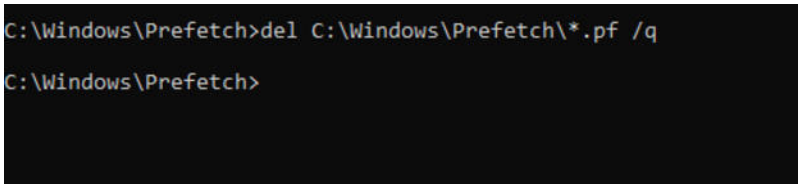


```
C:\Windows\Prefetch>dir
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: 682C-F9EC

 Directorio de C:\Windows\Prefetch

22/05/2026  10:19    <DIR>          .
22/05/2026  10:19    <DIR>          ..
19/05/2026  11:38             2.005 AM_BASE.EXE-3F70DC95.pf
20/05/2026  21:55             2.295 AM_BASE_PATCH1.EXE-B2ACC39D.pf
20/05/2026  21:55             2.091 AM_DELTA.EXE-78CAB3B0.pf
21/05/2026  10:13             2.175 AM_DELTA_PATCH_1.451.15.0.EXE-A694EBB5.pf
22/05/2026  10:17             2.044 AM_DELTA_PATCH_1.451.29.0.EXE-8137FAF6.pf
19/05/2026  11:37             2.268 AM_ENGINE.EXE-F1C956E4.pf
19/05/2026  11:45             2.056 AM_ENGINE_PATCH_1.1.26030.300-769F343C.pf
22/05/2026  10:10             7.175 AUDIODG.EXE-AB22E9A6.pf
22/05/2026  10:15            13.279 BACKGROUNDTASKHOST.EXE-05A8BF9D.pf
22/05/2026  10:15            15.613 BACKGROUNDTASKHOST.EXE-2E00C3F1.pf
18/05/2026  22:09            12.126 BACKGROUNDTASKHOST.EXE-7165C35C.pf
19/05/2026  11:37            15.471 BACKGROUNDTASKHOST.EXE-91FF0BA4.pf
18/05/2026  22:10            17.909 BACKGROUNDTRANSFERHOST.EXE-07EA5F06.pf
19/05/2026  11:40             8.001 BACKGROUNDTRANSFERHOST.EXE-621DBAF8.pf
18/05/2026  22:13             5.918 BYTECODEGENERATOR.EXE-FB938A53.pf
19/05/2026  12:02                80 cadrespri.7db
19/05/2026  11:51            24.225 CALCULATOR.EXE-928334E9.pf
19/05/2026  11:44            25.800 CHXSMARTSCREEN.EXE-061DFBA0.pf
22/05/2026  10:15             3.350 CMD.EXE-0BD30981.pf
20/05/2026  21:55             2.301 CMD.EXE-6D6290C5.pf
22/05/2026  10:15            10.832 CONHOST.EXE-0C6456FB.pf
22/05/2026  10:15            50.821 CONSENT.EXE-40419367.pf
18/05/2026  22:08            26.409 CREDENTIALUIBROKER.EXE-8CEDA3EB.pf
18/05/2026  22:09             4.380 CSRSS.EXE-F3C368CB.pf
21/05/2026  10:18             4.507 DEFRAG.EXE-3D9E8D72.pf
22/05/2026  10:15             9.572 DLLHOST.EXE-15CDDA9C.pf
22/05/2026  10:10             4.989 DLLHOST.EXE-3D723117.pf
22/05/2026  10:13             5.736 DLLHOST.EXE-4427C062.pf
22/05/2026  10:15             3.920 DLLHOST.EXE-4B6CB38A.pf
20/05/2026  21:56             8.330 DLLHOST.EXE-4F1B3E7E.pf
```

Nota. En la imagen se muestra el listado completo del directorio C:\Windows\Prefetch utilizando el comando dir desde el símbolo del sistema.

Figura 39

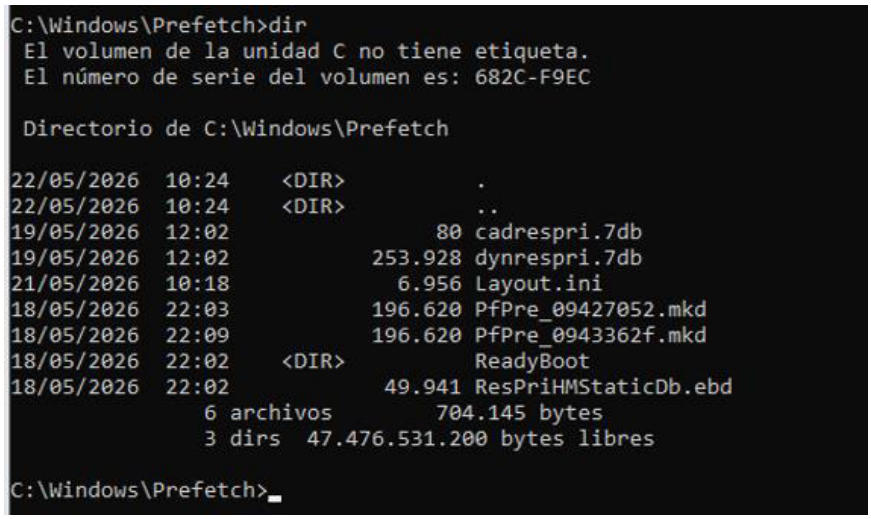
Ejecución del comando de eliminación.



Nota. En esta imagen se observa la ejecución del comando “del C:\Windows\Prefetch*.pf /q”, utilizado para eliminar los archivos almacenados en la carpeta Prefetch, aplicando la técnica antiforense de eliminación de artefactos.

Figura 40

Carpeta Prefetch posterior a la eliminación.



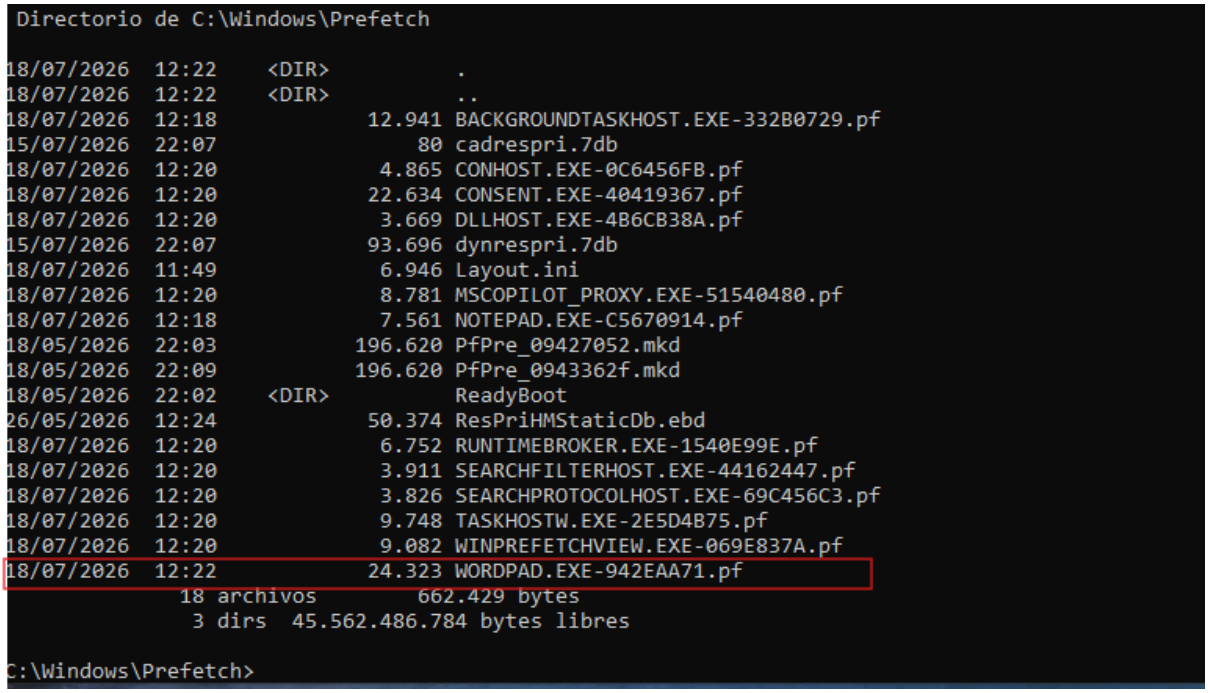
Nota. En la imagen se observa nuevamente el contenido de la carpeta C:\Windows\Prefetch después de aplicar la técnica antiforense de eliminación de archivos Prefetch. Se evidencia que los archivos .pf asociados a la ejecución de aplicaciones fueron eliminados exitosamente.

Recuperación de archivos prefetch

Como parte del laboratorio de recuperación, se ejecutó la aplicación **WordPad** con el propósito de generar un nuevo artefacto Prefetch dentro del sistema operativo Windows. Posteriormente, se accedió al directorio **C:\Windows\Prefetch** mediante la línea de comandos para verificar la creación del archivo correspondiente. En la Figura X se observa la presencia del archivo **WORDPAD.EXE-942EAA71.pf**, confirmando la correcta generación del artefacto que fue utilizado como evidencia de referencia para las fases posteriores de recolección, eliminación, recuperación y análisis comparativo.

Figura 41.

Generación e identificación del archivo Prefetch correspondiente a WordPad



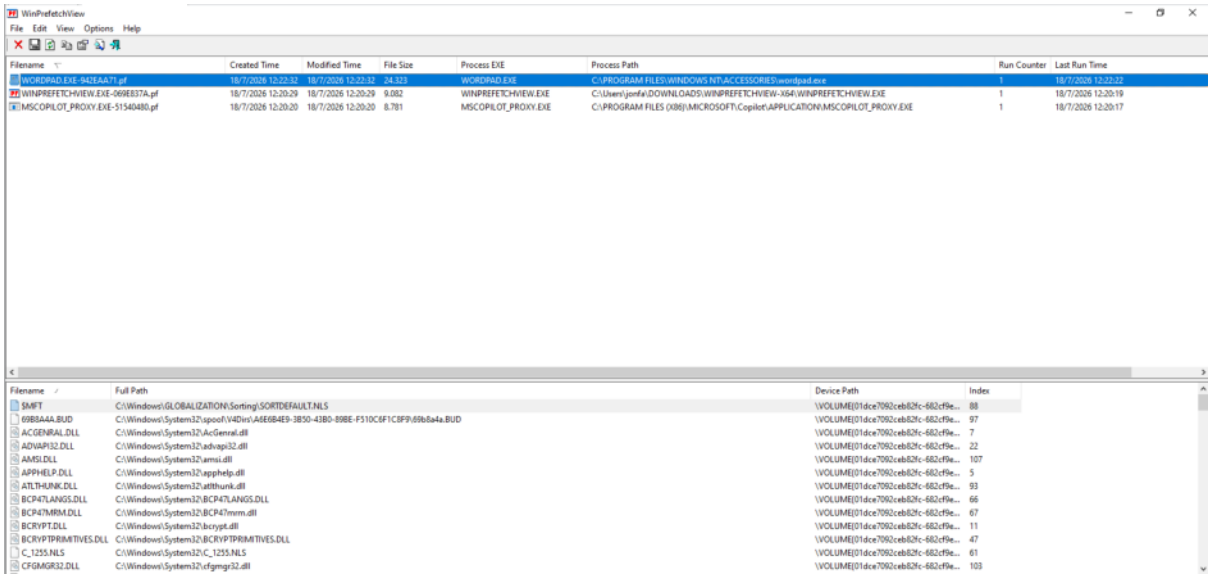
Nota. Captura de pantalla obtenida durante el laboratorio experimental, donde se evidencia la generación del archivo WORDPAD.EXE-942EAA71.pf dentro del directorio

C:\Windows\Prefetch, utilizado como artefacto de referencia para evaluar la recuperabilidad e integridad de la evidencia digital. Fuente: Elaboración propia.

Una vez confirmada la generación del archivo WORDPAD.EXE-942EAA71.pf, se realizó su análisis utilizando la herramienta WinPrefetchView, con el propósito de verificar la información forense almacenada en el artefacto antes de aplicar la técnica antiforense. Como se observa en la Figura X, el archivo Prefetch registra información relevante, como el nombre del proceso ejecutado, la ruta de la aplicación, el contador de ejecuciones, la fecha de la última ejecución y las bibliotecas dinámicas (DLL) cargadas durante su funcionamiento. Esta información constituyó la línea base para el análisis comparativo con el archivo recuperado en las etapas posteriores del laboratorio.

Figura 42.

Verificación del artefacto Prefetch de WordPad mediante WinPrefetchView

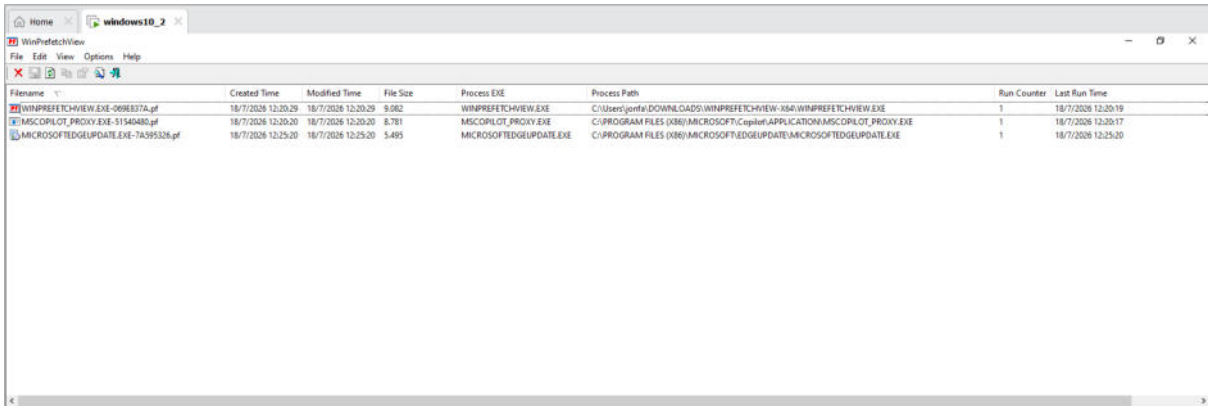


Nota. Captura de pantalla de la herramienta WinPrefetchView, donde se visualiza el archivo WORDPAD.EXE-942EAA71.pf y los principales metadatos forenses asociados a su ejecución, utilizados como referencia para el análisis comparativo de recuperabilidad e integridad de la evidencia digital. Fuente: Elaboración propia.

Una vez realizada la eliminación del archivo WORDPAD.EXE-942EAA71.pf, se efectuó una nueva verificación utilizando la herramienta WinPrefetchView con el propósito de confirmar que el artefacto ya no se encontraba disponible dentro del sistema operativo. Como se observa en la Figura X, el archivo Prefetch asociado a la ejecución de WordPad dejó de aparecer en el listado de artefactos, evidenciando que la técnica antiforense fue aplicada correctamente y que el registro de ejecución fue eliminado del directorio Prefetch. Esta verificación permitió establecer el estado del sistema antes de iniciar el proceso de recuperación del artefacto.

Figura 43.

Verificación de la eliminación del archivo Prefetch de WordPad



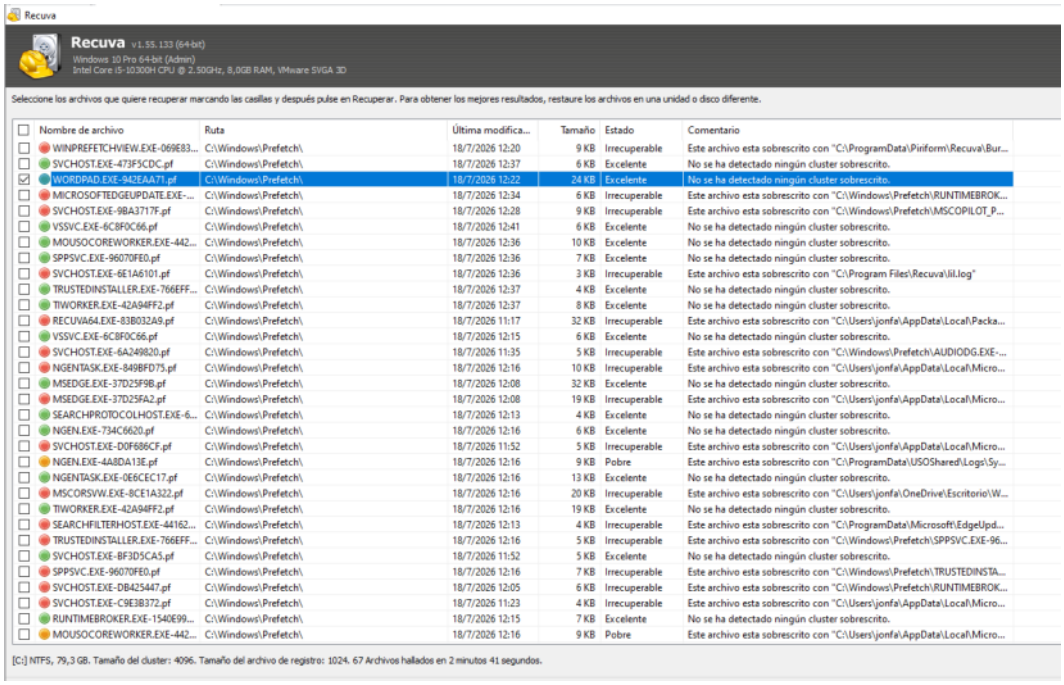
Nota. Captura de pantalla de WinPrefetchView posterior a la eliminación del archivo WORDPAD.EXE-942EAA71.pf, donde se confirma la ausencia del artefacto Prefetch

utilizado durante el laboratorio. Esta verificación permitió validar la correcta aplicación de la técnica antiforense antes de realizar la recuperación del archivo. Fuente: Elaboración propia.

Una vez aplicada la técnica antiforense, se inició el proceso de recuperación utilizando la herramienta Recuva. Se realizó un análisis del directorio donde originalmente se encontraba almacenado el archivo Prefetch, con el propósito de identificar artefactos eliminados que aún pudieran ser recuperados. Como se observa en la Figura X, la herramienta detectó el archivo WORDPAD.EXE-942EAA71.pf con estado "Excelente", indicando que el artefacto permanecía disponible para su recuperación y posterior análisis. Este resultado permitió continuar con la evaluación de la recuperabilidad del archivo y de la información forense contenida en él.

Figura 44.

Identificación del archivo Prefetch eliminado mediante la herramienta Recuva

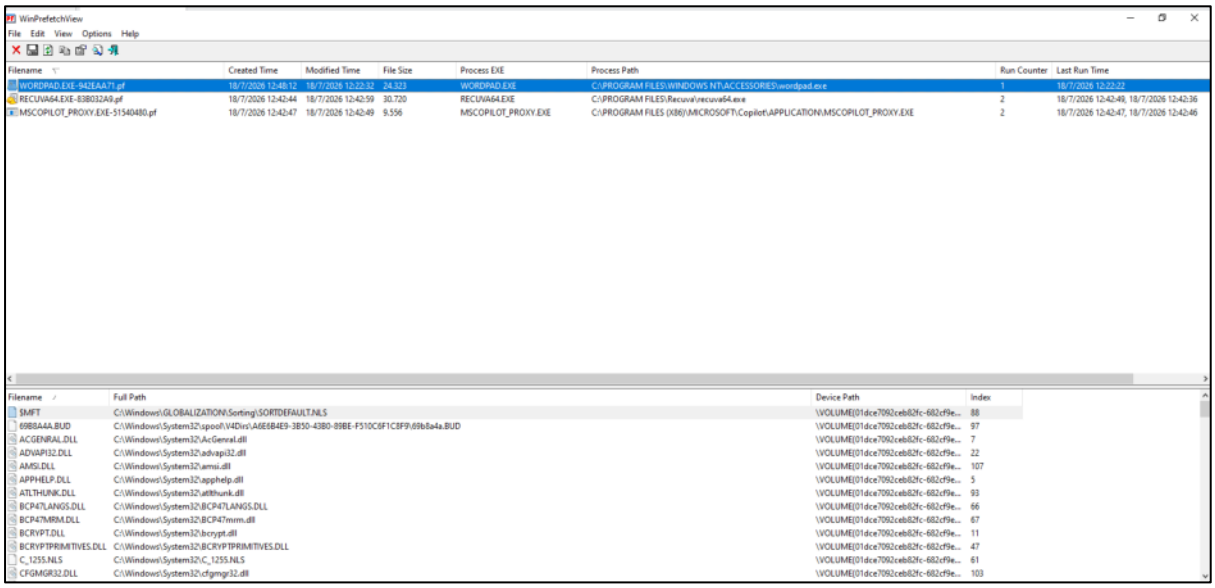


Nota. Captura de pantalla de la herramienta Recuva, donde se identifica el archivo WORDPAD.EXE-942EAA71.pf después de su eliminación. El estado "Excelente" indica que el artefacto no presentaba sobrescritura detectada y podía ser recuperado para su posterior análisis comparativo. Fuente: Elaboración propia.

Una vez recuperado el archivo WORDPAD.EXE-942EAA71.pf mediante la herramienta Recuva, se realizó una verificación utilizando WinPrefetchView con el propósito de comprobar que el artefacto podía ser interpretado correctamente y que la información forense permanecía accesible. Como se observa en la Figura X, la herramienta identificó nuevamente el archivo Prefetch, mostrando información como el nombre del proceso ejecutado, la ruta de la aplicación, el contador de ejecuciones y la fecha de la última ejecución. Esta verificación confirmó que el artefacto recuperado conservó información relevante para continuar con el análisis comparativo de su integridad respecto al archivo original.

Figura 45.

Verificación del archivo Prefetch recuperado mediante WinPrefetchView



Nota. Captura de pantalla de WinPrefetchView correspondiente al archivo WORDPAD.EXE-942EAA71.pf recuperado mediante Recuva. La herramienta reconoce el artefacto y presenta los principales metadatos asociados a su ejecución, permitiendo verificar la recuperación del archivo y continuar con la evaluación de la integridad de la evidencia digital. Fuente: Elaboración propia.

3.1.3.4. Artefactos afectados

La técnica antiforense de eliminación de archivos Prefetch impacta directamente a uno de los artefactos más utilizados durante las investigaciones forenses en sistemas operativos Windows: los archivos con extensión .pf. Estos archivos son generados automáticamente por el sistema y almacenan información relacionada con la ejecución de aplicaciones y procesos, permitiendo a los investigadores conocer qué programas fueron utilizados en un equipo y cuándo fueron ejecutados.

Al eliminar estos archivos, se pierde información valiosa para el análisis forense, como registros de ejecución de aplicaciones, fechas y horas de uso, frecuencia de ejecución

y otros datos que pueden contribuir a la reconstrucción de actividades realizadas por un usuario o por un posible atacante. Como consecuencia, se reduce la cantidad de evidencia disponible para determinar el comportamiento del sistema durante un período determinado.

Además, la ausencia de los archivos Prefetch dificulta la identificación de rastros de ejecución y la construcción de líneas temporales de eventos, ya que estos artefactos suelen ser una fuente importante para correlacionar actividades dentro del sistema operativo. Esto limita la capacidad del investigador para establecer con precisión qué aplicaciones fueron ejecutadas y en qué momento ocurrieron dichas acciones.

En la práctica se observó que la eliminación de los archivos Prefetch no implica la desaparición total de la evidencia digital. Otros artefactos del sistema operativo pueden conservar información relacionada con la actividad realizada, permitiendo recuperar indicios complementarios durante una investigación más profunda. Por esta razón, la eliminación de archivos Prefetch debe considerarse una técnica orientada a dificultar el análisis forense y reducir la evidencia disponible, mas no como un mecanismo capaz de eliminar completamente todos los rastros de actividad presentes en el sistema.

Tabla 9.

Artefactos forenses generados por los archivos Prefetch

Artefacto	**Descripción**
.pf	Archivos Prefetch generados por Windows
Execution traces	Rastros de ejecución de aplicaciones
Timestamps	Fechas y horas de ejecución
Historial de aplicaciones	Evidencia de programas ejecutados

Nota. La tabla presenta los principales artefactos forenses generados por los archivos Prefetch en sistemas operativos Windows.

3.1.3.5. Método de detección aplicado

Para detectar la aplicación de la técnica antiforense, se emplearon diferentes procedimientos de análisis forense orientados a la identificación de alteraciones en los artefactos Prefetch del sistema operativo Windows. Entre las actividades realizadas se incluyeron la verificación de la existencia de archivos **.pf**, la identificación de su ausencia posterior a la eliminación, la comparación entre las aplicaciones ejecutadas y los rastros de ejecución disponibles, así como la revisión de la información contenida en los artefactos mediante la herramienta **WinPrefetchView**.

Como parte del procedimiento de validación, también se verificó la existencia de posibles vacíos de evidencia ocasionados por la eliminación de los archivos Prefetch, los cuales constituyen un indicador de la aplicación de la técnica antiforense. Posteriormente, se empleó la herramienta Recuva para evaluar la posibilidad de recuperar los artefactos eliminados y determinar si la información forense permanecía disponible para su análisis.

Los resultados obtenidos permitieron confirmar que la ausencia de un archivo Prefetch dentro del sistema operativo constituye un indicio de alteración de la evidencia; sin embargo, también se comprobó que dicha ausencia no implica necesariamente la pérdida definitiva del artefacto. La recuperación y posterior verificación mediante WinPrefetchView demostraron que, en las condiciones del laboratorio, fue posible restaurar el archivo y acceder nuevamente a información relevante para el análisis forense.

3.1.4. Alternate data streams (ads)

3.1.4.1. Descripción de la Técnica

Los Alternate Data Streams (ADS) constituyen una característica nativa del sistema de archivos NTFS (New Technology File System) de Microsoft, originalmente diseñada para mantener compatibilidad con el sistema de archivos HFS (Hierarchical File System) de Macintosh al intercambiar metadatos de recursos. Técnicamente, NTFS organiza la información mediante registros de la Master File Table (MFT), donde cada archivo posee atributos $DATA. ADS permite asociar múltiples atributos $DATA a un único registro MFT, creando flujos de datos paralelos al flujo principal.

Desde la perspectiva antiforense, esta característica resulta especialmente relevante porque el Sistema Operativo no refleja el contenido ni el tamaño de los flujos alternos en las propiedades del archivo visibles al usuario final. Un archivo de 135 bytes puede contener internamente un flujo adicional de cualquier tamaño sin que el explorador de Windows, ni la mayoría de las herramientas de gestión de archivos convencionales, revelen dicha información, convirtiendo a ADS en un vector eficaz para el ocultamiento de datos o la persistencia silenciosa de código malicioso.

3.1.4.2. Herramientas Utilizadas

- Entorno de ejecución: Máquina Virtual Windows 11 (Oracle VirtualBox)
- Consola de administración: Windows PowerShell
- Suite de análisis forense: Autopsy Forensics Browser v4.23.1

3.1.4.3. Procedimiento Realizado

3.1.4.3.1. Fase de preparación y restricciones del sistema

Durante la fase inicial se detectó un control de integridad por parte de las ACLs (Access Control Lists) del sistema operativo. Al intentar manipular directamente el directorio C:\Windows\System32\, Windows PowerShell denegó la operación emitiendo una excepción UnauthorizedAccessException, lo que obligó a aislar el entorno experimental en una carpeta de trabajo dedicada dentro del perfil del usuario.

Este incidente fue resuelto parametrizando las rutas de forma absoluta y relativa estricta. A continuación, se detalla la secuencia de comandos ejecutados:

```
cd $HOME\Desktop
mkdir Laboratorio_Antiforense
cd Laboratorio_Antiforense
```

Posteriormente se generó el archivo portador notas_soporte.txt, inyectando texto que simula un recordatorio de mantenimiento. El tamaño físico inicial del archivo fue certificado en 135 bytes:

Out-File -FilePath .\notas_soporte.txt -InputObject "Recordatorio: El mantenimiento programado para el servidor de réplicas se ejecutará el fin de semana a las 23:00 de forma automática." -Encoding ascii Get-Item .\notas_soporte.txt | Select-Object Name, Length

3.1.4.3.2 Fase de Inyección e Impacto Antiforense

Se ejecutó la inyección del payload sensible a través del cmdlet Set-Content, vinculando un flujo de datos alterno nominado credenciales.txt con parámetros de conexión en texto plano:

```
Set-Content -Path .\notas_soporte.txt -Stream credenciales.txt
    -Value "DB_PROD_USER=sa;
    DB_PROD_PASS=ContrasenaMaestra2026!; IP=10.0.1.50"
```

La validación inmediata demostró el éxito de la técnica: al consultar nuevamente el atributo de tamaño Length, el sistema de archivos NTFS continuó reportando 135 bytes, invisibilizando para el usuario final los 60 bytes adicionales del flujo subyacente.

3.1.4.3.3 Fase de Detección en Vivo (Live Forensics)

La ocultación mediante ADS es ineficaz contra inspecciones directas basadas en llamadas al kernel del sistema de almacenamiento. Al invocar el parámetro wildcard -Stream * se revela la existencia del flujo alterno:

```
Get-Item .\notas_soporte.txt -Stream *
```

3.1.4.3.4 Fase de Análisis Forense Post-Mortem (Autopsy)

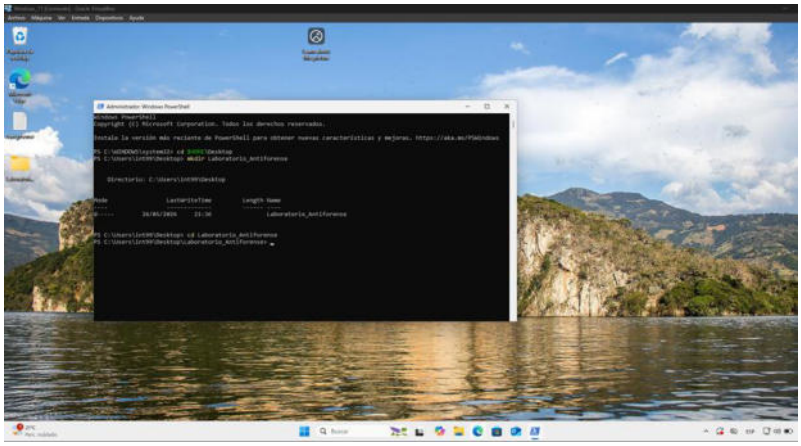
Se creó el caso pericial denominado Proyecto_ADS en Autopsy e importada la carpeta mediante la modalidad Logical Files. En una primera inspección sin módulos de ingesta, la herramienta leyó únicamente la dimensión lógica simple del archivo (135 bytes). La activación del módulo File Type Identification permitió indexar de forma automática la bifurcación de recursos, revelando el flujo oculto en la sección Data Artifacts.

3.1.4.4. Evidencia visual

A continuación, se presentan las capturas de pantalla documentadas durante la ejecución del laboratorio, organizadas según las fases del procedimiento.

Figura 46

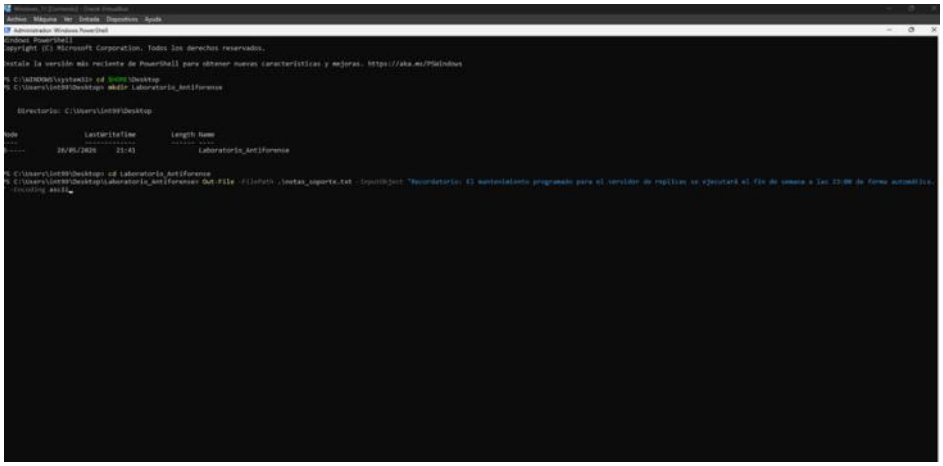
Creación del directorio de trabajo aislado.



Nota. Creación del directorio de trabajo aislado Laboratorio_Antiforense en la ruta de usuario mediante Windows PowerShell

Figura 47

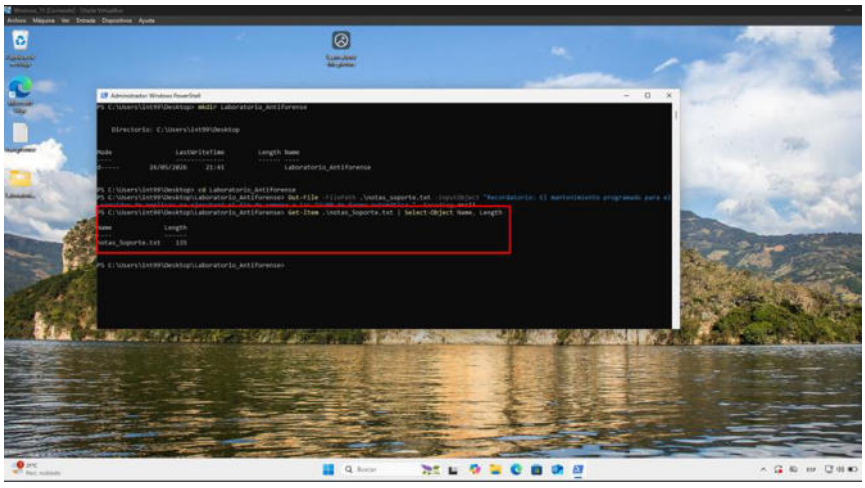
Ejecución del comando Out-File.



Nota. Ejecución del comando Out-File con codificación ASCII para generar el archivo portador base notas_soporte.txt.

Figura 48

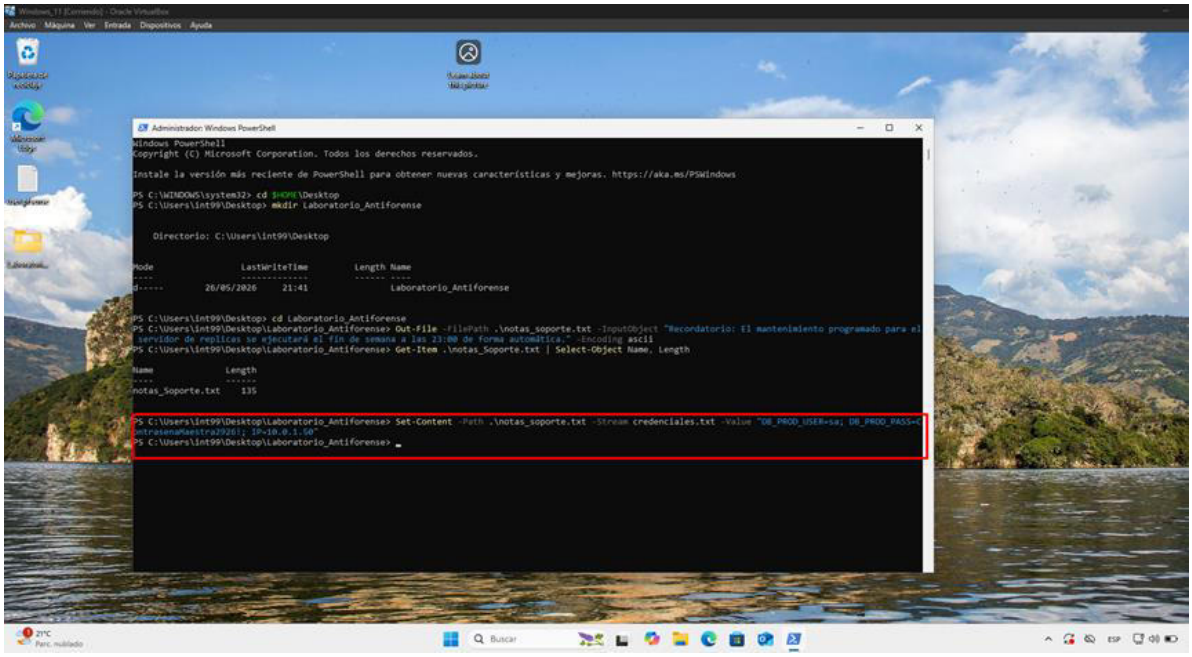
Auditoría inicial del archivo notas_soporte.txt.



Nota. Auditoría inicial del archivo notas_soporte.txt mediante el parámetro Length, certificando un tamaño base limpio de 135 bytes antes de la manipulación antiforense.

Figura 49

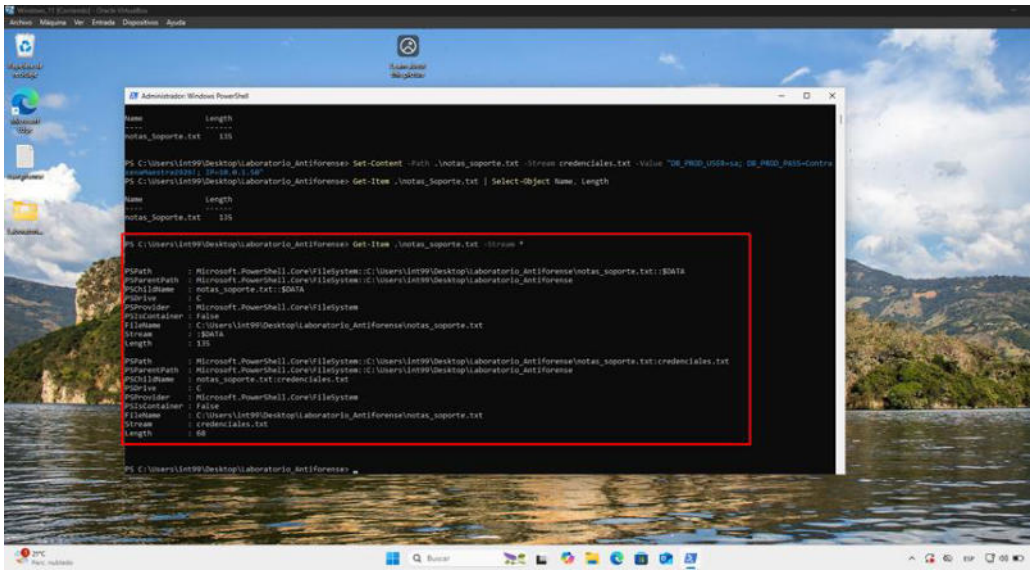
Inyección del payload confidencial.



Nota. Inyección del payload confidencial mediante el cmdlet Set-Content empleando el parámetro -Stream para crear el flujo de datos alternativo credenciales.txt.

Figura 50

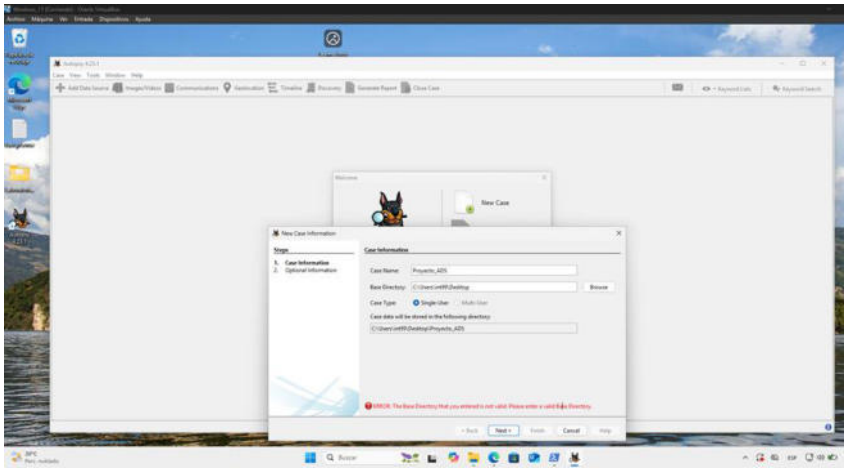
Validación de la persistencia.



Nota. Validación de la persistencia oculta mediante el parámetro -Stream *, exponiendo de manera diferenciada el descriptor original: $DATA y el flujo alternativo:credenciales.txt.

Figura 51

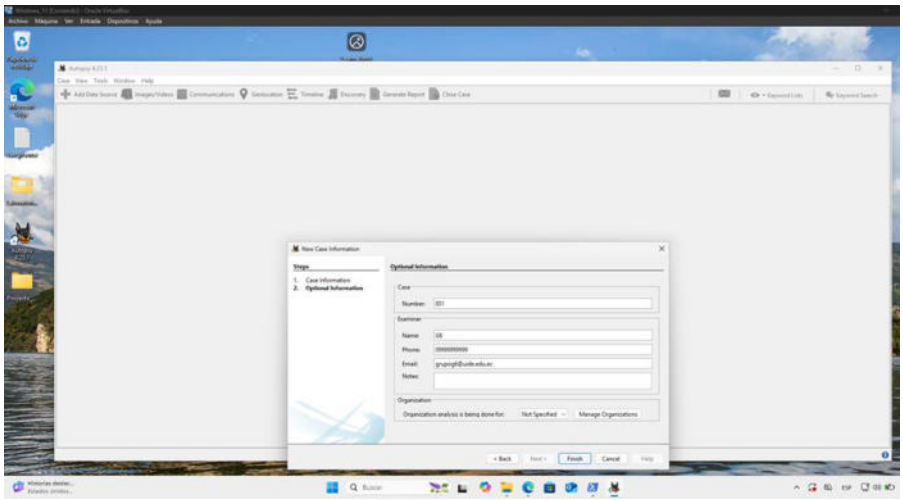
Inicialización del caso pericial en Autopsy.



Nota. Inicialización del caso pericial en Autopsy (v4.23.1) , registrando una alerta de validación en la ruta del directorio base debido a restricciones de permisos del sistema.

Figura 52

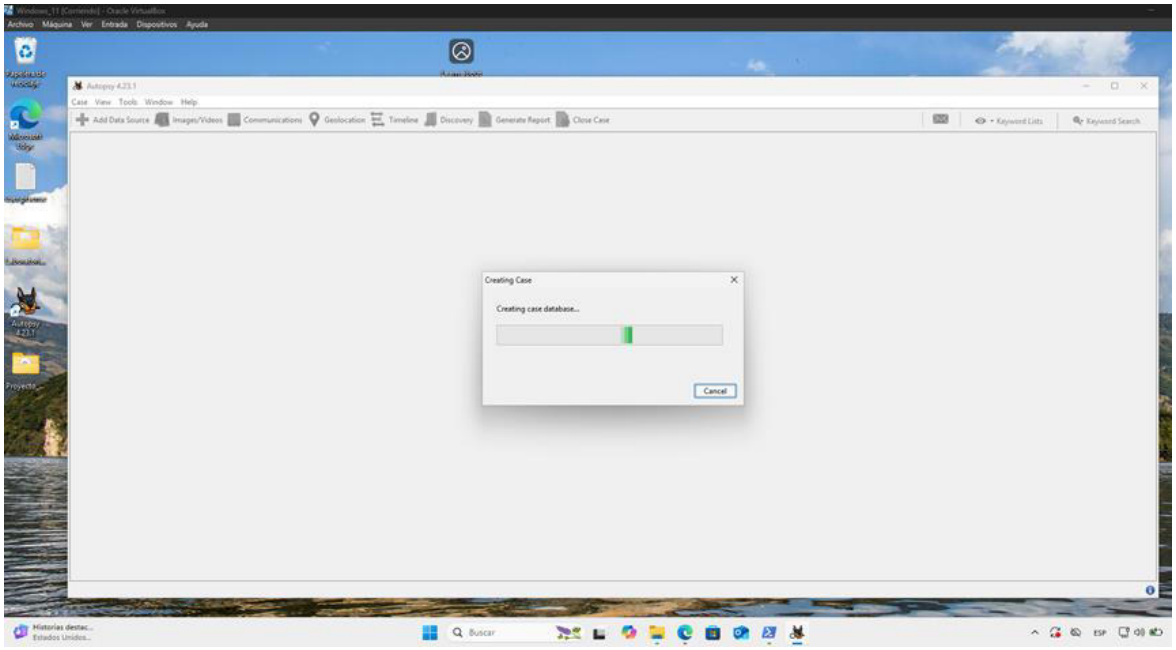
Configuración de información del caso.



Nota. Configuración de la información opcional del caso (Optional Information) en Autopsy, definiendo el identificador numérico de la investigación y los datos de contacto del examinador pericial.

Figura 53

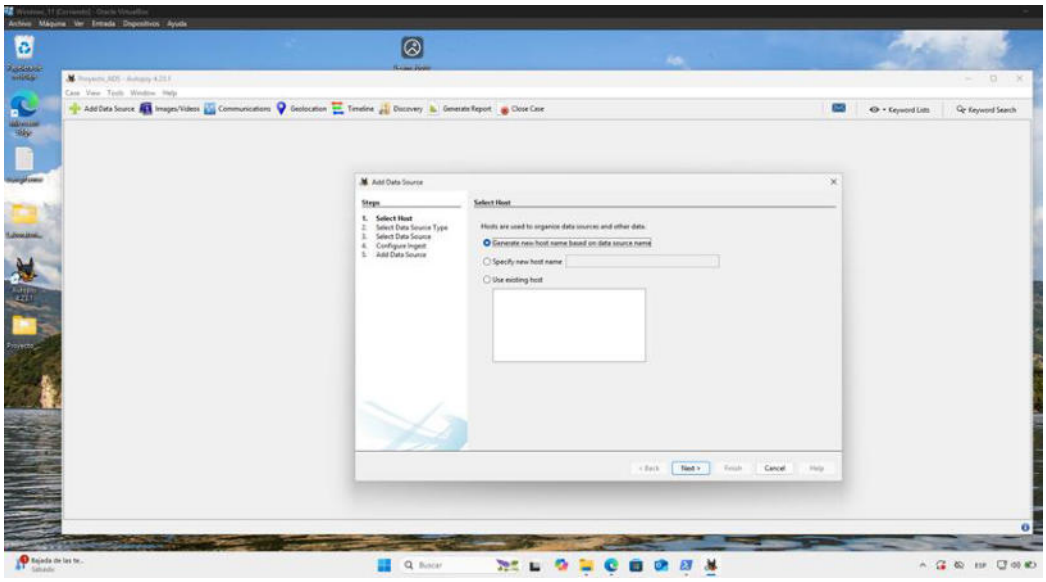
Generación automática de la base.

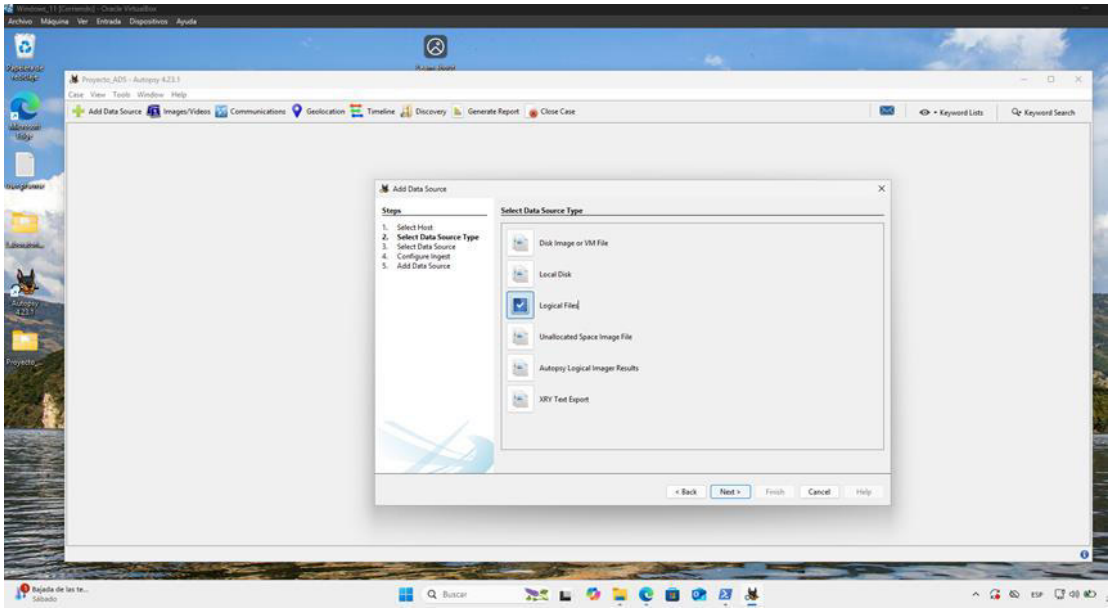


Nota. Generación automática de la base de datos local indexada por Autopsy para el almacenamiento centralizado de la evidencia del caso.

Figura 54

Configuración tipo de fuente de datos.

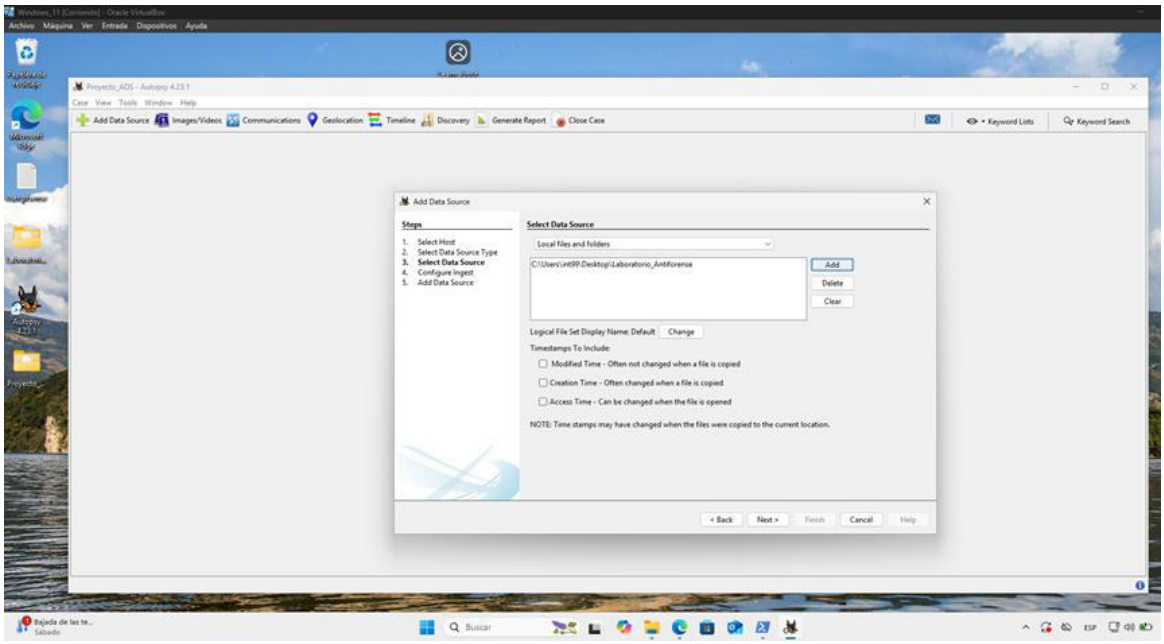




Nota. Selección del tipo de fuente de datos (Select Data Source Type) en Autopsy, optando por la modalidad de archivos lógicos (Logical Files) para el procesamiento directo de carpetas específicas.

Figura 55

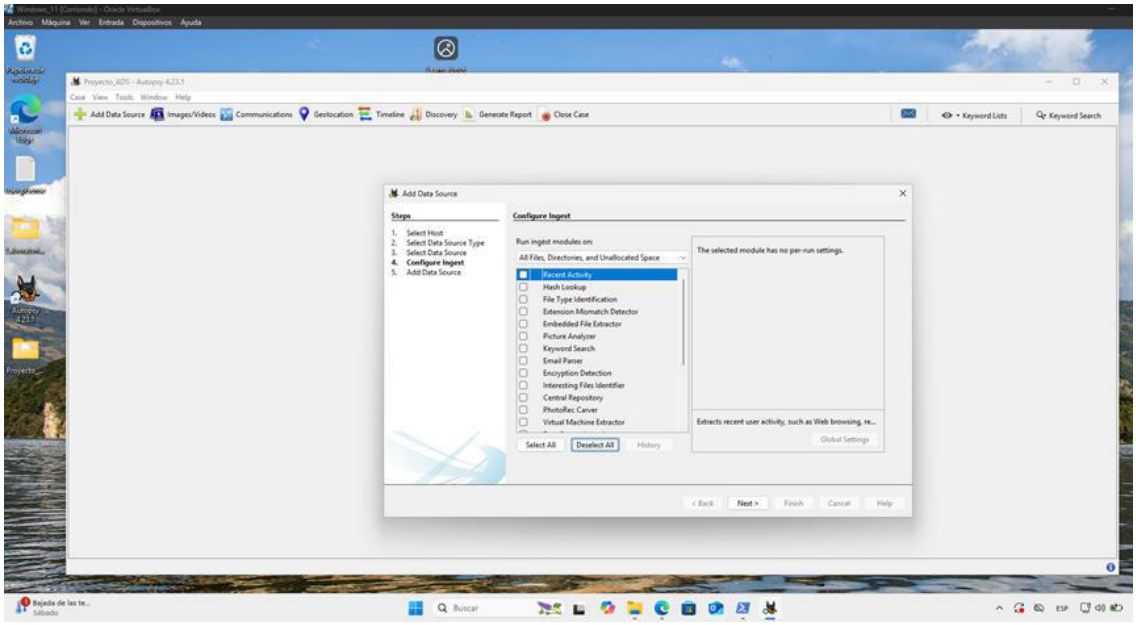
Ruta de origen de importación.



Nota. Especificación de la ruta origen del directorio Laboratorio_Antiforense en el panel Select Data Source para su importación lógica en el caso de estudio.

Figura 56

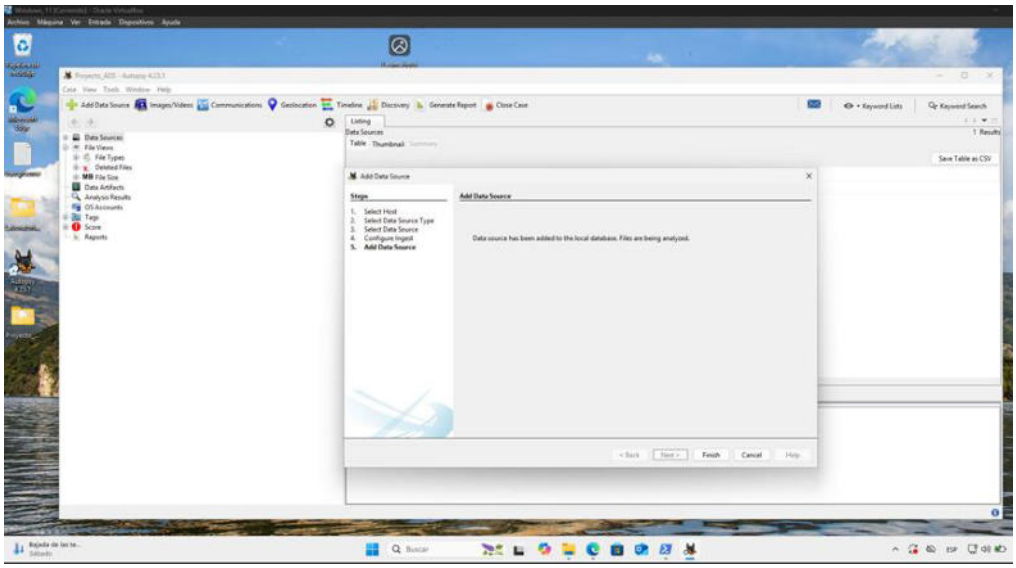
Configuración de los módulos de análisis.



Nota. Configuración de los módulos de análisis (ConFigura Ingest) en Autopsy, donde se optó por desmarcar las funciones automatizadas por defecto para proceder con una inspección estática manual y controlada de la estructura de archivos.

Figura 57

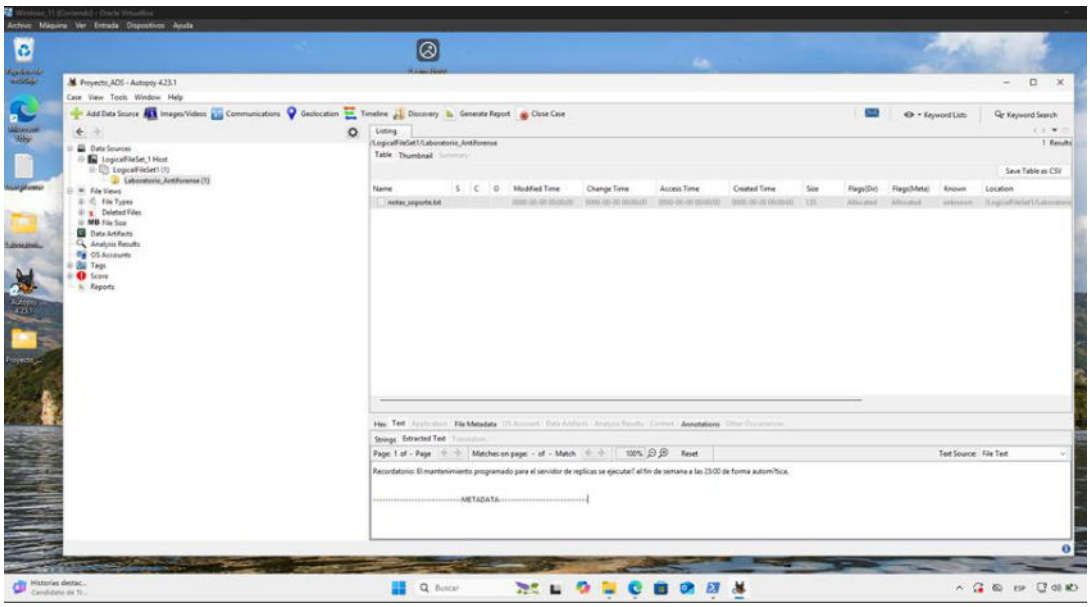
Confirmación de la adición en Autopsy.



Nota. Confirmación de la adición exitosa de la fuente de datos lógicos a la base de datos local de Autopsy antes de la apertura de la interfaz de visualización.

Figura 58

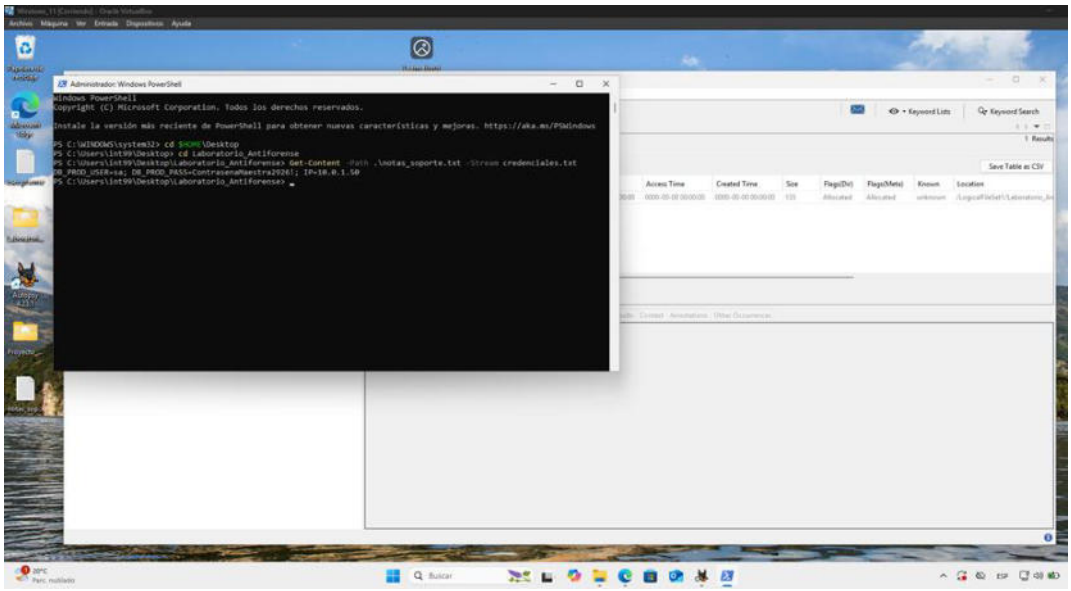
Interfaz inicial Autopsy.



Nota. Vista inicial de la interfaz de Autopsy mostrando la lectura del archivo lógico notas_soporte.txt, donde el visor de texto inferior (Text) expone únicamente el contenido visible de la fachada sin procesar los metadatos ocultos.

Figura 59

Extracción y lectura del payload.



Nota. Extracción y lectura del payload oculto en texto plano mediante el cmdlet Get-Content y el parámetro -Stream, exponiendo las credenciales de la base de datos de producción.

3.1.4.5. Artefactos Afectados

Los artefactos comprometidos durante la ejecución del laboratorio ADS se detallan a continuación:

Tabla 10.

Detalle de artefactos identificados

Artefacto	Tipo	Descripción	Tamaño Visible / Real
notas_soporte.txt	Archivo NTFS (flujo $DATA)	Archivo portador con texto de fachada	135 bytes / 135 bytes
notas_soporte.txt:credenciales.txt	Flujo ADS ($DATA alterno)	Payload con credenciales de BD en texto plano	0 bytes (oculto) / 60 bytes
Registro MFT	Estructura del sistema de archivos	Entrada MFT que contiene ambos flujos del archivo	N/A
Laboratorio_Antiforense\	Directorio NTFS	Contenedor del laboratorio experimental	Variable

3.1.4.6. Método de detección Aplicado

Se aplicaron dos métodos complementarios de detección forense:

3.1.4.6.1. Detección en Vivo mediante PowerShell

La detección directa se realizó mediante el parámetro -Stream * del cmdlet Get-Item, que consulta las estructuras $DATA a nivel de kernel, revelando todos los flujos asociados a un archivo independientemente de su visibilidad en el explorador:

```
Get-Item .\notas_soporte.txt -Stream *

Get-Content .\notas_soporte.txt -Stream credenciales.txt
```

3.1.4.6.2. Detección Post-Mortem mediante Autopsy

En el análisis forense post-mortem con Autopsy, la detección requirió la activación explícita del módulo de ingesta File Type Identification. Este módulo analiza las entradas de la MFT directamente, identificando y catalogando los flujos alternos en la sección Data Artifacts del caso pericial, generando artefactos auditables para la cadena de custodia.

3.1.5. Esteganografía

3.1.5.1. Descripción de la Técnica

La esteganografía (del griego στεγανός, "cubierto" y γράφειν, "escritura") es la disciplina que estudia las técnicas de comunicación encubierta mediante el ocultamiento de mensajes dentro de contenedores digitales aparentemente inocuos. A diferencia de la criptografía, que protege el contenido del mensaje haciendo imposible su lectura sin la clave correspondiente, la esteganografía oculta la propia existencia del mensaje.

La variante implementada en este laboratorio es la técnica de sustitución del Bit Menos Significativo (Least Significant Bit, LSB), aplicada sobre imágenes digitales JPEG. En esta técnica, los bits del mensaje secreto reemplazan a los bits de menor peso en los valores de los canales de color (R, G, B) de cada píxel. Dado que la alteración de un bit en el valor de un color produce una diferencia de intensidad imperceptible para el ojo humano (e.g., cambiar 254 a 255 en el canal verde), la imagen modificada resulta visualmente idéntica a la original. No obstante, a nivel matemático, la estructura de bits ha sido completamente alterada, lo que se manifiesta como una ruptura de la integridad criptográfica del archivo.

3.1.5.2. Herramientas Utilizadas

- Herramienta de ocultación: Steghide v0.5.1 (binario portátil para Windows, win32)
- Dependencias: cygwin1.dll, cygcrypt-4.dll (librerías de entorno Cygwin)
- Archivo contenedor: FOTO_CARNET_CV.jpeg (imagen de perfil corporativo en formato JPEG)
- Archivo de payload: exfiltracion.txt (token API simulado de producción)

- Verificación de integridad: Algoritmo SHA-256 nativo de PowerShell (Get-FileHash)

3.1.5.3. Procedimiento Realizado:

3.1.5.3.1. Fase de Preparación y Despliegue del Entorno

Se registró la descarga y descompresión de las librerías portables de Steghide junto con el archivo gráfico origen. Se estructuró una subcarpeta exclusiva Esteganografia dentro de la ruta principal del laboratorio, normalizando el espacio de nombres para aislar la ejecución y evitar conflictos de dependencias:

```
mkdir Esteganografia
cd Esteganografia
```

Se generó el payload confidencial denominado exfiltracion.txt, almacenando un token API simulado de producción que representa el tipo de información objetivo en un escenario real de exfiltración de datos corporativos.

3.1.5.3.2. Fase de Incrustación del Payload (Ocultamiento)

Mediante Steghide se ejecutó la incrustación del archivo exfiltracion.txt dentro de la imagen FOTO_CARNET_CV.jpeg, aplicando el algoritmo de sustitución LSB con cifrado AES-128 y una contraseña de protección:

```
steghide embed -cf FOTO_CARNET_CV.jpeg -sf exfiltracion.txt -p [password]
```

El proceso generó una imagen modificada visualmente idéntica a la original, pero con la estructura de bits internos alterada para contener el payload oculto.

3.1.5.3.3. Fase de Detección Forense — Análisis de Integridad (Hash SHA-256)

La detección pericial se realizó mediante el cálculo y comparación de firmas criptográficas SHA-256 de la imagen original y la imagen modificada. La diferencia entre ambos

hashes constituye evidencia irrefutable de manipulación del archivo a nivel de bits:

```
Get-FileHash .\FOTO_CARNET_CV_ORIGINAL.jpeg -Algorithm SHA256
Get-FileHash .\FOTO_CARNET_CV.jpeg -Algorithm SHA256
```

Los resultados obtenidos mostraron que la imagen original produce un hash SHA-256 iniciado con DA019C.…, mientras que la imagen modificada genera un hash completamente diferente iniciado con 8540A9..., confirmando la alteración matemática del archivo a pesar de la identidad visual.

3.1.5.3.4. Fase de Extracción Forense (Recuperación del Payload)

Para completar el ciclo del estegoanálisis, se realizó la extracción del archivo oculto utilizando Steghide con la contraseña conocida, simulando la perspectiva del investigador forense que ha identificado el archivo sospechoso y posee la clave de acceso:

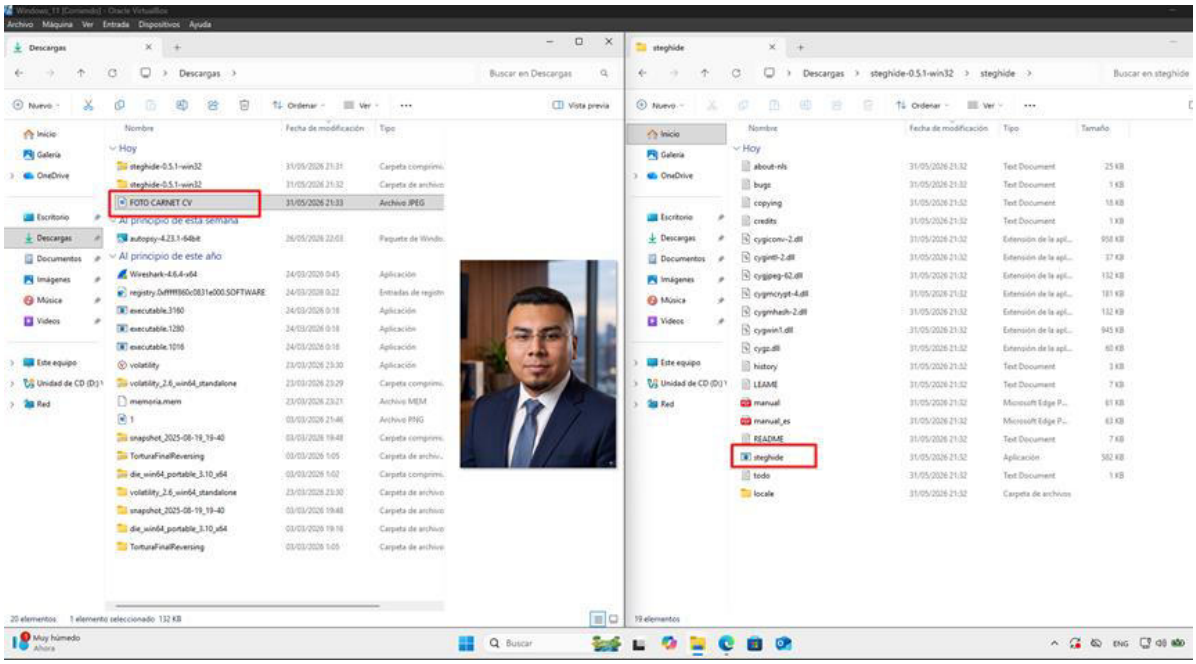
steghide extract -sf FOTO_CARNET_CV.jpeg -p [password] -xf payload_extraido.txt

3.1.5.4. Evidencia Visual

A continuación, se presentan las capturas de pantalla documentadas durante la ejecución del laboratorio de esteganografía.

Figura 60

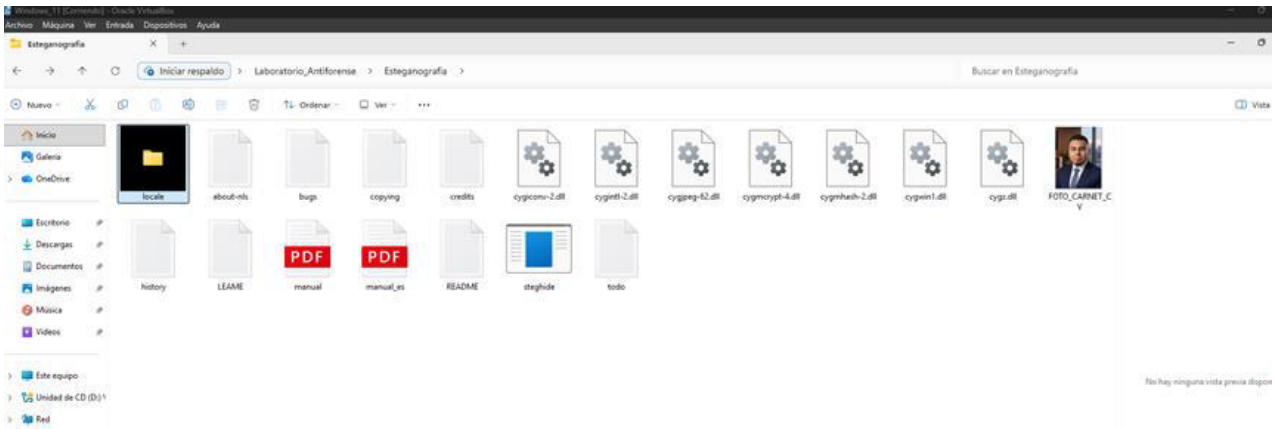
Preparación del entorno esteganografía.



Nota. Preparación del entorno para la segunda técnica, identificando el archivo portador gráfico original FOTO_CARNET_CV y el directorio de la herramienta esteganográfica portatil steghide.

Figura 61

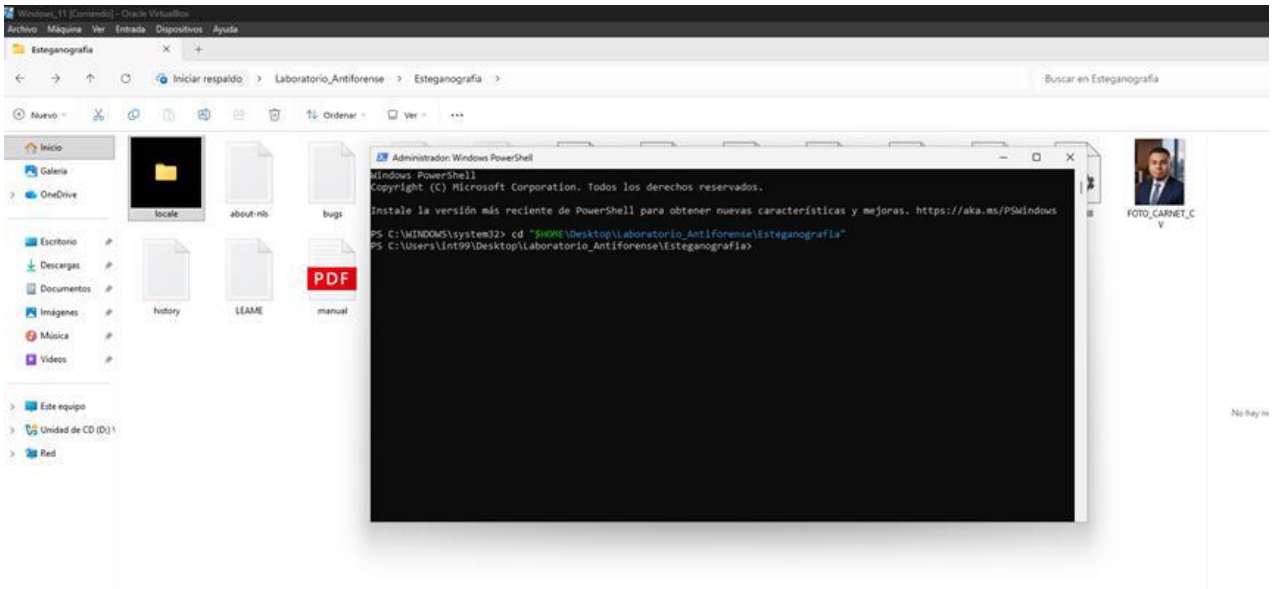
Estructuración y aislamiento.



Nota. Estructuración y aislamiento de los binarios de la suite steghide junto con el portador gráfico en la subcarpeta de destino Esteganografia.

Figura 62

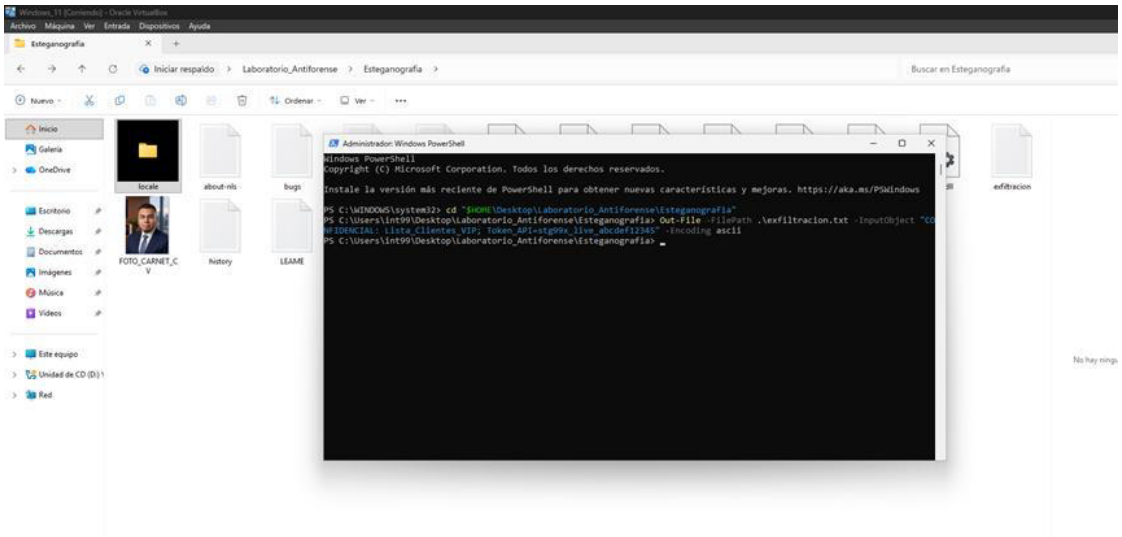
Ruta de la carpeta en PowerShell.



Nota. Apertura de Windows PowerShell en modo administrador y redireccionamiento de la consola hacia la ruta de la subcarpeta Esteganografia.

Figura 63

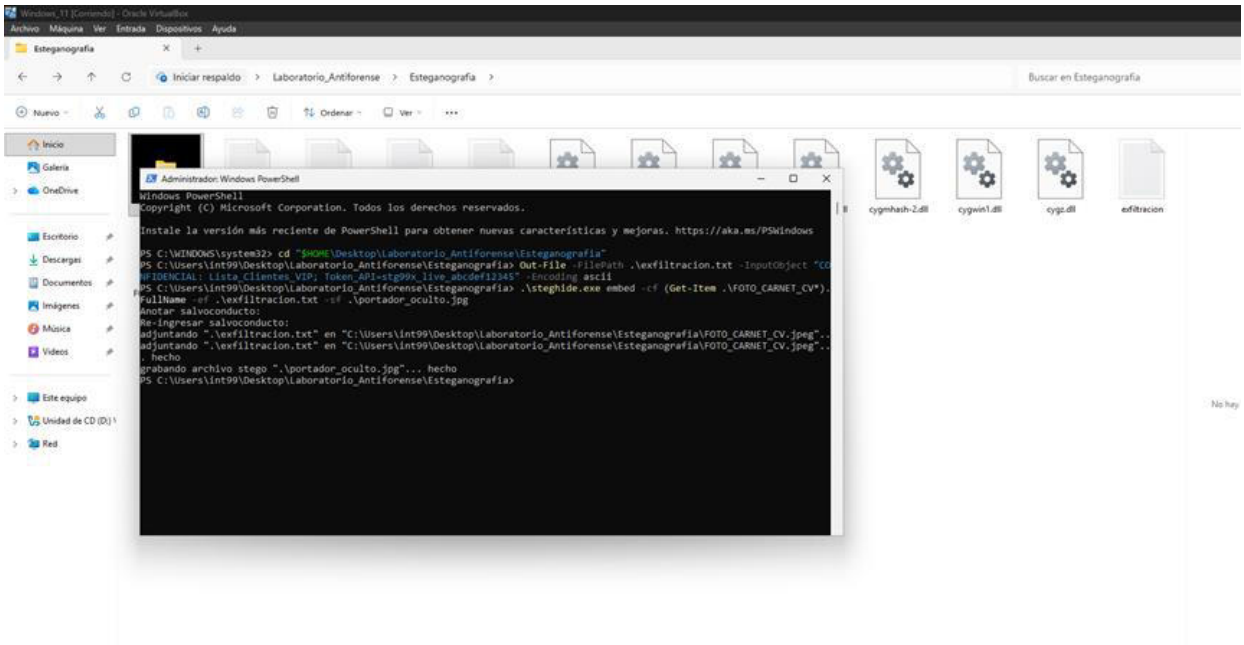
Creación del archivo de texto plano.



Nota. Creación del archivo de texto plano exfiltracion.txt con codificación ASCII para almacenar el payload confidencial (lista de clientes y token de API) antes del proceso de ocultamiento.

Figura 64

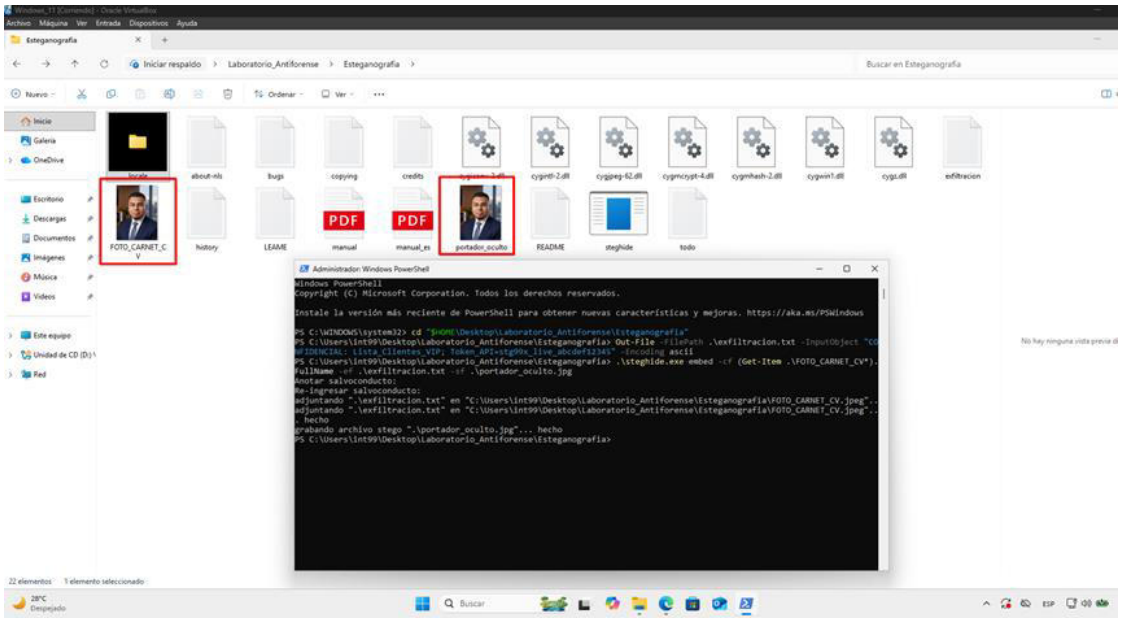
Inyección esteganográfica.



Nota. Ejecución e inyección esteganográfica exitosa mediante steghide, ocultando el archivo exfiltracion.txt dentro del portador gráfico original y grabando el resultado en portador_oculto.jpg.

Figura 65

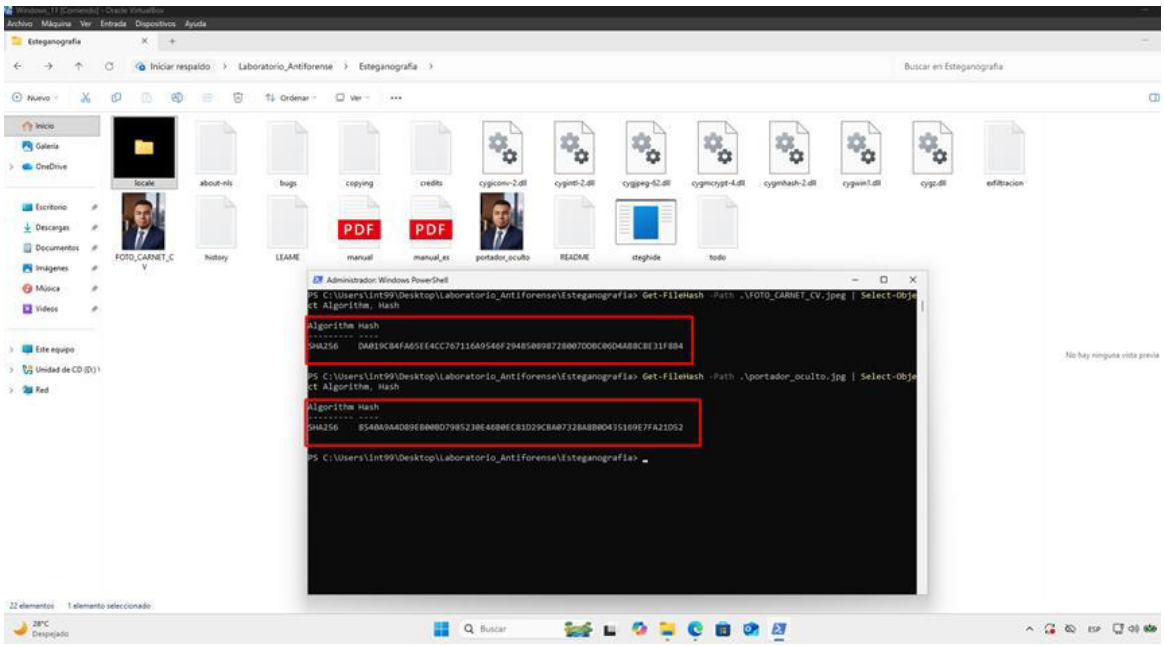
Análisis explorador de Windows.



Nota. Comparación visual en el Explorador de Windows entre la imagen original FOTO_CARNET_CV y el archivo generado portador_oculto, demostrando el principio de sigilo esteganográfico al no existir discrepancias macroscópicas detectables a simple vista.

Figura 66

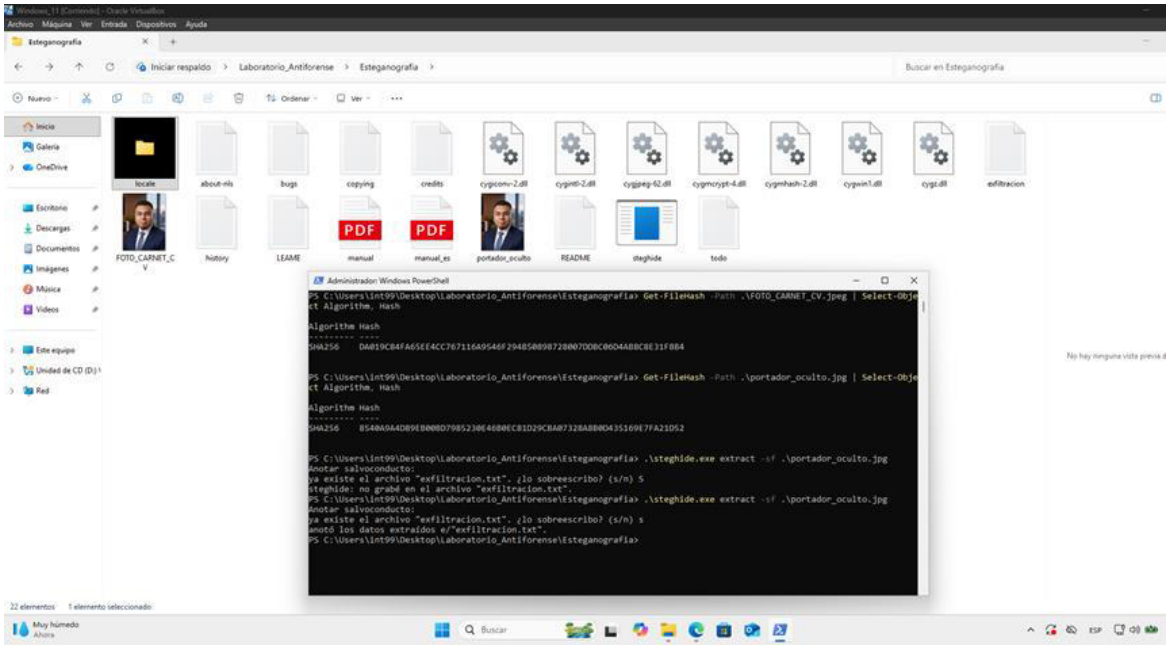
Verificación forense SHA-256.



Nota. Verificación forense matemática mediante el cálculo de firmas criptográficas con el cmdlet Get-FileHash, demostrando la ruptura absoluta de la integridad binaria (SHA-256) entre el portador original y la imagen con esteganografía.

Figura 67

Simulación de extracción forense.



Nota. Simulación de extracción forense mediante steghide con confirmación de sobreescritura de datos, recuperando con éxito el payload original (exfiltracion.txt) directamente desde los bits del portador modificado.

3.1.5.5. Artefactos Afectados

Los artefactos comprometidos durante el laboratorio de esteganografía se detallan a continuación:

Tabla 11.

Artefactos usados en esteganografía

Artefacto	Tipo	Estado	Observación Forense
FOTO_CARNET_CV.jpeg	Imagen JPEG portadora	Modificado (LSB)	Hash SHA-256 alterado

FOTO_CARNET_CV_ORIGINAL.jpeg	Imagen JPEG de referencia	Intacto	Hash SHA-256 original (DA019C...)
exfiltracion.txt	Payload (token API)	Oculto/Recuperado	Extraído íntegramente con Steghide
payload_extraido.txt	Artefacto forense	Recuperado	Evidencia de la extracción exitosa

3.1.5.6. Métodos de Detección Aplicados

Se aplicaron dos enfoques complementarios de detección forense para el análisis esteganográfico:

3.1.5.6.1. Análisis de Integridad Criptográfica (Hash SHA-256)

El método primario de detección consistió en el cálculo y comparación de firmas criptográficas SHA-256 mediante el cmdlet nativo Get-FileHash de PowerShell. Este enfoque permite detectar cualquier alteración a nivel de bits sin necesidad de conocer la contraseña ni el tipo de algoritmo esteganográfico utilizado. La discrepancia entre el hash de la imagen original y el de la imagen sospechosa constituye evidencia matemáticamente irrefutable de manipulación del archivo, aunque no revela el contenido oculto.

```
Get-FileHash -Path .\FOTO_CARNET_CV_ORIGINAL.jpeg -Algorithm SHA256
Get-FileHash -Path .\FOTO_CARNET_CV.jpeg -Algorithm SHA256
```

3.1.5.6.2. Extracción Dirigida mediante Steghide (Estegoanálisis con Sospecha)

El segundo enfoque simuló el estegoanálisis activo bajo sospecha fundamentada, donde el investigador dispone de la contraseña o la ha obtenido mediante otras técnicas (ingeniería social, análisis de memoria, etc.). La extracción exitosa mediante el parámetro extract de Steghide confirmó la presencia del payload y permitió su recuperación íntegra:

steghide extract -sf FOTO_CARNET_CV.jpeg -p [password] -xf resultado.txt

3.1.6. File Wiping

3.1.6.1. Descripción de la Técnica

El File Wiping es una técnica antiforense que consiste en la sobreescritura deliberada del contenido de archivos con datos aleatorios o patrones predefinidos antes de su eliminación, con el objetivo de impedir su recuperación mediante herramientas forenses digitales estándar.

A diferencia de una eliminación convencional donde el sistema operativo simplemente marca el espacio como disponible manteniendo los datos recuperables, el wiping destruye el contenido real del archivo, sus metadatos y sus marcas de tiempo, dificultando significativamente su reconstrucción.

En este laboratorio se evaluó inicialmente la efectividad del File Wiping mediante la herramienta Eraser, analizando la evidencia a través de FTK Imager, Autopsy y Recuva. Sin embargo, durante el proceso de validación se identificó que ninguna de estas herramientas logró recuperar archivos eliminados ni siquiera mediante métodos de eliminación convencional (sin aplicar wiping), lo cual generó la necesidad de verificar si dicho resultado correspondía a la efectividad real de las técnicas evaluadas o a una limitación del protocolo de detección empleado.

Con este fin, se incorporó la herramienta PhotoRec, la cual opera mediante reconocimiento de firmas de archivo (file carving) directamente sobre el disco físico, sin depender de la estructura del sistema de archivos ni de una adquisición forense previa. Se realizó primero una prueba de control consistente en la eliminación convencional de archivos

(sin wiping), confirmando que PhotoRec sí era capaz de recuperarlos en modo de análisis completo (Whole). Una vez validada la efectividad de esta herramienta, se procedió a aplicar el mismo procedimiento sobre los archivos sometidos a la técnica de File Wiping mediante Eraser y SDelete (Microsoft Sysinternals), con el objetivo de comparar de manera consistente los resultados obtenidos entre ambas herramientas de wiping.

3.1.6.2. Herramientas Utilizadas

Tabla 12.

Herramientas utilizadas durante el laboratorio experimental

Herramienta	Versión	Función dentro del laboratorio
Eraser	Última disponible	Aplicación del File Wiping sobre el archivo de prueba
PowerShell	Windows integrado	Generación de hashes MD5 y SHA1 pre y post wiping
FTK Imager	4.7.3.81	Adquisición forense — exportación lógica e imagen física
Autopsy	4.23.1	Plataforma de análisis forense digital
Recuva	4.23.1	Intento de recuperación con Deep Scan sector por sector
PhotoRec	7.2	Recuperación de archivos mediante carving

Nota. La tabla presenta las herramientas utilizadas durante el desarrollo experimental de la investigación. Fuente: Elaboración propia.

3.1.6.3. Procedimiento Realizado

3.1.6.3.1. File Wiping con Eraser

Se creó un archivo de prueba denominado prueba.txt en el escritorio del sistema Windows, incorporando contenido identificable para facilitar su rastreo posterior.

Antes de aplicar la técnica antiforense se calcularon los hashes MD5 y SHA1 para documentar la existencia e integridad del archivo original. Posteriormente, mediante la herramienta Eraser se ejecutó una tarea de borrado seguro sobre el archivo. Una vez completado el proceso, se verificó que el archivo no estuviera presente ni en el escritorio ni en la Papelera de Reciclaje. Adicionalmente, se comprobó la inexistencia de Shadow Copies mediante VSSAdmin. Finalmente, se realizaron adquisiciones lógica y física utilizando FTK Imager y se analizaron los resultados mediante Autopsy y Recuva para determinar la posibilidad de recuperación de la evidencia eliminada.

3.1.6.4. Evidencia Visual

3.1.6.4.1.1. Creación del archivo de prueba

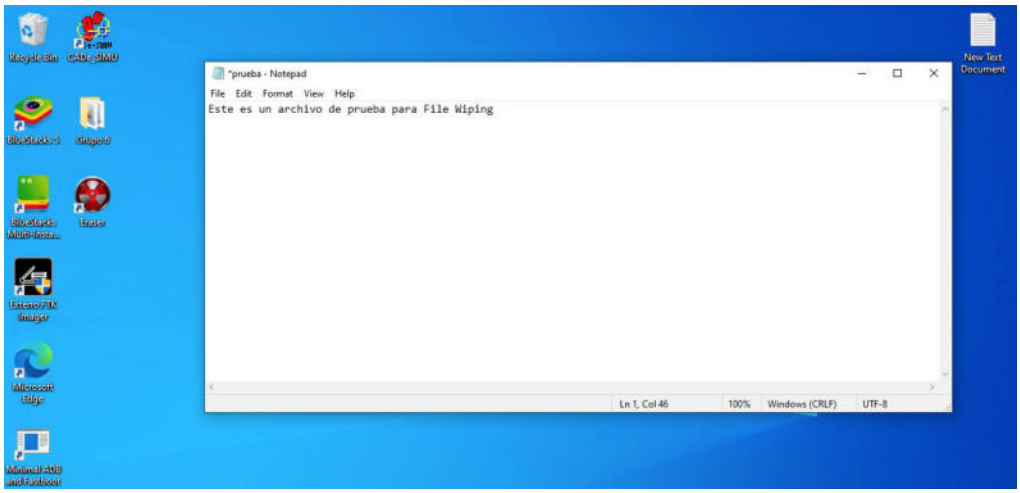
Se creó un archivo de texto plano llamado prueba.txt en el escritorio del sistema con contenido identificable para facilitar su búsqueda posterior durante el análisis forense:

Ruta: `C:\Users\User\Desktop\prueba.txt`

Contenido: texto identificable para análisis forense

Figura 68

Archivo prueba.txt.



Nota. La figura presenta la creación del archivo .txt para realizar puebas del File Wiping. Fuente: Elaboración propia.

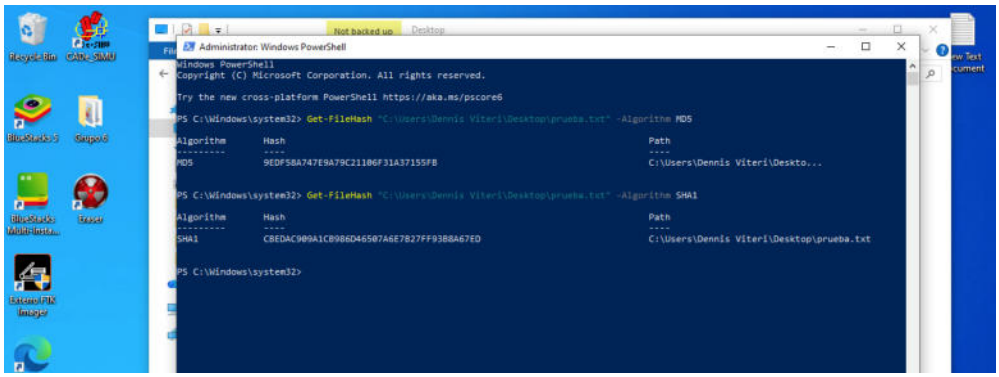
3.1.6.4.1.2. Obtención de hash antes del wiping

Antes de aplicar la técnica se generaron los hashes MD5 y SHA1 del archivo original mediante PowerShell para documentar su existencia e integridad:

```
Get-FileHash "C:\Users\User\Desktop\prueba.txt" -Algorithm MD5
Get-FileHash "C:\Users\User\Desktop\prueba.txt" -Algorithm SHA1
```

Figura 69

Hash del archivo prueba.txt.



Nota. La figura presenta la forma de implemenatar comandos en Powershell y adquirir el hash del archivo prueba.txt. Fuente: Elaboración propia.

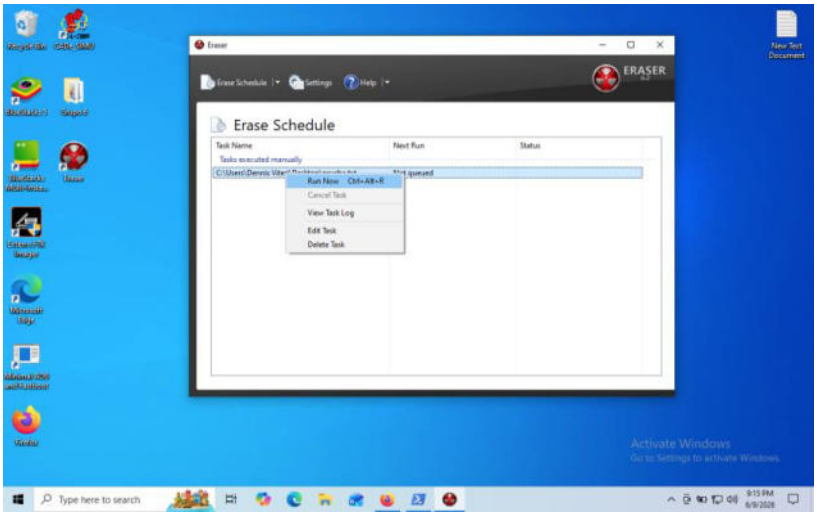
3.1.6.4.1.3. Aplicación del File Wiping con Eraser

Se ejecutó Eraser como Administrador y se configuró una nueva tarea de wiping:

```
Erase Schedule → New Task → Add Data →
File → prueba.txt → Run Now
```

Figura 70

Eraser.

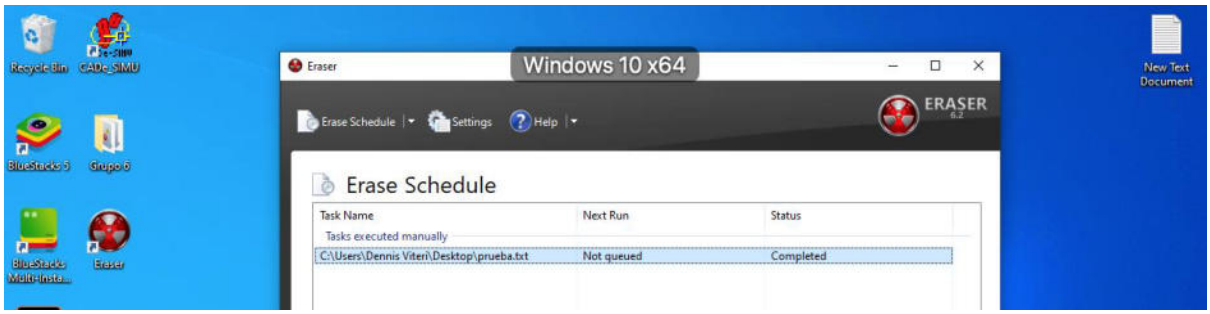


Nota. La figura presenta el entorno de la aplicacion de Eraser para poder eliminar el archivo prueba.txt de tal manera que no deje rastro en sistema. Fuente: Elaboración propia.

Eraser completó el proceso de sobreescritura y eliminación del archivo mostrando estado Completed.

Figura 71

Ejecución de Eraser.



Nota. La figura presenta la culminación del programa Eraser con status Completed Fuente: Elaboración propia.

3.1.6.4.1.4. Verificación de eliminación

Se verificó que el archivo prueba.txt no existía ni en el escritorio ni en la Papelera de Reciclaje, confirmando que el wiping eliminó completamente el archivo del sistema de archivos visible.

3.1.6.4.1.5. Verificación de Shadow Copies

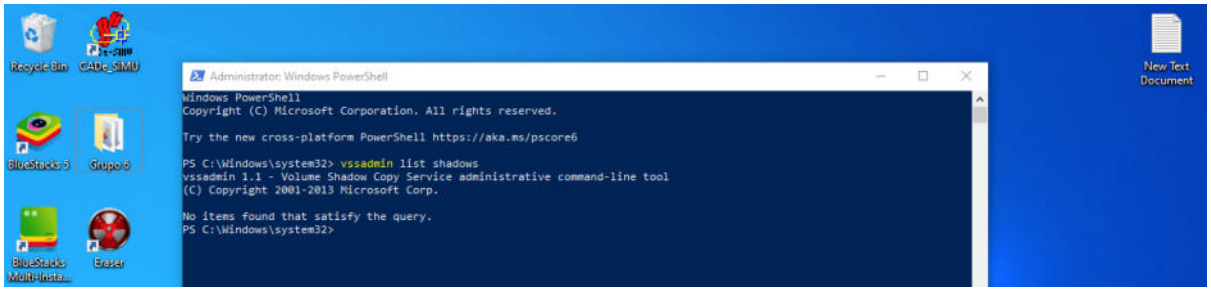
Se verificó la existencia de puntos de restauración mediante PowerShell:

```
vssadmin list shadows
```

El sistema no presentó Shadow Copies disponibles, descartando esta vía de recuperación.

Figura 72

Shadow Copies en Powershell.



Nota. La figura presenta como se realiza la ejecución del comando vssadmin list shadow.

Fuente: Elaboración propia.

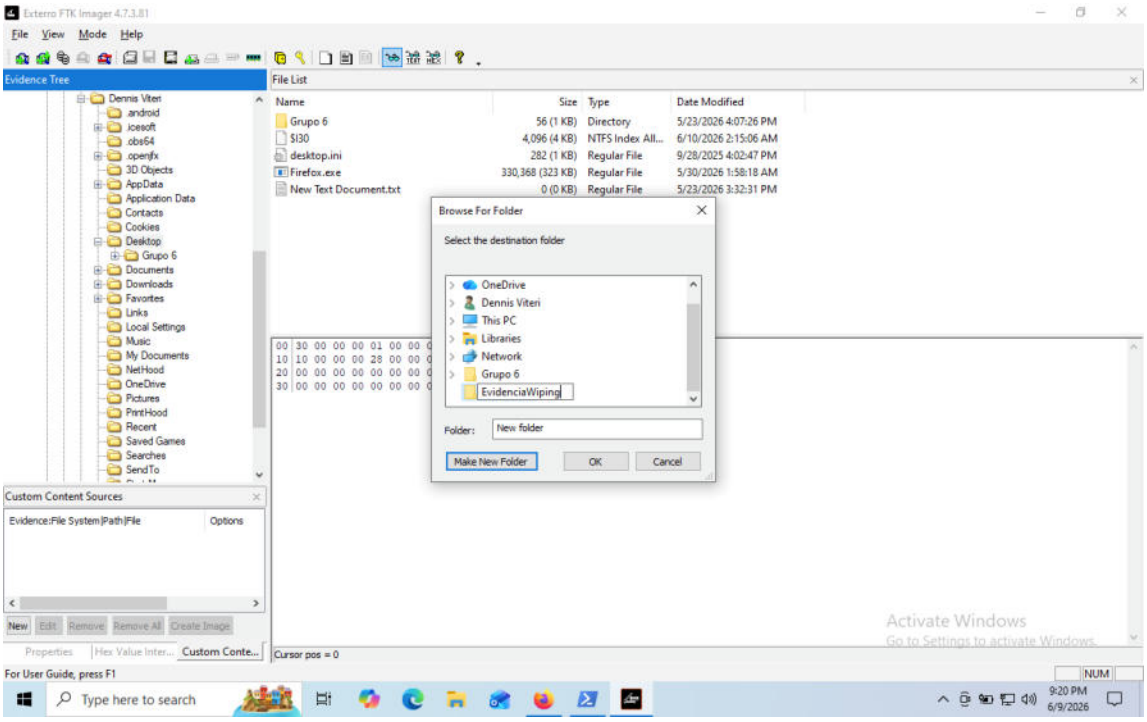
3.1.6.4.1.6. Adquisición lógica con FTK Imager

Se exportó el contenido de la carpeta del escritorio mediante adquisición lógica:

```
File → Add Evidence Item → Logical Drive → C: →
Desktop → Export Files → EvidenciaWiping
```

Figura 73

Exportación en FTK Imager.



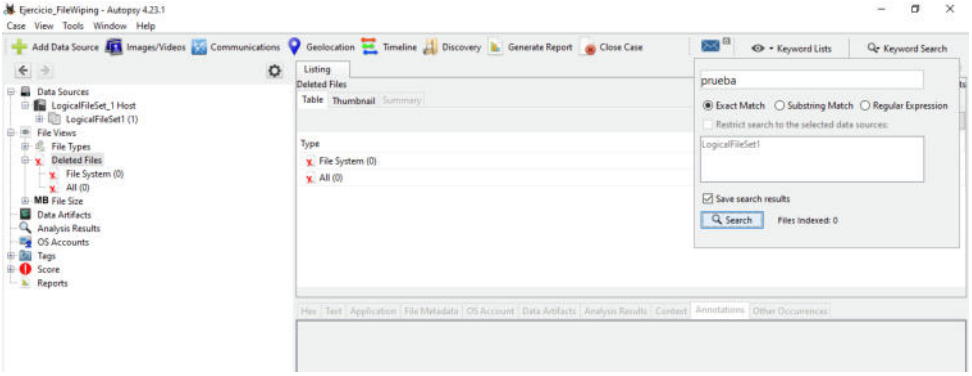
Nota. La figura presenta la forma de exportar el contenido del Escritorio donde se encontraba el archivo prueba.txt. Fuente: Elaboración propia.

3.1.6.4.1.7. Análisis lógico en Autopsy

Se creó el caso `Ejercicio_FileWiping` en Autopsy cargando los archivos exportados como Logical Files. Se realizaron búsquedas por keyword `prueba` y análisis de `Deleted Files` sin resultados.

Figura 74

Archivo exportado en Autopsy.



Nota. La figura presenta la aplicación de Autopsy para poder buscar desde el archivo exportado en FTK Imager . Fuente: Elaboración propia.

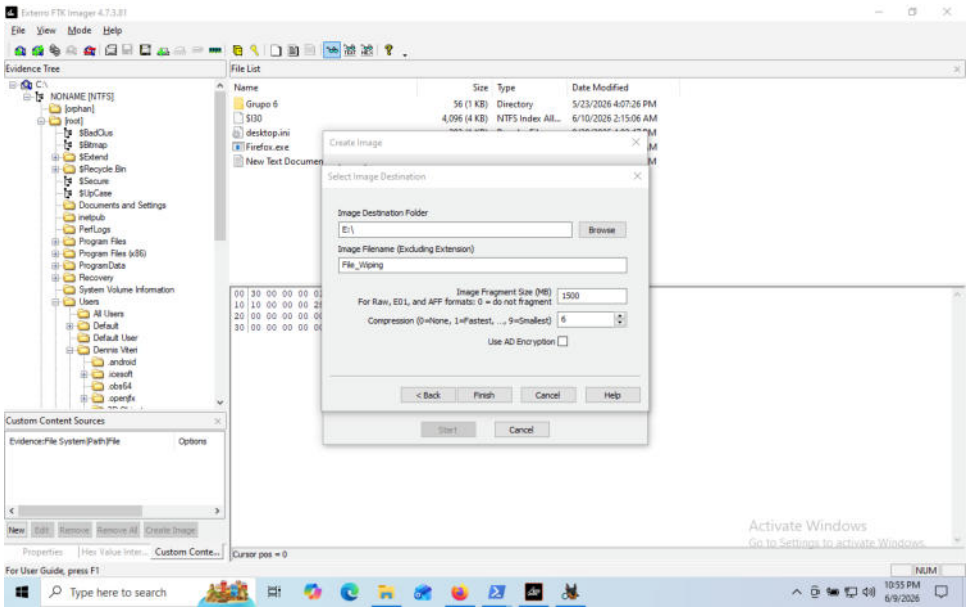
3.1.6.4.1.8. Adquisición física con FTK Imager

Se realizó imagen física completa del disco para un análisis más exhaustivo:

`File → Create Disk Image → Physical Drive → \\.\PHYSICALDRIVE0 → formato E01`

Figura 75

Imagen de disco en FTK Imager.



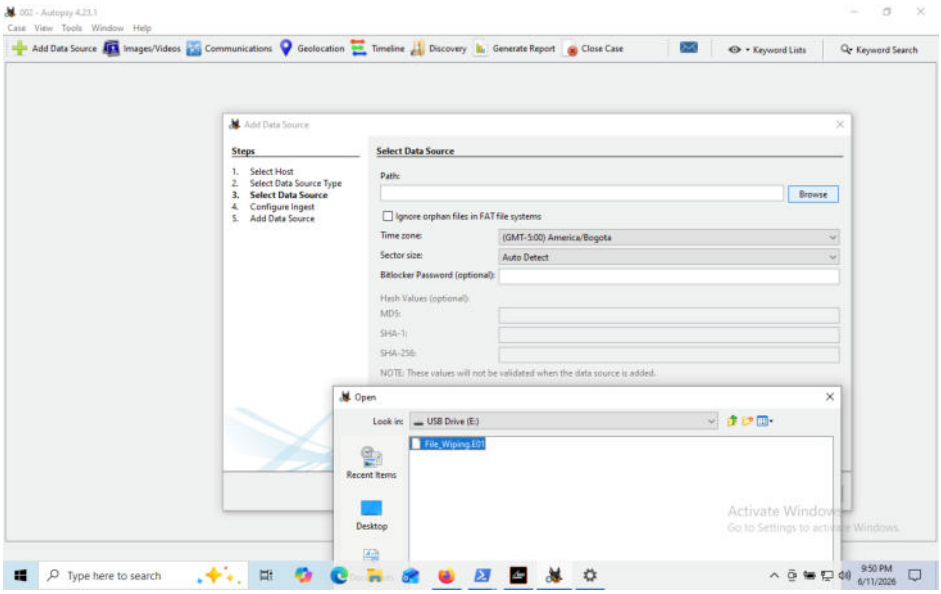
Nota. La figura presenta como la forma en realizar la exportación de una imagen de disco en este caso el disco C. Fuente: Elaboración propia.

3.1.6.4.1.9. Análisis físico en Autopsy

Se cargó la imagen física `.E01` en Autopsy como Disk Image activando todos los módulos disponibles incluyendo PhotoRec Carver y Deleted Files. Se realizaron búsquedas exhaustivas sin resultados para el archivo `prueba.txt`.

Figura 76

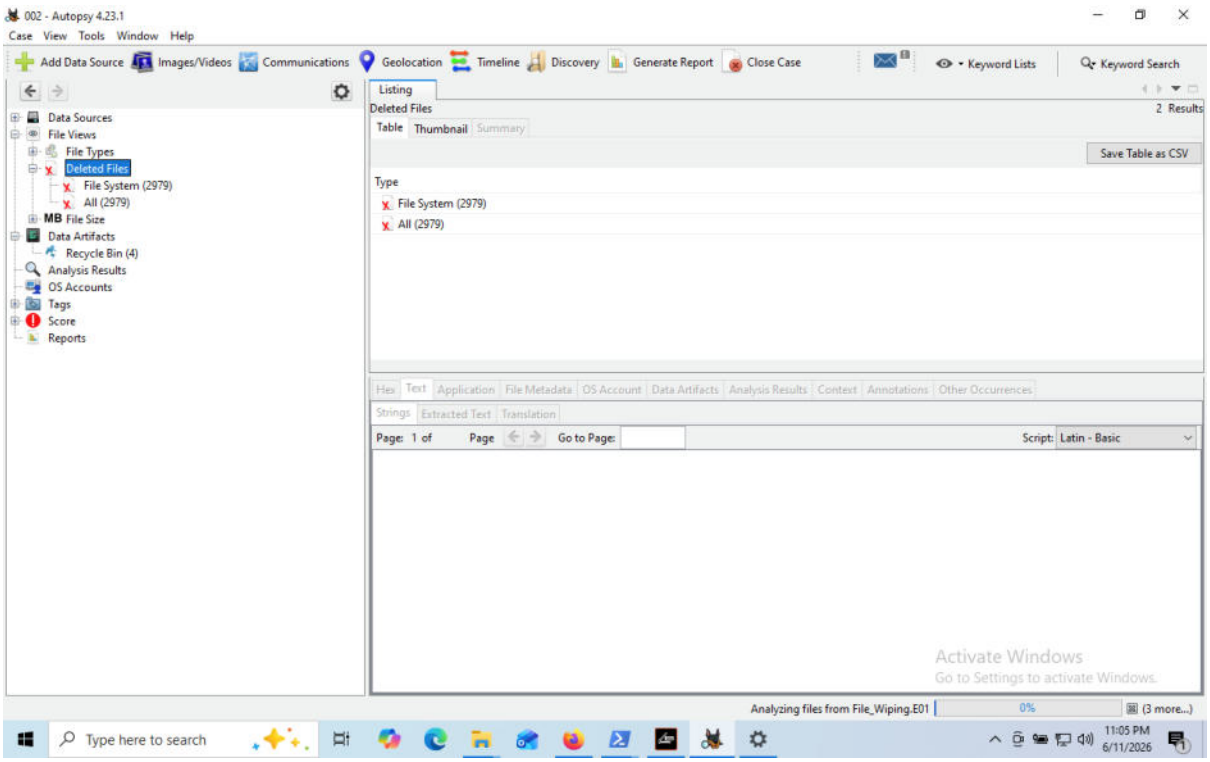
Autopsy con imagen de disco C.



Nota. La figura presenta como se usa la imagen de disco C para poder analizarlo de forma más detallada. Fuente: Elaboración propia.

Figura 77

Autopsy con imagen de disco local C.



Nota. La figura presenta la visualización de la imagen del disco C por lo que se tiene más información. Fuente: Elaboración propia.

3.1.6.4.1.10. Intento de recuperación con Recuva

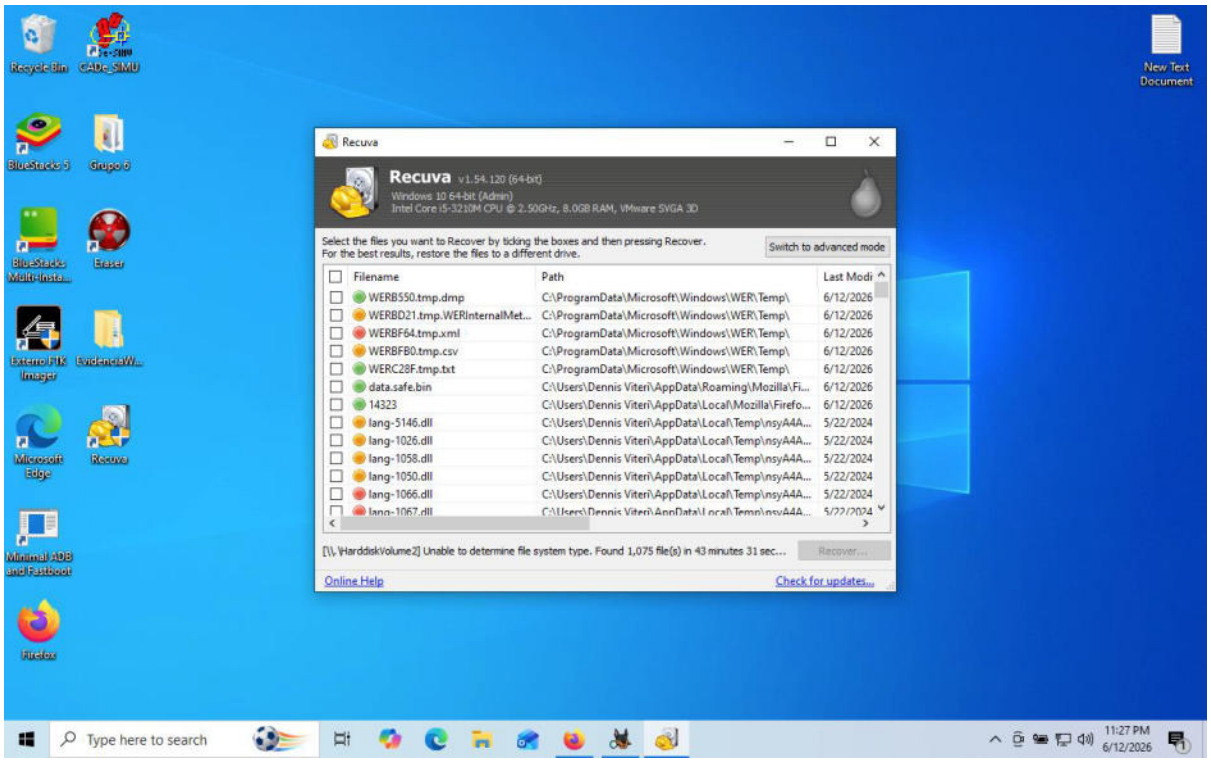
Se ejecutó Recuva como Administrador con Deep Scan activado sobre el directorio del escritorio:

```
All Files → Specific Location → Desktop →
Enable Deep Scan → Start
```

El escaneo sector por sector no encontró rastros recuperables del archivo wipeado.

Figura 78

Recuva



Nota. La figura presenta aplicación de Recuva una alternativa para analizar la imagen de disco C aparte de solo usar el Autopsy. Fuente: Elaboración propia.

Dado que ninguna de las herramientas y métodos aplicados en exportación lógica, imagen física y Recuva no se logró recuperar el archivo sometido a wiping, se planteó la necesidad de verificar si este resultado correspondía efectivamente a la efectividad de la técnica antiforense evaluada, o si, por el contrario, respondía a una limitación del protocolo de detección empleado en el entorno virtualizado utilizado para el laboratorio.

Con el fin de resolver esta incertidumbre, se diseñó una prueba de control basada en la eliminación convencional de archivos, sin la aplicación de ninguna técnica

antiforense, incorporando adicionalmente la herramienta PhotoRec como método de recuperación complementario.

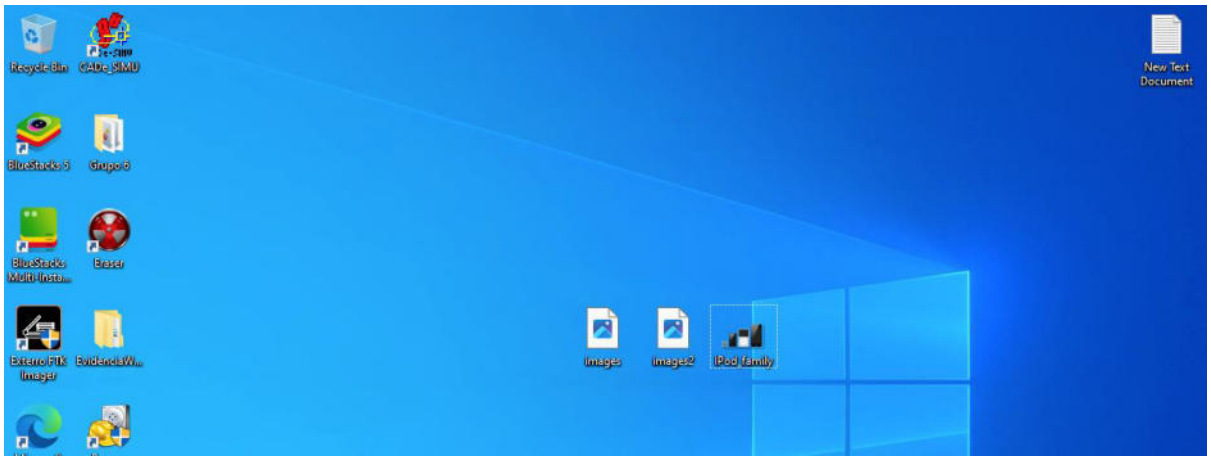
3.1.6.3.2 Prueba de control — Eliminación convencional con PhotoRec

3.1.6.3.2.1 Eliminación convencional de archivos

Se eliminaron mediante la combinación `Shift+Delete` 3 imágenes identificables ya que se descargó vía internet 2 imágenes formato .jpg y una imágen en formato .png, omitiendo deliberadamente el paso por la Papelera de Reciclaje.

Figura 79

Imágenes formato JPG y PNG.



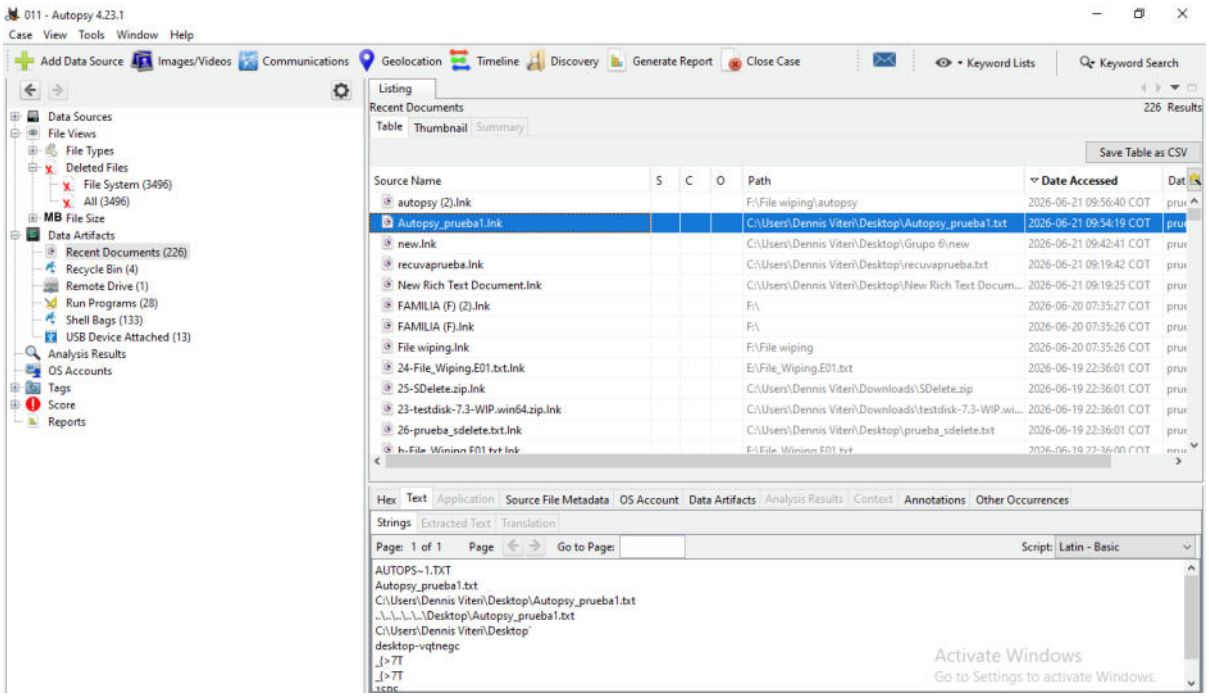
Nota. La figura presenta 3 imágenes en formato jpg y png para poder realizar pruebas de eliminación de forma tradicional Fuente: Elaboración propia.

3.1.6.3.2.2 Intento de recuperación con FTK Imager y Autopsy

Se exportó nuevamente el contenido del escritorio mediante FTK Imager y se analizó en Autopsy de forma lógia y física. No fue posible recuperar los archivos eliminados de forma convencional a pesar de que se realizó tanto lógica como física.

Figura 80

Análisis en Autopsy.



Nota. La figura presenta análisis del archivo exportado en FTK Imager en Autopsy

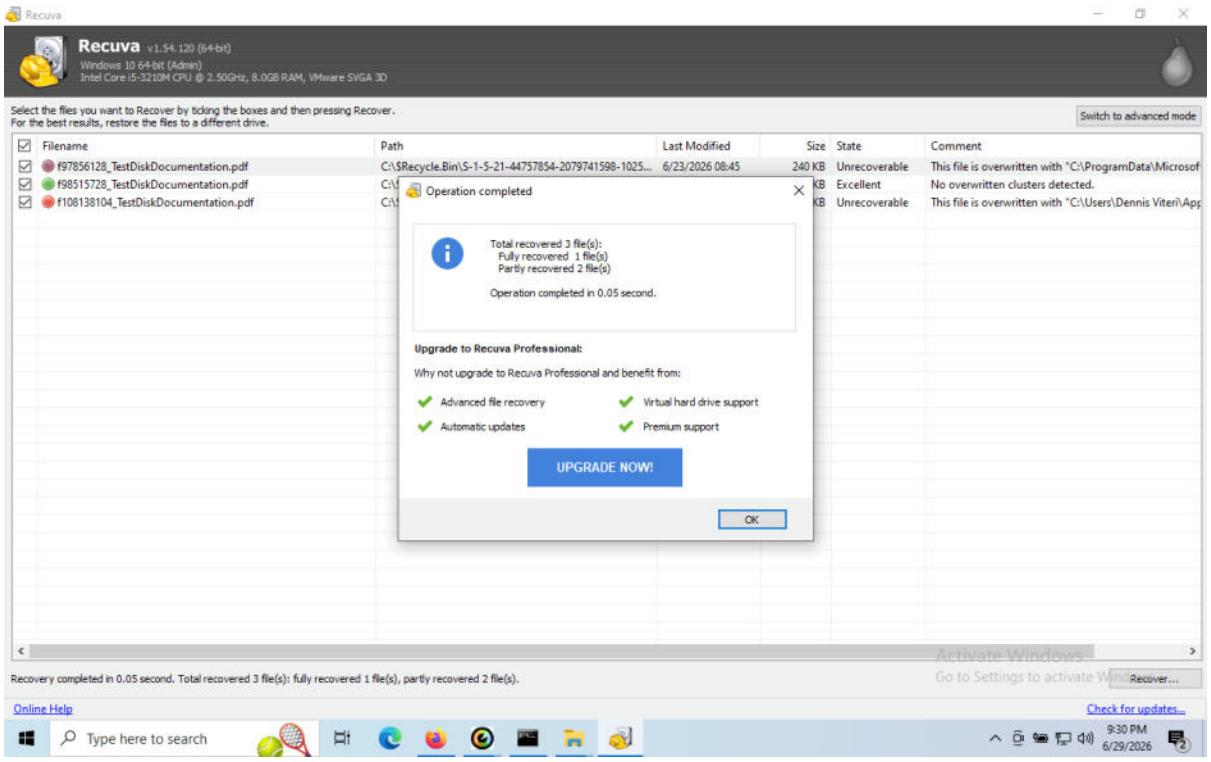
Fuente: Elaboración propia.

3.1.6.3.2.3 Intento de recuperación con Recuva

Se ejecutó Recuva con Deep Scan sobre el mismo directorio. El escaneo no logró recuperar ninguno de los archivos eliminados, reportando archivos encontrados, recupero archivos en formato pdf lo cual no eran los archivos eliminados.

Figura 81

Análisis en Recuva.



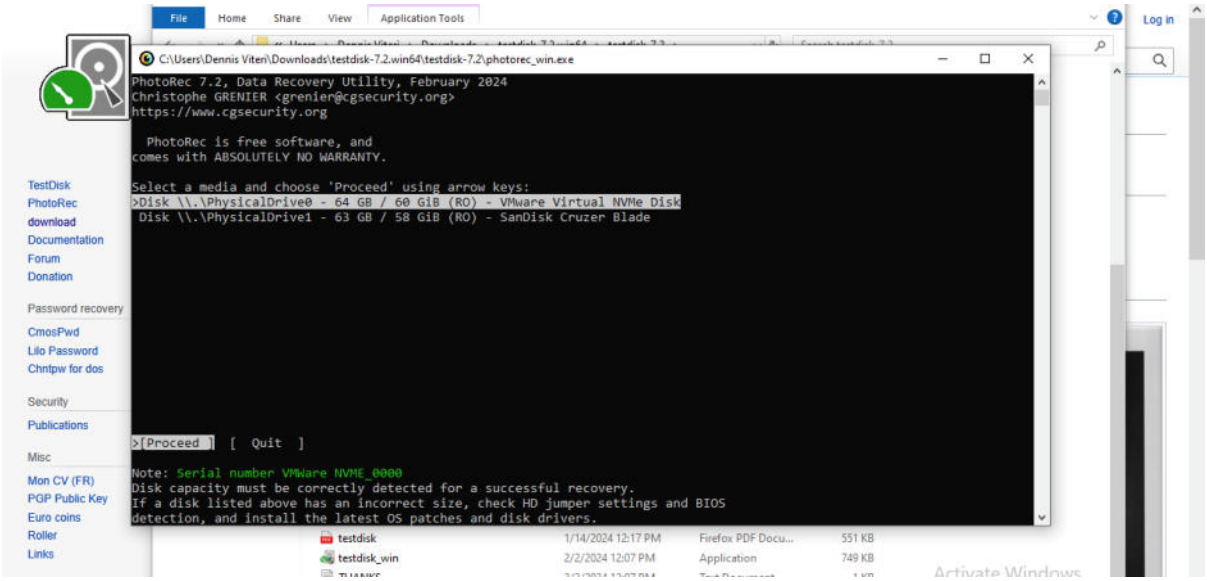
Nota. La figura presenta análisis de la papelera de reciclaje mediante el programa de Recuva. Fuente: Elaboración propia.

3.1.6.3.14 Ejecución de PhotoRec en modo Free

Se instaló la herramienta PhotoRec y se ejecutó sobre la partición de datos del disco `(\\.\PhysicalDrive0)`, seleccionando el modo de búsqueda Free, limitado al espacio marcado como disponible por el sistema de archivos. Este modo recupero archivos, pero no permitió recuperar los archivos eliminados ya que fue limitado.

Figura 82

PhotoRec.

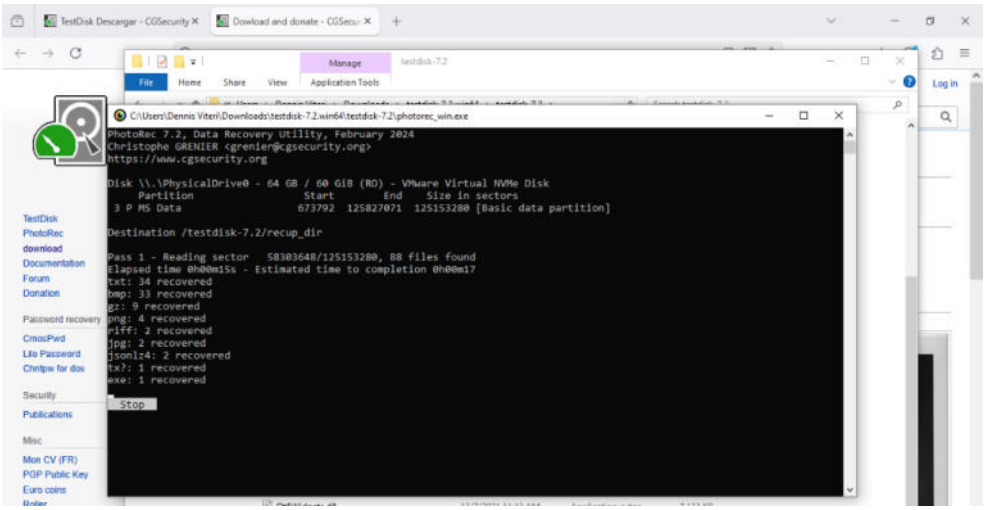


Nota. La figura presenta análisis del disco local C el cual se lo realizó con PhotoRec

Fuente: Elaboración propia.

Figura 83

Disco local C en PhotoRec.



Nota. La figura presenta el proceso de recuèración de los archivos con la aplicación de

PhotoRec Fuente: Elaboración propia.

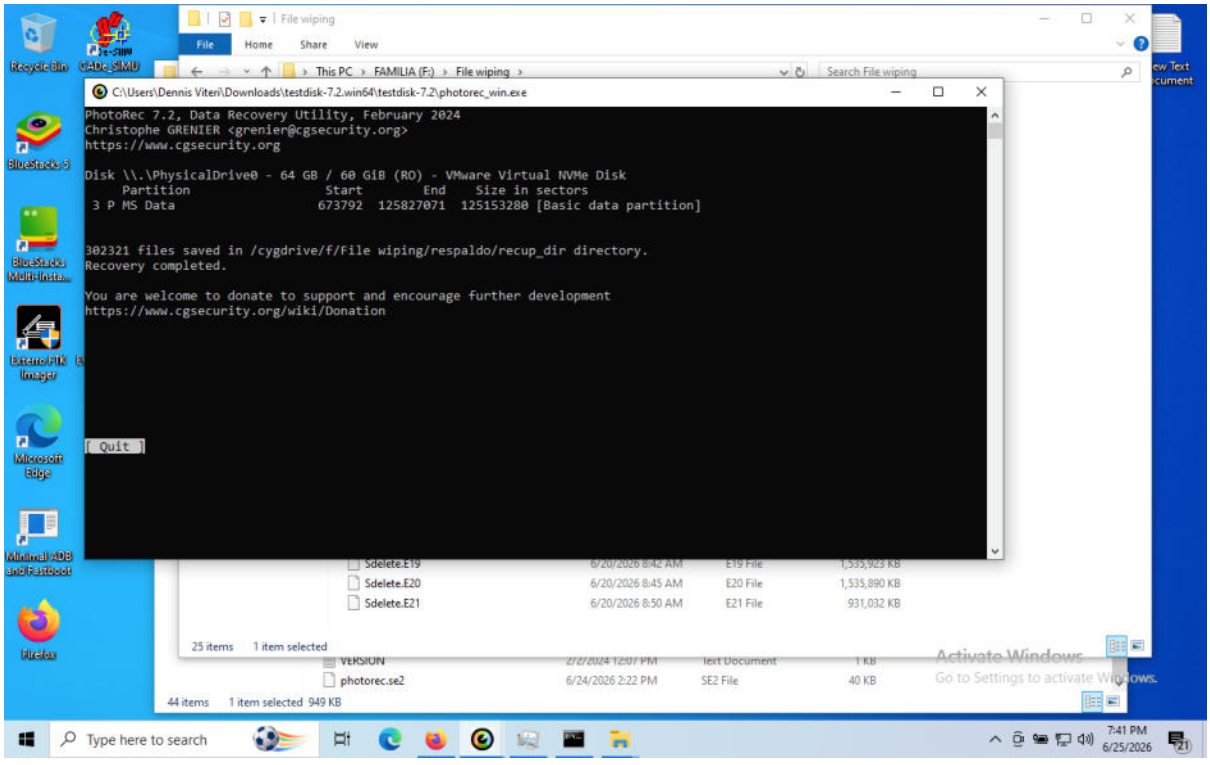
3.1.6.3.15 Ejecución de PhotoRec en modo Whole

Se repitió el procedimiento seleccionando el modo de búsqueda Whole, correspondiente al análisis completo del disco. La herramienta recuperó un total de 302,321 archivos, almacenados en la carpeta de destino:

`F:\File wiping\respaldo\recup_dir.`

Figura 84

Resultado del Disco local C en PhotoRec.

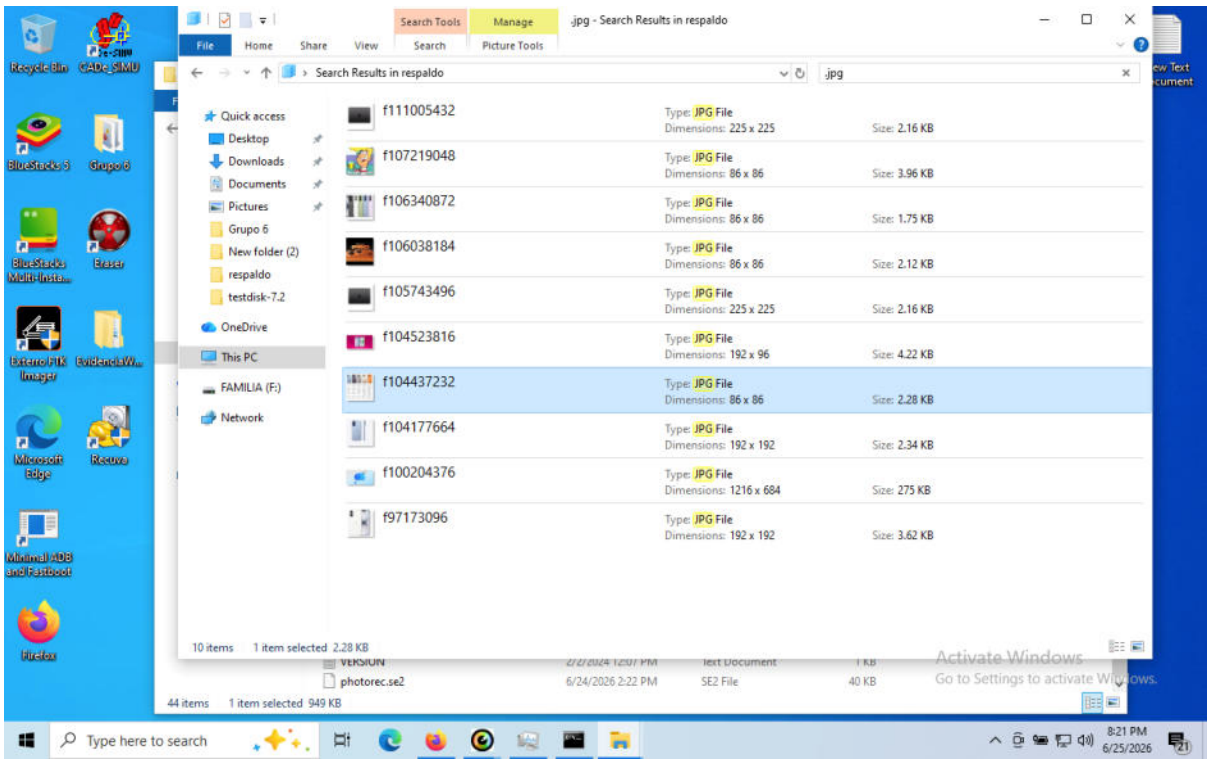


Nota. La figura presenta el proceso finalizado de la recuperación de los archivos con la aplicación de PhotoRec. Fuente: Elaboración propia.

Entre los archivos recuperados se identificaron imágenes personales correspondientes a las eliminadas durante la prueba de control, confirmando la efectividad del procedimiento. Los archivos recuperados no conservaron su nombre original, sino una nomenclatura autogenerada por la herramienta (por ejemplo, f104437232.jpg), debido a que el proceso de carving por firma reconstruye el contenido del archivo sin acceso a los metadatos originales del sistema de archivos.

Figura 85

Imágenes recuperadas en PhotoRec.



Nota. La figura presenta las imágenes recuperadas con la aplicación de PhotoRec

Fuente: Elaboración propia.

Este resultado confirmó que el protocolo de detección complementado con PhotoRec sí es capaz de recuperar evidencia eliminada cuando esto es técnicamente posible, validando que los resultados negativos obtenidos previamente con Eraser corresponden a la efectividad real de la técnica, y no a una limitación del proceso de análisis.

3.1.6.3.3 Ejecución de PhotoRec sobre el archivo wipeado con Eraser

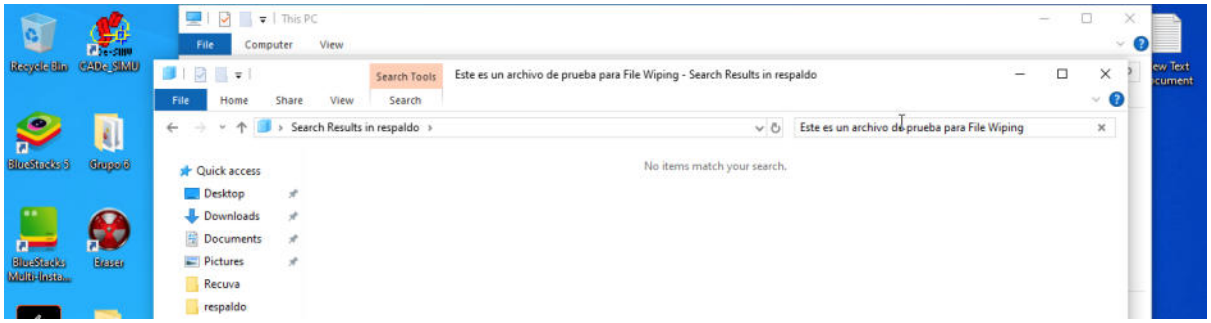
Una vez validada la efectividad del protocolo de detección mediante la prueba de control, se procedió a buscar el contenido del archivo prueba.txt entre los archivos recuperados mediante la ejecución de PhotoRec en modo Whole sobre el mismo disco físico (`\\.\PhysicalDrive0`), correspondiente a la misma sesión de recuperación utilizada para la prueba de control, descrita en el apartado 3.1.6.3.15. Dado que el proceso de carving por firma de archivo no conserva los nombres originales de los archivos recuperados, la búsqueda se realizó por contenido, utilizando el texto identificable incluido en el archivo original al momento de su creación.

Tras revisar el conjunto de 302,321 archivos recuperados almacenados en la carpeta de destino `F:\File wiping\respaldo\recup_dir`, no se localizó ningún archivo de texto que contuviera el contenido correspondiente a prueba.txt. Este resultado confirma que, a diferencia de los archivos eliminados de forma convencional los cuales sí fueron recuperables mediante PhotoRec, como se evidenció en la prueba de control, el contenido del archivo sometido a wiping mediante Eraser no fue recuperable ni siquiera mediante una herramienta de carving que opera directamente sobre el disco físico sin depender de la estructura del sistema de

archivos. Esto refuerza la efectividad de la técnica de File Wiping evaluada, dado que su resistencia se mantiene incluso frente a un método de recuperación que sí demostró ser efectivo en condiciones de eliminación estándar.

Figura 86

Búsqueda del archivo de File Wiping.

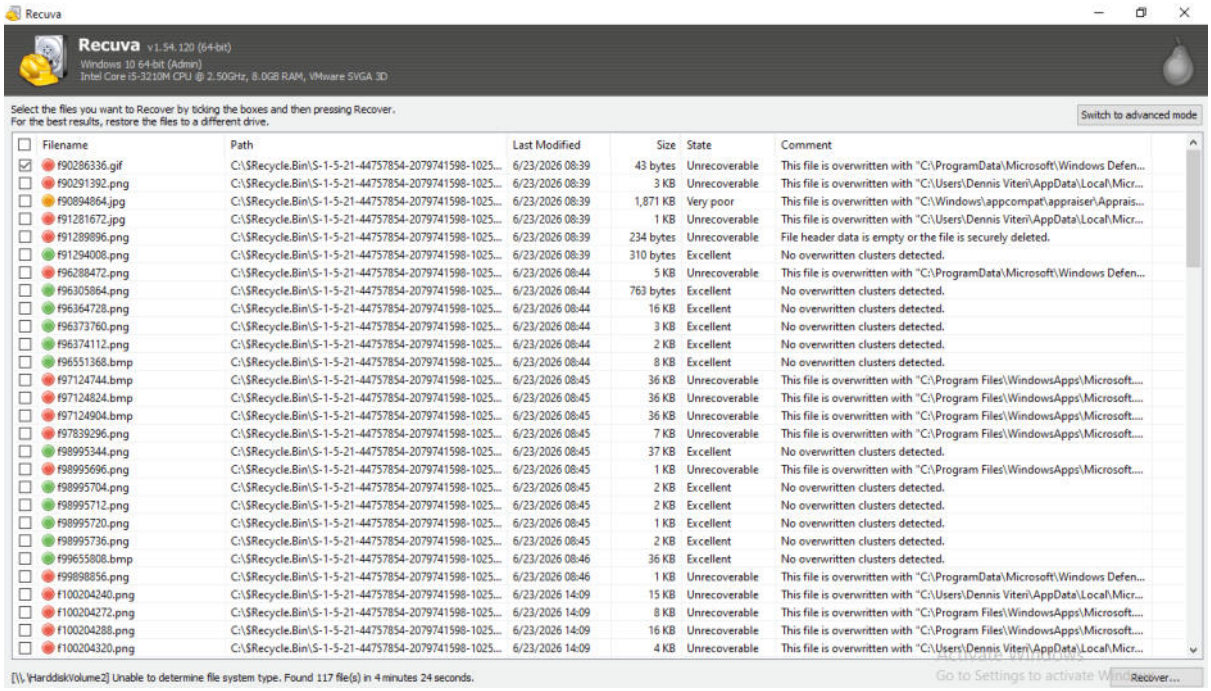


Nota. La figura presenta la búsqueda del texto que contenía el archivo el cual se hizo pruebas de File Wiping Fuente: Elaboración propia.

3.6.1.4 Evidencia visual (capturas)

Figura 87

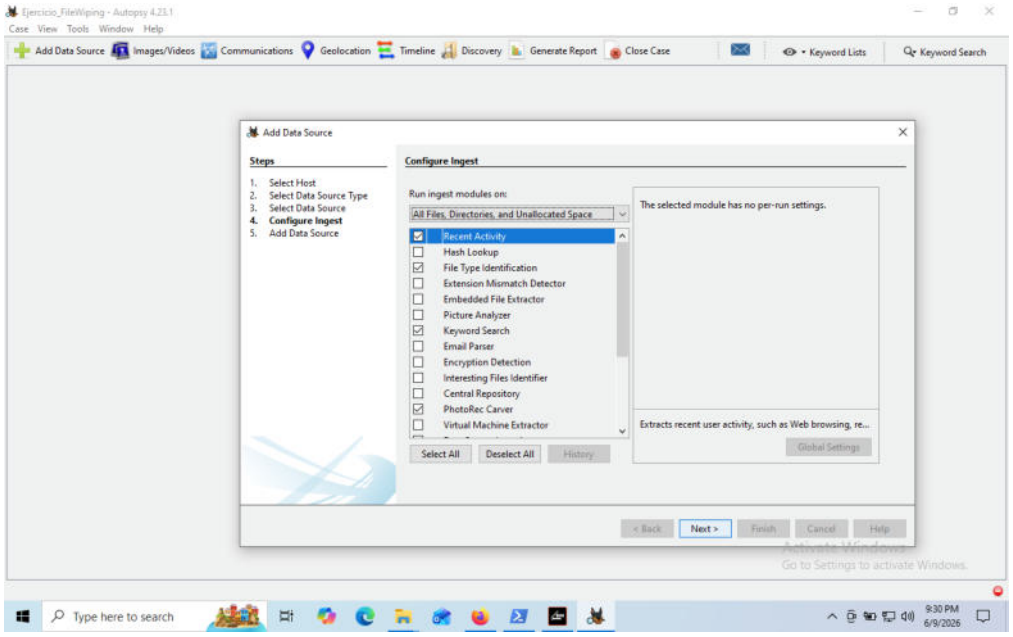
Búsqueda en Recuva.



Nota. La figura presenta la visualización de la interfaz del programa de Recuva. Fuente: Elaboración propia.

Figura 88

Configuración de Autopsy.

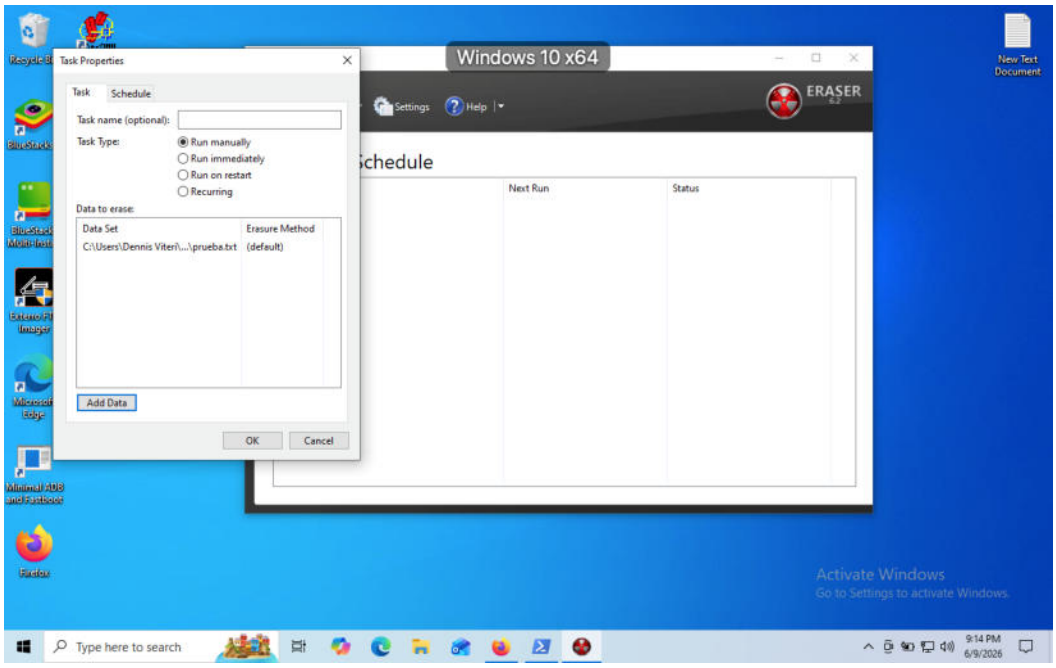


Nota. La figura presenta la visualización de los módulos que nosotros vamos a usar en

Autopsy para analizar el archivo exportado. Fuente: Elaboración propia.

Figura 89

Configuración de Eraser.



Nota. La figura presenta la visualización de la forma en configurar el programa Eraser.

Fuente: Elaboración propia.

3.1.6.5.Artefactos afectados.

Los artefactos comprometidos durante la ejecución del laboratorio de File Wiping se detallan a continuación:

Tabla 13.

Artefactos afectados durante el laboratorio de File Wiping

Artefacto	Tipo	Estado	Observación Forense
prueba.txt	Archivo de texto (.txt)	Sobreescrito y eliminado	Contenido destruido por sobreescritura; no recuperable mediante Autopsy, Recuva ni PhotoRec
Timestamps MAC de prueba.txt	Metadatos NTFS	Destruidos (0000-00-00)	Valores temporales nulos identificados en Autopsy; indicador indirecto de wiping
Hash MD5/SHA1 de prueba.txt	Función de integridad	No calculable post-wiping	Imposibilidad de verificar integridad del archivo tras la sobreescritura
Shadow Copies	Punto de restauración del sistema	No disponibles	Verificado mediante vssadmin list shadows; descartado como vía de recuperación
Archivos de imagen personales (prueba de control)	Archivos eliminados convencionalmente	Eliminados sin sobreescritura	Recuperados mediante PhotoRec modo Whole; no recuperables con Autopsy ni Recuva en entorno virtual

Nota. La tabla presenta los artefactos digitales generados o afectados durante la ejecución de la técnica de File Wiping con Eraser y la prueba de control con eliminación convencional, indicando su estado tras la aplicación de cada procedimiento y su recuperabilidad mediante las herramientas forenses empleadas. Fuente: Elaboración propia.

3.1.6.6.Método de detección aplicado

Se aplicaron cuatro métodos complementarios de detección forense, organizados en orden creciente de profundidad:

Exportación lógica mediante FTK Imager y análisis en Autopsy La primera etapa consistió en la exportación lógica de la carpeta del escritorio mediante FTK Imager y su análisis en Autopsy bajo la modalidad Logical Files. Se realizaron búsquedas mediante Keyword Search y revisión de Deleted Files. Este método no logró identificar ni recuperar el archivo prueba.txt ni fragmentos de su contenido, confirmando que la modalidad de exportación lógica no accede a sectores sobreescritos del disco.

Adquisición de imagen física mediante FTK Imager y análisis en Autopsy La segunda etapa consistió en la generación de una imagen física completa del disco en formato E01 mediante FTK Imager, verificada mediante hash MD5 y SHA1. La imagen fue analizada en Autopsy bajo la modalidad Disk Image, activando los módulos de Deleted Files y PhotoRec Carver integrado. Este método tampoco permitió recuperar el archivo wipeado, confirmando que la sobreescritura realizada por Eraser operó a un nivel suficientemente profundo para resistir el análisis forense de imagen física.

Intento de recuperación con Recuva (Deep Scan) Se ejecutó Recuva con la opción Deep Scan activada sobre el directorio del escritorio. El escaneo no logró recuperar el archivo prueba.txt, reportando 0 archivos encontrados, incluso tras múltiples ejecuciones del procedimiento.

Recuperación mediante PhotoRec (file carving directo sobre disco físico) Como método de validación del protocolo de detección, se ejecutó PhotoRec (versión 7.2) directamente sobre el disco físico (\\.\PhysicalDrive0) en modo de análisis completo (Whole), sin

depender de una adquisición forense previa mediante FTK Imager. A diferencia de los métodos anteriores, PhotoRec opera mediante reconocimiento de firmas de archivo (file carving), reconstruyendo archivos a partir del contenido de los sectores del disco sin depender de la estructura del sistema de archivos NTFS.

Para validar que este método sí es capaz de recuperar archivos cuando esto es técnicamente posible, se realizó previamente una prueba de control consistente en la eliminación convencional de archivos mediante Shift+Delete. PhotoRec en modo Whole recuperó exitosamente 302,321 archivos, entre los cuales se identificaron imágenes personales eliminadas durante la prueba de control, confirmando la efectividad del método en condiciones de eliminación estándar.

Posteriormente, el mismo procedimiento fue aplicado sobre el archivo wipeado con Eraser. La búsqueda del contenido original de prueba.txt entre los 302,321 archivos recuperados no arrojó resultados positivos, confirmando que el wiping destruyó el contenido del archivo de forma irreversible incluso frente a este método de recuperación más profundo.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

4.1.Pruebas de Concepto

4.1.1. Timestomping

4.1.1.1. Descripción de la prueba

Con el propósito de evaluar el impacto de la técnica de Timestomping sobre el análisis forense de sistemas de archivos NTFS, se ejecutó la modificación deliberada de los atributos temporales de un conjunto de archivos mediante comandos de PowerShell. La práctica tuvo como objetivo analizar el efecto de la alteración de las fechas de creación, modificación y último acceso sobre la reconstrucción cronológica de los eventos, así como evaluar la capacidad de las herramientas forenses para identificar evidencias compatibles con este tipo de manipulación.

La técnica se implementó utilizando un archivo legítimo como referencia temporal, cuyos atributos fueron clonados hacia un conjunto de archivos de prueba, permitiendo generar un escenario representativo de manipulación de metadatos ampliamente descrito en investigaciones sobre técnicas antiforenses.

4.1.1.2. Procedimiento aplicado

Inicialmente se creó un conjunto de archivos de prueba conformado por doce archivos, además de un archivo legítimo utilizado como referencia para la clonación de los atributos temporales. Posteriormente, mediante comandos de PowerShell, se modificaron los atributos Creation Time, Last Write Time y Last Access Time de los

archivos seleccionados, igualándolos a los valores temporales del archivo de referencia con el propósito de simular un escenario de Timestomping.

Finalizada la ejecución de la técnica, se realizó la adquisición forense de la evidencia mediante FTK Imager, obteniéndose una imagen lógica de los archivos utilizados durante el laboratorio y verificando su integridad mediante funciones hash MD5 y SHA1. Posteriormente, la evidencia fue analizada utilizando Autopsy, permitiendo una primera inspección de los metadatos de los archivos manipulados.

Con el objetivo de profundizar el análisis, se efectuó la extracción del archivo Master File Table ($MFT) mediante la herramienta KAPE, obteniéndose una copia íntegra del artefacto del sistema de archivos NTFS. Finalmente, el archivo $MFT fue procesado con MFTECmd, permitiendo comparar los atributos $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30) de los archivos analizados e identificar discrepancias y patrones temporales asociados a la manipulación de metadatos.

4.1.1.3. Artefactos generados

Tabla 14.

Artefactos generados durante la prueba de Timestomping

Artefacto	**Estado**
archivo_legitimo.txt	Utilizado como archivo de referencia para la clonación de atributos temporales.
Archivos manipulados (doc1.txt–doc10.txt, reporte.txt y evidencia_final.txt)	Marcas temporales modificadas mediante PowerShell; archivos recuperados y analizados.
Imagen lógica de la evidencia	Adquirida mediante FTK Imager y utilizada para el análisis forense.
Archivo $MFT	Extraído mediante KAPE y procesado con MFTECmd para el análisis de los metadatos NTFS.
Registros de la MFT	Analizados mediante MFTECmd, permitiendo comparar los atributos $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30).
Hash MD5/SHA1 de la evidencia	Calculados y verificados durante el proceso de adquisición forense.

Nota. La tabla presenta los principales artefactos digitales generados o afectados durante la ejecución de la técnica de Timestomping, así como su estado durante las fases de adquisición y análisis forense. Fuente: Elaboración propia.

4.1.2. Eliminación de Logs de Eventos

4.1.2.1. Descripción de la prueba

Se ejecutó la eliminación deliberada de los registros de eventos de seguridad de Windows mediante tres métodos distintos: Event Viewer (Clear Log), PowerShell (Clear-EventLog) y wevtutil (wevtutil cl), con el objetivo de evaluar la capacidad de las herramientas forenses estándar para detectar dicha manipulación.

4.1.2.2. Procedimiento aplicado

Previo a cada uno de los tres métodos se regeneró actividad en el sistema (apertura de aplicaciones, comandos administrativos), garantizando que cada prueba se ejecutara sobre un registro con contenido real. Posteriormente se realizó una exportación lógica del archivo Security.evtx mediante FTK Imager, complementada con una adquisición forense completa mediante imagen física en formato E01. La evidencia fue analizada en Autopsy en dos etapas: sobre los archivos exportados (Logical Files) y sobre la imagen física (Disk Image), validando en ambos casos la presencia del archivo Security.evtx y del Event ID 1102.

4.1.2.3. Artefactos generados

Tabla 15.

Artefactos generados durante la prueba de eliminación de logs de eventos

Artefacto	**Estado**
Security.evtx (copia lógica)	Eliminado el contenido visible; archivo recuperado y analizado
Security.evtx (dentro de imagen física E01)	Localizado y verificado en ruta original del sistema
Event ID 1102	Generado consistentemente en los tres métodos
Hash MD5/SHA1 del archivo	Calculado y verificado
Hash MD5/SHA1 de la imagen física	Calculado y verificado por FTK Imager

Nota. La tabla presenta los principales artefactos digitales generados o afectados durante la ejecución de la técnica de eliminación de logs de eventos, así como su estado tras el procedimiento experimental. Fuente: Elaboración propia.

4.1.3. Eliminación de archivos Prefetch

4.1.3.1. Descripción de la prueba

La presente prueba tuvo como objetivo analizar el impacto de la eliminación de archivos Prefetch sobre la evidencia digital disponible en un sistema Windows. Los archivos Prefetch son generados automáticamente por el sistema operativo y contienen información relacionada con la ejecución de aplicaciones, por lo que constituyen un artefacto de interés durante los procesos de análisis forense.

Para la realización de la prueba se ejecutaron diferentes aplicaciones con el fin de generar archivos Prefetch dentro del sistema. Posteriormente, mediante el uso de herramientas nativas de Windows, se verificó la presencia de estos archivos en el directorio C:\Windows\Prefetch y se registró la información correspondiente utilizando la herramienta WinPrefetchView.

Una vez confirmada la existencia de los archivos Prefetch, se procedió a eliminarlos manualmente para simular una técnica antiforense orientada a ocultar evidencia de ejecución de programas. Después de la eliminación, se realizó una nueva revisión del directorio Prefetch y de la información mostrada por WinPrefetchView con el propósito de verificar los cambios generados por la técnica aplicada.

Finalmente, se analizaron los resultados obtenidos para determinar el impacto de la eliminación de archivos Prefetch sobre la evidencia disponible y evaluar la capacidad de identificar la ejecución previa de aplicaciones tras la aplicación de la técnica antiforense.

Tabla 16.

Artefactos forenses generados por los archivos Prefetch

Artefacto	**Descripción**

.pf	Archivos Prefetch generados por Windows
Execution traces	Rastros de ejecución de aplicaciones
Timestamps	Fechas y horas de ejecución
Historial de aplicaciones	Evidencia de programas ejecutados

Nota. La tabla presenta los principales artefactos forenses generados por los archivos Prefetch en sistemas operativos Windows.

4.1.4. ADS (Alternate Data Streams)

4.1.4.1. Descripción de la prueba

La presente prueba tuvo como propósito evaluar la detectabilidad de la ocultación de información confidencial en el sistema de archivos NTFS mediante el uso de flujos alternos de datos (ADS). El diseño experimental buscó determinar la capacidad latente del sistema operativo Windows 11 para invisibilizar un payload de texto plano (credenciales.txt con parámetros de conexión simulados de bases de datos de producción) adosado a un archivo contenedor legítimo de fachada (notas_soporte.txt). La hipótesis central del experimento consistió en verificar si el subsistema de almacenamiento mantiene inalterados los metadatos visibles del archivo principal (longitud en bytes reportada al usuario) tras la inyección, midiendo el nivel de sigilo de la técnica frente a una inspección convencional y estableciendo las metodologías forenses estándar requeridas para su identificación.

4.1.4.2. Procedimiento Aplicado

- Aislamiento y preparación: Debido a las restricciones de seguridad impuestas por las Listas de Control de Acceso (ACLs) nativas del kernel en el directorio System32, se procedió a parametrizar una ruta de trabajo dedicada dentro del perfil del usuario

($HOME\Desktop\Laboratorio_Antiforense).4.1.4.3. Eliminación de Logs / Alteración de Evento.

- Generación del contenedor base: Se ejecutó el cmdlet Out-File en Windows PowerShell con codificación ASCII para generar el archivo portador notas_soporte.txt. Mediante la propiedad Length, se auditó y certificó su tamaño inicial en un valor estático de 135 bytes.
- Inyección del payload oculto: Se utilizó el cmdlet Set-Content empleando el parámetro específico -Stream para crear e inyectar el flujo de datos alternativo oculto nominado credenciales.txt, asignándole una cadena de datos equivalente a 60 bytes reales.
- Fase de Triage y Detección en Vivo: Se contrastó la visualización macroscópica del Explorador de Windows y comandos convencionales (donde el archivo reportaba seguir midiendo 135 bytes) frente a llamadas directas al kernel de archivos de bajo nivel mediante el parámetro wildcard Get-Item .\notas_soporte.txt -Stream *.
- Análisis Forense Post-Mortem: Se ejecutó la adquisición lógica de la carpeta y se procesó en la suite pericial Autopsy (v4.23.1). El flujo de trabajo requirió la activación manual y selectiva del módulo de ingesta File Type Identification para parsear los atributos $DATA de la Master File Table (MFT) y extraer el flujo secundario de forma íntegra.

4.1.4.4. Eliminación de Logs de Evento

La inyección y el aprovisionamiento nativo de un flujo alterno de datos (ADS) mediante comandos de PowerShell o la API de Windows no generan de forma predeterminada alertas automáticas, registros de auditoría ni Event IDs específicos en los canales

tradicionales del Visor de Eventos (Security, System o Application). Debido a esta ausencia de telemetría reactiva, la técnica no requiere de un proceso activo de deshabilitación o purga de logs para mantener su persistencia silenciosa.

Para contrarrestar esta falta de eventos y descubrir la alteración, el analista forense debe auditar las estructuras físicas y transaccionales del volumen NTFS a través de dos mecanismos estándar:

- Parseo de la MFT en Autopsy: Identificación de inconsistencias estructurales en el registro del archivo de la MFT, donde el módulo detecta la anomalía de un segundo atributo $DATA que posee un nombre asignado (bifurcación del ADS) frente al descriptor anónimo estándar del flujo primario.
- Correlación con el diario de cambios ($UsnJrnl): Extracción y análisis del archivo transaccional de bajo nivel $UsnJrnl. A pesar de no existir un log de eventos activo, el diario de transacciones de Windows registra de forma física irreversible las operaciones bajo la razón de control STREAM_CHANGE y DATA_EXTEND asociadas al ID del archivo portador, indexando cronológicamente el momento exacto en el que el flujo secundario fue inyectado en el disco.

4.1.5. Esteganografía

4.1.5.1. Descripción de la prueba

Esta prueba tuvo como objetivo evaluar el impacto técnico y el nivel de detección de la ocultación avanzada de contenido mediante esteganografía digital basada en la sustitución del Bit Menos Significativo (LSB) sobre archivos multimedia en entornos Windows. El experimento se diseñó para simular un vector real de exfiltración encubierta de datos sensibles, incrustando un payload crítico (exfiltracion.txt, que

contenía un token API simulado de producción) dentro de una imagen de perfil corporativo legítima en formato JPEG (FOTO_CARNET_CV.jpeg). La prueba buscó medir la resistencia de la técnica frente a la inspección visual humana y los controles perimetrales automatizados (como firmas de antivirus), validando la efectividad del análisis de integridad criptográfica diferencial como contramedida de detección forense primaria.

4.1.5.2. Procedimiento Aplicado

- Configuración del entorno: Se desplegó la utilidad portable de línea de comandos Steghide v0.5.1 junto con sus librerías de entorno (cygwin1.dll y cygcrypt-4.dll) dentro de una subcarpeta aislada nominada Esteganografia.
- Creación del Payload: Se generó el archivo de texto plano exfiltracion.txt utilizando codificación ASCII para almacenar la información confidencial objeto de la simulación.
- Incrustación esteganográfica: Se ejecutó el binario de ocultación mediante el comando steghide embed -cf FOTO_CARNET_CV.jpeg -sf exfiltracion.txt -p [password], aplicando el algoritmo de sustitución LSB combinado con un cifrado simétrico nativo AES-128 para proteger la carga útil.
- Evaluación de Invisibilidad: Se realizaron comparaciones macroscópicas directas en pantalla utilizando el Explorador de Windows entre la imagen original y la procesada (portador_oculto.jpg), comprobando que no existieran distorsiones de color ni alteraciones geométricas perceptibles.

- Análisis Forense Matemático: Se calculó la firma hash de ambos archivos mediante el cmdlet Get-FileHash aplicando el algoritmo SHA-256 para contrastar la ruptura de la integridad binaria.
- Ciclo de Extracción Forense: Se simuló el estegoanálisis activo recuperando el contenido oculto mediante la sintaxis steghide extract -sf portador_oculto.jpg -p [password] -xf payload_extraido.txt, verificando la integridad del mensaje recuperado

4.1.5.3. Eliminacion de Logs

Al igual que con la técnica ADS, la manipulación de la estructura de bits internos de un archivo multimedia mediante herramientas esteganográficas portables es una operación puramente en memoria y a nivel de usuario que no activa alertas en tiempo real, ni genera Event IDs indiciarios en los logs de auditoría nativos del sistema operativo Windows. Dado que las soluciones de antivirus convencionales no catalogan estos contenedores alterados como código malicioso, la técnica alcanza un alto nivel de evasión inicial en el endpoint. Para subsanar la ausencia de registros de eventos de Windows y evidenciar la manipulación del archivo multimedia, el procedimiento forense estándar se enfoca en las siguientes contramedidas:

- Ruptura de Integridad Criptográfica Diferencial: La inyección de datos en los canales de color altera irreversiblemente la composición binaria del archivo. El análisis forense identificó una discrepancia absoluta entre los hashes SHA-256, donde la imagen original generó una firma iniciada con DA019C... y el archivo esteganográfico produjo un hash completamente divergente iniciado con 8540A9.... Este cambio drástico es el principal indicador para alertar sobre la manipulación a sistemas de Monitoreo de Integridad de Archivos (FIM).

- Auditoría de Artefactos Secundarios de Ejecución: Aunque el log de Windows no registra el cambio interno del archivo, el uso de la herramienta externa steghide.exe deja "migajas digitales" en el sistema operativo. El perito forense puede correlacionar la actividad esteganográfica mediante el análisis de la colmena de registro Amcache.hve, las tablas Shimcache (Syscache) y el parseo de la MFT / $UsnJrnl, donde quedan registradas de forma cronológica las operaciones de lectura/escritura, físicas del ejecutable portable y del archivo exfiltracion.txt, exponiendo el nexo causal del ataque.

4.1.6. File Wiping

4. 1.6.1 Descripción de la prueba

Se aplicó sobreescritura segura sobre un archivo de prueba (prueba.txt) mediante la herramienta Eraser, con el objetivo de evaluar la resistencia de esta técnica frente a múltiples métodos de recuperación forense de distinta profundidad. El laboratorio se desarrolló sobre un disco virtual configurado como dinámicamente asignado (dynamically allocated).

4.1.6.2 Procedimiento aplicado

El procedimiento se desarrolló en tres fases progresivas. En la primera fase se aplicó el wiping con Eraser sobre el archivo prueba.txt, verificando su ausencia en el sistema activo y la papelera de reciclaje, y descartando la existencia de Shadow Copies. Posteriormente se realizó la adquisición de evidencia mediante FTK Imager en dos niveles: exportación lógica e imagen física completa en formato E01, analizando ambas fuentes en Autopsy sin lograr recuperar el archivo. En una segunda fase, ante la imposibilidad de recuperación mediante Autopsy y Recuva, se incorporó PhotoRec

como herramienta de validación del protocolo de detección. Se realizó una prueba de control con eliminación convencional (Shift+Delete), confirmando que PhotoRec en modo Whole sí recupera archivos eliminados de forma estándar (302,321 archivos recuperados), mientras que el mismo método no logró recuperar el contenido del archivo wipeado con Eraser, validando que la ausencia de recuperación responde a la efectividad real de la técnica antiforense.

4.1.6.3 Artefactos generados

Tabla 17.

Artefactos identificados en la técnica antiforense

Artefacto	**Estado**
Prueba.txt	Sobreescrito e irrecuperable en todos los métodos aplicados
Timestamps MAC	Destruidos (0000-00-00)
Hash MD5/SHA1	No calculable post-wiping
Jump List (.lnk)	Presente como rastro indirecto; sin contenido original
Archivos de control (eliminación normal)	Recuperados con PhotoRec modo Whole

4.2 Análisis de Resultados

4.2.1 Timestomping

Tabla 18.

Evaluación mediante métricas

Métrica		**Estado**
Detectabilidad	La modificación de los atributos temporales puede pasar desapercibida durante una revisión superficial; sin	6/10

	embargo, el análisis de la MFT mediante MFTECmd permitió identificar discrepancias entre los atributos $STANDARD_INFORMATION y $FILE_NAME, así como patrones temporales compatibles con la manipulación de metadatos.	
Trazabilidad	La correlación de los metadatos obtenidos mediante FTK Imager, Autopsy y MFTECmd permitió reconstruir parcialmente la secuencia de los eventos y establecer relaciones entre los archivos manipulados y el archivo legítimo utilizado como referencia.	8/10
Recuperabilidad	Aunque la técnica altera las marcas temporales visibles de los archivos, el análisis de la MFT permitió recuperar información suficiente para identificar indicadores compatibles con la manipulación. No obstante, la cronología original no puede reconstruirse completamente únicamente a partir de los metadatos modificados.	6/10
Integridad verificable	La evidencia adquirida conservó su integridad durante todo el proceso de adquisición y análisis, validándose mediante funciones hash MD5 y SHA1 antes de su procesamiento con las herramientas forenses.	10/10

4.2.1.1 Interpretación

La aplicación de la técnica de Timestomping mediante PowerShell demostró que es posible modificar de forma deliberada los atributos temporales de archivos almacenados en sistemas de archivos NTFS, alterando las fechas de creación, modificación y último acceso sin modificar el contenido de los archivos. Como consecuencia, los artefactos

manipulados adquirieron una cronología aparente diferente a la real, dificultando la reconstrucción de los eventos cuando el análisis se limita a la inspección de los metadatos visibles del sistema operativo.

La incorporación de un proceso de adquisición forense, mediante FTK Imager, complementado con la extracción del archivo Master File Table ($MFT) utilizando KAPE, permitió analizar la evidencia desde dos perspectivas diferentes: la observación de los metadatos de los archivos adquiridos y el examen de los registros internos del sistema de archivos NTFS. Posteriormente, el procesamiento del archivo $MFT mediante MFTECmd permitió comparar los atributos $STANDARD_INFORMATION (0x10) y $FILE_NAME (0x30), identificando discrepancias temporales y patrones repetitivos entre los archivos manipulados.

Los resultados obtenidos evidenciaron que los doce archivos sometidos a la técnica compartían atributos temporales coincidentes, mientras que el archivo archivo_legitimo.txt conservó sus valores originales al ser utilizado como referencia para la clonación de metadatos. Asimismo, la presencia de diferencias entre los atributos temporales almacenados en la MFT demostró que la modificación de los timestamps no elimina completamente los indicios forenses asociados a la manipulación, proporcionando información útil para el proceso de investigación.

En conclusión, la técnica de Timestomping constituye un mecanismo eficaz para alterar la cronología visible de los archivos y dificultar el análisis forense basado exclusivamente en los atributos temporales expuestos por el sistema operativo. Sin

embargo, la aplicación de una metodología que combine la adquisición adecuada de la evidencia, el análisis de la Master File Table y la correlación de múltiples fuentes de metadatos permite identificar indicadores consistentes de manipulación y fortalece la capacidad del investigador para detectar actividades antiforenses.

4.2.2 Eliminación de Logs de Eventos

Tabla 19.

Resultados obtenidos

Nota. La tabla presenta los principales resultados de verificación obtenidos durante la aplicación de la técnica de eliminación de logs de eventos, incluyendo la generación del Event ID 1102, la disponibilidad del archivo Security.evtx y la verificación de integridad mediante hash. Fuente: Elaboración propia.

Tabla 20.

Evaluación mediante métricas

Métrica		**Estado**
Detectabilidad	Se identificaron rastros de actividad mediante FTK Imager y Autopsy, incluyendo el archivo Security.evtx residual y artefactos Prefetch	10/10
Trazabilidad	Persistencia de artefactos asociados a PowerShell y herramientas administrativas, permitiendo correlacionar la actividad antiforense ejecutada	10/10
Recuperabilidad	El contenido eliminado del log no fue recuperable	6/10

	directamente; sin embargo, el archivo Security.evtx permaneció disponible en exportación lógica y en imagen física	
Integridad verificable	Se calculó y verificó el hash MD5/SHA1 del archivo Security.evtx y de la imagen física adquirida	10/10

4.2.2.1 Interpretación

La eliminación de logs de eventos mediante Event Viewer, PowerShell y wevtutil demostró ser una técnica antiforense de baja efectividad real frente al análisis forense estándar, a pesar de eliminar exitosamente el contenido visible del registro de Seguridad. El factor determinante es que Windows genera de manera automática el Event ID 1102 al ejecutarse cada uno de los tres métodos de limpieza, constituyendo un mecanismo de auditoría nativo que documenta la acción incluso cuando el atacante intenta eliminar el registro repetidamente.

La incorporación de una adquisición forense completa —mediante imagen física E01 verificada con hash MD5/SHA1— permitió validar este hallazgo en dos niveles independientes de evidencia: la copia obtenida por exportación lógica y la copia extraída directamente de la imagen física del disco. En ambos casos se confirmó la presencia del Event ID 1102, descartando que se trate de un artefacto exclusivo del método de adquisición utilizado.

En conclusión, esta técnica resulta altamente detectable y trazable, ya que ataca únicamente la disponibilidad del contenido del log activo, pero no impide la generación del evento de auditoría que documenta el propio acto de eliminación, ni la persistencia de artefactos colaterales del sistema operativo.

4.2.3 Eliminación de archivos Prefetch

4.2.3.1 Resultados obtenidos

Eliminación de Archivos Prefetch

Durante la práctica se pudo comprobar que la eliminación de archivos Prefetch es un procedimiento sencillo de ejecutar utilizando herramientas nativas de Windows. Después de generar evidencia mediante la ejecución de distintas aplicaciones y aplicar el comando de eliminación, se verificó que los archivos .pf asociados a dichas aplicaciones ya no se encontraban disponibles dentro de la carpeta Prefetch.

Al comparar el estado del sistema antes y después de aplicar la técnica, se evidenció una reducción de los rastros relacionados con la ejecución de programas. Esto dificulta la identificación de aplicaciones utilizadas y limita la información disponible para reconstruir la secuencia de eventos dentro del sistema.

Adicional, se observó que la eliminación de los archivos Prefetch no elimina completamente toda la evidencia digital. Aún pueden existir otros registros y artefactos del sistema que permitan inferir actividades realizadas previamente, por lo que esta técnica debe considerarse como un mecanismo para dificultar el análisis forense y no como una solución definitiva para ocultar toda la actividad del sistema.

4.2.3.2 Evaluación mediante métricas

Los resultados obtenidos evidencian que la eliminación de archivos Prefetch puede ejecutarse de forma rápida y sencilla, requiriendo un bajo nivel de complejidad técnica por parte del atacante. En menos de un minuto fue posible eliminar los archivos Prefetch asociados a distintas aplicaciones ejecutadas dentro del sistema, afectando

directamente la disponibilidad de evidencias relacionadas con la ejecución de programas.

De igual manera, se determinó que el impacto sobre el análisis forense es alto, ya que la ausencia de estos artefactos dificulta la reconstrucción de actividades realizadas en el sistema comprometido. A pesar de la eliminación de los archivos Prefetch, aún es posible encontrar rastros secundarios y artefactos residuales que pueden ser utilizados por un analista forense para identificar indicios de manipulación antiforense y reconstruir parcialmente los eventos ocurridos.

Tabla 21.

Resultados obtenidos de la eliminación de archivos Prefetch

Métrica	Valor medido	Resultado	Observación
Archivos Prefetch eliminados	Aproximadamente 100 archivos	Alto	La técnica permitió eliminar una cantidad significativa de archivos Prefetch generados por el sistema.
Aplicaciones afectadas	4 aplicaciones	Medio	La eliminación afectó la información asociada a las aplicaciones ejecutadas durante la prueba.
Tiempo de ejecución de la técnica	Menor a 1 minuto	Bajo	La técnica puede ejecutarse rápidamente utilizando herramientas nativas de Windows.
Dificultad de implementación	Baja	Baja	No requiere conocimientos avanzados ni herramientas especializadas para su ejecución.

Dificultad de detección	Media	Media	Aunque los archivos Prefetch son eliminados, pueden existir otros artefactos que permitan inferir la ejecución de aplicaciones.
Impacto sobre el análisis forense	Alto	Alto	La eliminación de archivos Prefetch reduce la disponibilidad de evidencia relacionada con la ejecución de programas.
Persistencia de rastros secundarios	Sí	Media	Permanecen artefactos alternativos que pueden ser utilizados para complementar la investigación forense.

Nota. Elaboración propia a partir de los resultados obtenidos durante la ejecución de la técnica de eliminación de archivos Prefetch en el entorno de pruebas.

4.2.3.3 Interpretación de resultados

La eliminación de archivos Prefetch constituye una técnica antiforense que puede resultar efectiva para dificultar el análisis de ejecución de aplicaciones en sistemas operativos Windows. Durante la práctica se evidenció que su implementación presenta un bajo nivel de complejidad y puede realizarse rápidamente mediante el uso de herramientas nativas del sistema, lo que facilita su utilización por parte de un atacante.

El análisis forense realizado permitió comprobar que la eliminación de estos archivos no implica la desaparición total de la evidencia digital. A pesar de la ausencia de archivos Prefetch, fue posible identificar otros artefactos y rastros residuales dentro del sistema

operativo, los cuales pueden aportar información relevante durante una investigación forense.

Esto demuestra la importancia de aplicar metodologías de correlación de evidencia y análisis integral de artefactos, ya que un único mecanismo de ocultamiento no garantiza la eliminación completa de los rastros de actividad. En conclusión, esta técnica tiene un impacto considerable sobre la reconstrucción de eventos y líneas temporales, aunque puede ser detectada mediante un análisis forense especializado y la identificación de evidencias secundarias presentes en el sistema.

Tabla 22.

Escala de efectividad antiforense aplicada a la eliminación de archivos Prefetch

Criterio de evaluación	**Puntuación (0-10)**	**Clasificación**	**Interpretación**
Recuperación de evidencia	8	Alta	La eliminación de archivos Prefetch reduce significativamente la evidencia relacionada con la ejecución de aplicaciones.
Trazabilidad	5	Media	Aunque los archivos Prefetch son eliminados, permanecen artefactos alternativos que permiten inferir actividad previa.
Dificultad de detección	6	Media	La ausencia de archivos Prefetch puede identificarse mediante análisis forense, aunque requiere correlación con otros artefactos.
Tiempo de análisis	4	Baja	La verificación de la técnica no demanda un incremento significativo en el tiempo de investigación.

Nota. Elaboración propia a partir de los resultados obtenidos durante la ejecución de la técnica de eliminación de archivos Prefetch.

4.2.4 ADS (Alternate Data Streams)

Resultados Obtenidos:

- Los resultados del laboratorio ADS confirmaron las siguientes conclusiones técnicas:
- La técnica ADS logró ocultar 60 bytes de payload bajo un archivo portador de 135 bytes, manteniendo el tamaño reportado invariable ante herramientas convencionales.
- El Sistema Operativo Windows 11 no genera ninguna alerta ni notificación visual al crear o acceder a un flujo ADS mediante PowerShell.
- El Explorador de Windows, el Administrador de Tareas y la consola cmd.exe no revelan la presencia del flujo alterno bajo inspección estándar.
- La detección mediante PowerShell (Get-Item -Stream *) es inmediata y no requiere herramientas externas, aunque presupone conocimiento técnico del investigador.
- Autopsy requirió la activación manual del módulo File Type Identification para revelar los ADS; sin este módulo, la herramienta mostró únicamente el flujo principal.
- El flujo alterno fue recuperado íntegramente sin pérdida de datos, confirmando la viabilidad de ADS como vector de exfiltración de información.

Evaluación de Métricas:

Tabla 23.
A continuación, se presentan las métricas cuantitativas de evaluación del laboratorio ADS:

Métrica	**Valor Medido**	**Resultado**	**Observación**
Tamaño archivo portador	135 bytes	Certificado	Tamaño $DATA primario
Tamaño flujo ADS	60 bytes	Oculto	No visible en propiedades estándar
Ratio ocultamiento	44.4%	Efectivo	60/135 = 0.444
Tiempo inyección ADS	< 1 segundo	Inmediato	Set-Content en tiempo real

Detección cmd.exe	No detectado	Fallido	dir /R requerido
Detección PowerShell	Detectado	Exitoso	Get-Item -Stream *
Detección Autopsy (sin módulos)	No detectado	Fallido	Solo flujo primario
Detección Autopsy (con módulos)	Detectado	Exitoso	File Type Identification
Integridad del payload recuperado	100%	Íntegro	Sin pérdida de datos
Alertas del SO generadas	0	Nulas	Sin notificación al usuario

Escala de Efectividad Antiforense:

Tabla 24.

Escala de efectividad antiforense aplicada a la técnica ADS

Criterio de Evaluación	**Puntuación (0-10)**	**Clasificación**	**Justificación**
Invisibilidad ante usuario final	9/10	Alta	No detectable sin conocimiento técnico
Invisibilidad ante herramientas básicas	8/10	Alta	Explorer, cmd, taskmgr no lo revelan
Resistencia a detección forense	4/10	Media-Baja	Detectable con PS y Autopsy+módulos
Facilidad de implementación	9/10	Alta	Un solo cmdlet de PowerShell
Capacidad de payload	6/10	Media	Limitado al FS NTFS (no transferible)
PROMEDIO GLOBAL	7.2/10	Alta-Media	Efectivo en entornos no especializados

Interpretación de Resultados:

Los resultados del laboratorio de ADS evidencian una técnica antiforense con alta efectividad en entornos no especializados, pero con vulnerabilidades inherentes frente a análisis forense técnico. La puntuación global de 7.2/10 refleja su relevancia como vector real de amenaza en escenarios donde el personal de TI no dispone de herramientas o conocimientos forenses avanzados.

La incapacidad del Sistema Operativo de generar alertas ante la creación o acceso a flujos ADS representa el vector de riesgo más significativo desde la perspectiva de la seguridad corporativa, dado que permite la exfiltración de datos sensibles sin activar mecanismos de DLP (Data Loss Prevention) que no estén específicamente configurados para monitorizar estructuras de bajo nivel del sistema de archivos NTFS.

Sin embargo, la técnica presenta limitaciones estructurales críticas: los flujos ADS no se transfieren al copiar archivos entre sistemas de archivos distintos de NTFS (como FAT32 o ExFAT), lo que restringe su alcance de persistencia. Adicionalmente, herramientas forenses estándar como Autopsy, una vez configuradas correctamente, detectan la bifurcación con total fiabilidad.

Desde la perspectiva de la respuesta a incidentes, la recomendación técnica es incorporar en los procesos de auditoría la ejecución periódica de Get-Item -Recurse -Stream * sobre directorios críticos, así como la activación sistemática de módulos de ingesta especializados en cualquier análisis con Autopsy que involucre volúmenes NTFS.

4.2.5 Esteganografía

Resultados Obtenidos:

Los resultados del laboratorio de esteganografía confirmaron las siguientes conclusiones

técnicas:

- La técnica LSB de Steghide logró incrustar exitosamente el payload exfiltracion.txt dentro de la imagen JPEG portadora sin producir ninguna diferencia visual detectable por el ojo humano.
- El hash SHA-256 de la imagen portadora fue completamente alterado tras la incrustación, pasando de DA019C... a 8540A9..., lo que representa una ruptura total de integridad matemática.
- La imagen modificada supera sin problema las inspecciones visuales convencionales, incluyendo comparaciones directas entre la imagen original y la modificada en pantalla.
- La detección basada únicamente en inspección visual resultó ineficaz, lo que confirma la viabilidad del método como técnica de exfiltración encubierta.
- El análisis de hashes SHA-256 detectó la manipulación de forma inmediata y determinante, requiriendo únicamente comandos nativos de PowerShell sin herramientas adicionales.
- La extracción forense recuperó el payload íntegramente sin pérdida de datos, confirmando la efectividad bidireccional de la técnica (ocultamiento y recuperación).
- El cifrado integrado de Steghide (AES-128 por defecto) añade una capa adicional de protección al payload, haciendo ineficaz la extracción sin la contraseña correcta.

Evaluación Mediante Métricas

Tablas de Métricas Técnicas

Tabla 25.

Tablas de Métricas Técnicas, técnica esteganografía

Métrica	Valor Medido	Resultado	Observación
Hash SHA-256 imagen original	DA019C... (inicio)	Referencia	Firmado antes de la operación
Hash SHA-256 imagen modificada	8540A9... (inicio)	Alterado	Cambio total de firma criptográfica
Detección visual humana	No detectado	Ineficaz	Imágenes visualmente idénticas
Detección por hash SHA-256	Detectado	Exitoso	Inmediato con Get-FileHash
Extracción sin contraseña	No recuperado	Fallido	AES-128 activo por defecto
Extracción con contraseña	Recuperado 100%	Exitoso	Payload íntegro sin pérdidas
Alertas del SO generadas	0	Nulas	Sin notificación en tiempo real
Alertas de antivirus	0	Nulas	No detectado como malicioso
Tiempo de ocultamiento	< 3 segundos	Inmediato	Steghide en tiempo real
Integridad del payload extraído	100%	Íntegra	Sin corrupción de datos

Escala de Efectividad Antiforense

Tabla 26.

Escala de efectividad antiforense aplicada a la técnica AD

Criterio de Evaluación	Puntuación (0-10)	Clasificación	Justificación
Invisibilidad visual	10/10	Máxima	Imágenes indistinguibles para humanos
Invisibilidad ante SO y AV	9/10	Alta	Sin alertas del sistema o antivirus
Resistencia a detección forense	6/10	Media	Detectable por hash, no por visual
Facilidad de implementación	8/10	Alta	Herramienta portable, un comando

Portabilidad del payload	9/10	Alta	Imagen transferible por cualquier medio
Seguridad del payload (cifrado)	8/10	Alta	AES-128 integrado en Steghide
PROMEDIO GLOBAL	8.3/10	Alta	Efectiva incluso ante investigadores

Interpretación de Resultados

Los resultados del laboratorio de esteganografía revelan una técnica antiforense con mayor efectividad global (8.3/10) que la técnica ADS (7.2/10), lo que puede atribuirse principalmente a dos factores diferenciadores: la portabilidad del contenedor y el cifrado integrado del payload.

A diferencia de los flujos ADS, que quedan destruidos al copiar el archivo a sistemas de archivos no NTFS, una imagen JPEG con contenido esteganográfico mantiene su payload independientemente del medio de transferencia (correo electrónico, almacenamiento USB, servicios en la nube, mensajería instantánea). Esto amplía significativamente el vector de exfiltración y la dificultad de detección en los controles perimetrales de la organización.

La limitación técnica más relevante de la técnica es la ruptura de integridad criptográfica que introduce: cualquier proceso de monitorización de integridad de archivos (FIM, File Integrity Monitoring) que registre hashes SHA-256 de imágenes corporativas detectará la manipulación de forma automática. Esta es la contramedida defensiva más eficiente y de menor coste computacional frente a la esteganografía LSB.

La incapacidad de los sistemas antivirus convencionales para detectar el archivo como malicioso subraya la necesidad de incorporar herramientas de estegoanálisis estadístico (como StegExpose o análisis de histogramas de color) en los procesos de seguridad perimetral de organizaciones con alto riesgo de amenazas internas. La esteganografía representa un vector de amenaza infradetectado en la mayoría de los marcos de seguridad corporativa actuales, lo que justifica su inclusión como técnica de estudio prioritaria en los programas de ciberseguridad forense.

4.2.6 File Wiping

Tabla 27.

Resultados obtenidos

Verificación	**Autopsy Lógico**	**Autopsy Físico**	**Recuva**	**PhotoRec**
Archivo recuperado	No	No	No	No
Contenido recuperado	No	No	No	No
Timestamps preservados	No (0000-00-00)	No (0000-00-00)	No (0000-00-00)	No (0000-00-00)
Hash verificable	No calculable	No calculable	No calculable	No calculable
Shadow Copies disponibles	No existen	No existen	No existen	No existen
Archivos de control recuperados	No aplica	No aplica	No	Sí (302,321 archivos)

Tabla 28.

Evaluación mediante métricas

Métrica	Resultado	Estado	
Detectabilidad	El archivo wipeado no fue detectable ni recuperable por ninguna herramienta estándar aplicada	1/10	
Trazabilidad	Se identificaron rastros indirectos: timestamps destruidos (0000-00-00) y Jump List (.lnk) que confirman la existencia previa del archivo, sin contener su contenido original	4/10	
Recuperabilidad	Imposible recuperar el contenido en ninguno de los cuatro métodos aplicados, incluyendo PhotoRec en modo Whole sobre el disco físico completo	1/10	
Integridad verificable	Hash MD5/SHA1 no calculable post-wiping; sin posibilidad de verificar la integridad del archivo eliminado	1/10	

4.2.2.1 Interpretación

El File Wiping mediante Eraser demostró ser una técnica antiforense de alta efectividad, confirmada a través de cuatro métodos de detección progresivos: exportación lógica, imagen física completa, escaneo de recuperación dedicado (Recuva Deep Scan) y carving directo sobre disco físico (PhotoRec modo Whole). En ninguno de los cuatro escenarios fue posible recuperar el archivo prueba.txt ni fragmentos reconocibles de su contenido.

La incorporación de la prueba de control representa un elemento metodológico clave: al confirmar que PhotoRec sí recupera archivos eliminados de forma convencional (302,321 archivos en la prueba de control), se descarta que los resultados negativos obtenidos con Eraser respondan a una limitación del proceso de análisis forense, y se

atribuyen directamente a la efectividad de la técnica antiforense evaluada.

Los únicos indicadores detectables de manera indirecta fueron la presencia de timestamps con valor 0000-00-00 00:00:00 en todos los campos temporales del archivo y la persistencia de un artefacto Jump List (.lnk) en el sistema operativo. Estos rastros no permiten reconstruir el contenido original del archivo, pero sí constituyen una firma reconocible de que se ejecutó una operación de sobreescritura deliberada sobre ese espacio del sistema de archivos.

En comparación con la eliminación de logs de eventos — técnica que genera automáticamente el Event ID 1102, lo que la hace altamente detectable — el File Wiping no activa ningún mecanismo de auditoría nativo del sistema operativo, lo que explica su puntuación significativamente más baja en todas las métricas de detectabilidad evaluadas.

En conclusión, el File Wiping con Eraser representa una técnica antiforense considerablemente más robusta que la eliminación de logs, cuya efectividad se mantiene incluso frente a métodos de recuperación que operan directamente sobre el disco físico sin depender de la estructura del sistema de archivos. Los únicos rastros identificables son de carácter indirecto y no permiten la reconstrucción de la evidencia original.

4.3 Análisis comparativo de resultados

4.3.1 Valoración obtenida para cada técnica

Una vez finalizadas las pruebas de las seis técnicas antiforenses, se realizó una comparación de los resultados obtenidos utilizando las mismas métricas definidas en el protocolo

experimental. Esto permitió evaluar cada técnica bajo los mismos criterios y analizar las diferencias observadas en su impacto sobre la evidencia digital y en su nivel de detectabilidad.

Para cada técnica se consideraron las métricas de recuperación de evidencia, persistencia del rastro digital, dificultad de detección y tiempo de análisis. Con base en estos criterios se asignó una puntuación que facilitó la comparación de los resultados obtenidos durante la investigación.

Tabla 29.

Valoración obtenida por cada una de las técnicas antiforenses implementada

Técnica	Recuperación de evidencia	Trazabilidad	Dificultad de detección	Tiempo de análisis	Promedio
Timestomping	8	7	6	6	6,75
Eliminación de logs	7	5	6	5	5,75
Eliminación de archivos Prefetch	8	5	6	5	6,00
ADS	5	8	8	7	7,00
Esteganografía	5	8	8	8	7,25
File Wiping	9	8	9	8	8,50

Nota. Elaboración propia a partir de los resultados obtenidos durante las pruebas realizadas. La puntuación de cada técnica corresponde al promedio de los criterios de evaluación definidos en la investigación y fue utilizada para establecer su nivel de efectividad antiforense.

4.3.2 Escala de interpretación de resultados

Tabla 30.

Escala de interpretación de la efectividad antiforense

Rango de puntuación	Nivel de efectividad antiforense	Interpretación
1,0 – 2,0	Muy baja	La técnica genera un impacto mínimo sobre la evidencia digital y puede detectarse fácilmente durante el análisis forense.
2,1 – 4,0	Baja	La técnica produce alteraciones limitadas y conserva suficientes rastros para facilitar su identificación.
4,1 – 6,0	Media	La técnica dificulta parcialmente el análisis forense, aunque todavía es posible identificar evidencia y artefactos residuales mediante un análisis detallado.
6,1 – 8,0	Alta	La técnica afecta de forma importante la evidencia digital, dificultando su detección y recuperación durante el análisis forense.
8,1 – 10,0	Muy alta	La técnica presenta una elevada capacidad antiforense, reduciendo significativamente la posibilidad de detectar y recuperar evidencia relevante.

Nota. Escala elaborada para interpretar las puntuaciones obtenidas durante la evaluación comparativa de las técnicas antiforenses implementadas en el entorno experimental.

4.3.3 Clasificación general de las técnicas implementadas

Tabla 31.
Interpretación **de las técnicas antiforenses**

Técnica	Promedio	Clasificación

Timestomping	6,75	Alta
Logs	5,75	Media
Prefetch	6,00	Media
ADS	7,00	Alta
Esteganografía	7,25	Alta
File Wiping	8,50	Muy alta

Nota. Elaboración propia. La clasificación final se obtuvo aplicando la escala de interpretación definida en la Tabla 31 al promedio alcanzado por cada técnica. Esta clasificación permitió comparar de manera uniforme el nivel de efectividad antiforense observado durante la investigación.

4.3.4 Interpretación de resultados

En términos generales, File Wiping obtuvo la puntuación más alta, alcanzando un nivel de efectividad antiforense muy alto, debido al impacto que genera sobre la recuperación de la evidencia digital. Por su parte, ADS, Esteganografía y Timestomping fueron clasificadas con un nivel alto, ya que dificultan el análisis forense, aunque durante las pruebas fue posible identificar algunos rastros residuales. Finalmente, las técnicas de eliminación de logs de eventos y eliminación de archivos Prefetch obtuvieron una clasificación media, evidenciando que, aunque alteran la evidencia digital, aún dejan artefactos que pueden ser identificados mediante un análisis detallado.

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

A través de las pruebas realizadas fue posible analizar el comportamiento de diferentes técnicas antiforenses aplicadas en sistemas Windows y observar el efecto que generan sobre la evidencia digital. Cada técnica presentó características particulares y afectó distintos artefactos del sistema, permitiendo identificar diferencias en su impacto y nivel de detectabilidad.

La aplicación de técnicas como la eliminación de logs de eventos, eliminación de archivos Prefetch, ADS, esteganografía y file wiping permitió comprobar que es posible dificultar el trabajo de análisis forense mediante la alteración, ocultación o eliminación de información. Sin embargo, en la mayoría de los casos se identificaron rastros residuales o artefactos alternativos que permitieron inferir la actividad realizada.

Los resultados obtenidos evidenciaron que no todas las técnicas presentan el mismo nivel de efectividad antiforense. Algunas afectan directamente la disponibilidad de la evidencia, mientras que otras se enfocan en ocultar información o dificultar su identificación. Esto permitió observar diferencias importantes en la forma en que cada técnica impacta el proceso de análisis forense.

La evaluación realizada mediante las métricas definidas en la investigación facilitó la comparación de las técnicas estudiadas bajo los mismos criterios de análisis. Esto permitió obtener una visión más clara de su comportamiento y reducir la subjetividad al momento de interpretar los resultados obtenidos durante las pruebas.

Finalmente, la matriz de evaluación desarrollada permitió organizar y comparar los resultados de manera estructurada, identificando las técnicas que generan un mayor impacto

sobre la evidencia digital y aquellas que presentan mayores niveles de detectabilidad. Los resultados obtenidos demuestran la importancia de complementar el análisis forense con diferentes fuentes de evidencia y no depender únicamente de un artefacto específico durante una investigación.

5.2. Contribuciones

El presente trabajo aporta una forma estructurada de evaluar la detectabilidad de técnicas antiforenses en sistemas Windows mediante un entorno de laboratorio controlado. La implementación de las pruebas permitió observar el comportamiento de cada técnica y analizar el impacto que produce sobre la evidencia digital y los principales artefactos del sistema.

Como parte de la investigación, se desarrolló una matriz de evaluación que reúne métricas como la recuperación de evidencia, la persistencia del rastro digital, la dificultad de detección y el tiempo de análisis. Esta matriz permitió comparar las técnicas implementadas utilizando los mismos criterios de evaluación, facilitando el análisis de los resultados obtenidos.

Otro aporte de este trabajo es la documentación del procedimiento utilizado para implementar y analizar las seis técnicas antiforenses seleccionadas. Esto permite que el laboratorio pueda ser reproducido en futuras investigaciones o utilizado como material de apoyo en actividades académicas relacionadas con la informática forense.

Finalmente, la investigación demuestra que el uso de herramientas nativas de Windows, junto con WinPrefetchView, permite identificar diferentes rastros asociados a la aplicación de técnicas antiforenses. Los resultados obtenidos muestran que es posible realizar este tipo de análisis sin depender exclusivamente de herramientas forenses comerciales, ofreciendo una alternativa práctica para entornos de laboratorio y de formación

5.3. Recomendaciones

Se recomienda que los procesos de análisis forense no se basen únicamente en un tipo de artefacto o fuente de evidencia, ya que durante las pruebas realizadas se observó que varias técnicas antiforenses afectan información específica del sistema sin eliminar completamente todos los rastros de actividad.

Es importante complementar el análisis mediante la correlación de diferentes artefactos digitales, con el fin de identificar inconsistencias o evidencias residuales que puedan indicar la aplicación de técnicas antiforenses. Este enfoque permite obtener una visión más completa de los eventos ocurridos en el sistema.

Se recomienda que los investigadores forenses mantengan actualizados sus procedimientos y metodologías de análisis, considerando que muchas técnicas antiforenses pueden ejecutarse utilizando herramientas de fácil acceso o incluso funcionalidades nativas del sistema operativo.

Para futuras investigaciones, se recomienda ampliar el número de técnicas analizadas e incorporar otros artefactos de Windows que permitan evaluar escenarios más complejos y cercanos a entornos reales. De igual manera, podría considerarse la aplicación simultánea de varias técnicas antiforenses para analizar su impacto combinado sobre la evidencia digital.

Finalmente, se recomienda continuar desarrollando criterios y métricas que permitan evaluar de forma objetiva la detectabilidad de las técnicas antiforenses, con el propósito de fortalecer los procesos de análisis forense y facilitar la comparación de resultados entre diferentes estudios.

6. REFERENCIAS BIBLIOGRÁFICAS

Alji, M., & Chougdali, K. (2019). *Detection of timestamps tampering in NTFS using machine learning*. *Procedia Computer Science, 160*, 778–784. https://www.researchgate.net/publication/337434341_Detection_of_Timestamps_Tampering_in_NTFS_using_Machine_Learning

Autopsy Digital Forensics. (2024). *Autopsy user documentation* (Version 4.23). Basis Technology. https://sleuthkit.org/autopsy/docs/

Bejtlich, R. (2005). *The Tao of network security monitoring: Beyond intrusion detection*. Addison-Wesley Professional.

Carvey, H. (2020). *Windows forensic analysis toolkit* (5th ed.). Syngress.

Carrier, B. (2005). *File system forensic analysis*. Addison-Wesley Professional.

Casey, E. (2024). *Digital evidence and computer crime: Forensic science, computers, and the Internet* (4th ed.). Academic Press.

Cheddad, A., Condell, J., Curran, J., & McKevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal Processing, 90*(3), 727–752. https://doi.org/10.1016/j.sigpro.2009.08.010

Farmer, D., & Venema, W. (2005). *Forensic discovery*. Addison-Wesley Professional.

Fridrich, J. (2009). *Steganography in digital media: Principles, algorithms and applications*. Cambridge University Press. https://doi.org/10.1017/CBO9781139192903

Garfinkel, S. L. (2007). Anti-forensics: Techniques, detection and countermeasures. *Proceedings of the 2nd International Conference on Information Warfare and Security (ICIW)*, 77–84. https://apps.dtic.mil/sti/citations/ADA576161

González Arias, R., Bermejo Higuera, J., Rainer Granados, J. J., Bermejo Higuera, J. R., & Sicilia Montalvo, J. A. (2024). Systematic review: Anti-forensic computer techniques. *Applied Sciences, 14*(12), 5302. https://doi.org/10.3390/app14125302

Harris, R. (2006). Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. *Digital Investigation, 3*(1), 44–49. https://doi.org/10.1016/j.diin.2006.06.008

International Organization for Standardization. (2012). *Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence (ISO/IEC 27037:2012).* https://www.iso.org/standard/44381.html

International Organization for Standardization. (2015). *Information technology — Security techniques — Guidelines for the analysis and interpretation of digital evidence (ISO/IEC 27042:2015).* https://www.iso.org/standard/44465.html

Jones, K. J., Bejtlich, R., & Rose, C. W. (2006). *Real digital forensics: Computer security and incident response*. Addison-Wesley Professional.

Joo, D., Lee, J., & Jeong, D. (2023). A reference database of Windows artifacts for file-wiping tool execution analysis. *Journal of Forensic Sciences, 68*(3), 856–870. https://doi.org/10.1111/1556-4029.15240

Kessler, G. C. (2023). *Steganography: Hiding data within data*. Gary Kessler Associates. https://www.garykessler.net/library/steganography.html

Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). *The art of memory forensics: Detecting malware and threats in Windows, Linux, and Mac memory*. Wiley.

Lyle, J. R., Guttman, B., Butler, J., Sauerwein, K., Reed, C., & Lloyd, C. (2022). *Digital investigation techniques: A NIST scientific foundation review (NIST IR 8354).* National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8354

Microsoft. (2024). *del*. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/del

Microsoft. (2024). *Get-ChildItem*. Microsoft Learn. https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-childitem

Microsoft. (2024). *Windows performance documentation*. Microsoft Learn. https://learn.microsoft.com/en-us/windows

Microsoft Corporation. (2023). *Alternate data streams in NTFS*. Microsoft Learn. https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-fscc/efbfe127-73ad-4140-9967-ec6500e66d5e

Microsoft Corporation. (2025). *NTFS Alternate Data Streams and PowerShell Management*. Microsoft Learn Documentation. https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-item

Microsoft Corporation. (2025). *USN_RECORD_V2 structure and change journal control codes*. Microsoft Learn. https://learn.microsoft.com/en-us/windows/win32/api/winioctl/ns-winioctl-usn_record_v2

MITRE Corporation. (2026). *Defense Evasion Tactic (TA0005).* MITRE ATT&CK®. https://attack.mitre.org/tactics/TA0005/

MITRE Corporation. (2026). *Hide Artifacts (T1564).* MITRE ATT&CK®. https://attack.mitre.org/techniques/T1564/

MITRE Corporation. (2026). *Hide Artifacts: Alternate Data Streams (T1564.004).* MITRE ATT&CK®. https://attack.mitre.org/techniques/T1564/004/

MITRE Corporation. (2026). *Indicator Removal (T1070).* MITRE ATT&CK®. https://attack.mitre.org/techniques/T1070/

MITRE Corporation. (2026). *Indicator Removal: File Deletion (T1070.004).* MITRE ATT&CK®. https://attack.mitre.org/techniques/T1070/004/

National Institute of Standards and Technology. (2014). *Guide to integrating forensic techniques into incident response (NIST Special Publication 800-86).* U.S. Department of Commerce. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf

National Institute of Standards and Technology. (2015). *Guidelines for Media Sanitization (NIST Special Publication 800-88 Rev. 1).* U.S. Department of Commerce. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-88r1.pdf

Nelson, B., Phillips, A., & Steuart, C. (2019). *Guide to computer forensics and investigations* (6th ed.). Cengage Learning.

VMware, Inc. (2024). *VMware Workstation Pro documentation*. https://docs.vmware.com/en/VMware-Workstation-Pro/

Yaacoub, J.-P. A., Noura, H. N., Salman, O., & Chehab, A. (2021). Digital forensics vs. anti-digital forensics: Techniques, limitations and recommendations. *arXiv*. https://doi.org/10.48550/arXiv.2103.17028

7. Anexos

Con el propósito de garantizar la reproducibilidad de los experimentos desarrollados durante la presente investigación, se pone a disposición del tribunal un repositorio que contiene las máquinas virtuales utilizadas en la implementación de las técnicas antiforenses, los laboratorios desarrollados y la documentación complementaria empleada durante las pruebas experimentales.

Repositorio de evidencias del laboratorio:

https://1024terabox.com/s/1xstG9MvErb2DaMlwiViPvg