FOMENTO DE LA PROTECCIÓN DE LOS SISTEMAS OPERATIVOS A TRAVÉS DEL ANÁLISIS AVANZADO DE MALWARE

Wilmer Guillermo Howard Abarca
Universidad CENFOTEC, San José, Costa Rica
e-mail: whowarda@ucenfotec.ac.cr

I. ABSTRACT

El análisis de malware constituye una de las actividades más relevantes dentro del campo de la ciberseguridad, debido al crecimiento constante de amenazas dirigidas a sistemas operativos modernos. En particular, los entornos Windows representan uno de los objetivos principales de los atacantes debido a su amplia adopción en entornos empresariales y personales. En este contexto, el presente estudio propone una guía técnica para el análisis de malware basada en la utilización de múltiples herramientas especializadas y técnicas de investigación digital.

La investigación se fundamenta en la aplicación de diferentes tipos de análisis, incluyendo análisis estático, ingeniería inversa, análisis dinámico, análisis de red y análisis forense de memoria. Cada una de estas técnicas permite examinar el comportamiento del malware desde diferentes perspectivas y facilitar la identificación de indicadores de compromiso (IOC). Para ello se implementó un laboratorio controlado basado en máquinas virtuales aisladas, donde se ejecutaron muestras de malware utilizando herramientas especializadas que permiten observar cambios en el sistema, comunicaciones de red y actividades sospechosas.

Adicionalmente, los resultados obtenidos durante el análisis se relacionan con el framework MITRE ATT&CK, el cual permite clasificar las tácticas y técnicas utilizadas por los atacantes en ataques reales. De esta forma, se logra contextualizar el comportamiento del malware dentro de un marco de referencia estandarizado utilizado ampliamente en la industria de la ciberseguridad. Como resultado, el estudio presenta una metodología estructurada que permite a analistas de seguridad comprender mejor el funcionamiento del malware y mejorar los procesos de detección y análisis de amenazas en sistemas Windows.

PALABRAS CLAVE: Análisis de malware, Ciberseguridad, Sistemas operativos Windows, Indicadores de compromiso (IoC) y Análisis forense de memoria.

II. INTRODUCCIÓN

El malware representa una de las amenazas más importantes dentro del panorama actual de la ciberseguridad. Este tipo de software malicioso está diseñado para comprometer la confidencialidad, integridad o disponibilidad de los sistemas informáticos, permitiendo a los atacantes ejecutar acciones como robo de información, espionaje, sabotaje o control remoto de dispositivos. Debido a la creciente sofisticación de estas amenazas, el análisis de

malware se ha convertido en una disciplina esencial para comprender su funcionamiento y desarrollar mecanismos de defensa efectivos.

El análisis de malware consiste en estudiar el comportamiento y las características de programas sospechosos con el objetivo de identificar su funcionalidad, los mecanismos utilizados para comprometer sistemas y los posibles indicadores de compromiso asociados con la infección. Este proceso puede realizarse mediante diversas técnicas que permiten examinar el software malicioso desde diferentes perspectivas, incluyendo análisis estático, dinámico, análisis de red e investigación forense de memoria (Fortinet, s. f.-a; Al-Sofyani et al., 2023; Wong et al., 2021).

El análisis estático se enfoca en examinar un archivo ejecutable sin necesidad de ejecutarlo en el sistema, lo que permite identificar elementos estructurales del programa como encabezados binarios, librerías importadas, cadenas de texto y recursos internos del ejecutable. Este tipo de análisis permite obtener información preliminar sobre el comportamiento potencial del malware sin el riesgo de activar código malicioso (Gillis, 2020; Talukder & Talukder, 2020).

Por otro lado, el análisis dinámico consiste en ejecutar el malware dentro de un entorno controlado con el fin de observar su comportamiento en tiempo real. Durante este proceso se monitorean actividades como la creación de procesos, modificaciones en el registro del sistema, cambios en archivos del sistema y conexiones de red generadas por el malware. Este enfoque permite identificar acciones que no pueden observarse únicamente mediante el análisis estático (Liu et al., 2022; Wong et al., 2021).

Adicionalmente, el análisis de red permite examinar el tráfico generado por el malware durante su ejecución, lo que facilita la identificación de comunicaciones con servidores externos, dominios sospechosos o infraestructuras de comando y control utilizadas por los atacantes. Asimismo, el análisis forense de memoria permite detectar malware que se ejecuta directamente en la memoria del sistema o que intenta ocultarse del sistema operativo (Snort, s. f.; Suricata, s. f.; Redwan et al., 2024).

Sin embargo, el proceso de análisis de malware también presenta diversos desafíos. Entre ellos se encuentran las técnicas de evasión utilizadas por los atacantes, como mecanismos anti-debugging, detección de entornos virtualizados o malware que reside únicamente en memoria. Estas técnicas buscan dificultar el trabajo de los analistas y evitar la detección de las amenazas durante el proceso de investigación.

Ante este panorama, resulta fundamental contar con metodologías estructuradas y herramientas especializadas que permitan realizar análisis completos y sistemáticos del

malware. En este sentido, el presente estudio propone una guía técnica para el análisis de malware en sistemas Windows que integra diferentes técnicas de análisis y herramientas especializadas dentro de un laboratorio controlado.

III. METODOLOGÍA

La presente investigación se desarrolló mediante un enfoque experimental orientado al análisis de malware en sistemas Windows mediante el uso de herramientas especializadas y entornos de laboratorio controlados. El objetivo principal consistió en evaluar diferentes técnicas y herramientas de análisis de malware con el fin de identificar indicadores de compromiso y comprender el comportamiento de muestras maliciosas.

El estudio contempla el análisis de diversas técnicas utilizadas en la investigación de malware, incluyendo análisis estático, ingeniería inversa, análisis dinámico, análisis de red y análisis forense de memoria. Cada una de estas técnicas permite estudiar diferentes aspectos del comportamiento del malware y facilita la obtención de evidencia digital durante el proceso de análisis.

El análisis estático: es una técnica fundamental en el estudio de malware que consiste en examinar un archivo ejecutable sin necesidad de ejecutarlo, lo que permite identificar características internas de forma segura. Para ello, se utilizan herramientas como visores de cadenas, por ejemplo Strings (Sysinternals), FLOSS (FireEye) o BinText, que permiten extraer texto incrustado en el binario, como rutas, URLs o comandos ocultos. A partir de este análisis, es posible identificar indicadores de compromiso (IoC), tales como direcciones IP sospechosas (por ejemplo, 192.168.x.x o IPs públicas asociadas a C2), dominios maliciosos (como malicious-domain.com), claves de registro alteradas (por ejemplo, HKCU\Software\Microsoft\Windows\CurrentVersion\Run), nombres de archivos inusuales, hashes de archivos (MD5, SHA256) o cadenas relacionadas con funciones sensibles del sistema (Gillis, 2020; Talukder & Talukder, 2020).

La ingeniería inversa: constituye una técnica esencial dentro del análisis de malware, ya que permite descomponer y examinar el funcionamiento interno de un software malicioso sin necesidad de acceder a su código fuente original. Mediante el uso de herramientas especializadas, como desensambladores y depuradores, es posible identificar estructuras internas, instrucciones ejecutables y comportamientos ocultos en el código. A través de este proceso, los analistas pueden comprender la lógica de propagación, los mecanismos de evasión (por ejemplo, la ofuscación o polimorfismo) y las intenciones del atacante, lo que facilita la clasificación del malware (por ejemplo, rootkits o worms) y la creación de firmas o reglas de detección más efectivas (Fox, 2023; Collado, s. f.; Micucci, 2024; Talukder & Talukder, 2020).

El análisis dinámico: es una técnica clave en el estudio de malware que consiste en ejecutar el software malicioso en un entorno controlado para observar su comportamiento en tiempo real. Este enfoque permite identificar acciones concretas que no son evidentes en el análisis estático, como la creación de procesos sospechosos, modificaciones en el registro del sistema, establecimiento de mecanismos de persistencia y comunicación con servidores externos. A partir de este análisis, se pueden detectar indicadores de compromiso (IoC) como direcciones IP y dominios asociados a centros de comando y control (C2), cambios en claves críticas del sistema, generación de archivos temporales o ejecución de comandos inusuales (Liu et al., 2022; Wong et al., 2021; Al-Sofyani et al., 2023).

El análisis de red: es una técnica fundamental en el estudio de malware que se enfoca en examinar el tráfico generado por un software malicioso para identificar comunicaciones sospechosas y patrones de comportamiento en la red. A través de este análisis, es posible detectar indicadores de compromiso (IoC) como direcciones IP externas, dominios maliciosos, solicitudes DNS inusuales, uso de puertos no estándar o conexiones cifradas hacia servidores de comando y control (C2). Asimismo, se pueden identificar patrones de comportamiento como comunicaciones periódicas tipo beaconing, transferencias de datos anormales que sugieren exfiltración de información, o intentos de propagación lateral dentro de la red (Snort, s. f.; Suricata, s. f.; Talukder & Talukder, 2020).

El análisis forense de memoria: es una técnica crítica en el estudio de malware que se centra en examinar el contenido de la memoria RAM para identificar evidencias volátiles que no quedan registradas en el disco. Este enfoque permite detectar indicadores de compromiso (IoC) como procesos ocultos, inyecciones de código en memoria, DLLs cargadas de forma anómala, credenciales en texto claro y conexiones de red activas que podrían pasar desapercibidas en otros tipos de análisis. Asimismo, facilita la identificación de patrones de comportamiento como técnicas de process hollowing, code injection, escalamiento de privilegios y ejecución de malware sin archivo (fileless malware) (Redwan et al., 2024; Al-Sofyani et al., 2023; Wong et al., 2021).

IV. RESULTADOS

Los resultados obtenidos a partir de la implementación de la propuesta evidencian que la correcta selección y combinación de herramientas especializadas permite cubrir de manera integral las diferentes fases del análisis de malware, demostrando que cada categoría de herramientas cumple un rol específico dentro del laboratorio. En particular, estas herramientas facilitan la recolección de evidencias, la identificación de comportamientos maliciosos y la generación de indicadores de compromiso, lo que permite estructurar un

proceso de análisis más completo, preciso y alineado con las necesidades actuales de la ciberseguridad.

1. Herramientas de virtualización (soporte del laboratorio)

Las herramientas VirtualBox y VMware Workstation demostraron ser fundamentales para la ejecución controlada de muestras maliciosas, permitiendo crear entornos aislados donde se pueden integrar el resto de herramientas. Su principal aporte dentro de los resultados es la capacidad de reproducir escenarios de análisis de forma segura y repetible, sirviendo como base para todas las demás fases del proceso.

2. Herramientas de análisis estático

En esta categoría, herramientas como PEiD, PEview, PE Explorer, BinText y Resource Hacker permitieron obtener información estructural relevante del ejecutable. Los resultados muestran que:

- PEiD: permitió identificar el uso de packers, compiladores y firmas conocidas, lo que facilita detectar técnicas de evasión como ofuscación o empaquetamiento del malware.
- PEview: permitió analizar la estructura interna del archivo PE, incluyendo encabezados, secciones y tablas, lo que ayuda a detectar anomalías en la composición del ejecutable.
- PE Explorer: facilitó la revisión de dependencias, librerías importadas y recursos del sistema, permitiendo identificar APIs potencialmente sospechosas utilizadas por el malware.
- BinText: permitió extraer cadenas de texto incrustadas en el binario, como direcciones IP, URLs, rutas de archivos o comandos, útiles como indicadores de compromiso.
- Resource Hacker: permitió examinar recursos internos del ejecutable, como imágenes, configuraciones o archivos embebidos, donde pueden ocultarse componentes maliciosos adicionales.

Estas herramientas en conjunto permitieron generar una primera caracterización técnica de las muestras analizadas, aportando información clave antes de ejecutar el malware.

3. Herramientas de ingeniería inversa

La herramienta principal identificada en los resultados es IDA Pro, la cual permitió desensamblar el código binario. Su uso facilitó:

- La identificación de funciones relevantes dentro del malware.
- El análisis de llamadas a APIs del sistema.
- La comprensión del flujo de ejecución del programa.

Los resultados muestran que esta herramienta es crítica para profundizar en la lógica interna del malware y complementar los hallazgos obtenidos en el análisis estático.

4. Herramientas de análisis dinámico

Las herramientas Process Monitor, Process Explorer, Regshot y Capture BAT permitieron observar el comportamiento del malware durante su ejecución. Según los resultados:

- Process Monitor: permitió registrar en detalle la actividad del sistema durante la ejecución del malware, incluyendo accesos a archivos, modificaciones en el registro y creación de procesos, lo que facilita identificar comportamientos sospechosos y acciones realizadas por la muestra.
- Process Explorer: facilitó la visualización de los procesos activos, sus relaciones jerárquicas y dependencias, permitiendo detectar procesos inusuales, inyecciones de código o ejecución de procesos secundarios maliciosos.
- Regshot: permitió comparar el estado del registro del sistema antes y después de la ejecución, identificando claves modificadas, creadas o eliminadas, lo cual es clave para detectar mecanismos de persistencia.
- Capture BAT: permitió registrar cambios realizados en el sistema durante la ejecución del malware, incluyendo modificaciones en archivos y actividades del sistema, facilitando la reconstrucción del comportamiento de la muestra

El uso combinado de estas herramientas permitió correlacionar cambios en el sistema y reconstruir la actividad del malware en tiempo real, lo cual es consistente con lo descrito en estudios de análisis dinámico donde múltiples herramientas trabajan de forma complementaria.

5. Herramientas de análisis de red

La herramienta principal identificada es Wireshark, la cual permitió capturar y analizar el tráfico generado por el malware. Los resultados evidencian que:

- Se pueden identificar conexiones externas sospechosas.
- Se facilita el análisis de paquetes en detalle.

- Se detectan patrones de comunicación con posibles servidores de comando y control.

Este tipo de herramientas es ampliamente reconocido en la literatura como esencial para el análisis de tráfico malicioso y la identificación de comunicaciones ocultas.

6. Herramientas de análisis forense de memoria

Las herramientas DumpIt, FastDump, Memoryze, Rekall y Volatility permitieron analizar volcados de memoria RAM. Los resultados muestran que:

- DumpIt: permitió generar volcados completos de la memoria RAM del sistema, facilitando la captura del estado del sistema en el momento de la ejecución del malware para su posterior análisis.
- FastDump: permitió obtener imágenes de memoria de forma rápida y eficiente, optimizando el proceso de adquisición de evidencia en entornos controlados.
- Memoryze: facilitó la identificación de procesos activos en memoria, incluyendo aquellos que no son visibles mediante herramientas tradicionales del sistema, lo que permite detectar actividad maliciosa oculta.
- Rekall: permitió realizar análisis forense avanzado sobre los volcados de memoria, identificando estructuras del sistema, procesos, conexiones y artefactos relevantes asociados al malware.
- Volatility: permitió profundizar en el análisis de memoria RAM mediante plugins especializados, facilitando la detección de inyección de código, procesos ocultos y comportamiento malicioso residente en memoria.

Resultado clave: el uso conjunto de estas herramientas permitió identificar técnicas avanzadas como malware fileless y procesos ocultos, evidenciando actividades que no dejan rastros en el disco y que solo pueden detectarse mediante análisis de memoria.

7. Integración de herramientas y resultados

Finalmente, los resultados reflejan que ninguna herramienta por sí sola es suficiente. La combinación de herramientas de diferentes categorías permite:

- Correlacionar evidencia entre múltiples fuentes.
- Validar hallazgos obtenidos en distintas fases.
- Generar indicadores de compromiso más completos.

Esta integración facilita posteriormente el mapeo con MITRE ATT&CK y la generación de inteligencia de amenazas, consolidando un enfoque estructurado y efectivo para el análisis de malware dentro del laboratorio propuesto.

V. DISCUSIÓN

Los resultados obtenidos en la implementación de la guía de análisis de malware evidencian que el enfoque propuesto, basado en la combinación de múltiples herramientas especializadas, es consistente con lo establecido en la literatura científica reciente. Diversos estudios destacan que el análisis efectivo del malware no puede depender de una sola técnica, sino que requiere la integración de enfoques estáticos, dinámicos, de red y forenses para obtener una visión completa del comportamiento malicioso.

En el caso del análisis estático, las herramientas utilizadas (PEiD, PEview, PE Explorer, BinText y Resource Hacker) permitieron identificar características estructurales relevantes del ejecutable. Sin embargo, los resultados también reflejan una limitación ampliamente documentada: la efectividad de este tipo de herramientas puede verse reducida ante técnicas avanzadas de ofuscación, empaquetamiento o polimorfismo. Esto coincide con lo señalado en los estudios revisados, donde se indica que el análisis estático, aunque rápido y seguro, puede no revelar completamente el comportamiento real del malware.

Por otro lado, el análisis dinámico, apoyado en herramientas como Process Monitor, Process Explorer, Regshot y Capture BAT, permitió observar cambios reales en el sistema durante la ejecución del malware. Los resultados obtenidos demuestran que este enfoque es fundamental para identificar acciones como modificaciones en el registro, creación de procesos y alteraciones en archivos. No obstante, también se identifican desafíos importantes, como la posibilidad de que el malware detecte entornos controlados o utilice técnicas anti-análisis para ocultar su comportamiento, lo cual ha sido ampliamente documentado en investigaciones recientes.

En relación con el análisis de red, el uso de Wireshark permitió identificar patrones de comunicación y tráfico sospechoso generado por el malware. Este tipo de análisis es clave para detectar conexiones con servidores de comando y control (C2), lo cual coincide con la literatura que posiciona el análisis de tráfico como un componente esencial en la identificación de campañas maliciosas y exfiltración de datos. Sin embargo, el uso de cifrado en las comunicaciones representa una limitación importante, ya que puede dificultar la interpretación del contenido del tráfico.

El análisis forense de memoria, mediante herramientas como DumpIt, Memoryze, Rekall y Volatility, demostró ser uno de los componentes más relevantes del análisis,

especialmente para detectar técnicas avanzadas como malware fileless o procesos ocultos. Los resultados coinciden con estudios que destacan la importancia del análisis de memoria para identificar amenazas que no dejan rastros en el sistema de archivos, lo cual representa una tendencia creciente en el desarrollo de malware moderno.

Adicionalmente, la integración de los resultados mediante el framework MITRE ATT&CK permitió contextualizar el comportamiento del malware dentro de tácticas y técnicas reales utilizadas por atacantes. Esto aporta un valor significativo al análisis, ya que no solo permite identificar qué hace el malware, sino también entender el “cómo” y el “por qué” desde una perspectiva estratégica. Tal como se indica en el documento base, este mapeo facilita la generación de inteligencia de amenazas y mejora la capacidad de respuesta ante incidentes.

En conjunto, la discusión evidencia que la propuesta planteada en el PIA01 logra integrar de manera efectiva herramientas y técnicas de análisis de malware, alineándose con las mejores prácticas del campo. No obstante, también se identifican desafíos relevantes, como la evasión de entornos virtualizados, el uso de técnicas anti-forenses y la creciente complejidad del malware. Estos hallazgos refuerzan la necesidad de continuar evolucionando los laboratorios de análisis, incorporando automatización, inteligencia artificial y entornos más realistas que permitan enfrentar amenazas cada vez más sofisticadas.

VI. CONCLUSIONES

El presente estudio permitió evidenciar que el análisis de malware constituye una disciplina fundamental dentro del campo de la ciberseguridad, especialmente en entornos Windows, donde la alta adopción de estos sistemas los convierte en un objetivo prioritario para los atacantes. A través del desarrollo de un laboratorio controlado y la aplicación de múltiples técnicas de análisis, se logró comprender de manera integral el comportamiento de muestras maliciosas y los mecanismos utilizados para comprometer sistemas.

1. Uno de los principales hallazgos de la investigación es que ninguna técnica de análisis, por sí sola, es suficiente para identificar completamente el comportamiento del malware. En este sentido, la combinación de análisis estático, ingeniería inversa, análisis dinámico, análisis de red y análisis forense de memoria permite obtener una visión más completa y precisa, facilitando la identificación de indicadores de compromiso y patrones de comportamiento malicioso.

2. Asimismo, los resultados demostraron que el uso de herramientas especializadas dentro de un entorno virtualizado es clave para garantizar un análisis seguro, controlado y reproducible. Estas herramientas permiten no solo observar el

comportamiento del malware, sino también correlacionar evidencias provenientes de diferentes fuentes, fortaleciendo el proceso de análisis y mejorando la calidad de los hallazgos obtenidos.

3. Por otra parte, se identificó que el análisis forense de memoria juega un papel crítico en la detección de amenazas avanzadas, como el malware sin archivo (fileless) y procesos ocultos, los cuales no dejan rastros en el disco y representan un desafío significativo para los métodos tradicionales de detección.

4. Adicionalmente, la integración de los resultados con el framework MITRE ATT&CK permitió contextualizar las acciones del malware dentro de tácticas y técnicas reales utilizadas por los atacantes, aportando un valor estratégico al análisis y facilitando la generación de inteligencia de amenazas.

5. No obstante, el estudio también evidenció diversas limitaciones, como la presencia de técnicas de evasión por parte del malware, incluyendo mecanismos anti-análisis, detección de entornos virtualizados y el uso de cifrado en comunicaciones, lo cual dificulta la observación completa de su comportamiento.

6. Finalmente, se concluye que es necesario continuar evolucionando los laboratorios de análisis de malware, incorporando nuevas tecnologías como la automatización, la inteligencia artificial y entornos más realistas, con el fin de enfrentar la creciente sofisticación de las amenazas. La metodología propuesta en este estudio representa una base sólida para fortalecer los procesos de análisis, detección y respuesta ante incidentes en sistemas operativos Windows.

VII. BIBLIOGRAFÍA

Bhat, S. (2013, November 19). *IDA Pro introduction*. SecPod Technologies. https://www.secpod.com/blog/introduction-to-ida-pro/

Collado, C. (s. f.). *Malware polimórfico y metamórfico: Diferencias y funcionamiento*. CyberBrainers. https://www.cyberbrainers.com/malware-polimorfico-y-metamorfico-diferencias-y-funcionamiento/

Contrast Security. (s. f.). *Code injection attacks: Identification and prevention*. https://www.contrastsecurity.com/glossary/code-injection

CrowdStrike Inc. (s. f.). *Hybrid analysis*. https://www.maltego.com/transform-hub/hybrid-analysis/

Fortinet. (s. f.-a). *¿Qué es el análisis de malware? Tipos y etapas del análisis de malware*. https://www.fortinet.com/lat/resources/cyberglossary/malware-analysis

Fortinet. (s. f.-b). *What is malware analysis? Types and stages of malware analysis*. https://www.fortinet.com/resources/cyberglossary/malware-analysis

Fox, N. (2023, April 6). *How to use Ghidra to reverse engineer malware*. Varonis. https://www.varonis.com/blog/how-to-use-ghidra

Fox, N. (2023, June 16). *PeStudio overview: Setup, tutorial and tips*. Varonis. https://www.varonis.com/blog/pestudio

Gillis, A. S. (2020, July 31). *What is static analysis (static code analysis)?* TechTarget. https://www.techtarget.com/whatis/definition/static-analysis-static-code-analysis

HxD. (s. f.). *HxD hex editor - Free download*. https://hxd.en.download.it/

Jain, N. (2023, July 7). *¿Qué es el diseño de investigación cualitativa? Definición, tipos, métodos y buenas prácticas*. IdeaScale. https://ideascale.com/es/blogs/diseno-de-investigacion-cualitativa/

Logit.io. (s. f.). *A guide to Cuckoo sandbox*. https://logit.io/blog/post/cuckoo-sandbox/

Maria, A., Gomez, M., Stancill, B., & Raabe, M. (2022, December 5). *FLARE VM: A FLAREytale open to the public*. Google Cloud. https://cloud.google.com/blog/topics/threat-intelligence/flarevm-open-to-public

Martínez, C. (2024, November 22). *Analyze and detect signatures in PE files easily: PEiD for Windows*. Uptodown. https://peid.en.uptodown.com/windows

Micucci, M. (2024, February 2). *La ofuscación de código: Un arte que reina en la ciberseguridad*. WeLiveSecurity. https://www.welivesecurity.com/es/recursos-herramientas/ofuscacion-de-codigo-arte-ciberseguridad/

Mitchell, A. (2025, May 3). *Google VirusTotal: Premium features, API, Chronicle, enterprise*. Lumifi Cyber. https://www.lumificyber.com/blog/google-virustotal/

mh-nexus. (s. f.). *Freeware programs*.
https://mh-nexus.de/en/programs.php

Oberhumer, M., Molnár, L., & Reiser, J. F. (s. f.). *UPX: The ultimate packer for executables*.
https://upx.github.io/

Oracle. (s. f.). *VirtualBox*.
https://www.oracle.com/virtualization/virtualbox/

Panay, P. (2021, June 24). *Introducing Windows 11*. Windows Experience Blog.
https://blogs.windows.com/windowsexperience/2021/06/24/introducing-windows-11/

Pistelli, E. (2012). *Explorer Suite*. NTCore.
https://ntcore.com/explorer-suite/

Security Onion Docs. (s. f.). *Zeek*.
https://docs.securityonion.net/en/2.4/zeek.html

Snort. (s. f.). *What is Snort?*
https://www.snort.org/

Suricata. (s. f.). *Suricata*.
https://suricata.io/our-story/suricata/

Williams, M. (s. f.). *Investigate suspect files with Exeinfo PE*. BetaNews.
https://betanews.com/2015/06/26/investigate-suspect-files-with-exeinfo-pe/

Zeltser, L. (s. f.). *REMnux*. SANS Institute.
https://www.sans.org/tools/remnux/

Talukder, S., & Talukder, Z. (2020). *A survey on malware detection and analysis tools*.
https://www.researchgate.net/publication/339816480

Al-Sofyani, S., Alelayani, A., Al-zahrani, F., & Monshi, R. (2023). *A survey of malware forensics analysis techniques and tools*. IEEE.
https://ieeexplore.ieee.org/document/10085474

Wong, M. Y., Landen, M., Antonakakis, M., Blough, D. M., Redmiles, E. M., & Ahamad, M. (2021). *An inside look into the practice of malware analysis*. ACM.
https://doi.org/10.1145/3460120.3484759

Amira, A., Derhab, A., Karbab, E. B., & Nouali, O. (2023). *A survey of malware analysis using community detection algorithms*. ACM.
https://doi.org/10.1145/3610223

Liu, S., Feng, P., Wang, S., Sun, K., & Cao, J. (2022). *Enhancing malware analysis sandboxes with emulated user behavior*. ScienceDirect.
https://www.sciencedirect.com/science/article/pii/S0167404822000128

Vasani, V., Bairwa, A. K., Joshi, S., Pljonkin, A., & Kaur, M. (2023). *Comprehensive analysis of advanced techniques and vital tools for detecting malware intrusion*. *Electronics, 12*(20), 4299.
https://www.mdpi.com/2079-9292/12/20/4299

Aminu, S. A., Sufyanu, Z., Sani, T., & Idris, A. (2020). *Evaluating the effectiveness of antivirus evasion tools against Windows platform*. *FUDMA Journal of Sciences*.
https://fjs.fudutsinma.edu.ng/index.php/fjs/article/view/27

Redwan, K., Akhter, J., & Sayed, A. S. (2024). *Forensic tools for Windows forensic analysis: A comprehensive review*. ResearchGate.