

Maestría en

CIBERSEGURIDAD

Trabajo Previo a la Obtención de Título de

Magister en Ciberseguridad

AUTORES:

Alberto André Carrasco Ruiz

Dennisse Mishel Guartan Salinas

María Fernanda Pacheco Tulcanaza

Jeniffer Judith Urquiza Olivo

Alexis Israel Veloz Jaya

TUTORES:

Iván Reyes Chacón

Alejandro Cortés López

TEMA:

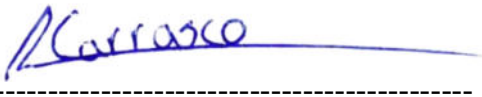
Perfil Mínimo de Detección en Windows Contra Ransomware Black

Basta: Diseño y Análisis en Laboratorio

Certificación de Autoría

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

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



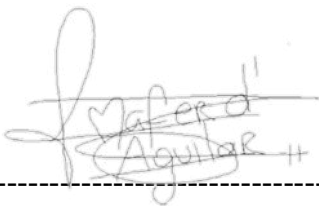
Firma

Alberto André Carrasco Ruiz



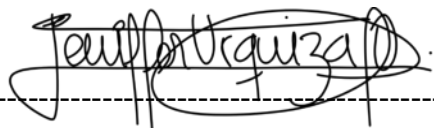
Firma

Dennisse Mishel Guartan Salinas



Firma

María Fernanda Pacheco Tulcanaza



Firma

Jeniffer Judith Urquiza Olivo



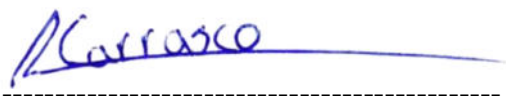
Firma

Alexis Israel Veloz Jaya

Autorización de Derechos de Propiedad Intelectual

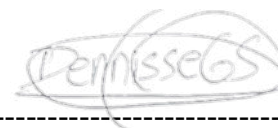
Nosotros, **Alberto André Carrasco Ruiz, Dennisse Mishel Guartan Salinas, María Fernanda Pacheco Tulcanaza, Jeniffer Judith Urquiza Olivo y Alexis Israel Veloz Jaya**, en calidad de autores del trabajo de investigación titulado ***Perfil Mínimo de Detección en Windows Contra Ransomware Black Basta: Diseño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

D. M. Quito, septiembre 2025.



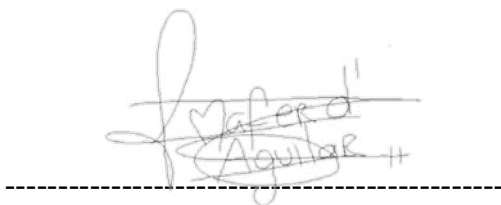
Firma

Alberto André Carrasco Ruiz



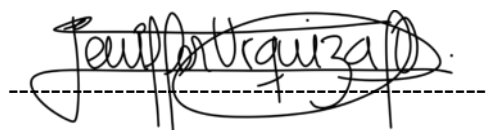
Firma

Dennisse Mishel Guartan Salinas



Firma

María Fernanda Pacheco Tulcanaza



Firma

Jeniffer Judith Urquiza Olivo

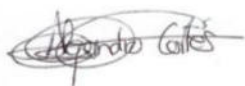
A handwritten signature in blue ink, appearing to read 'Alexis Israel Veloz Jaya', is positioned above a horizontal dashed line.

Firma

Alexis Israel Veloz Jaya

Aprobación de Dirección y Coordinación del Programa

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



Alejandro Cortés L.

Maestría en Ciberseguridad



Iván Reyes Ch.

Maestría en Ciberseguridad

Dedicatoria

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

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

Agradecimientos

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

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

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

Resumen

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

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

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

Abstract

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

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

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

Tabla de Contenido

Certificación de Autoría	i
Acuerdo de Confidencialidad	v
Aprobación de Dirección y Coordinación del Programa	vi
Dedicatoria.....	vii
Agradecimientos	viii
Resumen	ix
Abstract	x
Capítulo 1	33
1. Introducción.....	33
1.1. Definición del Proyecto	33
1.2. Justificación e Importancia del Trabajo de Investigación	33
1.3. Alcance	34
1.4. Objetivos.....	35
1.4.1. Objetivo General.....	35
1.4.2. Objetivo Específico.....	35
Capítulo 2	36
2. Revisión de Literatura	36
2.1. Estado del Arte.....	36
2.1.1. Ransomware Black Basta: Técnicas vs Evidencia en Laboratorio.....	36
2.1.2. Limitaciones Presentes en la Literatura Actual.	36
2.2. Marco Teórico.....	37
2.2.1. ¿Qué es un Ransomware?.	37

2.2.2. Ransomware Black Basta.....	38
2.2.2.1. Herramientas Utilizadas por Black Basta.	39
2.2.2.2. Cifrado de Archivos que Utiliza Black Basta.	40
2.2.2.3. Técnicas TTP Utilizadas por Black Basta.....	40
2.2.3. Artefactos en Windows.....	42
2.2.4. Artefactos de Black Basta.	42
2.2.5. El Análisis Forense en la Detección de Artefactos.....	43
2.2.6. Herramientas de Análisis Dinámico y Forense.	44
2.2.6.1. Virtual Box.....	44
2.2.6.2. Kali Linux.	44
2.2.6.3. Ubuntu.....	44
2.2.6.4. Sysmon.....	45
2.2.6.5. PowerShell.	45
2.2.6.6. Wireshark.	45
2.2.6.7. Velociraptor.....	45
2.2.6.8. Winpmem.....	46
2.2.6.9. Autoruns.....	46
2.2.6.10. Procmon.	46
2.2.6.11. Regshot.....	46
2.2.6.12. Volatility.	47

2.2.6.13. Autopsy.....	47
2.2.6.14. Reglas YARA.....	47
2.2.6.15. Windows Process Hacker.....	48
2.2.7. Herramientas de Análisis Estático.....	48
2.2.7.1. Detect It Easy.....	48
2.2.7.2. Ghidra.....	49
2.2.8. ¿Qué es “Baseline” y “Snapshot”?.....	49
2.2.9 Servidores Web y Arquitecturas de Manejo de Peticiones.....	50
2.2.9.1. Nginx.....	50
2.2.9.2. Fcgiwrap.....	50
Capítulo 3	51
3. Desarrollo	51
3.1. Desarrollo del Trabajo	51
3.1.1. Escenario Planteado Para Infección del Usuario.....	51
3.1.2. Topología del Entorno de Laboratorio.....	52
3.1.3. Componentes del Laboratorio y Justificación Técnica.....	53
3.1.3.1. VM OPBASTA - EvilWebServer01 (Servidor Malicioso).....	53
3.1.3.2. VM OPBASTA - ThreatAnalysis01 (Máquina de Análisis).....	55
3.1.3.3. VM OPBASTA - EmployeeHost01 (Máquina Víctima).....	56
3.1.4. Herramientas Implementadas.....	58
3.1.4.1. Scripts Utilizados Para Recolección de Evidencias.....	59

3.1.5. Muestras Black Basta.....	64
3.1.6. Medidas de Seguridad Implementadas.	65
3.1.7. Despliegue del Laboratorio.	66
3.1.7.1. VM OPBASTA - EvilWebServer01 (Servidor Malicioso).....	66
3.1.7.2. VM OPBASTA - ThreatAnalysis01 (Máquina de Análisis).....	76
3.1.7.3. VM OPBASTA - EmployeeHost01 (Máquina Víctima).	98
3.1.7.4. Portal “Operation Basta” - Plataforma Integrada de Distribución Segura de Muestras y Herramientas Forenses (Valor Agregado).....	117
3.2. Ejecución del Ransomware Black Basta en VM OPBASTA - EmployeeHost01	122
3.2.1. Preparación de VM OPBASTA - EmployeeHost01 Antes de la Ejecución del Ransomware.	122
3.2.2. Ejecución del Ransomware Black Basta.	126
3.3. Recolección de Evidencias Tras la Ejecución del Ransomware.....	128
Capítulo 4	133
4. Análisis de Resultados	133
4.1. Pruebas de Concepto	133
4.1.1. Análisis Estático.	133
4.1.1.1. Funcionamiento del Ransomware.	133
4.1.1.2. Revisión del Ransomware con la Herramienta Ghidra.....	135
4.1.2. Análisis Dinámico.....	138
4.1.2.1. Comportamiento en Tiempo Real de Black Basta.	138
4.2. Interpretación de Resultados.....	144

4.2.1. Análisis Estático.	144
4.2.1.1. Arquitectura General de Ejecución.....	145
4.2.1.2. Implementación Criptográfica con Enteros Grandes y RSA-Like.	145
4.2.1.3. Integración Criptográfica y Cifrado Híbrido.....	148
4.2.1.4. Inicialización de Parámetros Criptográficos y Claves Internas.	149
4.2.1.5. Enumeración de Volúmenes y Rutas en Windows.....	150
4.2.1.6. Planificación Multihilo y Cola de Trabajo.....	152
4.2.1.7. Rutina de Cifrado por Archivo.....	155
4.2.1.8. Manipulación del Wallpaper de Fondo.....	161
4.2.1.9. Manipulación de Servicios y Detección de Entornos Virtualizados. .	161
4.2.1.10. Creación de Readme.txt.	162
4.2.1.11. Métricas Finales, Tiempo y GB Cifrados.	164
4.2.2. Análisis Dinámico.....	165
4.2.2.1. Carga de Imágenes o Librerías Criptográficas.	165
4.2.2.2. Creación Anómala de Threads.	167
4.2.2.3. Manipulación del Wallpaper de Fondo.....	168
4.2.2.4. Descubrimiento del Entorno y Carpetas del Usuario.	168
4.2.2.5. Comprobaciones de Estado del Sistema SafeBoot, SmartLocker y Session Manager.....	169
4.2.2.6. Consultas a Image File Execution Options y Compatibilidad.	171

4.2.2.7. Actividad Enorme de Sistema de Archivos.	171
4.2.2.8. Creación y Acceso a Archivos Temporales en AppData\Local\Temp.	173
4.2.2.9. Enumeración del Sistema de Archivos.	174
4.2.2.10. Búsqueda de Archivos Sensibles.	176
4.2.2.11. Proceso de Encriptación de Archivos.	176
4.2.3. Análisis de Artefactos .json.	178
4.2.3.1. Elastic.EventLogs.Sysmon.json.	178
4.2.3.2. Windows.NTFS.Timestomp.json.	180
4.2.3.3. Windows.Packs.Persistence%2FStartup Items.json.	181
4.2.3.4. Generic.Forensic.Timeline.json.	182
4.2.3.5. Windows.Forensics.Lnk.json.	184
4.2.4. Identificación de Evidencias Adicionales y Verificación de Integridad.	185
4.2.5. Análisis de Archivos EVTX.	188
4.2.5.1. Microsoft-Windows-Sysmon/Operational.evtx.	189
4.2.6. Análisis de Archivos PML.	194
4.2.6.1. Adquisición de Evidencia Volátil.	195
4.2.7. Correlación de Hallazgos.	203
4.2.7.1. Identificación de Estrategia de Evasión.	203
4.2.7.2. Descubrimiento de la Fase de “Staging”.	204

4.2.7.3. Objetivo de Cifrado.	204
4.2.7.4. Fuerza Bruta en Notas de Rescate.	204
4.2.7.5. Reconstrucción del Flujo Táctico.	204
4.2.7.6. Alineación con el Marco MITRE ATT&CK.	205
4.2.8. Análisis de la Cobertura Táctica.	207
4.2.9. Análisis del Volcado de Memoria.	207
4.2.9.1. Identificación y Aislamiento del Proceso (Triage).	207
4.2.9.2. Ejecución de Comandos Para Descartar Falsos Positivos.	209
4.2.9.3. Análisis de Inyección de Código.	210
4.2.9.4. Extracción Forense del Espacio de Memoria.	211
4.2.10. Análisis de Otras Ejecuciones del Ransomware.	215
4.2.10.1 Análisis de archivo PML.	215
4.2.10.2. Fingerprinting del Sistema de Archivos.	216
4.2.10.3. Mutación de IOCs.	217
4.2.10.4. Matriz de Diferencias y Similitudes entre Variantes de Black Basta.	219
4.2.10.5. Análisis Forense de Memoria.	220
4.2.10.6. Análisis de Contenido de Memoria.	222
4.3. Perfil Mínimo de Black Basta.	225
4.3.1. Información General de Black Basta.	225

4.3.2. Comportamiento Observado.....	225
4.3.2.1. Procesos.	225
4.3.2.2. Modificaciones del Sistema.	225
4.3.2.3. Comportamiento del Cifrado.....	226
4.3.3. Funciones Utilizadas por Black Basta.	226
4.3.4. Actividades Registradas.	227
4.3.5. Artefactos Forenses.....	229
4.3.5.1. Lista de Artefactos Analizados.....	229
4.3.5.2. Artefactos Principales e Información Obtenida.	229
4.3.6. Técnicas MITRE ATT&CK.....	231
T1059.003.....	231
T1211.....	231
T1070.001.....	231
T1083.....	232
T1490.....	232
T1486.....	232
4.3.7. Propuesta de Perfil de Detección (IOAs).	232
4.3.7.1. Detección de Actividad Anti-Forense (Alta Fidelidad).	232
4.3.7.2. Detección de Evasión y Reconocimiento.....	233
4.3.7.3. Detección de Artefactos de “Staging” (Persistencia Visual).	235

4.3.8. Resumen del Perfil Defensivo.....	238
4.4 Falsos Positivos.....	238
Capítulo 5	240
5. Conclusiones y Recomendaciones	240
5.1. Conclusiones de Análisis Estático.....	240
5.2. Conclusiones de Análisis Dinámico.....	241
5.3. Análisis de Artefactos.....	242
5.4. Perfil de Detección Mínimo Black Basta	243
5.5. Recomendaciones Generales	244
Referencias	246
Apéndices	249

Lista de Tablas

Tabla 1 <i>Artefactos principales de Black Basta</i>	42
Tabla 2 <i>Herramientas implementadas en cada máquina virtual del Laboratorio</i>	58
Tabla 3 <i>Lista de Hash SHA-256 de cinco muestras de ransomware Black Basta Utilizadas</i>	64
Tabla 4 <i>Lista de Hash SHA-256 de Archivos EVTX del primer directorio</i>	185
Tabla 5 <i>Lista de Hash SHA-256 de Archivos EVTX del segundo directorio</i>	186
Tabla 6 <i>Lista de Hash SHA-256 de Archivos PML de Otras evidencias recopiladas</i>	188
Tabla 7 <i>Relación del comportamiento observado (eliminación de Shadow copies) con MITRE ATT&CK</i>	192
Tabla 8 <i>Relación del comportamiento observado (creación de notas de rescate) con MITRE ATT&CK</i>	193
Tabla 9 <i>Alineación del comportamiento observado con el marco MITRE ATT&CK</i>	205
Tabla 10 <i>Matriz de Diferencias y Similitudes entre Variantes de Black Basta</i>	219

Lista de Figuras

Figura 1 <i>Topología del Entorno de Laboratorio (VMs en Red Interna)</i>	52
Figura 2 <i>Selección de muestras de ransomware Black Basta desde el sitio web vx- underground.org</i>	65
Figura 3 <i>Instalación del sistema operativo Ubuntu para VM OPBASTA - EvilWebServer01</i>	66
Figura 4 <i>Instalación finalizada del sistema operativo Ubuntu para VM OPBASTA - EvilWebServer01</i>	67
Figura 5 <i>Instalación de herramienta nginx en VM OPBASTA - EvilWebServer01</i>	68
Figura 6 <i>Creación de directorio que aloja las muestras</i>	68
Figura 7 <i>Repositorio de muestras de Black Basta</i>	69
Figura 8 <i>Archivos de muestras transferidos de directorio</i>	69
Figura 9 <i>Archivos de muestras renombradas en el directorio creado</i>	70
Figura 10 <i>Creación de directorio destinado a la web maliciosa</i>	70
Figura 11 <i>Interfaz de Microsoft Teams a falsificar</i>	71
Figura 12 <i>Descarga en .html de la Interfaz de Microsoft Teams</i>	71
Figura 13 <i>Modificación del index.html de Microsoft Teams</i>	72
Figura 14 <i>Modificación del botón de descarga de Microsoft Teams</i>	72
Figura 15 <i>Creación del Script para la selección de las muestras aleatorias de Black Basta</i>	73
Figura 16 <i>Instalación de herramienta fcgiwrap</i>	73
Figura 17 <i>Reinicio del servicio nginx</i>	74
Figura 18 <i>Creación del servidor Web malicioso finalizado</i>	75
Figura 19 <i>Comprobación de descarga de la muestra Black Basta renombrada</i>	75

Figura 20	<i>Segunda comprobación de descarga de la muestra Black Basta renombrada</i>	76
Figura 21	<i>Descarga del sistema operativo Ubuntu para VM OPBASTA - ThreatAnalysis01</i>	76
Figura 22	<i>Configuración del modo de instalación del sistema operativo Ubuntu</i>	77
Figura 23	<i>Configuración de usuario y contraseña del SO Ubuntu</i>	77
Figura 24	<i>Configuración de características del SO Ubuntu</i>	78
Figura 25	<i>Proceso de instalación del SO Ubuntu</i>	78
Figura 26	<i>Mensaje de reinicio al culminar la instalación</i>	79
Figura 27	<i>Sistema operativo Ubuntu instalado en VM OPBASTA - ThreatAnalysis01</i>	79
Figura 28	<i>Instalación de dependencias para obtener mejoras en VM OPBASTA - ThreatAnalysis01</i>	80
Figura 29	<i>Continuación del proceso de instalación de dependencias</i>	80
Figura 30	<i>Instalación de mejoras de integración</i>	81
Figura 31	<i>Equipo listo para usarse</i>	81
Figura 32	<i>Actualización de paquetes en repositorios</i>	82
Figura 33	<i>Descarga de la última versión de Velociraptor para Linux</i>	83
Figura 34	<i>Redirección de la carpeta donde se invoca Velociraptor</i>	83
Figura 35	<i>Ejecución del binario de Velociraptor</i>	84
Figura 36	<i>Configuración del servidor de Velociraptor</i>	85
Figura 37	<i>Configuración de rutas, tiempo de expiración y otras opciones del servidor de Velociraptor</i>	86
Figura 38	<i>Configuración de características de red y puertos del servidor Velociraptor</i>	87
Figura 39	<i>Configuración de usuario y contraseña para autenticación en la GUI de Velociraptor</i>	88

Figura 40 Finalización de la configuración del servidor de Velociraptor	88
Figura 41 Comprobación de creación del archivo server.config.yaml de Velociraptor	89
Figura 42 Creación del servicio de systemd	89
Figura 43 Instalación del paquete .deb de Velociraptor	90
Figura 44 Inicialización de los servicios del servidor de Velociraptor	90
Figura 45 Interfaz gráfica de Velociraptor	91
Figura 46 Modificación del archivo de configuración del servidor de Velociraptor	91
Figura 47 Reinicio del servidor de Velociraptor	92
Figura 48 Creación del agente recolector de artefactos de Velociraptor.....	92
Figura 49 Descarga de la última versión de Ghidra desde el repositorio oficial	93
Figura 50 Descompresión del archivo descargado de Ghidra.....	94
Figura 51 Instalación de dependencias de Java	94
Figura 52 Establecimiento de la variable de entorno JAVA_HOME	94
Figura 53 Aceptación de acuerdo de usuario antes de ejecutar Ghidra.....	95
Figura 54 Ejecución de Ghidra en VM OPBASTA - ThreatAnalysis01	95
Figura 55 Interfaz de Ghidra abierta en VM OPBASTA - ThreatAnalysis01	96
Figura 56 Descarga de herramienta Detect It Easy	96
Figura 57 Instalación de dependencias libfuse2.....	97
Figura 58 Ejecución de Detect It Easy en VM OPBASTA - ThreatAnalysis01	98
Figura 59 Configuración de la máquina virtual Windows para VM OPBASTA - EmployeeHost01	99
Figura 60 Máquina virtual Windows VM OPBASTA - EmployeeHost01 instalada en VirtualBox	99

Figura 61 <i>Desactivación de protecciones de Windows en VM OPBASTA - EmployeeHost01</i> ..	100
Figura 62 <i>Desactivación de actualizaciones de Windows Update en VM OPBASTA - EmployeeHost01</i>	100
Figura 63 <i>Desactivación de Windows Defender desde el Editor de Registro</i>	101
Figura 64 <i>Desactivación del Firewall de Windows Defender</i>	102
Figura 65 <i>Descarga de herramienta Symon</i>	103
Figura 66 <i>Vista del archivo de Sysmon descomprimido</i>	103
Figura 67 <i>Descarga del archivo de configuración de Sysmon</i>	104
Figura 68 <i>Instalación de Sysmon de acuerdo con archivo de configuración</i>	104
Figura 69 <i>Verificación de instalación de Sysmon</i>	105
Figura 70 <i>Descarga de herramienta Regshot</i>	105
Figura 71 <i>Vista del archivo Regshot descomprimido</i>	106
Figura 72 <i>Descarga de herramienta Procmon</i>	106
Figura 73 <i>Vista del archivo Procmon descomprimido</i>	107
Figura 74 <i>Descarga de herramienta Process Hacker</i>	108
Figura 75 <i>Ejecución del archivo descargado processhacker-2.39-setup</i>	108
Figura 76 <i>Instalación de herramienta Process Hacker</i>	109
Figura 77 <i>Descarga de Velociraptor para Windows</i>	109
Figura 78 <i>Creación de servidor para transferencia de archivos entre VM ThreatAnalysis01 y VM EmployeeHost01</i>	110
Figura 79 <i>Vista del servidor con los archivos a descargar</i>	110
Figura 80 <i>Archivos descargados desde el servidor de transferencia</i>	111
Figura 81 <i>Instalación del cliente de Velociraptor</i>	111

Figura 82 <i>Reemplazo del archivo client.config.yaml</i>	112
Figura 83 <i>Visualización del cliente de Velociraptor desde la interfaz gráfica del servidor</i>	113
Figura 84 <i>Reporte entregado por el cliente de Velociraptor sobre la máquina Windows</i>	113
Figura 85 <i>Herramientas de análisis reubicadas en AnalysisTools de System32</i>	114
Figura 86 <i>Snapshot de VM OPBASTA - EvilWebServer01 previo a la ejecución del ransomware</i>	115
Figura 87 <i>Snapshot de VM OPBASTA - ThreatAnalysis01 previo a la ejecución del ransomware</i>	115
Figura 88 <i>Snapshot de VM OPBASTA - EmployeeHost01 previo a la ejecución del ransomware</i> <i>(parte 1)</i>	116
Figura 89 <i>Snapshot de VM OPBASTA - EmployeeHost01 previo a la ejecución del ransomware</i> <i>(parte 2)</i>	116
Figura 90 <i>Laboratorio completo con las tres máquinas</i>	117
Figura 91 <i>Portal “Operation Basta”- Plataforma Integrada de Distribución Segura de</i> <i>Muestras y Herramientas Forenses</i>	118
Figura 92 <i>Portal “Operation Basta”- Lista de muestras de malware disponibles a descargar</i>	119
Figura 93 <i>Portal “Operation Basta”- Sección de herramientas forenses disponibles</i>	120
Figura 94 <i>Portal “Operation Basta”- Sección de herramientas de ingeniería inversa disponibles</i>	120
Figura 95 <i>Vista del laboratorio completo y operativo</i>	121
Figura 96 <i>Verificación de que Sysmon se encuentra activo</i>	122
Figura 97 <i>Primer Snapshot desde Regshot antes de la ejecución del ransomware</i>	123
Figura 98 <i>Primer shot de Regshot</i>	123

Figura 99 <i>Finalización del primer shot Regshot</i>	124
Figura 100 <i>Primer shot de Regshot guardado en directorio dentro de System32</i>	124
Figura 101 <i>Ejecución de Procmon con permisos de administrador</i>	125
Figura 102 <i>Ejecución de Process Hacker con permisos de administrador</i>	126
Figura 103 <i>Ejecución de la muestra de Black Basta desde PowerShell</i>	126
Figura 104 <i>Vista de la ejecución de Black Basta y de la recepción de actividad en las herramientas</i>	127
Figura 105 <i>Archivos encriptados por Black Basta y nota de readme</i>	127
Figura 106 <i>Segundo shot posterior a la ejecución de Black Basta</i>	128
Figura 107 <i>Proceso de comparación de primer y segundo shot obtenidos</i>	128
Figura 108 <i>Registros de Procmon posterior a la ejecución del ransomware</i>	129
Figura 109 <i>Archivos obtenidos de las herramientas Regshot y Procmon</i>	129
Figura 110 <i>Selección del Collector en la interfaz gráfica de Velociraptor</i>	130
Figura 111 <i>Selección de artefactos que serán recolectados</i>	131
Figura 112 <i>Configuración de los parámetros del colector</i>	131
Figura 113 <i>Configuración de Specify Resources del colector</i>	132
Figura 114 <i>Revisión de los artefactos capturados por el colector</i>	132
Figura 115 <i>Análisis estático de la muestra de Black Basta con herramienta Detect It Easy (DIE)</i>	134
Figura 116 <i>Revisión de la entropía de la muestra de Black Basta en DIE</i>	134
Figura 117 <i>Revisión del porcentaje de empaquetado de la muestra de Black Basta en la herramienta DIE</i>	135
Figura 118 <i>Análisis de la muestra de Black Basta en la herramienta Ghidra</i>	136

Figura 119 Mensaje de confirmación del análisis de la muestra de Black Basta en la herramienta Ghidra	136
Figura 120 Inicio del análisis de la muestra de Black Basta en la herramienta Ghidra.....	136
Figura 121 Finalización del análisis de la muestra de Black Basta en la herramienta Ghidra	137
Figura 122 Código descompilado de Black Basta en la herramienta Ghidra	137
Figura 123 Estado inicial de los archivos antes de la ejecución de Black Basta.....	138
Figura 124 Revisión de archivo readme.txt generado por Black Basta.....	139
Figura 125 Encriptación de los archivos realizada por Black Basta	139
Figura 126 Archivos encriptados al finalizar la ejecución de Black Basta	140
Figura 127 Archivos renombrados tras la encriptación de Black Basta	140
Figura 128 Cambio de fondo de pantalla registrado tras la ejecución de Black Basta	141
Figura 129 Creación de archivo readme.txt en el escritorio.....	141
Figura 130 Revisión del archivo readme.txt generado por Black Basta	142
Figura 131 Revisión de los archivos que se encuentran en la papelería de reciclaje	142
Figura 132 Revisión del estado de los archivos que se encuentran en las carpetas de Windows	143
Figura 133 Revisión de los archivos que se encuentran en las subcarpetas de Windows	143
Figura 134 Mensaje de finalización de la ejecución de Black Basta con la cantidad de encriptados	144
Figura 135 Análisis del código de la muestra de Black Basta obtenidas de Ghidra.....	144
Figura 136 Revisión de código en Decompile: FUN_0041b250 en Ghidra	146
Figura 137 Revisión de código en Decompile: FUN_0041b540 en Ghidra	147
Figura 138 Revisión de código en Decompile: FUN_0041a410 en Ghidra	147

Figura 139 <i>Revisión de código en Decompile: FUN_0041bce0 en Ghidra</i>	148
Figura 140 <i>Revisión de código en Decompile: FUN_0040d160 en Ghidra</i>	149
Figura 141 <i>Análisis de código en Decompile: FUN_0040d160 en Ghidra</i>	150
Figura 142 <i>Revisión de código en Decompile: FUN_0040ca70 en Ghidra (parte 1)</i>	151
Figura 143 <i>Revisión de código en Decompile: FUN_0040ca70 en Ghidra (parte 2)</i>	151
Figura 144 <i>Revisión de código en Decompile: FUN_0040ca70 en Ghidra (parte 3)</i>	152
Figura 145 <i>Revisión de código en Decompile: FUN_0040b840 en Ghidra (parte 1)</i>	153
Figura 146 <i>Revisión de código en Decompile: FUN_0040b840 en Ghidra (parte 2)</i>	154
Figura 147 <i>Revisión de código en Decompile: FUN_0040b840 en Ghidra (parte 3)</i>	154
Figura 148 <i>Revisión de código en Decompile: FUN_0040bff0 en Ghidra (parte 1)</i>	155
Figura 149 <i>Revisión de código en Decompile: FUN_0040bff0 en Ghidra (parte 2)</i>	156
Figura 150 <i>Fragmento del código de la función FUN_0040bff0 donde se evaden loops de encriptación</i>	156
Figura 151 <i>Fragmento del código de la función FUN_0040bff0 para renombrar archivos (parte 1)</i>	156
Figura 152 <i>Fragmento del código de la función FUN_0040bff0 para renombrar archivos (parte 2)</i>	157
Figura 153 <i>Fragmento del código de la función FUN_0040bff0 para renombrar archivos (parte 3)</i>	157
Figura 154 <i>Fragmento del código de la función FUN_0040bff0 para validación de origen</i>	157
Figura 155 <i>Fragmento del código de la función FUN_0040bff0 para el renombre de archivos</i>	158
Figura 156 <i>Fragmento del código de la función FUN_0040d500</i>	158

Figura 157 Fragmento del código de la función FUN_0040bff0 para la generación de clave .	159
Figura 158 Fragmento del código de la función FUN_0040d130.....	159
Figura 159 Fragmento del código de la función FUN_0040bff0 donde llama a la función FUN_00415060	160
Figura 160 Fragmento del código de la función FUN_00415060.....	160
Figura 161 Fragmento del código de la función FUN_0040ae10.....	161
Figura 162 Revisión de código en Decompiler: FUN_00401180 en Ghidra	162
Figura 163 Fragmento del código de la función FUN_0040b090.....	163
Figura 164 Detalle del código de la función FUN_0040b090	163
Figura 165 Fragmento del código de la función FUN_0040ae10.....	164
Figura 166 Fragmento del código de la función FUN_0040ae10 que muestra la cantidad de encriptados	164
Figura 167 Proceso de carga de imágenes y librerías criptográficas por parte de la muestra.	166
Figura 168 Carga de imágenes y creación de threads.....	167
Figura 169 Llamado al módulo de Desktop Wallpaper	168
Figura 170 Procesos de lectura sobre claves de Explorer y del perfil	169
Figura 171 Consultas de claves del sistema por parte del ransomware.....	170
Figura 172 Comprobación de llamado a Image File Executions Options	171
Figura 173 Análisis realizado con la herramienta Procmon	172
Figura 174 Total de eventos registrados por Procmon.....	172
Figura 175 Registros de creación y acceso a archivos temporales (parte 1)	173
Figura 176 Registros de creación y acceso a archivos temporales (parte 2)	173
Figura 177 Creación y cierre de archivos en unidades de disco detectadas	175

Figura 178 Registro de búsqueda del ransomware de archivos que cifrar	175
Figura 179 Registros de procesos de encriptación de archivos	177
Figura 180 Registros de procesos de encriptación de archivos	177
Figura 181 Registros de eventos en Sysmon (parte 1)	178
Figura 182 Registros de eventos en Sysmon (parte 2)	178
Figura 183 Registros de eventos en Sysmon por ejecución del Binario	179
Figura 184 Registros de eventos ID 13 en Sysmon por ejecución del Binario.....	179
Figura 185 Registro de evento ID 5 en Sysmon por terminación de proceso	179
Figura 186 Registro de eventos en artefacto Windows.NTFS.Timestamp extraído por Velociraptor.....	180
Figura 187 Registro de eventos en artefacto Windows.Packs.Persistence%2FStartup Items ...	182
Figura 188 Registro de eventos en artefacto Generic.Forensic.Timeline.....	183
Figura 189 Registro de eventos en artefacto Windows.Forensics.Lnk	184
Figura 190 Eventos ID 1 y 11 capturados por herramienta Sysmon generados por Black Basta	189
Figura 191 Revisión del detalle de los eventos generados	190
Figura 192 Registros de la creación de notas de rescate obtenidas de Sysmon	193
Figura 193 Análisis de archivo procmon_20251114.pml filtrado por blackbasta.exe.....	195
Figura 194 Análisis de archivo procmon_20251114_121545.pml filtrado por Process Start ..	196
Figura 195 Análisis de archivo procmon_20251114_121545.pml filtrado por Detail contains vssadmin	197
Figura 196 Análisis de archivo procmon_20251114_121545.pml llamadas para eliminar Volume shadow Copies	197

Figura 197 <i>Análisis de archivo procmon_20251114_121545.pml filtrado por Temp y por WriteFile.....</i>	198
Figura 198 <i>Análisis de archivo procmon_20251114_121545.pml filtrado por Temp.....</i>	198
Figura 199 <i>Análisis de resultados del archivo procmon_20251114_121545.pml tras creación de readme.txt</i>	199
Figura 200 <i>Actividad sostenida de creación de notas de rescate (readme.txt) paralela al proceso de cifrado.....</i>	200
Figura 201 <i>Análisis del archivo procmon_20251114_121545.pml por corrupción de evidencias</i>	202
Figura 202 <i>Análisis del volcado de memoria en Volatility3 mediante el uso del plugin Windows.pslist.....</i>	208
Figura 203 <i>Identificación del PID 956 del proceso blackbasta.exe en el volcado de memoria.</i>	208
Figura 204 <i>Validación del origen del PID 956 en el volcado de memoria usando Volatility3..</i>	209
Figura 205 <i>Verificación de persistencia a la red mediante el uso del plugin Windows.netscan</i>	210
Figura 206 <i>Extracción del espacio de memoria mediante el plugin Windows.memmap</i>	211
Figura 207 <i>Generación del archivo pid.956.dmp</i>	211
Figura 208 <i>Procesamiento del archivo de volcado mediante el uso del comando findstr</i>	212
Figura 209 <i>Hallazgo de la ruta del ejecutable dentro del archivo analizado.....</i>	213
Figura 210 <i>Hallazgo de referencias explícitas a biblioteca rmclient.dll dentro del archivo analizado.....</i>	214
Figura 211 <i>Identificación de accesos a directorio C:\Logs</i>	215
Figura 212 <i>Identificación de operaciones de consulta sobre el directorio C:\BOOTNXT</i>	216
Figura 213 <i>Identificación de interacciones con Windows Error Reporting.....</i>	217

Figura 214 Identificación de operaciones SetRenameInformationFile.....	217
Figura 215 Identificación de operaciones QueryStandardInformationFile.....	218
Figura 216 Identificación de procesos activos a través del plugin windows.pslist	221
Figura 217 Análisis de la Tabla de Importación de Direcciones utilizando el plugin windows.dlllist.....	221
Figura 218 Identificadores encontrados en strings	223

Capítulo 1

1. Introducción

1.1. Definición del Proyecto

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

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

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

1.2. Justificación e Importancia del Trabajo de Investigación

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

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

1.3. Alcance

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

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

1.4. Objetivos

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

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

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

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

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

Capítulo 2

2. Revisión de Literatura

2.1. *Estado del Arte*

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

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

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

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

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

2.2. Marco Teórico

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

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

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

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

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

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

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

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

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

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

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

2.2.2.3. Técnicas TTP Utilizadas por Black Basta.

- **T1059.003 - Command and Scripting Interpreter: Windows Command Shell.-** Esta técnica es identificada porque el atacante usa el intérprete de comandos del sistema operativo, como por ejemplo cmd.exe para poder enviar instrucciones de forma directa. Mediante el intérprete lanza comandos que modifican el sistema, eliminan copias de seguridad, cambian archivos de direcciones o ejecutan programas. Se usa porque genera menor sospecha y evade soluciones de antivirus y EDR (Command and Scripting Interpreter: Windows Command Shell (T1059.003). MITRE ATT&CK., 2024).
- **T1562.001 - Impair Defenses: Disable or Modify Tools.-** Con esta técnica las herramientas de seguridad son deshabilitadas o modificadas para así evadirlas. El atacante busca modificar claves de registro, usar binarios legítimos, detener servicios. En este caso para Black Basta intenta deshabilitar Windows Defender. Con la aplicación de esta técnica se aumenta la probabilidad de que el ransomware logre cifrar archivos sin ser detectado, y reduce la capacidad de recuperación o intervención del sistema (Brumm, 2021).
- **T1204 – User Execution.-** Para comenzar con la infección es necesario que el usuario ejecute de forma manual un archivo o aplicación maliciosa. Esta técnica usa ingeniería social con el objetivo de que la víctima descargue el programa y lo ejecute. Black Basta utiliza

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

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

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

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

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

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

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

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

Tabla 1

Artefactos principales de Black Basta

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Capítulo 3

3. Desarrollo

3.1. Desarrollo del Trabajo

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

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

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

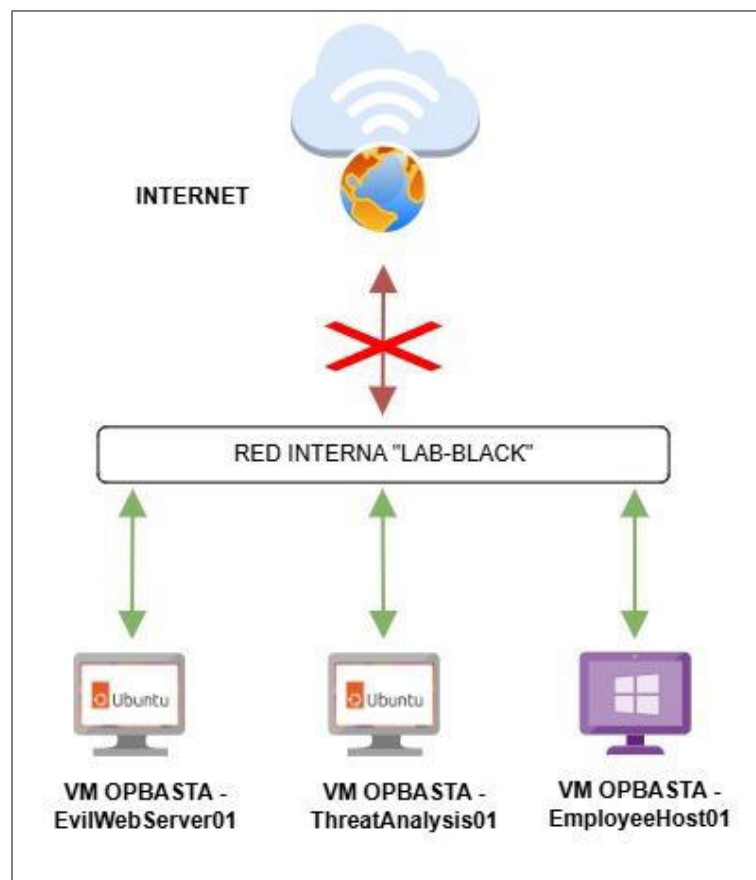
Cuando el usuario da clic en “Descargar”, obtiene un ejecutable que parece legítimo, pero que corresponde a una muestra del ransomware Black Basta. Al ejecutar el archivo descargado, los siguientes procesos maliciosos se despliegan en segundos: eliminación de copias de volumen (T1490), modificación de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

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

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

Figura 1

Topología del Entorno de Laboratorio (VMs en Red Interna)



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Tabla 2

Herramientas implementadas en cada máquina virtual del Laboratorio

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

Process Hacker
Velociraptor (Agente)
Winpmem

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

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

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

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

```
$procmon = "C:\Tools\Procmon.exe"

if (-not (Test-Path $procmon)) {

    Write-Error "Procmon.exe no encontrado en $procmon. Ajusta la
ruta."

    exit 1
}
```

```

    }

    $time = (Get-Date).ToString("yyyyMMdd_HH:mm:ss")

    $outPml = "\\10.10.10.20\forensic\procmon_$time.pml"

    # Start Procmon capturing quietly and write to backing file

    Start-Process -FilePath $procmon -ArgumentList

"/BackingFile",$outPml,"/AcceptEula","/Quiet" -WindowStyle Hidden -PassThru |

Out-Null

    Start-Sleep -Milliseconds 500

    Write-Output "Procmon started. Run the ransomware NOW. Will stop
capture automatically in 60 seconds."

    # Wait enough time for ransomware to run and generate activity

    Start-Sleep -Seconds 60

    # Stop Procmon (safe) and ensure file flushed

    Start-Process -FilePath $procmon -ArgumentList "/Terminate" -Wait -

WindowStyle Hidden

    Write-Output "Procmon stopped. PML should be at $outPml"

```

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

```

# Configuración de Camuflaje y Red

$SafeToolPath = "C:\Windows\sysdump.exe"

```

```
$MalwareName = "blackbasta"

$RemoteShare = "\\10.10.10.20\forensic"

# Definición de API Nativa para Suspensión de Hilos (P/Invoke)

$code = @"

using System;

using System.Runtime.InteropServices;

public class Utils {

    [DllImport("kernel32.dll")] public static extern IntPtr
OpenThread(int dwDesiredAccess, bool bInheritHandle, int dwThreadId);

    [DllImport("kernel32.dll")] public static extern int
SuspendThread(IntPtr hThread);

    [DllImport("kernel32.dll")] public static extern int
CloseHandle(IntPtr hObject);

}

"@

Add-Type -TypeDefinition $code

# Bucle de Detección y Respuesta

while ($true) {

    $target = Get-Process -Name $MalwareName -ErrorAction
SilentlyContinue

    if ($target) {

        $id = $target.Id

        # VENTANA CRÍTICA: 1.2 Segundos

        # Tiempo calculado para permitir el "Unpacking" en memoria

        Start-Sleep -Milliseconds 1200

        # 1. Congelamiento del Malware (Stop Execution)

        $target.Threads | ForEach-Object {

            $hThread = [Utils]::OpenThread(0x0002, $false, $_.Id)

            [Utils]::SuspendThread($hThread)

        }

    }

}
```

```

        [Utils]::CloseHandle($hThread)
    }

    # 2. Conexión Segura y Volcado
    net use Z: $RemoteShare /user:Guest "" | Out-Null

    $out = "Z:\MEM_1200ms_$(Get-Date -Format 'HHmmss').raw"

    Start-Process -FilePath $SafeToolPath -ArgumentList "$out" -

Wait

    break
}

Start-Sleep -Milliseconds 50
}

```

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

```

$sample = "C:\Users\isra\Downloads\m1\blackbasta.exe"

$destRoot = "\\10.10.10.20\forensic"

$timestamp = Get-Date -Format "yyyyMMdd_HHmmss"

$dest = Join-Path $destRoot "evtx_$timestamp"

New-Item -ItemType Directory -Path $dest -Force | Out-Null

# Logs que se quieren obtener, pero con búsqueda flexible

$targets = @(
    "Application",
    "Security",
    "System",

```

```

        "Sysmon",
        "PowerShell"
    )

Write-Host "Detectando nombres reales de logs..."

$available = wevtutil el

$resolved = @{}

foreach ($t in $targets) {
    $match = $available | Where-Object { $_ -match $t }

    if ($match) {
        Write-Host "? Encontrado para '$t': $match"

        $resolved[$t] = $match
    } else {
        Write-Host "? No existe ningún log que coincida con '$t'"
    }
}

Write-Host "`nEjecutando BlackBasta..."

Start-Process $sample

Start-Sleep -Seconds 8

Write-Host "`nExportando logs existentes..."

foreach ($key in $resolved.Keys) {
    foreach ($logName in $resolved[$key]) {

        $san = $logName.Replace("/", "_")

        $outfile = Join-Path $dest "$san.evtx"

        try {
            wevtutil epl "$logName" "$outfile"

            Write-Host "? Exportado -> $logName"
        }
    }
}

```

```

        catch {
            Write-Host "? Falló -> $logName"
        }
    }
}

Write-Host "`nExportación completada en: $dest"

```

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

Tabla 3

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

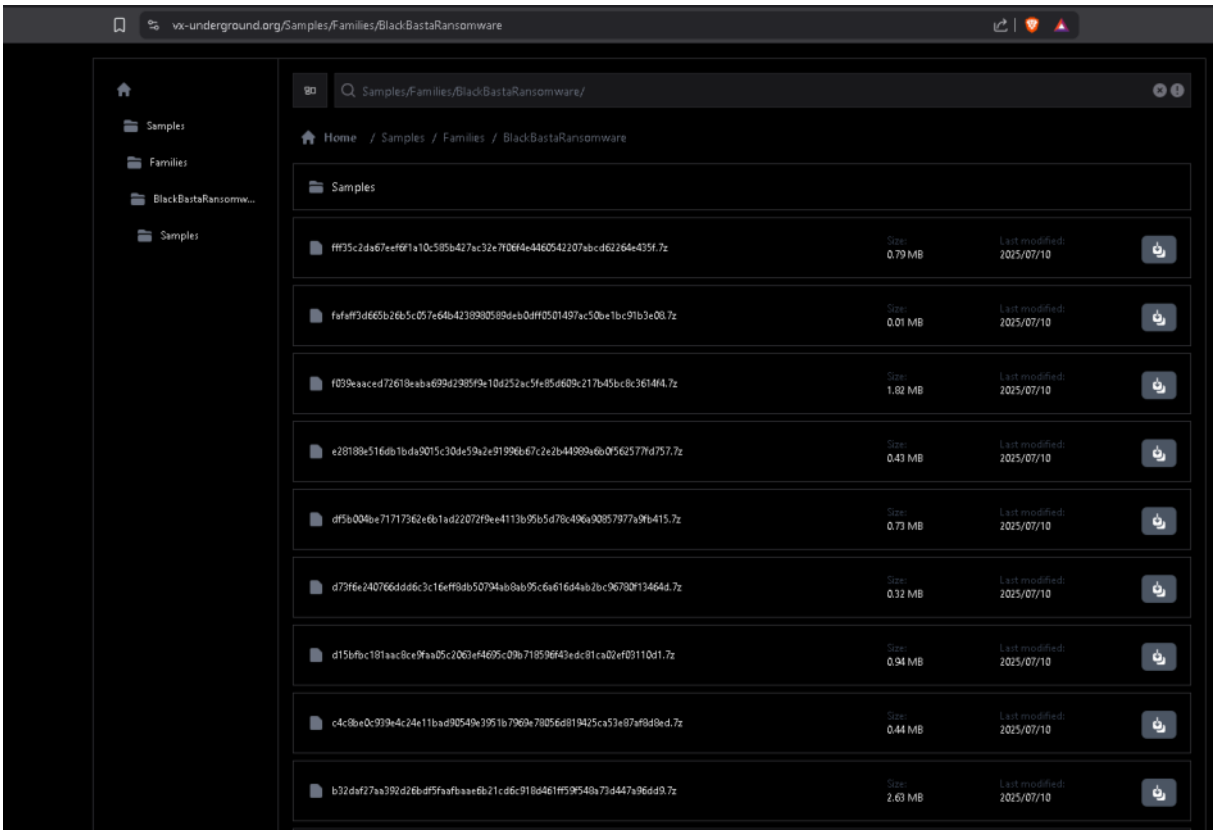
Nº Muestra	Hash SHA-256 de Muestra Del Ransomware Black Basta
M1	df5b004be71717362e6b1ad22072f9ee4113b95b5d78c496a90857977a9fb415
M2	e28188e516db1bda9015c30de59a2e91996b67c2e2b44989a6b0f562577fd757
M3	f039eaaced72618eaba699d2985f9e10d252ac5fe85d609c217b45bc8c3614f4
M4	fafaff3d665b26b5c057e64b4238980589deb0dff0501497ac50be1bc91b3e08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

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

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

Figura 2

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



3.1.6. Medidas de Seguridad Implementadas.

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

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

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

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

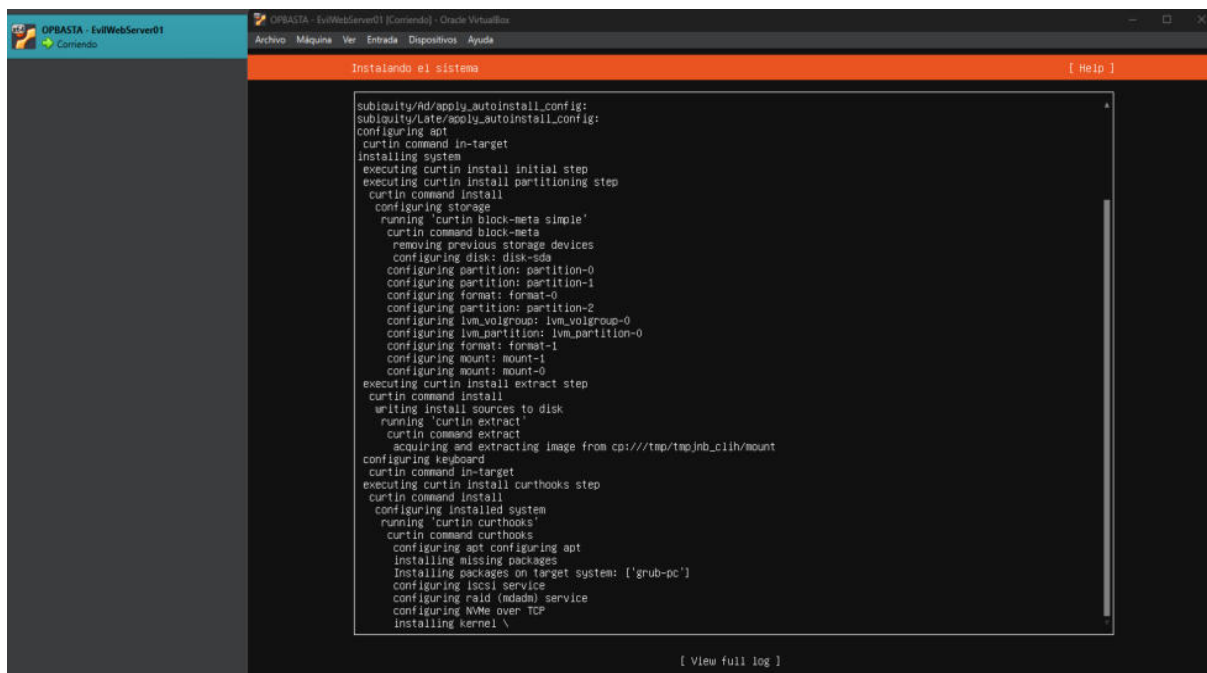
3.1.7. Despliegue del Laboratorio. Como primera instancia, se crearon las máquinas virtuales a utilizar, ya descritas en apartados anteriores.

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

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

Figura 3

Instalación del sistema operativo Ubuntu para VM OPBASTA - EvilWebServer01



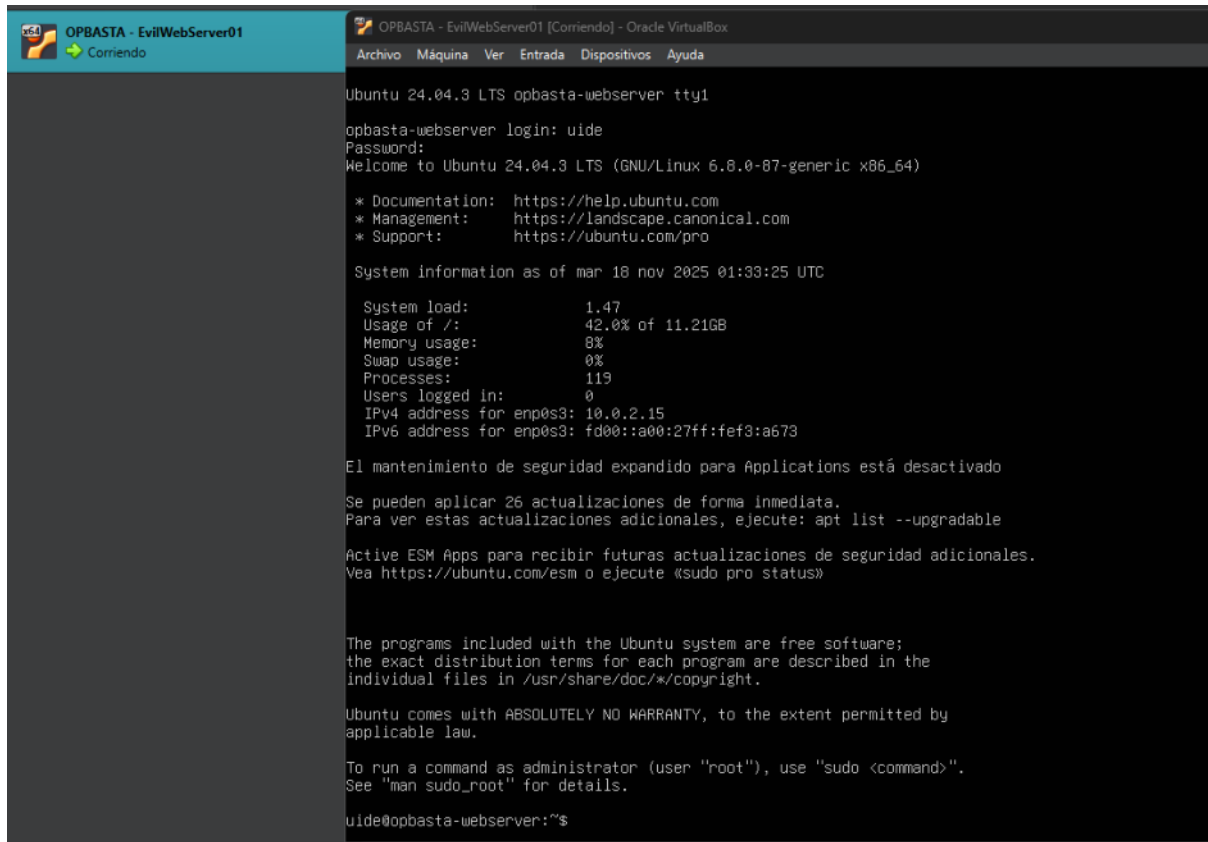
```
OPBASTA - EvilWebServer01 [Contenedor] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Instalando el sistema [ Help ]

subiquity/Ad/apply_autoinstall_config:
subiquity/Late/apply_autoinstall_config:
configuring apt
curtin command in-target
installing system
executing curtin install initial step
executing curtin install partitioning step
curtin command install
configuring storage
running 'curtin block-meta simple'
curtin command block-meta
removing previous storage devices
configuring disk: disk-0da
configuring partition: partition-0
configuring partition: partition-1
configuring format: format-0
configuring partitions partition-2
configuring lvm_voigroup: lvm_voigroup-0
configuring lvm_partition: lvm_partition-0
configuring format: format-1
configuring mount: mount-1
configuring mount: mount-0
executing curtin install extract step
curtin command install
writing install sources to disk
running 'curtin extract'
curtin command extract
acquiring and extracting image from cp:///tmp/jnb_cli/mount
configuring keyboard
curtin command in-target
executing curtin install curthooks step
curtin command install
configuring installed system
running 'curtin curthooks'
curtin command curthooks
configuring apt configuring apt
installing missing packages
Installing packages on target system: ['grub-pc']
configuring iscsi service
configuring raid (mdadm) service
configuring Nfme over TCP
installing kernel \
```

Figura 4

Instalación finalizada del sistema operativo Ubuntu para VM OPBASTA - EvilWebServer01



```
OPBASTA - EvilWebServer01 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Ubuntu 24.04.3 LTS opbasta-webserver tty1

opbasta-webserver login: uide
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of mar 18 nov 2025 01:33:25 UTC

System load:          1.47
Usage of /:            42.0% of 11.21GB
Memory usage:         8%
Swap usage:           0%
Processes:            119
Users logged in:      0
IPv4 address for enp0s3: 10.0.2.15
IPv6 address for enp0s3: fd00:a00:27ff:fe3:a673

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 26 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

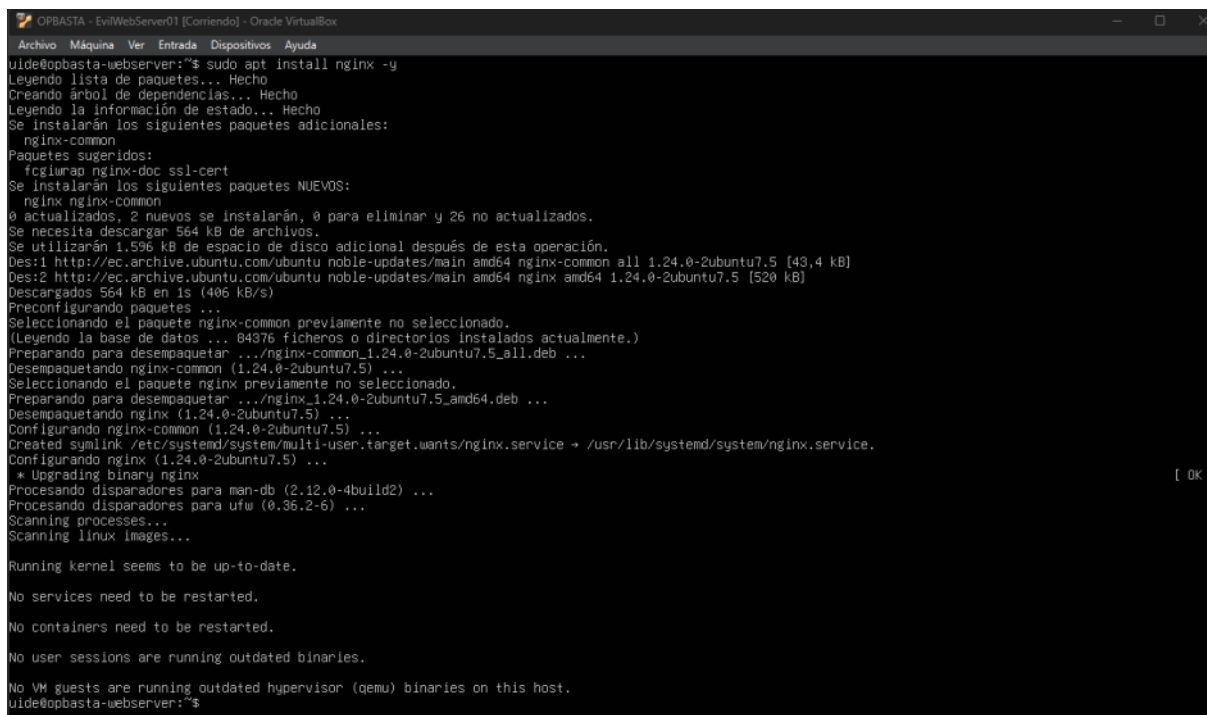
uide@opbasta-webserver:~$
```

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

```
sudo apt update
sudo apt install nginx -y
```

Figura 5

Instalación de herramienta nginx en VM OPBASTA - EvilWebServer01



```
OPBASTA - EvilWebServer01 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
uide@opbasta-webserver:~$ sudo apt install nginx -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  nginx-common
Paquetes sugeridos:
  fcgiwrap nginx-doc ssl-cert
Se instalarán los siguientes paquetes NUEVOS:
  nginx nginx-common
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 26 no actualizados.
Se necesita descargar 564 kB de archivos.
Se utilizarán 1.596 kB de espacio de disco adicional después de esta operación.
Des1: http://ec.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.5 [43,4 kB]
Des2: http://ec.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.5 [520 kB]
Descargados 564 kB en 1s (406 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete nginx-common previamente no seleccionado.
(Leyendo la base de datos ... 84376 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../nginx-common_1.24.0-2ubuntu7.5_all.deb ...
Desempaquetando nginx-common (1.24.0-2ubuntu7.5) ...
Seleccionando el paquete nginx previamente no seleccionado.
Preparando para desempaquetar .../nginx_1.24.0-2ubuntu7.5_amd64.deb ...
Desempaquetando nginx (1.24.0-2ubuntu7.5) ...
Configurando nginx-common (1.24.0-2ubuntu7.5) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Configurando nginx (1.24.0-2ubuntu7.5) ...
 * Upgrading binary nginx
Procesando disparadores para man-db (2.12.0-4build2) ...
Procesando disparadores para ufw (0.36.2-6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

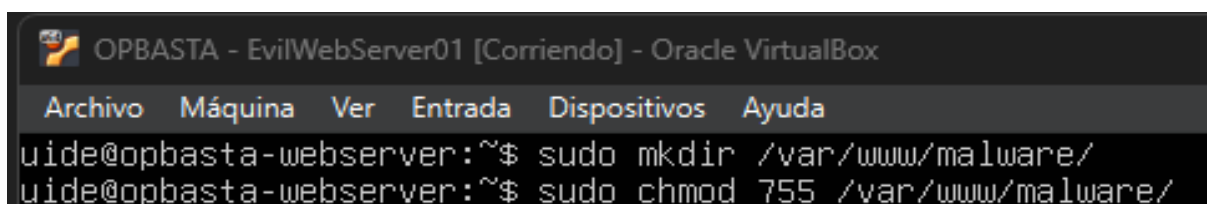
No VM guests are running outdated hypervisor (qemu) binaries on this host.
uide@opbasta-webserver:~$
```

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

```
sudo mkdir -p /var/www/malware/
sudo chmod 755 /var/www/malware/
```

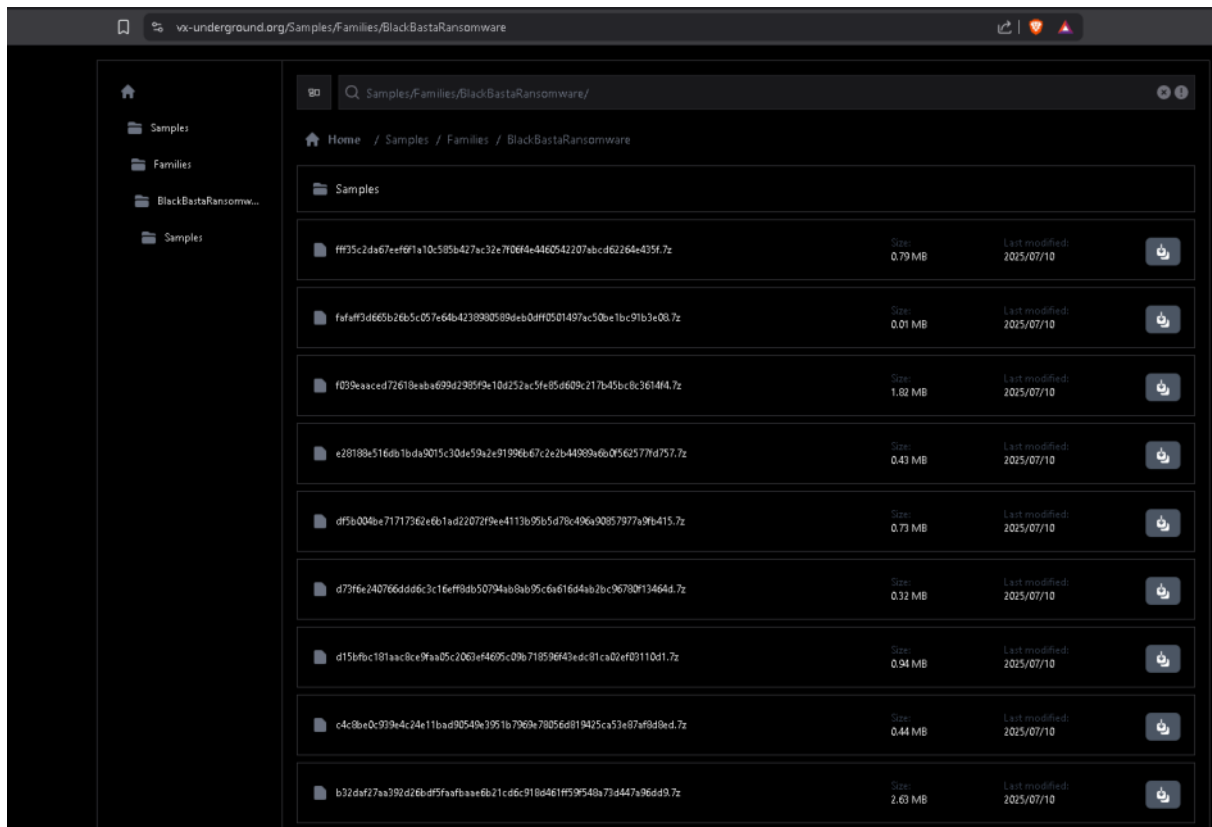
Figura 6

Creación de directorio que aloja las muestras



```
OPBASTA - EvilWebServer01 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
uide@opbasta-webserver:~$ sudo mkdir /var/www/malware/
uide@opbasta-webserver:~$ sudo chmod 755 /var/www/malware/
```

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

Figura 7*Repositorio de muestras de Black Basta*

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

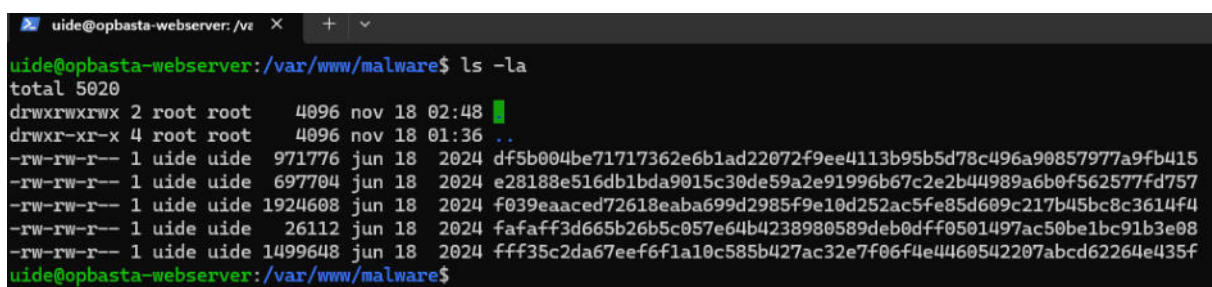
Figura 8*Archivos de muestras transferidos de directorio*

Figura 9

Archivos de muestras renombradas en el directorio creado

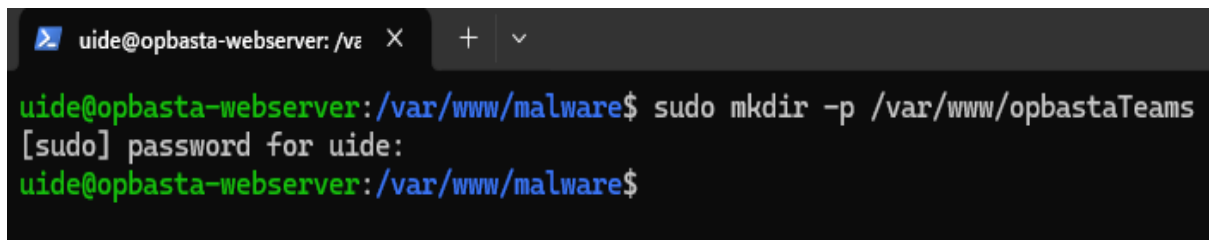
```
uide@opbasta-webserver:/var/www/malware$ mv df5b004be71717362e6b1ad22072f9ee4113b95b5d78c496a90857977a9fb415 TeamsOffline.exe
uide@opbasta-webserver:/var/www/malware$ mv e28188e516db1bda9015c30de59a2e91996b67c2e2b44989a6b0f562577fd757 MicrosoftTeams.exe
uide@opbasta-webserver:/var/www/malware$ mv f039eaaced72618eaba699d2985f9e10d252ac5fe85d609c217b45bc8c3614f4 MSTeamsInstaller.exe
uide@opbasta-webserver:/var/www/malware$ mv fafaaff3d665b26b5c057e64b4238980589deb0dfff0501497ac50be1bc91b3e08 MSTeamsOnline.exe
uide@opbasta-webserver:/var/www/malware$ mv fff35c2da67eef6f1a10c585b427ac32e7f06f4e4460542207abcd62264e435f MSTeams_Setup.exe
uide@opbasta-webserver:/var/www/malware$ ls -la
total 5020
drwxrwxrwx 2 root root 4096 nov 18 02:52
drwxr-xr-x 4 root root 4096 nov 18 01:36 ..
-rw-rw-r-- 1 uide uide 697704 jun 18 2024 MicrosoftTeams.exe
-rw-rw-r-- 1 uide uide 1924608 jun 18 2024 MSTeamsInstaller.exe
-rw-rw-r-- 1 uide uide 26112 jun 18 2024 MSTeamsOnline.exe
-rw-rw-r-- 1 uide uide 1499648 jun 18 2024 MSTeams_Setup.exe
-rw-rw-r-- 1 uide uide 971776 jun 18 2024 TeamsOffline.exe
```

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

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

Figura 10

Creación de directorio destinado a la web maliciosa

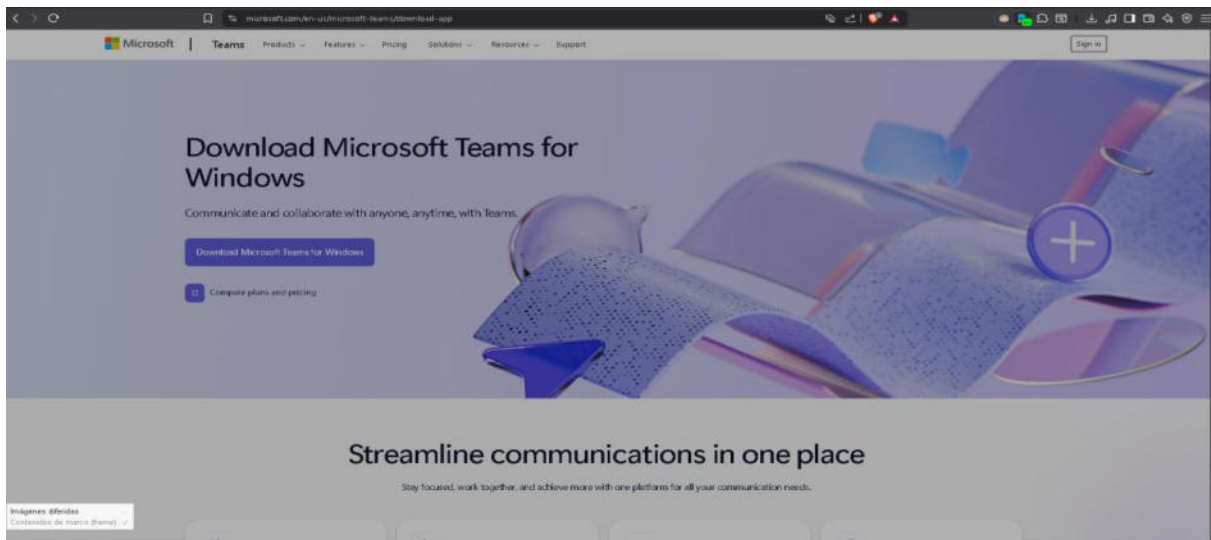


```
uide@opbasta-webserver: /var/www/malware$ sudo mkdir -p /var/www/opbastaTeams
[sudo] password for uide:
uide@opbasta-webserver: /var/www/malware$
```

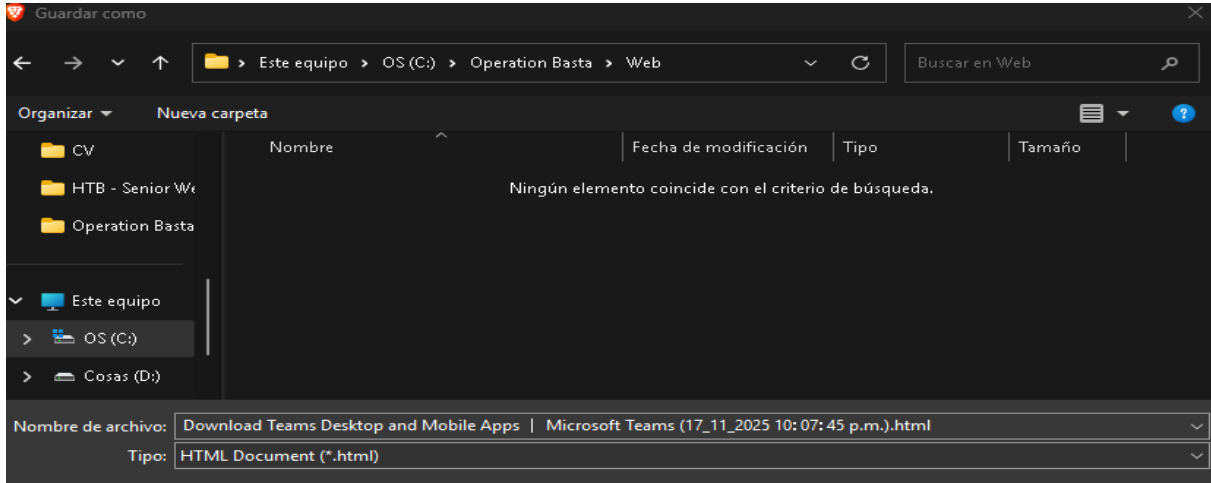
Se usó la extensión “SingleFile” disponible en Chrome para descargar una versión idéntica del front de la página web a falsificar; en este laboratorio se usó la interfaz de descarga de Microsoft Teams. Una vez que se descargó el index.html, se movió la ubicación hacia el directorio raíz de la web.

Figura 11

Interfaz de Microsoft Teams a falsificar

**Figura 12**

Descarga en .html de la Interfaz de Microsoft Teams



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

Figura 13*Modificación del index.html de Microsoft Teams*

```

</header>
</div>
</div>
</div>
<statics.teams.cdn.office.net/production-windows-x86/lkg/MSTeamsSetup.exe data-bi-sn=1 aria-label="Download Microsoft Teams for Windows" data-
<span class=popover__content>
<span class=popover__label>Try another option if you're having trouble with the QR code</span>

<button class=popover__info id=34ff6baa-120f-458d-95f6-e8a91ce6f0c0 aria-controls=popover-window-34ff6baa-120f-458d-95f6-e8a91ce6f0c0 data-
<div class="ocr-icon ocr-icon--size-xs ocr-icon-svg--info-filled" aria-hidden=true role=presentation>
<svg viewBox="0 0 24 24" xmlns=http://www.w3.org/2000/svg role=img alt>
<title class=sr-only>More information</title>

```

Figura 14*Modificación del botón de descarga de Microsoft Teams*

```

-ct="Component Button" id=action-oc21be href=/download data-
popover data-placement=right aria-expanded=false aria-haspopup

```

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

El código que contiene el script es el siguiente:

```

#!/usr/bin/env bash

MALWARE_DIR="/var/www/malware"

# Seleccionar archivo aleatorio

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

FULLPATH="$MALWARE_DIR/$FILE"

# Enviar cabeceras CGI válidas

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

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

echo ""

# Enviar el contenido binario

```

```
cat "$FULLPATH"

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

Figura 15

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

```
uide@opbasta-webserver:/var/www/opbastateams$ nano /usr/local/bin/random_basta.sh
uide@opbasta-webserver:/var/www/opbastateams$ sudo nano /usr/local/bin/random_basta.sh
uide@opbasta-webserver:/var/www/opbastateams$ sudo chmod +x /usr/local/bin/random_basta.sh
```

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

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

```
sudo apt install fcgiwrap

sudo systemctl enable fcgiwrap

sudo systemctl start fcgiwrap
```

Figura 16

Instalación de herramienta fcgiwrap

```
uide@opbasta-webserver:/var/www/opbastateams$ sudo apt install fcgiwrap
sudo systemctl enable fcgiwrap
sudo systemctl start fcgiwrap
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libfcgi-bin libfcgi0t64
Se instalarán los siguientes paquetes NUEVOS:
  fcgiwrap libfcgi-bin libfcgi0t64
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 25 no actualizados.
Se necesita descargar 55,9 kB de archivos.
Se utilizarán 193 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

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

El código que contiene el script es el siguiente:

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

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

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

Figura 17

Reinicio del servicio nginx



```
uide@opbasta-webserver:/var/www/opbastateams$ sudo nginx -t  
sudo systemctl reload nginx  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

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

Figura 18

Creación del servidor Web malicioso finalizado

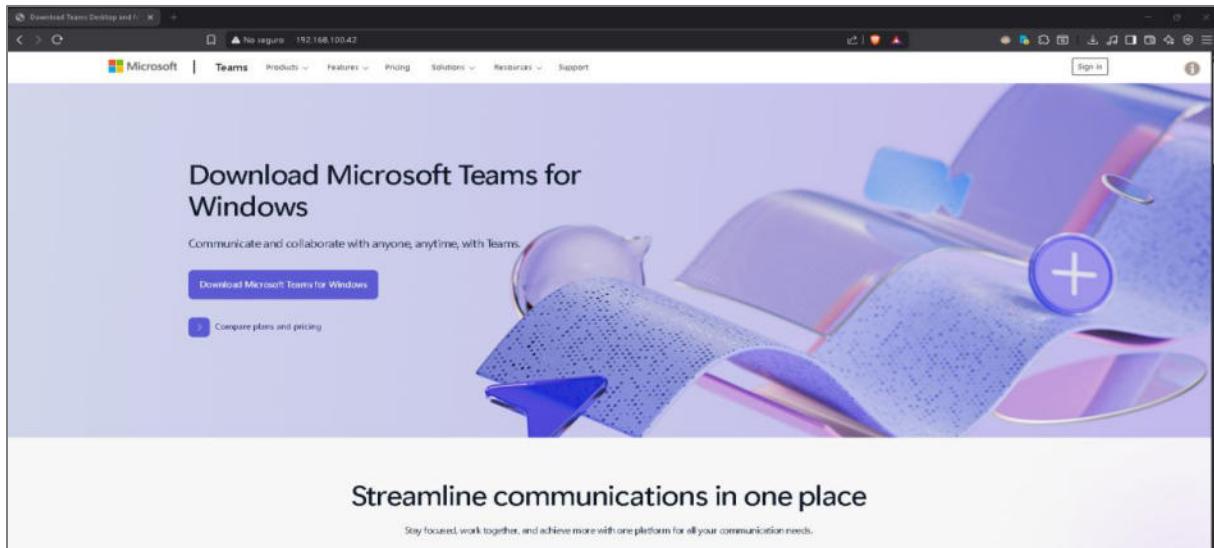


Figura 19

Comprobación de descarga de la muestra Black Basta renombrada

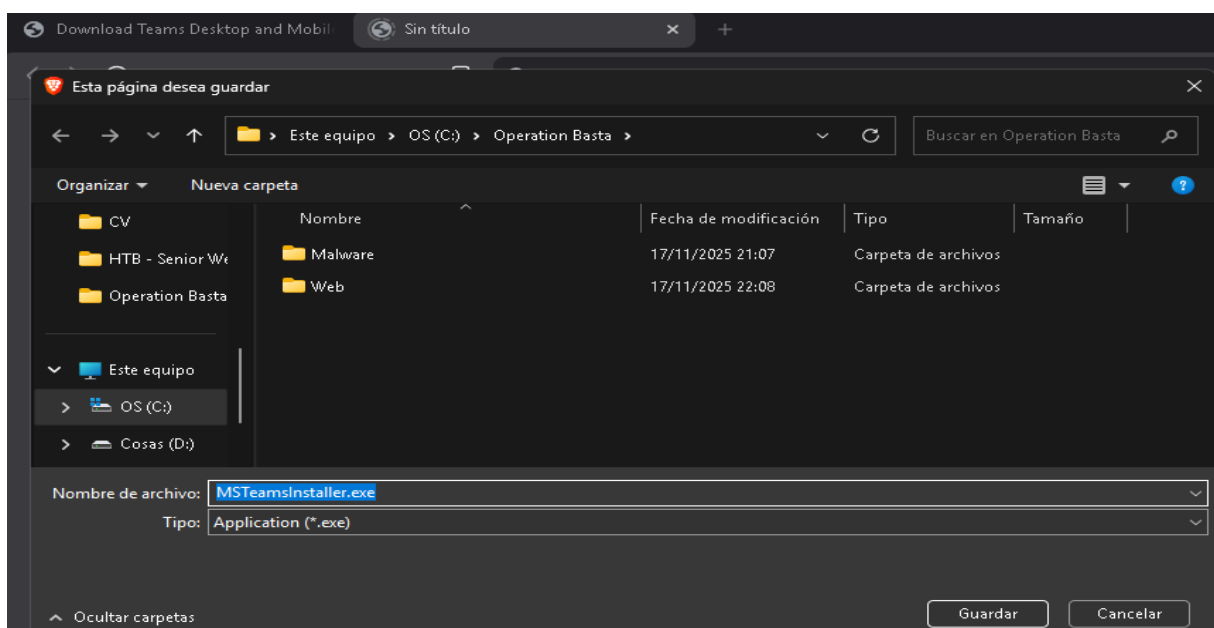
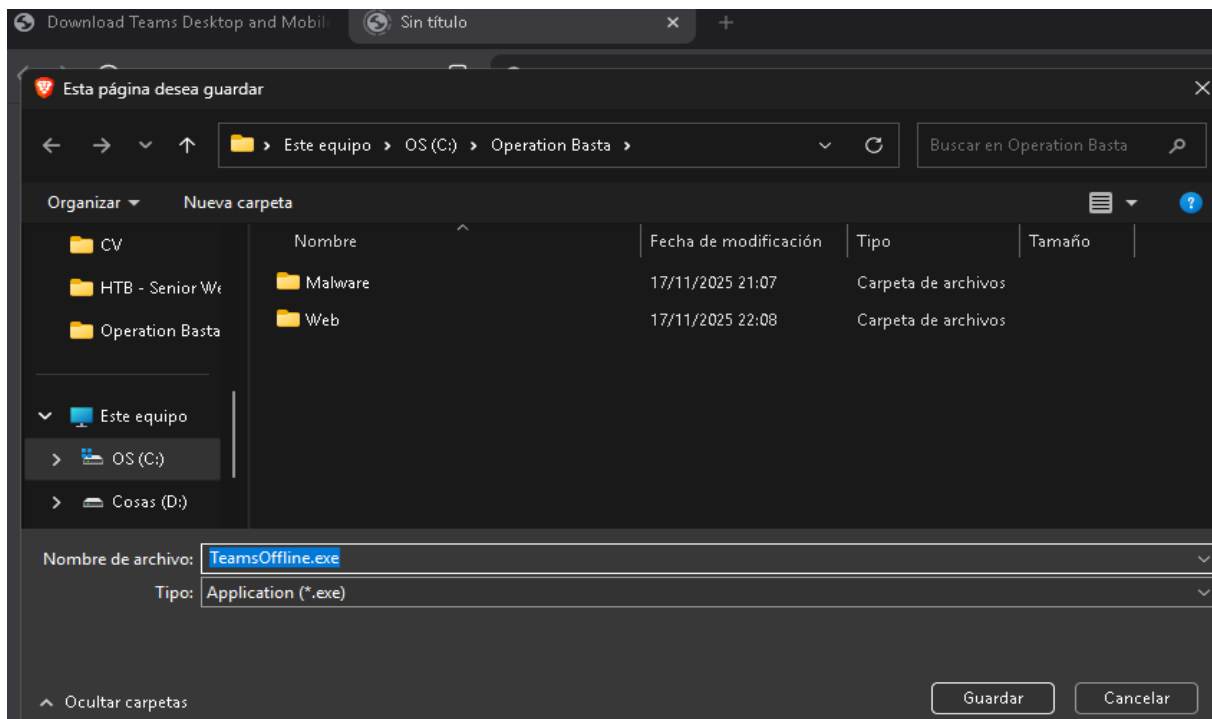


Figura 20

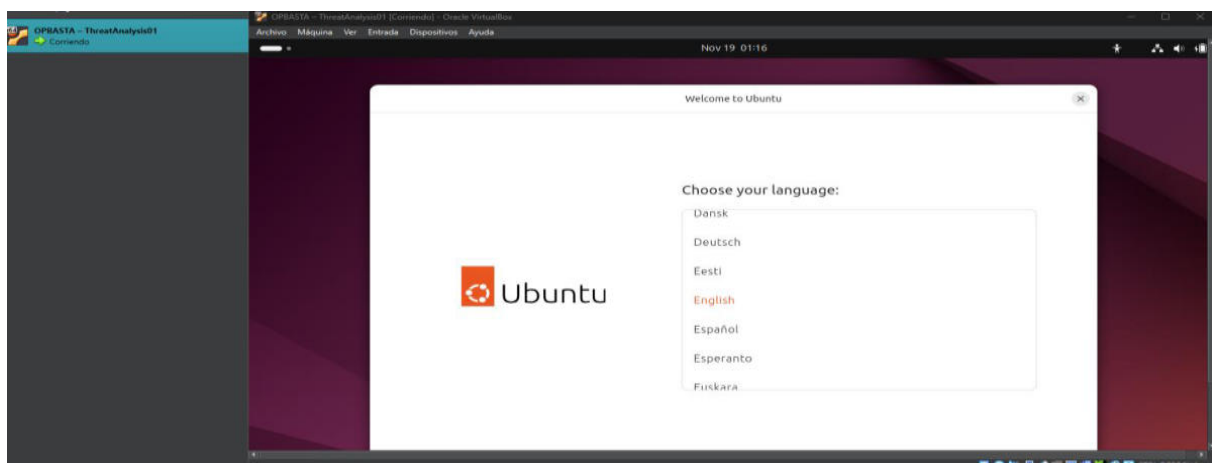
Segunda comprobación de descarga de la muestra Black Basta renombrada



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

Figura 21

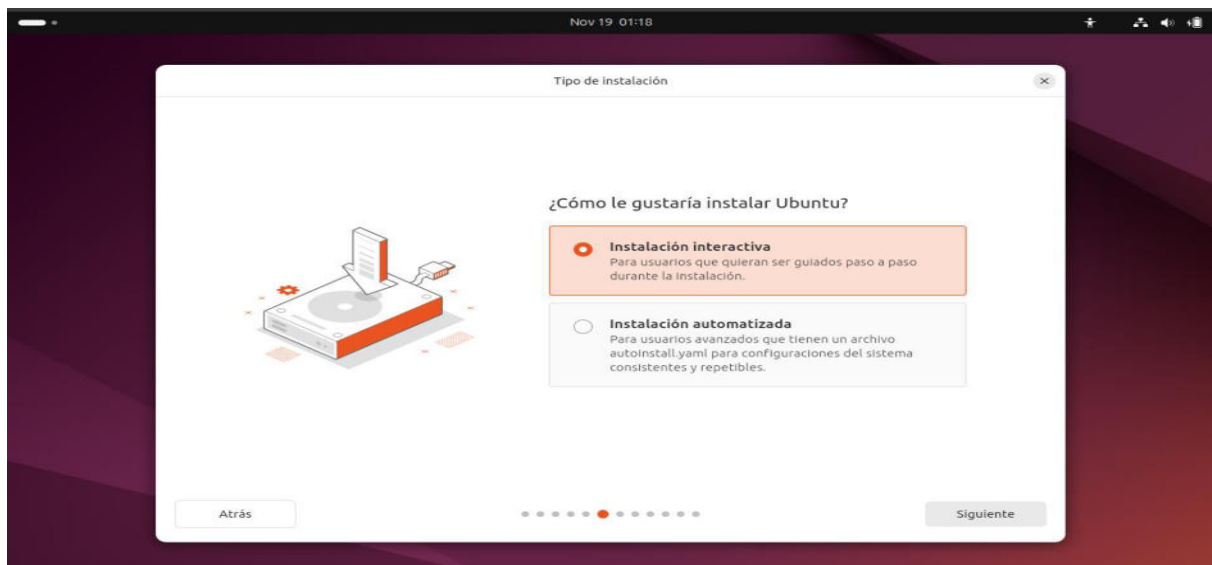
Descarga del sistema operativo Ubuntu para VM OPBASTA - ThreatAnalysis01



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

Figura 22

Configuración del modo de instalación del sistema operativo Ubuntu



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

Figura 23

Configuración de usuario y contraseña del SO Ubuntu

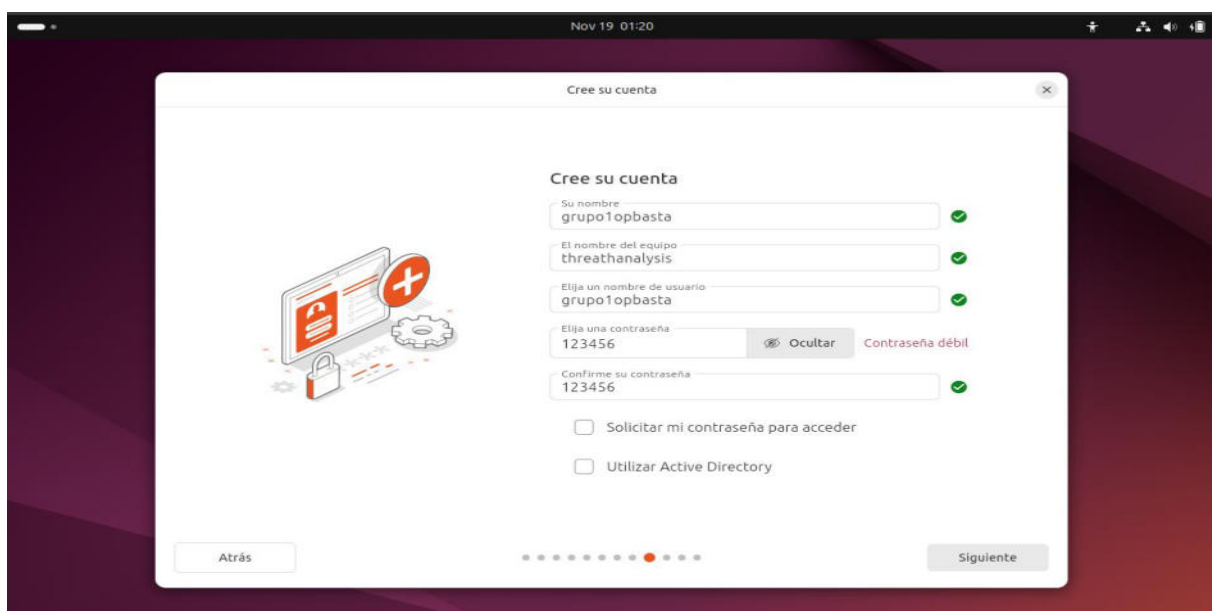
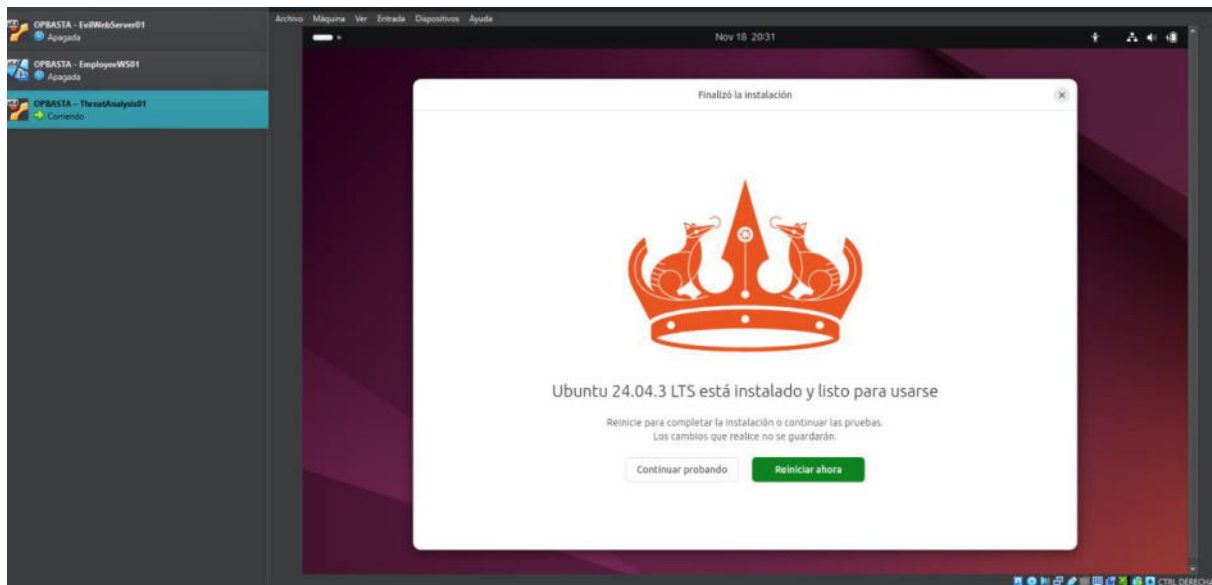


Figura 26

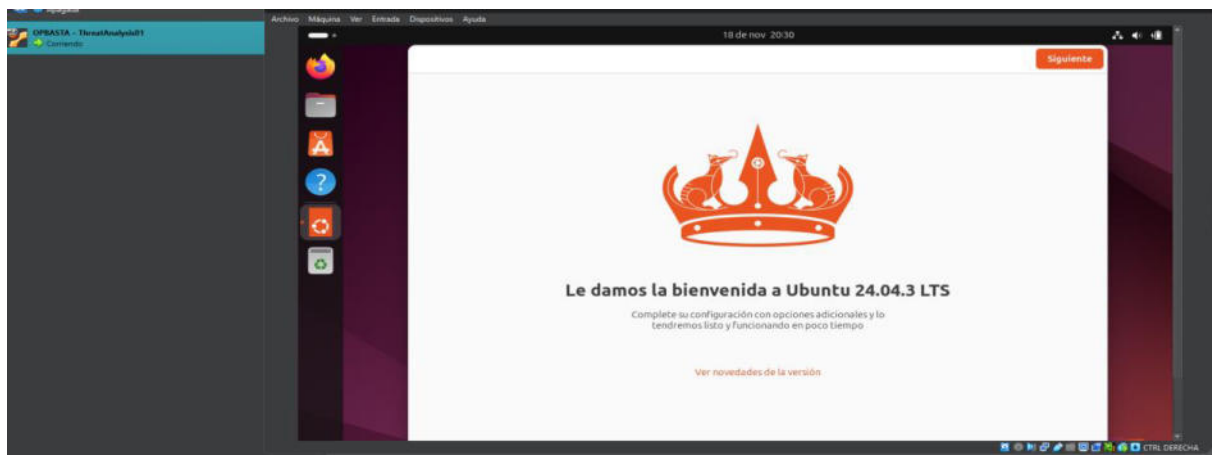
Mensaje de reinicio al culminar la instalación



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

Figura 27

Sistema operativo Ubuntu instalado en VM OPBASTA - ThreatAnalysis01



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

```
sudo apt update
```

```

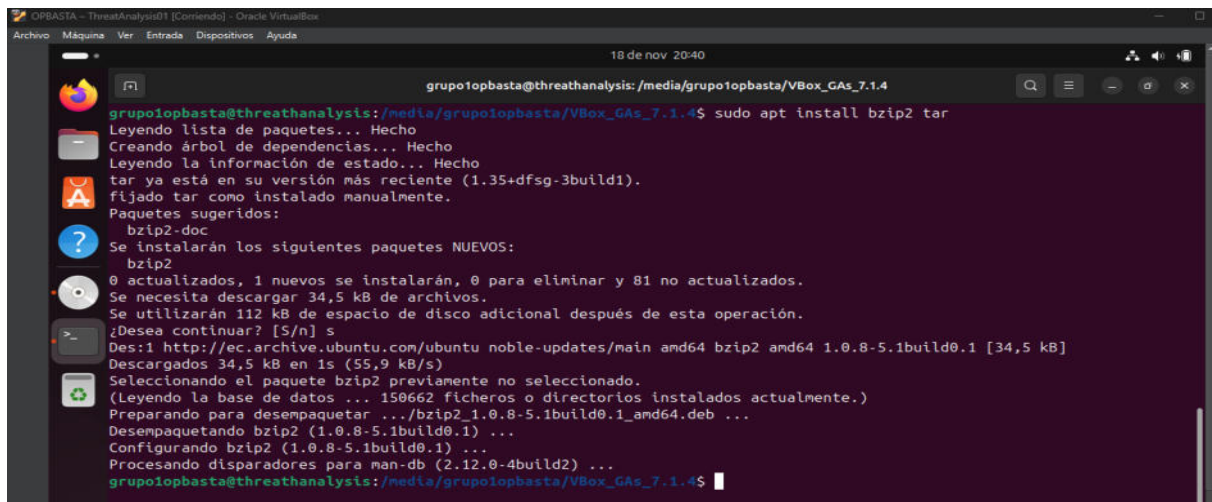
sudo apt install bzip2 tar

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

```

Figura 28

Instalación de dependencias para obtener mejoras en VM OPBASTA - ThreatAnalysis01



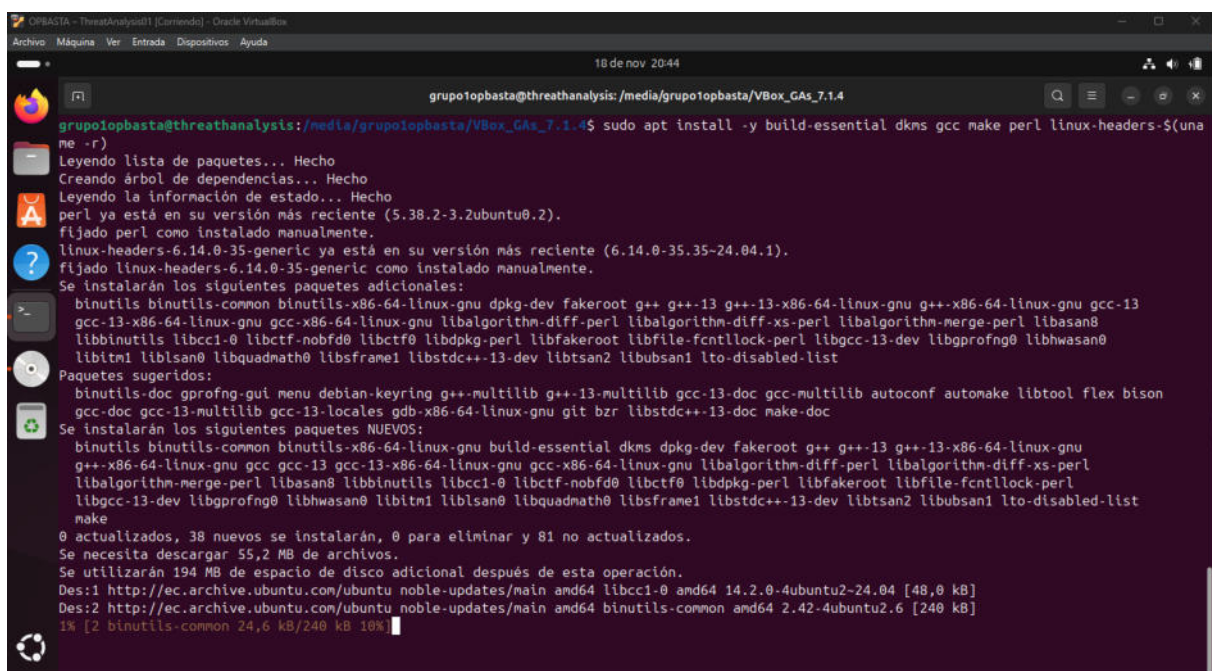
```

OPBASTA - ThreatAnalysis01 [Comando] - Oracle VM VirtualBox
18 de nov 20:40
grupo1opbasta@threatanalysis: /media/grupo1opbasta/VBox_GAs_7.1.4
grupo1opbasta@threatanalysis:/media/grupo1opbasta/VBox_GAs_7.1.4$ sudo apt install bzip2 tar
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
tar ya está en su versión más reciente (1.35+dfsg-3build1).
fijado tar como instalado manualmente.
Paquetes sugeridos:
  bzip2-doc
Se instalarán los siguientes paquetes NUEVOS:
  bzip2
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 81 no actualizados.
Se necesita descargar 34,5 kB de archivos.
Se utilizarán 112 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu noble-updates/main amd64 bzip2 amd64 1.0.8-5.1build0.1 [34,5 kB]
Descargados 34,5 kB en 1s (55,9 kB/s)
Seleccionando el paquete bzip2 previamente no seleccionado.
(Leyendo la base de datos ... 150662 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../bzip2_1.0.8-5.1build0.1_amd64.deb ...
Desempaquetando bzip2 (1.0.8-5.1build0.1) ...
Configurando bzip2 (1.0.8-5.1build0.1) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
grupo1opbasta@threatanalysis:/media/grupo1opbasta/VBox_GAs_7.1.4$

```

Figura 29

Continuación del proceso de instalación de dependencias



```

OPBASTA - ThreatAnalysis01 [Comando] - Oracle VM VirtualBox
18 de nov 20:44
grupo1opbasta@threatanalysis: /media/grupo1opbasta/VBox_GAs_7.1.4
grupo1opbasta@threatanalysis:/media/grupo1opbasta/VBox_GAs_7.1.4$ sudo apt install -y build-essential dkms gcc make perl linux-headers-$(uname -r)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
perl ya está en su versión más reciente (5.38.2-3.2ubuntu0.2).
fijado perl como instalado manualmente.
linux-headers-6.14.0-35-generic ya está en su versión más reciente (6.14.0-35.35~24.04.1).
fijado linux-headers-6.14.0-35-generic como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc-13 gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libbinutils libbcl1-0 libctf-nobfd0 libctf0 libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-13-dev libgprofng0 libhwasan0 libitm1 liblsan0 libquadmath0 libstdc++-13-dev libstdc++13 libubsan1 lto-disabled-list
Paquetes sugeridos:
  binutils-doc gprofng-gui menu debian-keyring g++-multilib g++-13-multilib gcc-13-doc gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-13-multilib gcc-13-locales gdb-x86-64-linux-gnu git bzr libstdc++13-doc make-doc
Se instalarán los siguientes paquetes NUEVOS:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dkms dpkg-dev fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc-13 gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libbinutils libbcl1-0 libctf-nobfd0 libctf0 libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-13-dev libgprofng0 libhwasan0 libitm1 liblsan0 libquadmath0 libstdc++13-dev libstdc++13 libubsan1 lto-disabled-list make
0 actualizados, 38 nuevos se instalarán, 0 para eliminar y 81 no actualizados.
Se necesita descargar 55,2 MB de archivos.
Se utilizarán 194 MB de espacio de disco adicional después de esta operación.
Des:1 http://ec.archive.ubuntu.com/ubuntu noble-updates/main amd64 libbcl1-0 amd64 14.2.0-4ubuntu2-24.04 [48,0 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu noble-updates/main amd64 binutils-common amd64 2.42-4ubuntu2.6 [240 kB]
1% [2 binutils-common 24,6 kB/240 kB 10%]

```

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

Figura 30

Instalación de mejoras de integración

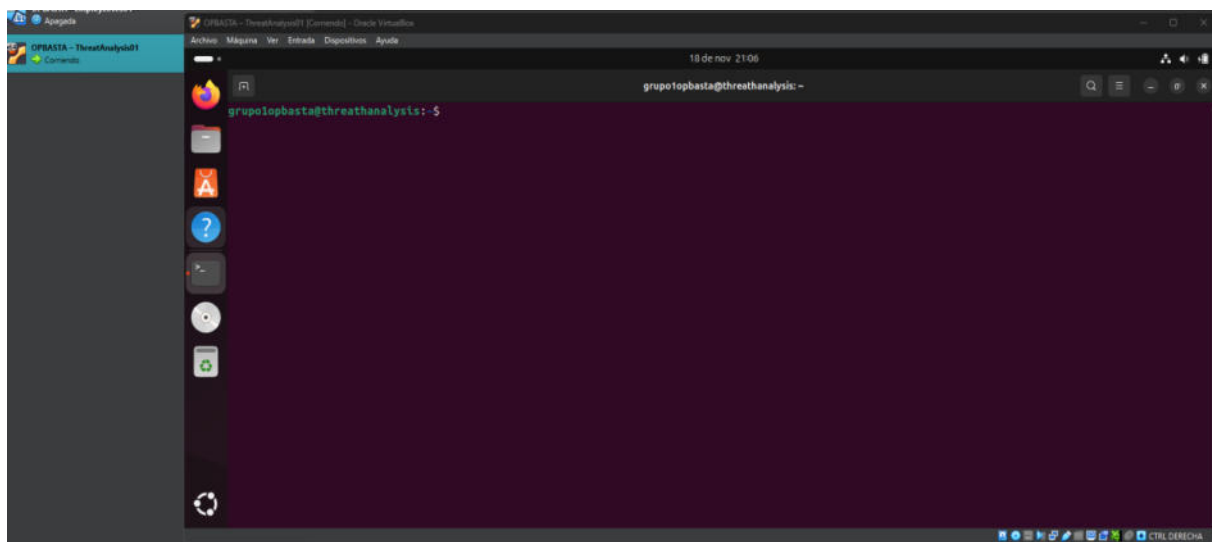
```
grupotopbasta@threatanalysis:~$ sudo apt-get install virtualbox-guest-x11
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  virtualbox-guest-utils
Se instalarán los siguientes paquetes NUEVOS:
  virtualbox-guest-utils virtualbox-guest-x11
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 81 no actualizados.
Se necesita descargar 728 kB de archivos.
Se utilizarán 2.691 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 virtualbox-guest-utils amd64 7.0.16-dfsg-2ubuntu1.1 [437 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 virtualbox-guest-x11 amd64 7.0.16-dfsg-2ubuntu1.1 [292 kB]
Descargados 728 kB en 39s (18,8 kB/s)
Seleccionando el paquete virtualbox-guest-utils previamente no seleccionado.
(Leyendo la base de datos ... 152832 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../virtualbox-guest-utils_7.0.16-dfsg-2ubuntu1.1_amd64.deb ...
Desempaquetando virtualbox-guest-utils (7.0.16-dfsg-2ubuntu1.1) ...
Seleccionando el paquete virtualbox-guest-x11 previamente no seleccionado.
Preparando para desempaquetar .../virtualbox-guest-x11_7.0.16-dfsg-2ubuntu1.1_amd64.deb ...
Desempaquetando virtualbox-guest-x11 (7.0.16-dfsg-2ubuntu1.1) ...
Configurando virtualbox-guest-utils (7.0.16-dfsg-2ubuntu1.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/virtualbox-guest-utils.service → /usr/lib/systemd/system/virtualbox-guest-utils.service.
Could not execute systemctl: at /usr/bin/deb-systemd-invoke line 148.
Configurando virtualbox-guest-x11 (7.0.16-dfsg-2ubuntu1.1) ...

Fichero de configuración '/etc/X11/Xsession.d/98vboxadd-xclient'
==> Fichero en el sistema creado por usted o por algún script.
==> Fichero también en el paquete.
¿Qué quisiera hacer al respecto?  Sus opciones son:
Y o I : instalar la versión del desarrollador del paquete
N o O : conservar la versión que tiene instalada actualmente
D      : mostrar las diferencias entre versiones
Z      : ejecutar un intérprete de órdenes para examinar la situación
La acción por omisión es conservar la versión actual.
*** 98vboxadd-xclient (Y/I/N/O/D/Z) [por omisión=N] ?
```

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

Figura 31

Equipo listo para usarse

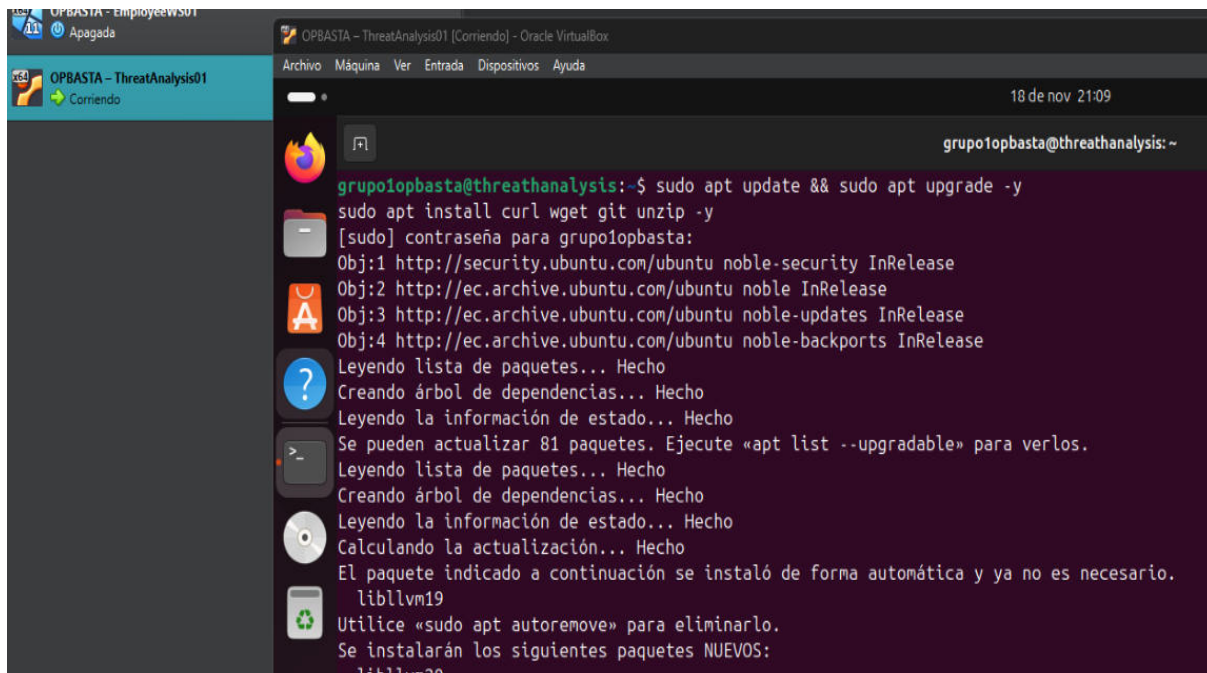


Se procedió con la instalación de todas las herramientas necesarias. Como primera herramienta, se instaló Velociraptor, el cual actúa como un “mini EDR”, ya que su objetivo principal es recolectar toda la data posible de cada movimiento realizado en el equipo Windows víctima. Los siguientes comandos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

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

Figura 32

Actualización de paquetes en repositorios



The screenshot shows a terminal window titled "OPBASTA - ThreatAnalysis01 [Corriendo] - Oracle VirtualBox". The terminal output shows the execution of the command `sudo apt update && sudo apt upgrade -y`. The output includes the following lines:

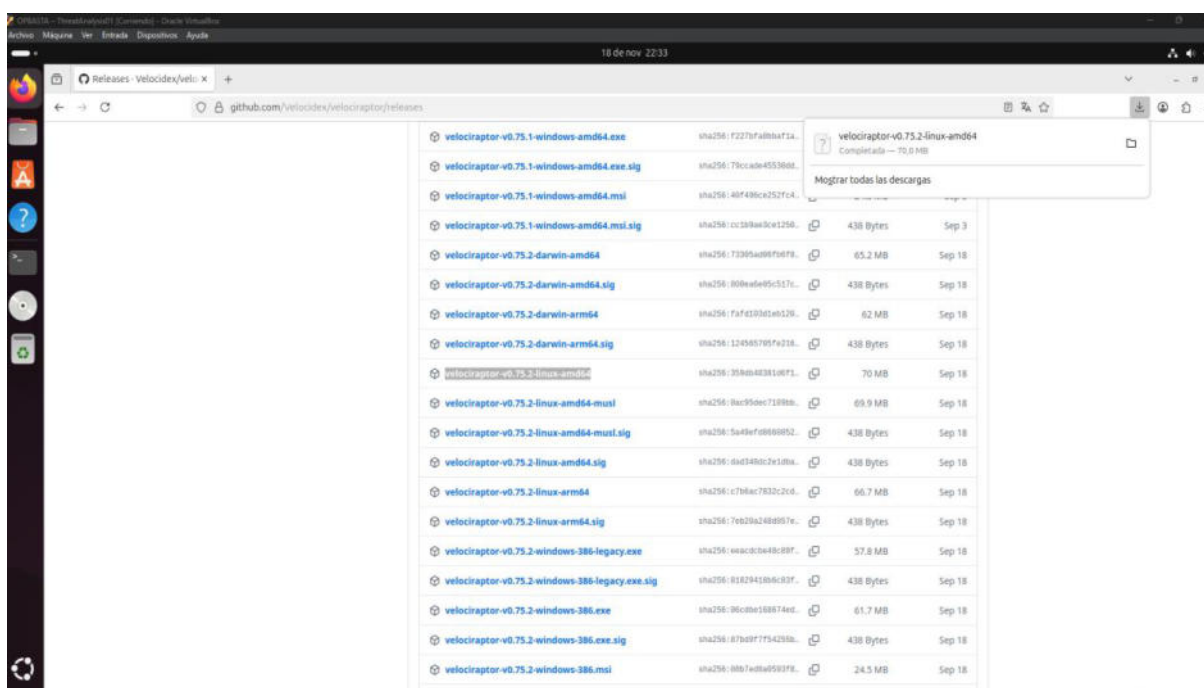
```
grupo10pbasta@threathanalysis:~$ sudo apt update && sudo apt upgrade -y  
sudo apt install curl wget git unzip -y  
[sudo] contraseña para grupo10pbasta:  
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease  
Obj:2 http://ec.archive.ubuntu.com/ubuntu noble InRelease  
Obj:3 http://ec.archive.ubuntu.com/ubuntu noble-updates InRelease  
Obj:4 http://ec.archive.ubuntu.com/ubuntu noble-backports InRelease  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Se pueden actualizar 81 paquetes. Ejecute «apt list --upgradable» para verlos.  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Calculando la actualización... Hecho  
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.  
libllvm19  
Utilice «sudo apt autoremove» para eliminarlo.  
Se instalarán los siguientes paquetes NUEVOS:  
libllvm20
```

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

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

Figura 33

Descarga de la última versión de Velociraptor para Linux

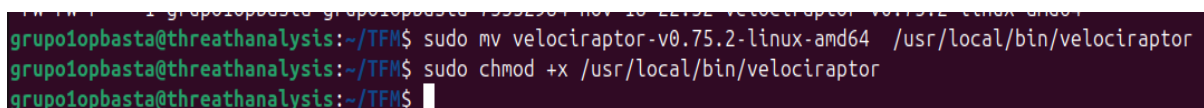


Luego de la descarga, se movió el binario de Velociraptor a una de las rutas del PATH para poder invocarlo desde cualquier directorio en consola, se usaron los comandos:

```
sudo mv velociraptor-v0.75.2-linux-amd64 /usr/local/bin/velociraptor  
sudo chmod +x /usr/local/bin/Velociraptor
```

Figura 34

Redirección de la carpeta donde se invoca Velociraptor



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

Figura 35*Ejecución del binario de Velociraptor*

```

grupo10pbasta@threathanalysis:~/TFM$ velociraptor
usage: velociraptor [<flags>] <command> [<args> ...]

An advanced incident response and monitoring agent.

Flags:
  -h, --[no-]help           Show context-sensitive help (also try --help-long and --help-man).
  --remap=REMAP             A remapping configuration file for dead disk analysis.
  --[no-]nobanner          Suppress the Velociraptor banner
  --[no-]debug             Enables debug and profile server.
  --debug_port=6060        Port for the debug server.
  -c, --config=CONFIG       The configuration file.
  --embedded_config=EMBEDDED_CONFIG
                           Extract the embedded configuration from this file.
  -a, --api_config=API_CONFIG The API configuration file.
  -o, --config_override=CONFIG_OVERRIDE
                           A json object to override the config.
  --runas=RUNAS            Run as this username's ACLs
  --definitions=DEFINITIONS A directory containing artifact definitions
  --[no-]nocolor           Disable color output
  -v, --[no-]verbose       Enabled verbose logging.
  --profile=PROFILE        Write profiling information to this file.
  --profile_duration=PROFILE_DURATION
                           Generate a profile file for each period in seconds.
  --trace=TRACE            Write trace information to this file.
  --[no-]trace_vql         Enable VQL tracing.
  --logfile=LOGFILE        Write to this file as well
  --tempdir=TEMPDIR        Write all temp files to this directory
  --[no-]prompt            Present a prompt before exit
  --max_wait=10            Maximum time to queue results.
  --timezone=TIMEZONE      Default encoding timezone (e.g. Australia/Brisbane). If not set we use UTC

Commands:

```

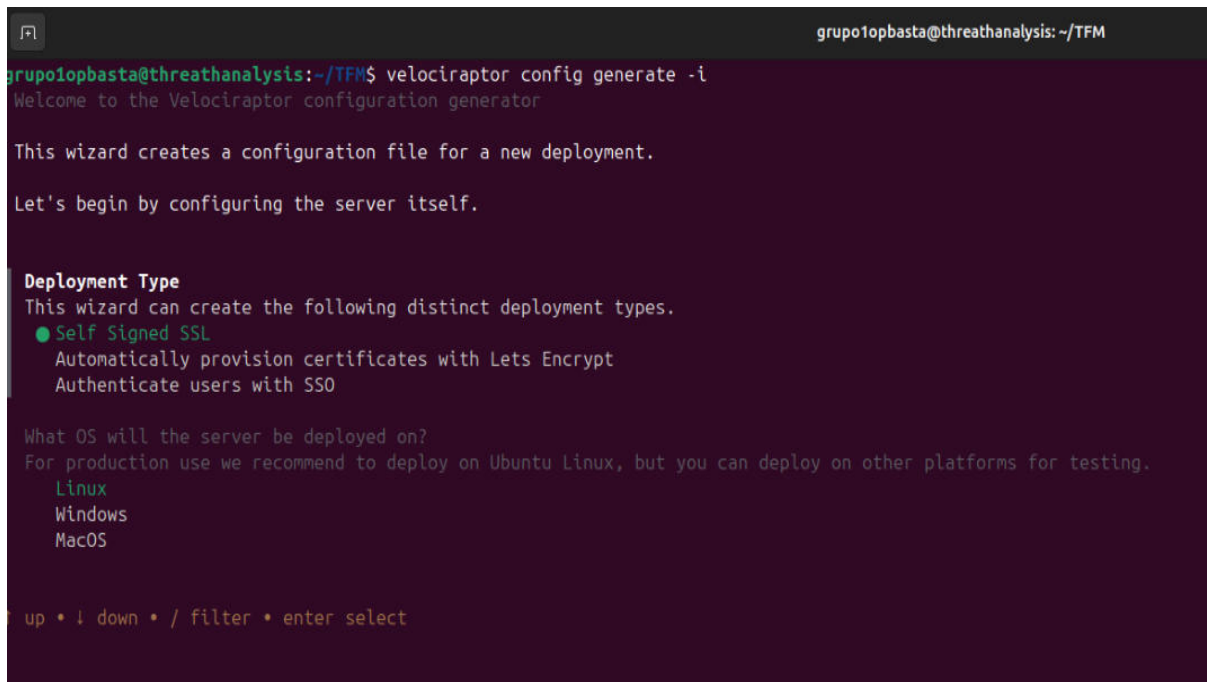
Inmediatamente, se configuró Velociraptor para que funcione como servidor y reciba toda la información que se recolecte desde un agente. Se utilizó el siguiente comando:

```
velociraptor config generate -i
```

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

Figura 36

Configuración del servidor de Velociraptor



```
grupo10pbasta@threathanalysis: ~/TFM
grupo10pbasta@threathanalysis:~/TFM$ velociraptor config generate -i
Welcome to the Velociraptor configuration generator

This wizard creates a configuration file for a new deployment.

Let's begin by configuring the server itself.

Deployment Type
This wizard can create the following distinct deployment types.
● Self Signed SSL
  Automatically provision certificates with Lets Encrypt
  Authenticate users with SSO

What OS will the server be deployed on?
For production use we recommend to deploy on Ubuntu Linux, but you can deploy on other platforms for testing.
Linux
Windows
MacOS

up • ↓ down • / filter • enter select
```

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

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

Figura 37

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

```
grupo10pbasta@threathanalysis:~/TFM$ velociraptor config generate -i

The datastore directory is where Velociraptor will store all files.

Path to the datastore directory.
The datastore directory is where Velociraptor will store all files.
This should be located on a partition large enough to contain all data you are likely to collect.
Make sure there is sufficient disk space available!
> /opt/velociraptor

Path to the logs directory.
Velociraptor will write logs to this directory. By default it resides within the datastore directory but you can place it anywhere.
> logs

Internal PKI Certificate Expiration
By default internal certificates are issued for 1 year.

If you expect this deployment to exist part one year you might
consider extending the default validation.
  1 Year
  2 Years
  10 Years

Use registry for client writeback?
Traditionally Velociraptor uses files to store client state on all operating systems.

You can instead use the registry on Windows. NOTE: It is your responsibility to ensure the registry keys used are properly secured!

By default we use HKLM\SOFTWARE\Velocidex\Velociraptor

Yes No

+/- toggle • shift+tab back • enter submit • y Yes • n No
```

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

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

Figura 38

Configuración de características de red y puertos del servidor Velociraptor

```
grupo10pbasta@threatanalysis:~/TFM$ velociraptor config generate -i

This section configures the server's network parameters.

What is the public DNS name of the Master Frontend?
Clients will connect to the Frontend using this public name.
> localhost

DNS Type
In order for the server to be reachable from the internet, you must have DNS configured.
  None - Configure DNS manually
  NOIP
  CloudFlare

Would you like to try the new experimental websocket comms?
Websocket is a bidirectional low latency communication protocol supported by
most modern proxies and load balancers. This method is more efficient and
portable than plain HTTP. Be sure to test this in your environment.

  Yes  No

Enter the frontend port to listen on.
> 8000

Enter the port for the GUI to listen on.
> 8889

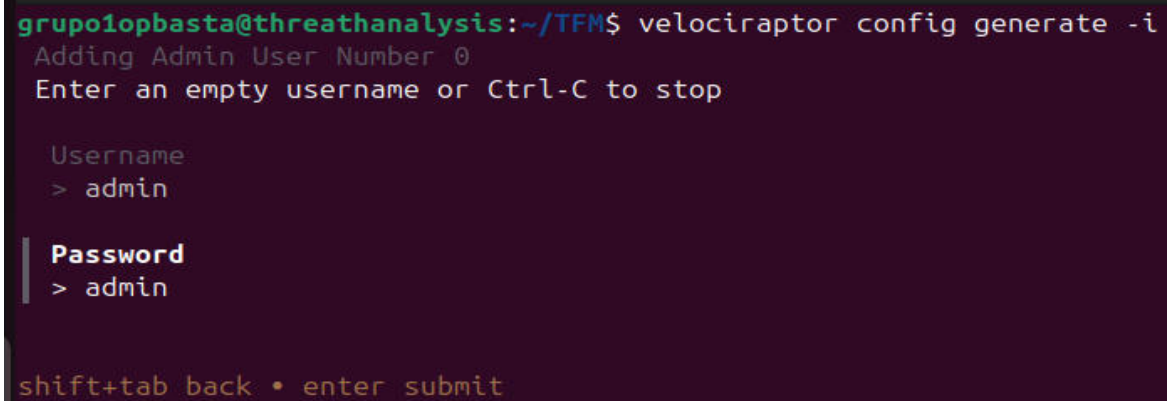
shift+tab back • enter submit
```

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

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

Figura 39

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



```
grupo10pbasta@threathanalysis:~/TFM$ velociraptor config generate -i
Adding Admin User Number 0
Enter an empty username or Ctrl-C to stop

Username
> admin

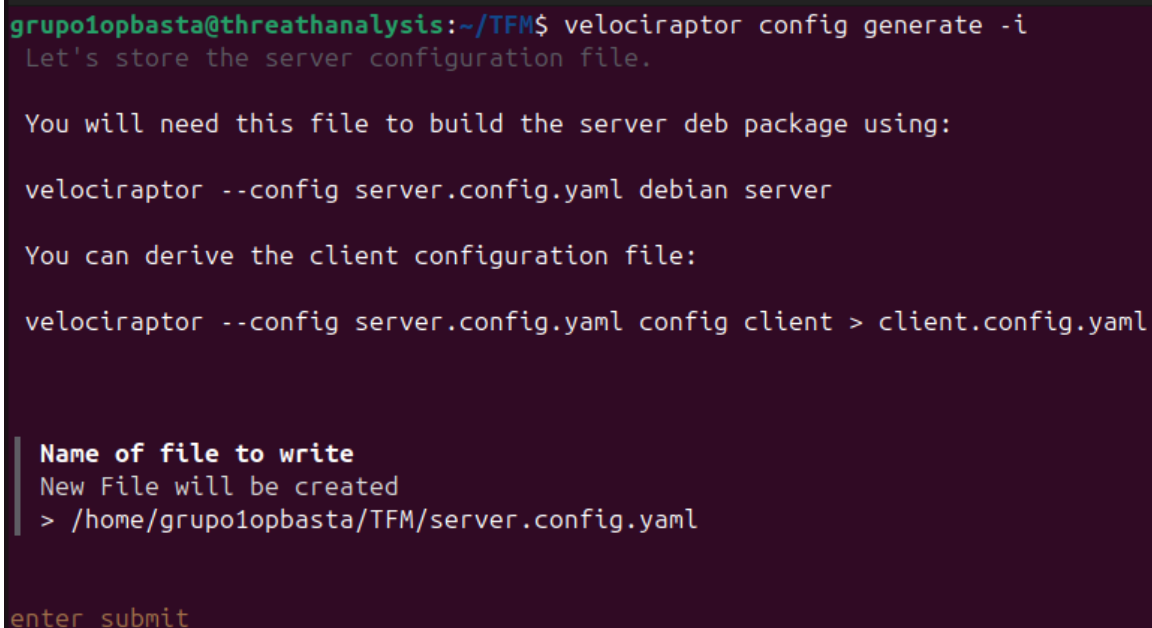
Password
> admin

shift+tab back • enter submit
```

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

Figura 40

Finalización de la configuración del servidor de Velociraptor



```
grupo10pbasta@threathanalysis:~/TFM$ velociraptor config generate -i
Let's store the server configuration file.

You will need this file to build the server deb package using:

velociraptor --config server.config.yaml debian server

You can derive the client configuration file:

velociraptor --config server.config.yaml config client > client.config.yaml

Name of file to write
New File will be created
> /home/grupo10pbasta/TFM/server.config.yaml

enter submit
```

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

Figura 41

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

```
grupo1opbasta@threathanalysis:~/TFM$ ls -la
total 24
drwxrwxr-x  2 grupo1opbasta grupo1opbasta  4096 nov 18 22:54 .
drwxr-x--- 16 grupo1opbasta grupo1opbasta  4096 nov 18 22:28 ..
-rw-----  1 grupo1opbasta grupo1opbasta 12935 nov 18 22:54 server.config.yaml
grupo1opbasta@threathanalysis:~/TFM$
```

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

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

Figura 42

Creación del servicio de systemd

```
OPBASTA - ThreatAnalysis01 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
19 de nov  20:22
grupo1opbasta@threathanalysis: ~/TFM
grupo1opbasta@threathanalysis:~/TFM$ sudo velociraptor --config server.config.yaml debian server
[sudo] contraseña para grupo1opbasta:
[[{
  "OSPath": "/home/grupo1opbasta/TFM/velociraptor-server-0.75.2.amd64.deb"
}]
grupo1opbasta@threathanalysis:~/TFM$
```

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

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

Figura 43

Instalación del paquete .deb de Velociraptor

```
grupoiofbasta@threathanalysis:~/TFM$ sudo dpkg -i velociraptor-server-0.75.2.amd64.deb
Seleccionando el paquete velociraptor-server previamente no seleccionado.
(Leyendo la base de datos ... 153963 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar velociraptor-server-0.75.2.amd64.deb ...
Desempaquetando velociraptor-server (0.75.2) ...
Configurando velociraptor-server (0.75.2) ...
info: Seleccionando un GID del rango 100 a 999 ...
info: Añadiendo el grupo 'velociraptor' (GID 124) ...
info: Seleccionando un UID del rango 100 a 999 ...
info: Añadiendo el usuario del sistema 'velociraptor' (UID 122) ...
info: Añadiendo un nuevo usuario 'velociraptor' (UID 122) con grupo 'velociraptor' ...
info: No se crea el directorio personal '/etc/velociraptor'.
Created symlink /etc/systemd/system/multi-user.target.wants/velociraptor_server.service → /etc/systemd/system/velociraptor_server.service.
grupoiofbasta@threathanalysis:~/TFM$
```

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

```
sudo systemctl enable velociraptor_server
sudo systemctl start velociraptor_server
sudo systemctl status velociraptor_server
```

Figura 44

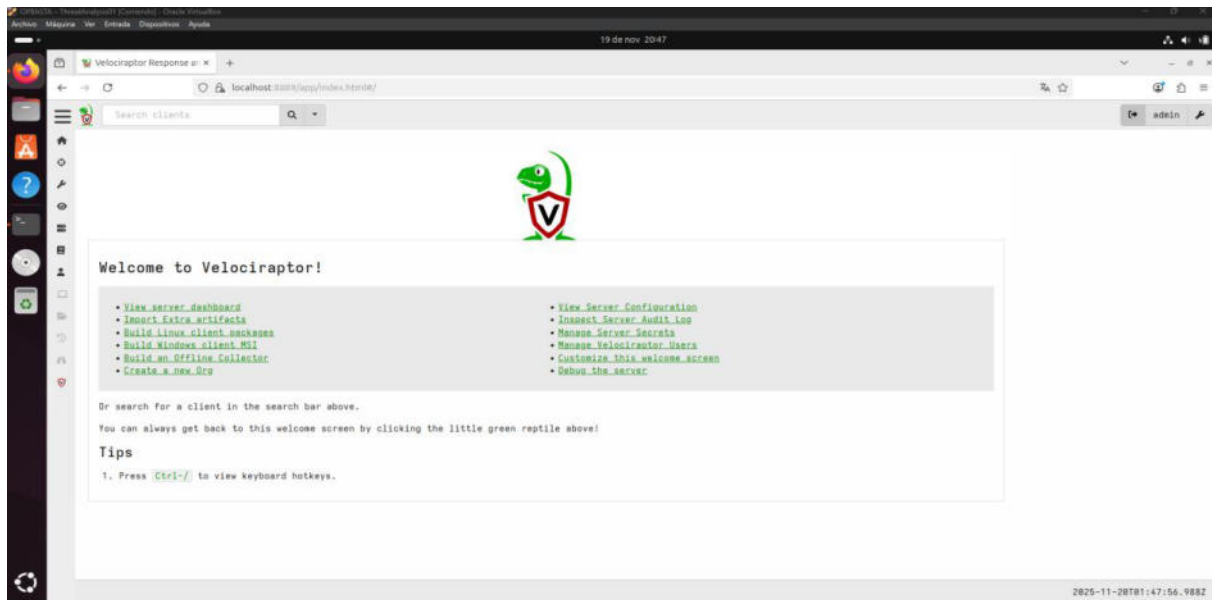
Inicialización de los servicios del servidor de Velociraptor

```
grupoiofbasta@threathanalysis:~/TFM$ sudo systemctl enable velociraptor_server
sudo systemctl start velociraptor_server
sudo systemctl status velociraptor_server
[sudo] contraseña para grupoiofbasta:
● velociraptor_server.service - Velociraptor server
   Loaded: loaded (/etc/systemd/system/velociraptor_server.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-11-19 20:24:04 -05; 21min ago
     Main PID: 4098 (velociraptor)
        Tasks: 22 (limit: 5198)
      Memory: 60.2M (peak: 80.9M)
         CPU: 8.810s
       CGroup: /system.slice/velociraptor_server.service
               └─4098 /usr/local/bin/velociraptor --config /etc/velociraptor/server.config.yaml frontend
                  └─4114 /usr/local/bin/velociraptor --config /etc/velociraptor/server.config.yaml frontend
```

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

Figura 45

Interfaz gráfica de Velociraptor



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

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

Figura 46

Modificación del archivo de configuración del servidor de Velociraptor

```
API:
  bind_address: 192.168.56.103
  bind_port: 8001
  bind_scheme: tcp
  pinned_gw_name: GRPC_GW
GUI:
  bind_address: 192.168.56.103
  bind_port: 8889
  gw_certificate: |
```

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

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

Figura 47

Reinicio del servidor de Velociraptor

```
grupo1opbasta@threathanalysis:~/TFM$ sudo systemctl restart velociraptor_server.service
grupo1opbasta@threathanalysis:~/TFM$ systemctl status velociraptor_server.service
● velociraptor_server.service - Velociraptor server
   Loaded: loaded (/etc/systemd/system/velociraptor_server.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-11-19 22:20:19 -05; 4s ago
     Main PID: 5352 (velociraptor)
        Tasks: 19 (limit: 5198)
       Memory: 71.3M (peak: 75.4M)
          CPU: 1.770s
      CGroup: /system.slice/velociraptor_server.service
              └─5352 /usr/local/bin/velociraptor --config /etc/velociraptor/server.config.yaml frontend
                 └─5363 /usr/local/bin/velociraptor --config /etc/velociraptor/server.config.yaml frontend
```

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

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

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

Figura 48

Creación del agente recolector de artefactos de Velociraptor

```
grupo1opbasta@threathanalysis:~/TFM$ sudo velociraptor --config /etc/velociraptor/server.config.yaml config client > client.config.yaml
grupo1opbasta@threathanalysis:~/TFM$ ls -la
total 28248
drwxrwxr-x 2 grupo1opbasta grupo1opbasta 4096 nov 19 22:29 .
drwxr-x--- 16 grupo1opbasta grupo1opbasta 4096 nov 19 22:20 ..
-rw-rw-r-- 1 grupo1opbasta grupo1opbasta 2692 nov 19 22:29 client.config.yaml
-rw----- 1 grupo1opbasta grupo1opbasta 12946 nov 19 22:04 server.config.yaml
-rw----- 1 root root 28893764 nov 19 22:04 velociraptor-server-0.75.5.amd64.deb
grupo1opbasta@threathanalysis:~/TFM$
```

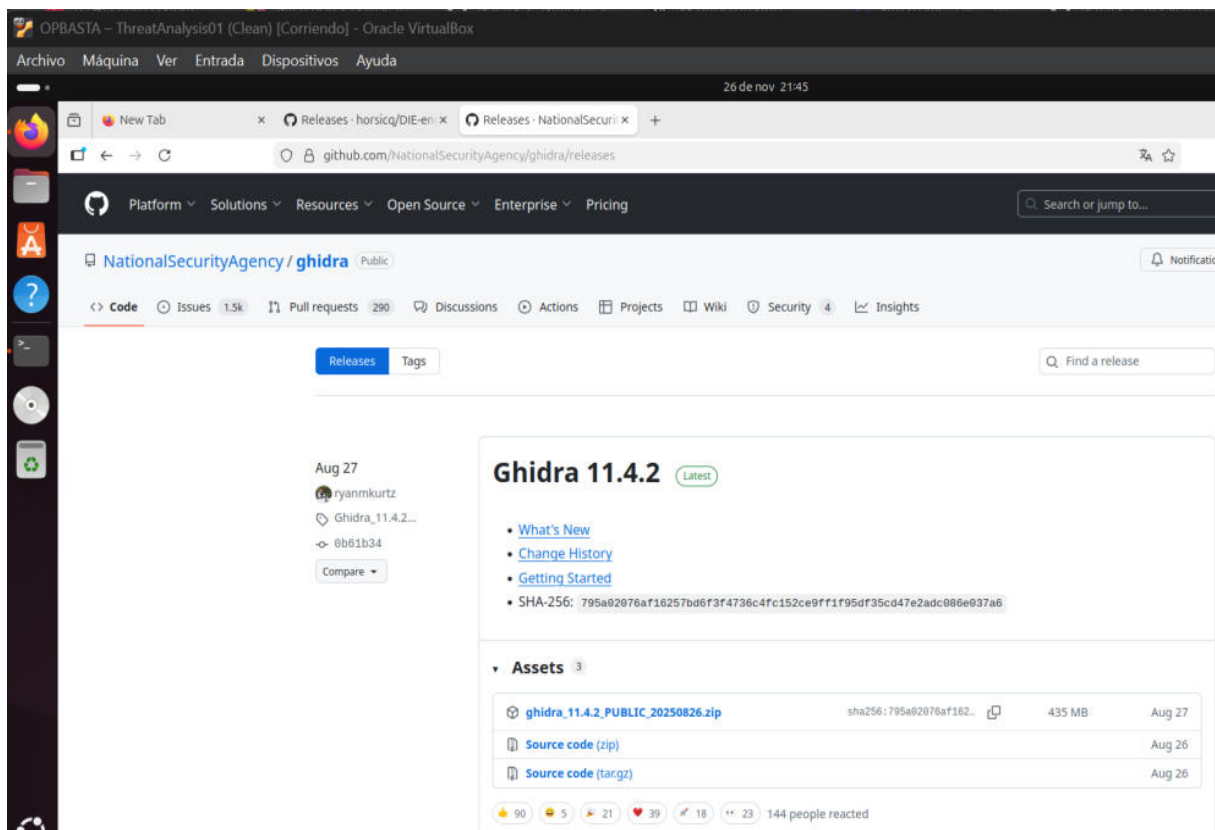
El archivo generado “client.config.yaml” se lo usa al momento de instalar el agente en nuestra máquina víctima VM OPBASTA - EmployeeHost01.

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

<https://github.com/NationalSecurityAgency/ghidra/releases>

Figura 49

Descarga de la última versión de Ghidra desde el repositorio oficial

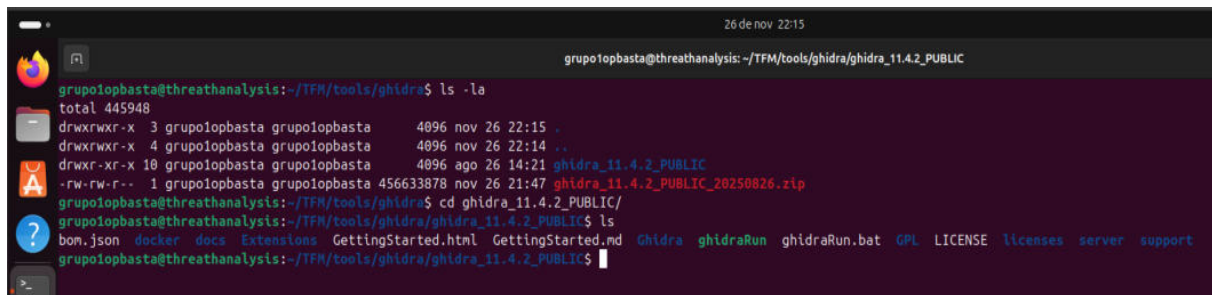


Una vez descargado, se descomprimió el archivo utilizando el comando:

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

Figura 50

Descompresión del archivo descargado de Ghidra



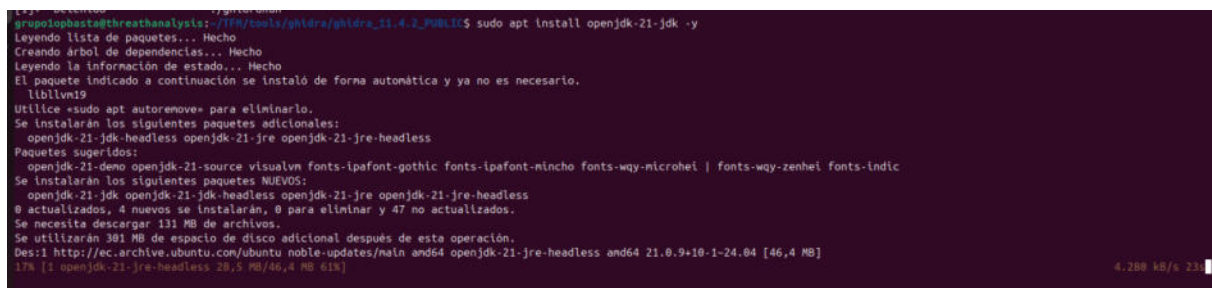
```
grupo10pbasta@threathanalysis: ~/TFM/tools/ghidra/ghidra_11.4.2_PUBLIC
grupo10pbasta@threathanalysis:~/TFM/tools/ghidra$ ls -la
total 445948
drwxrwxr-x 3 grupo10pbasta grupo10pbasta 4096 nov 26 22:15 .
drwxrwxr-x 4 grupo10pbasta grupo10pbasta 4096 nov 26 22:14 ..
drwxr-xr-x 10 grupo10pbasta grupo10pbasta 4096 ago 26 14:21 ghidra_11.4.2_PUBLIC
-rw-rw-r-- 1 grupo10pbasta grupo10pbasta 456633878 nov 26 21:47 ghidra_11.4.2_PUBLIC_20250826.zip
grupo10pbasta@threathanalysis:~/TFM/tools/ghidra$ cd ghidra_11.4.2_PUBLIC/
grupo10pbasta@threathanalysis:~/TFM/tools/ghidra/ghidra_11.4.2_PUBLIC$ ls
bon.json  docker  docs  Extensions  GettingStarted.html  GettingStarted.md  Chidra  ghidraRun  ghidraRun.bat  GPL  LICENSE  licenses  server  support
grupo10pbasta@threathanalysis:~/TFM/tools/ghidra/ghidra_11.4.2_PUBLIC$
```

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

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

Figura 51

Instalación de dependencias de Java



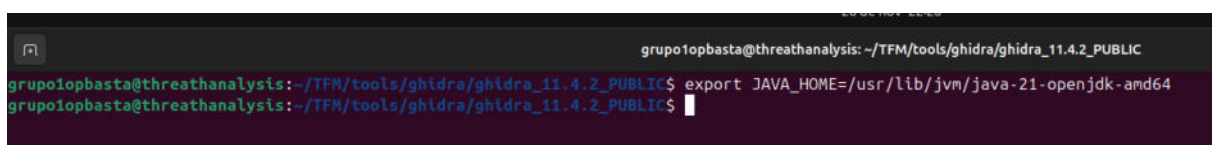
```
grupo10pbasta@threathanalysis:~/TFM/tools/ghidra/ghidra_11.4.2_PUBLIC$ sudo apt install openjdk-21-jdk -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
liblvm2
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
openjdk-21-jdk-headless openjdk-21-jre openjdk-21-jre-headless
Paquetes sugeridos:
openjdk-21-demo openjdk-21-source visualvm fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei fonts-Indic
Se instalarán los siguientes paquetes NUEVOS:
openjdk-21-jdk openjdk-21-jdk-headless openjdk-21-jre openjdk-21-jre-headless
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 47 no actualizados.
Se necesita descargar 131 MB de archivos.
Se utilizarán 381 MB de espacio de disco adicional después de esta operación.
Des:1 http://ec.archive.ubuntu.com/ubuntu noble-updates/main amd64 openjdk-21-jre-headless amd64 21.0.9+10-1-24.04 [46,4 MB]
[7% [1 openjdk-21-jre-headless 28,5 MB/46,4 MB 61%] 4.280 kB/s 23s]
```

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

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

Figura 52

Establecimiento de la variable de entorno JAVA_HOME



```
grupo10pbasta@threathanalysis:~/TFM/tools/ghidra/ghidra_11.4.2_PUBLIC$ export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
grupo10pbasta@threathanalysis:~/TFM/tools/ghidra/ghidra_11.4.2_PUBLIC$
```

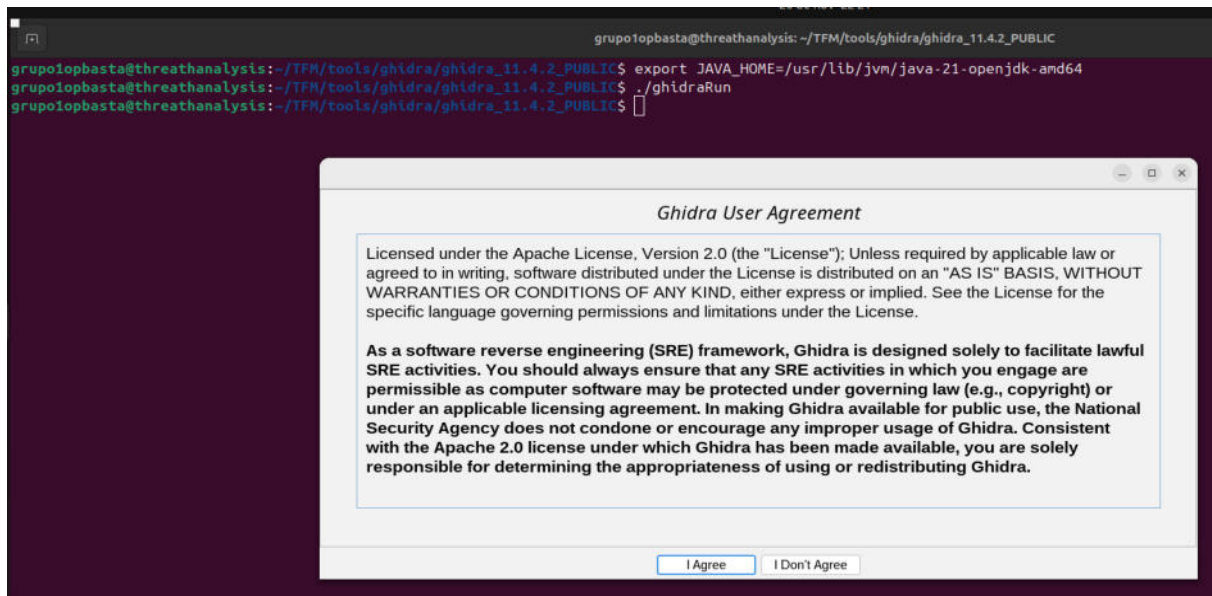
Adicionalmente, se otorgaron permisos de ejecución al archivo “ghidraRun” y finalmente, se lo ejecutó:

```
chmod +x ghidraRun
```

```
./ghidraRun
```

Figura 53

Aceptación de acuerdo de usuario antes de ejecutar Ghidra

**Figura 54**

Ejecución de Ghidra en VM OPBASTA - ThreatAnalysis01

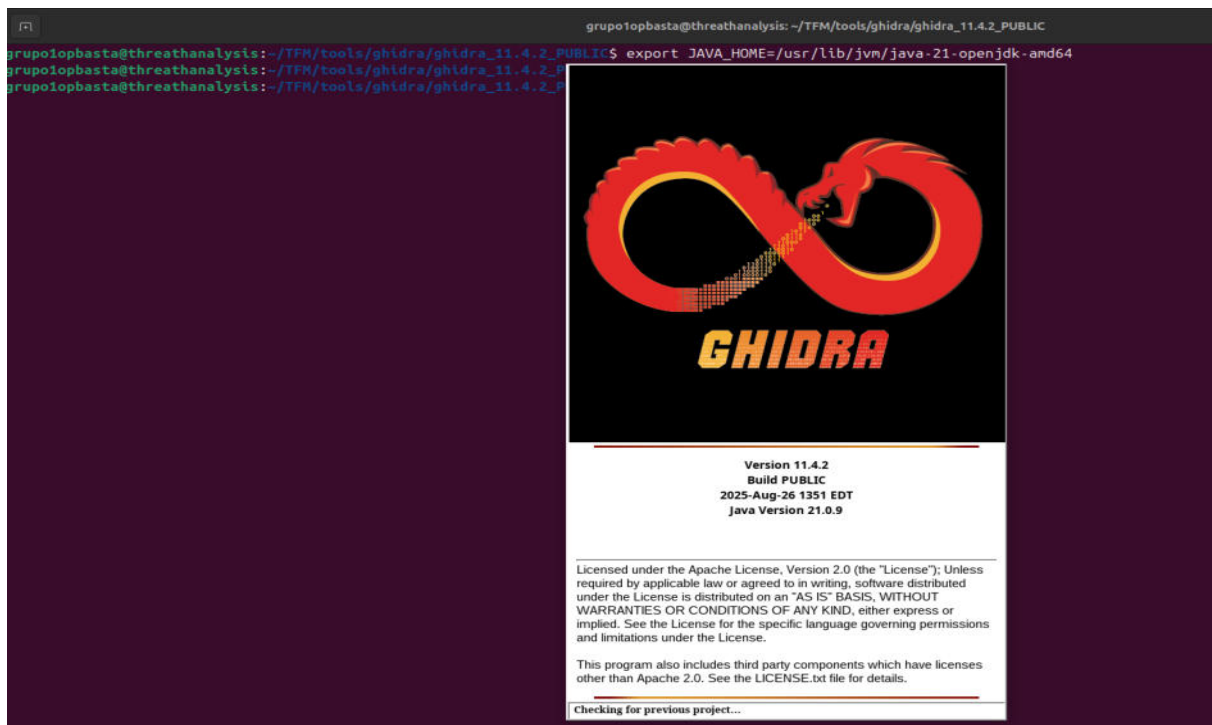


Figura 55

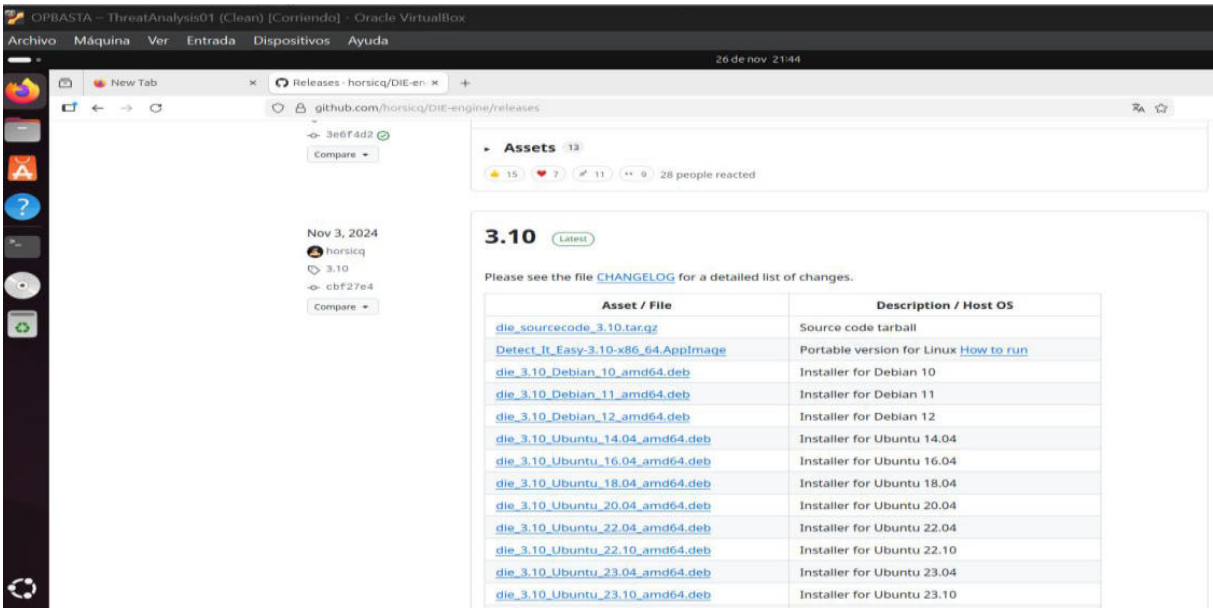
Interfaz de Ghidra abierta en VM OPBASTA - ThreatAnalysis01



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

Figura 56

Descarga de herramienta Detect It Easy

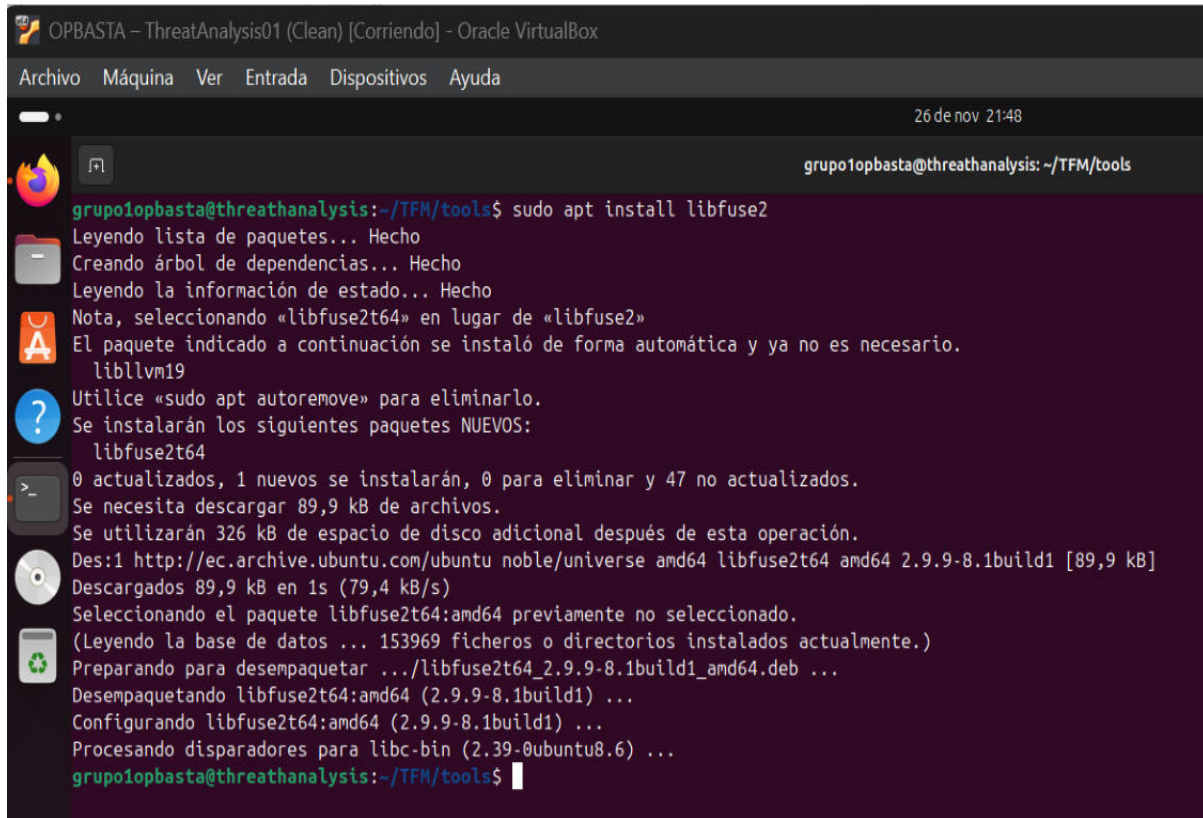


Se continuó con la instalación de dependencias para poder ejecutar el archivo de tipo AppImage. Se utilizó el siguiente comando:

```
sudo apt install libfuse2
```

Figura 57

Instalación de dependencias libfuse2



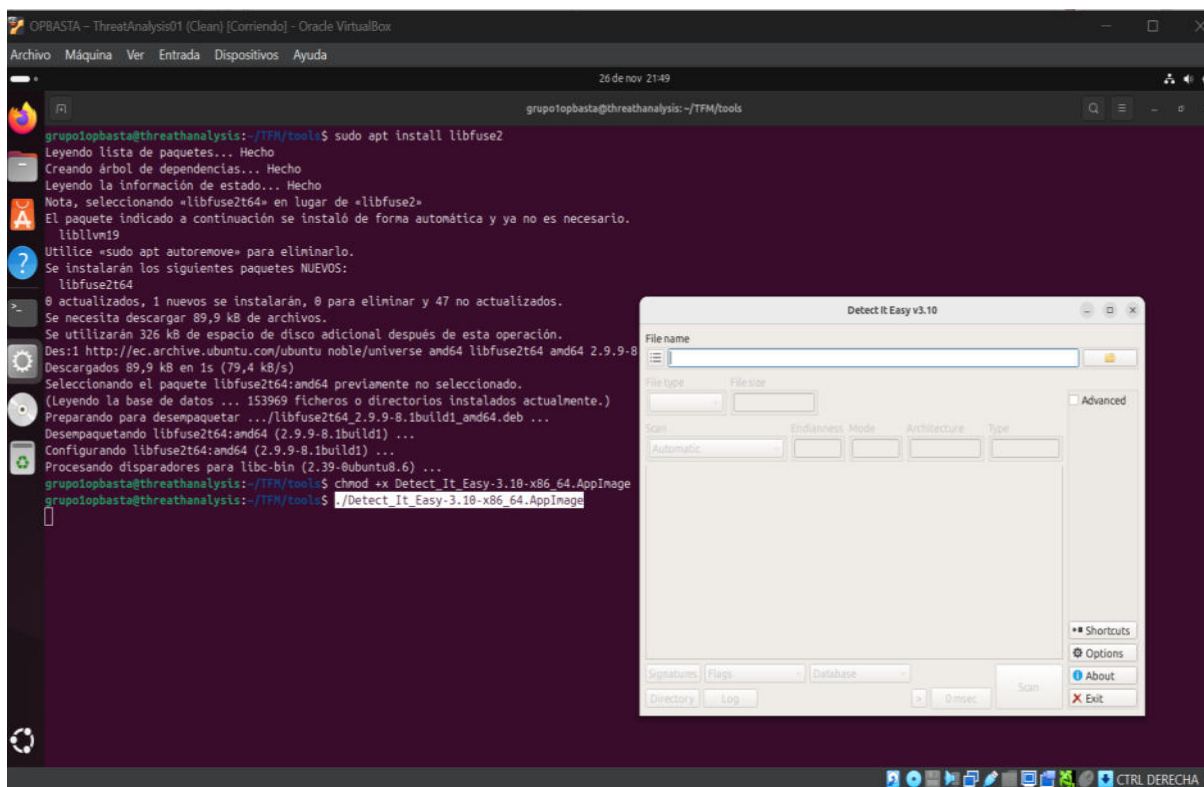
```
OPBASTA - ThreatAnalysis01 (Clean) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
26 de nov 21:48
grupo1opbasta@threathanalysis: ~/TFM/tools
grupo1opbasta@threathanalysis:~/TFM/tools$ sudo apt install libfuse2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Nota, seleccionando «libfuse2t64» en lugar de «libfuse2»
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes NUEVOS:
libfuse2t64
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 47 no actualizados.
Se necesita descargar 89,9 kB de archivos.
Se utilizarán 326 kB de espacio de disco adicional después de esta operación.
Des:1 http://ec.archive.ubuntu.com/ubuntu noble/universe amd64 libfuse2t64 amd64 2.9.9-8.1build1 [89,9 kB]
Descargados 89,9 kB en 1s (79,4 kB/s)
Seleccionando el paquete libfuse2t64:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 153969 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libfuse2t64_2.9.9-8.1build1_amd64.deb ...
Desempaquetando libfuse2t64:amd64 (2.9.9-8.1build1) ...
Configurando libfuse2t64:amd64 (2.9.9-8.1build1) ...
Procesando disparadores para libc-bin (2.39-0ubuntu8.6) ...
grupo1opbasta@threathanalysis:~/TFM/tools$
```

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

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

Figura 58

Ejecución de Detect It Easy en VM OPBASTA - ThreatAnalysis01

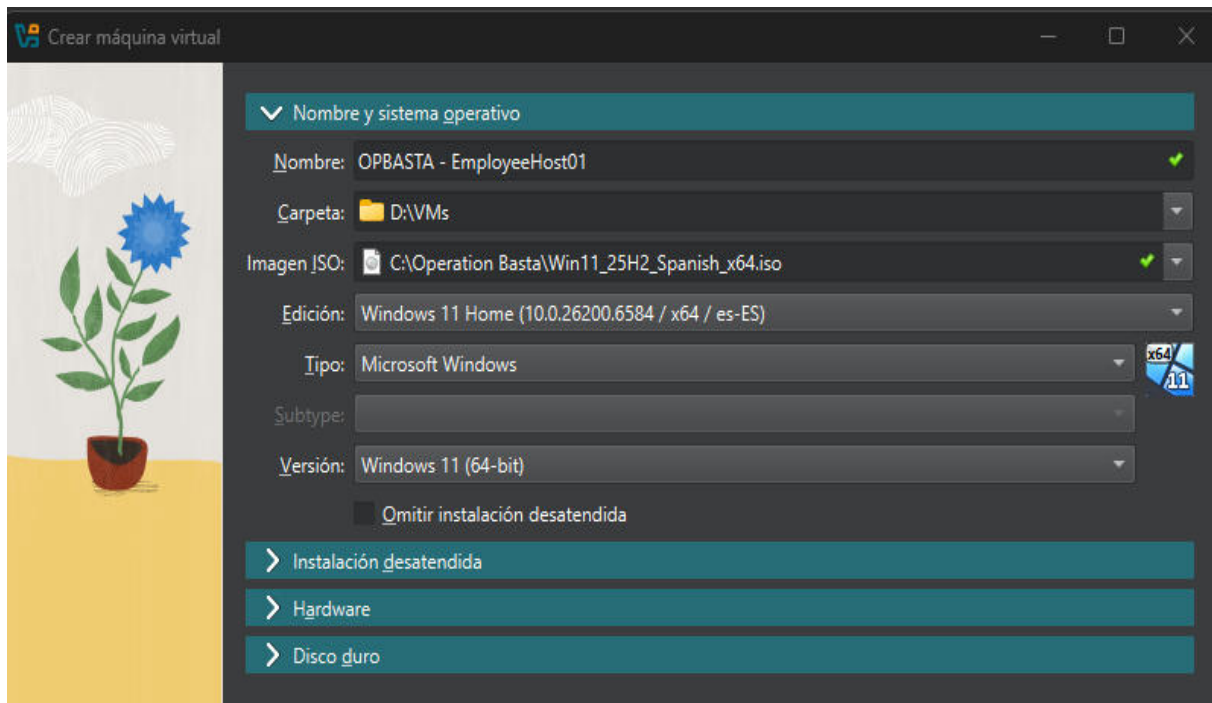


3.1.7.3. VM OPBASTA - EmployeeHost01 (Máquina Víctima). En este equipo se realizó la descarga e instalación del sistema operativo Windows con ISO de Windows 11 versión 25H2, obtenida desde la página oficial de Microsoft: <https://www.microsoft.com/en-us/software-download/windows11>

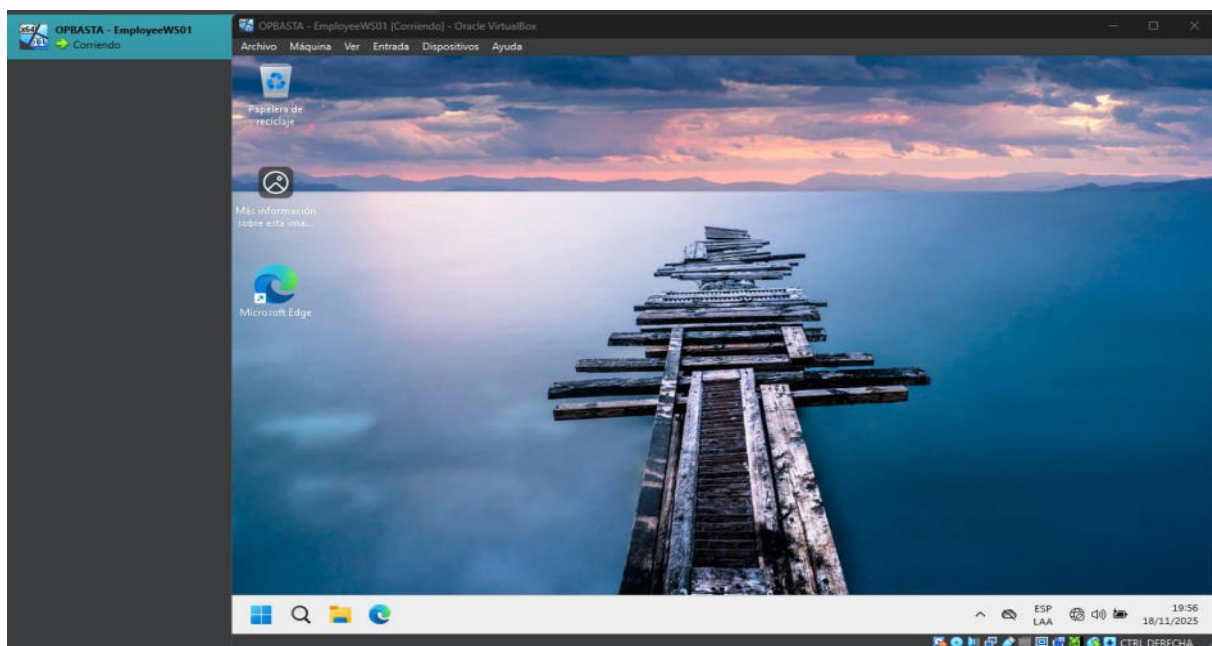
NOTA: En este punto se usó una configuración de red en modo adaptador en NAT para descargar desde Internet las herramientas necesarias a fin de adecuar la máquina; posteriormente, se configuró todo el laboratorio en una red interna virtual.

Figura 59

Configuración de la máquina virtual Windows para VM OPBASTA - EmployeeHost01

**Figura 60**

Máquina virtual Windows VM OPBASTA - EmployeeHost01 instalada en VirtualBox



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

Figura 61

Desactivación de protecciones de Windows en VM OPBASTA - EmployeeHost01

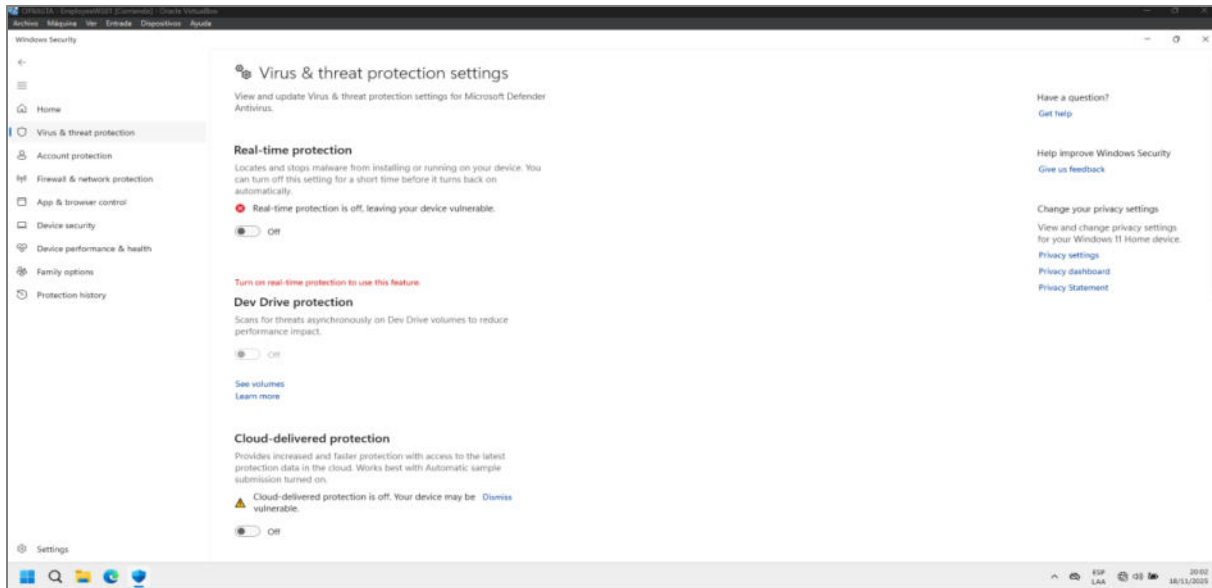
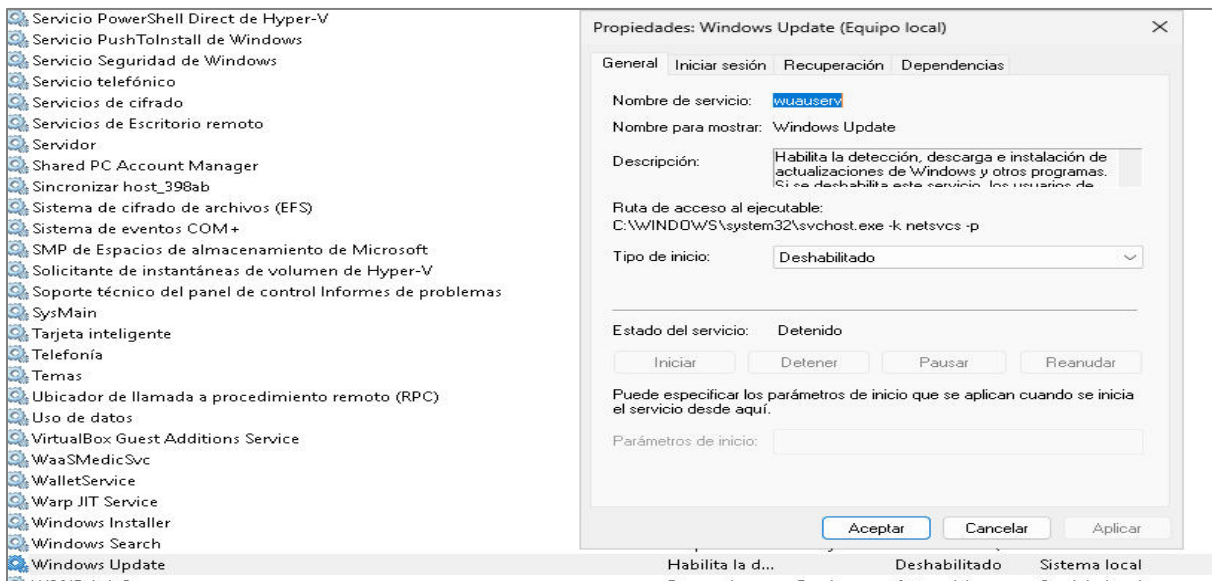


Figura 62

Desactivación de actualizaciones de Windows Update en VM OPBASTA - EmployeeHost01

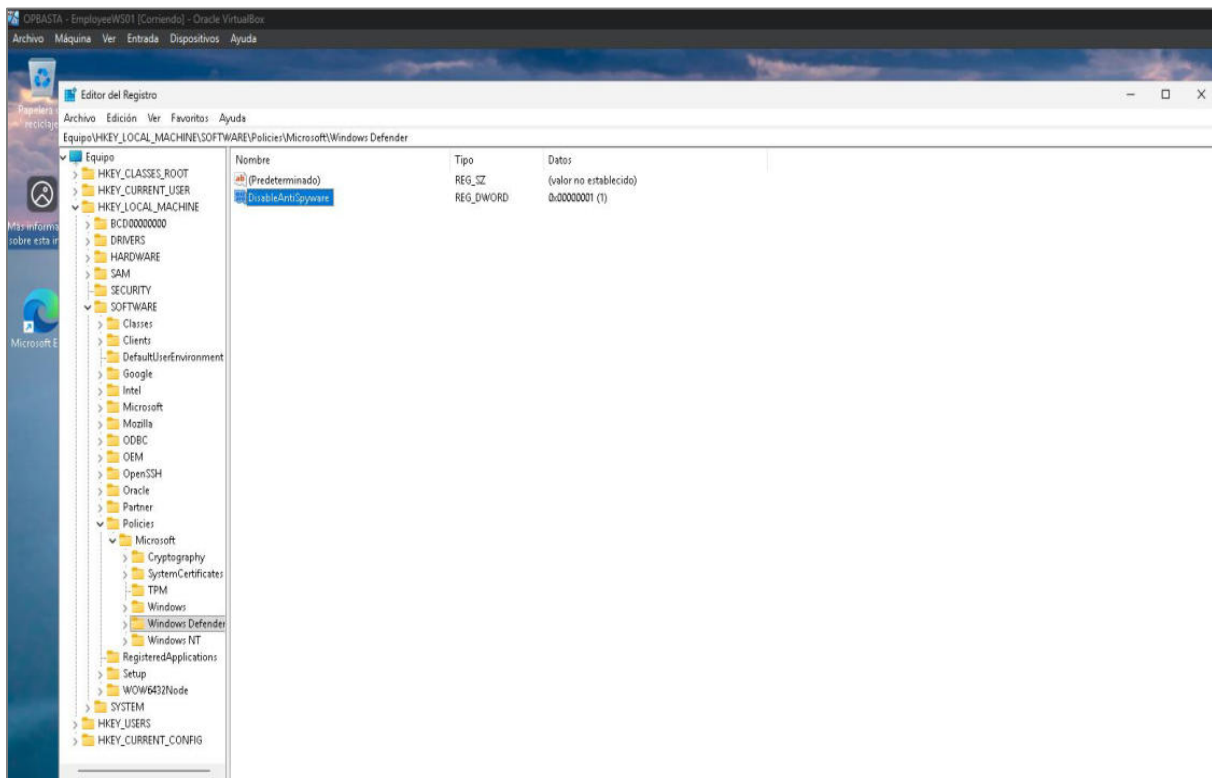


Para desactivar permanentemente la protección de Windows Defender y que no se vuelva a iniciar cuando se apague el equipo, se modificó el registro

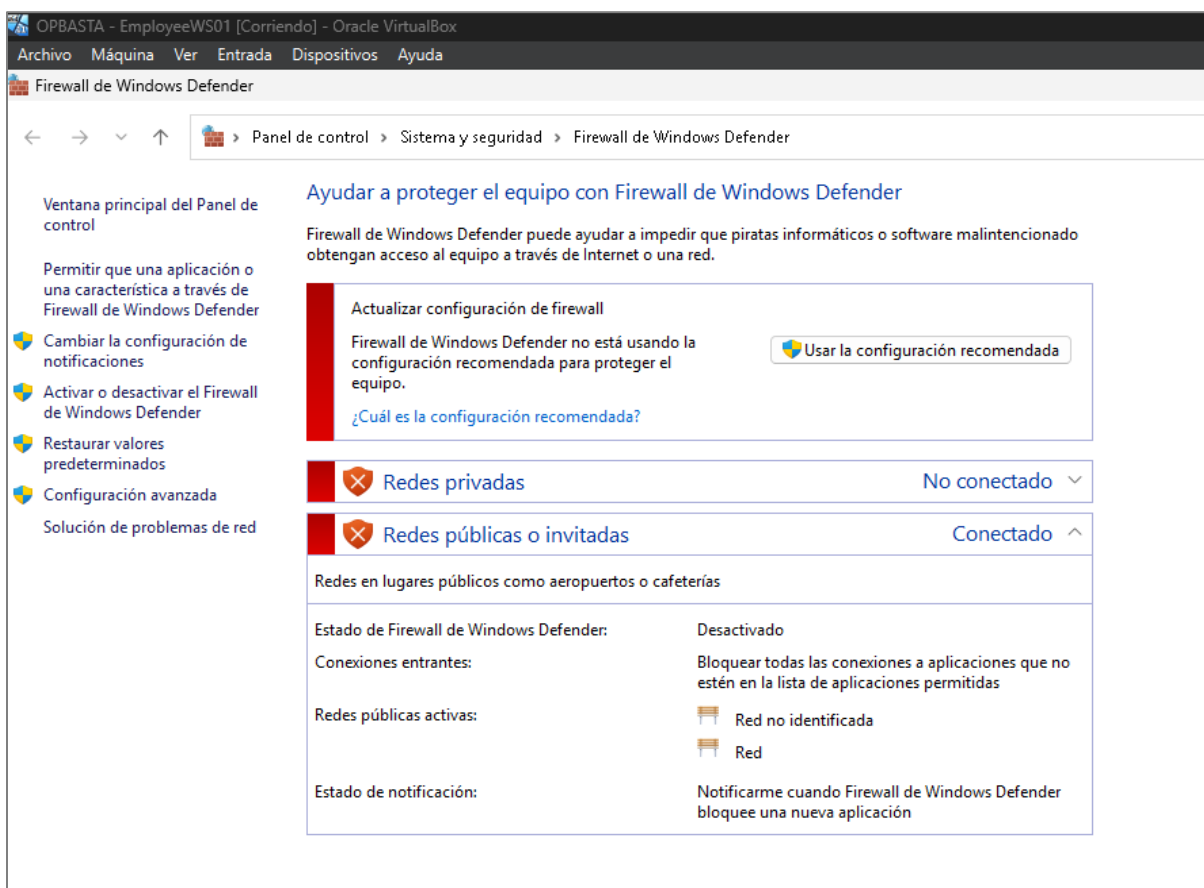
“HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender” añadiendo una clave de 32 bits llamada “DisableAntiSpyware” y se asignó el valor “1”.

Figura 63

Desactivación de Windows Defender desde el Editor de Registro



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

Figura 64*Desactivación del Firewall de Windows Defender*

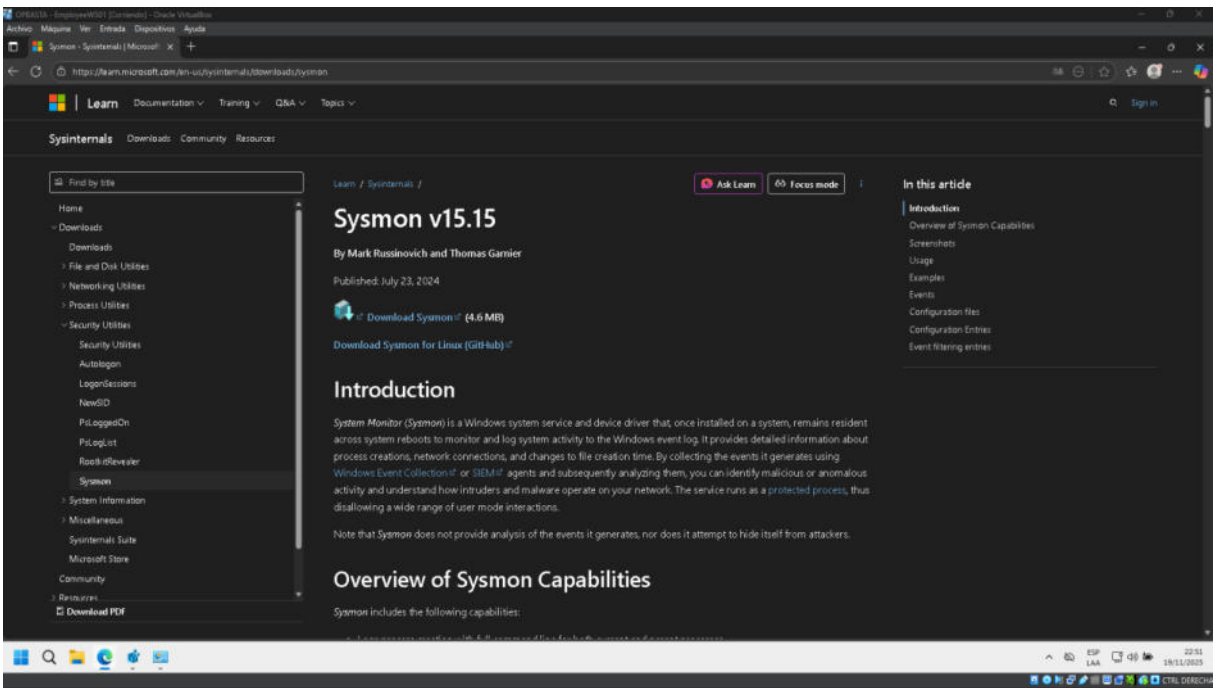
A continuación, se procedió con la instalación de todas las herramientas necesarias para ejecutar un análisis en mayor profundidad cuando el ransomware Black Basta sea ejecutado.

Como primera herramienta, se descargó Sysmon desde la web oficial:

<https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon> ya que esta herramienta brinda telemetría en profundidad que permite conocer lo que está sucediendo por detrás en el OS, lo cual por defecto no se guarda en registros de Windows.

Figura 65

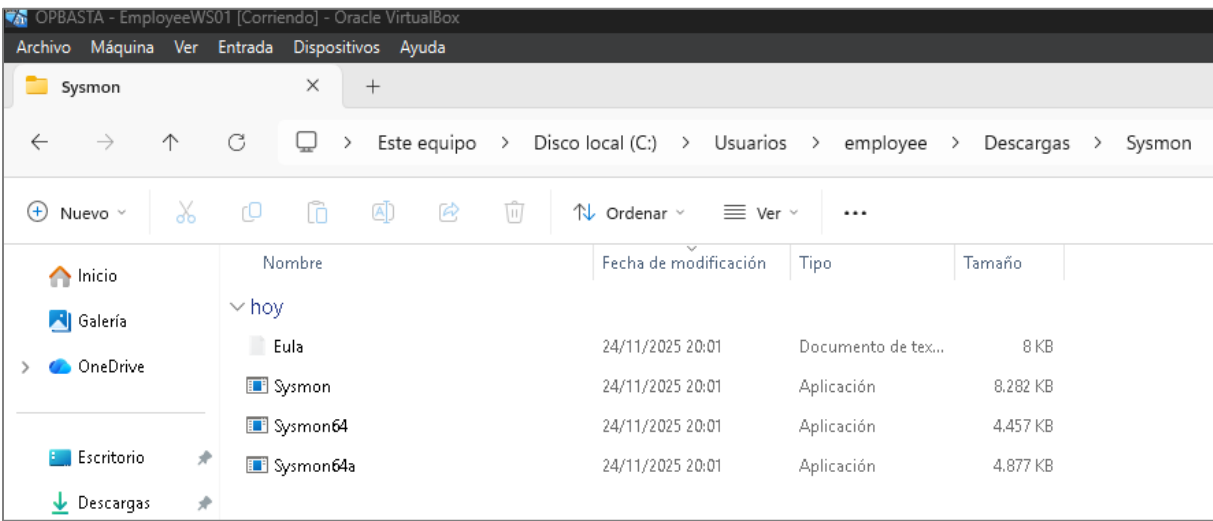
Descarga de herramienta Sysmon



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

Figura 66

Vista del archivo de Sysmon descomprimido

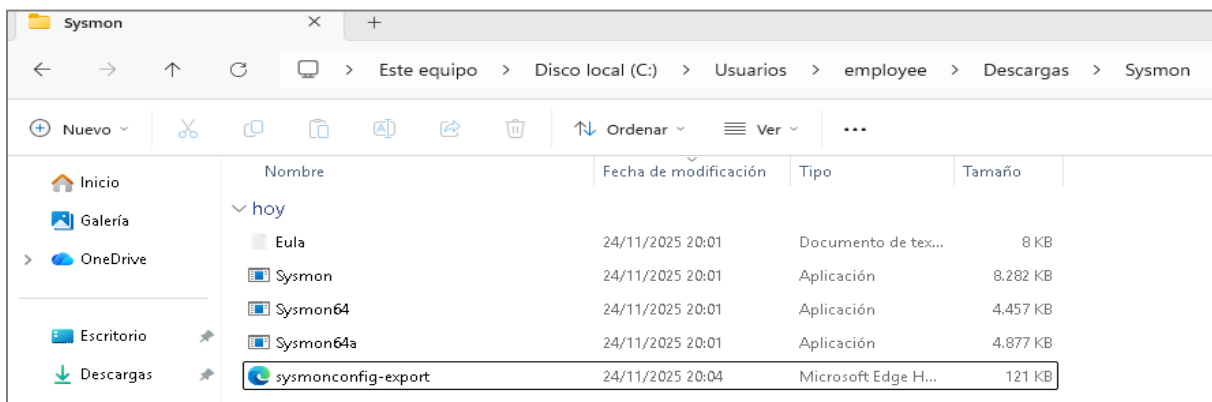


Antes de la instalación Sysmon se descargó un archivo preconfigurado, el cual ya tiene establecido parámetros para detectar los eventos de seguridad más importantes y evitar ruido; para este proyecto se escogió el archivo más utilizado en la comunidad, el enlace es el siguiente:

<https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml>

Figura 67

Descarga del archivo de configuración de Sysmon



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

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

Figura 68

Instalación de Sysmon de acuerdo con archivo de configuración

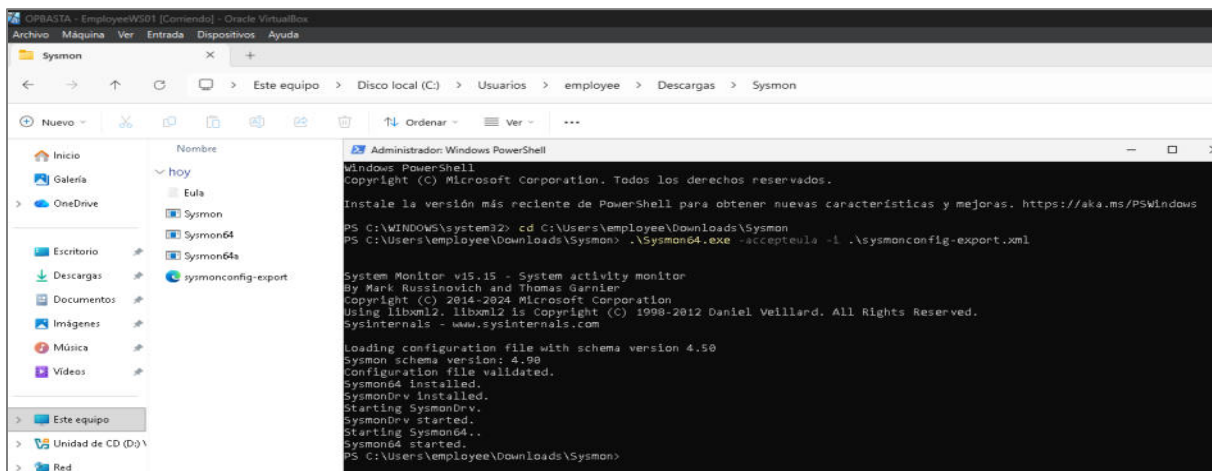
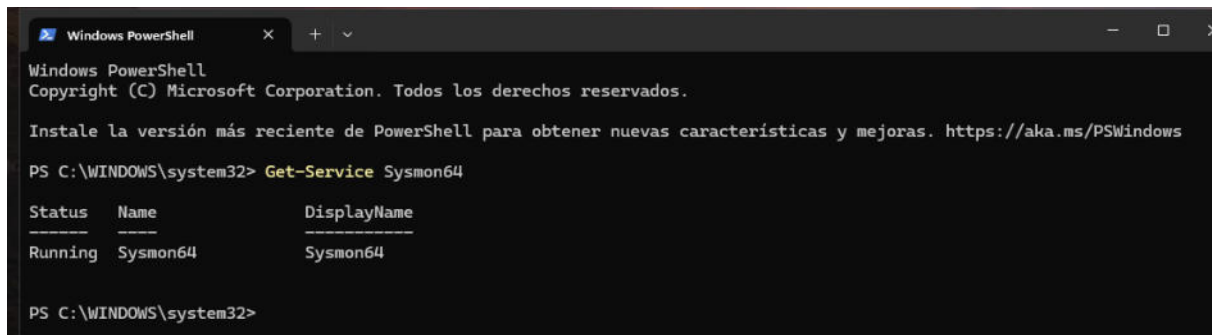


Figura 69

Verificación de instalación de Sysmon



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> Get-Service Sysmon64

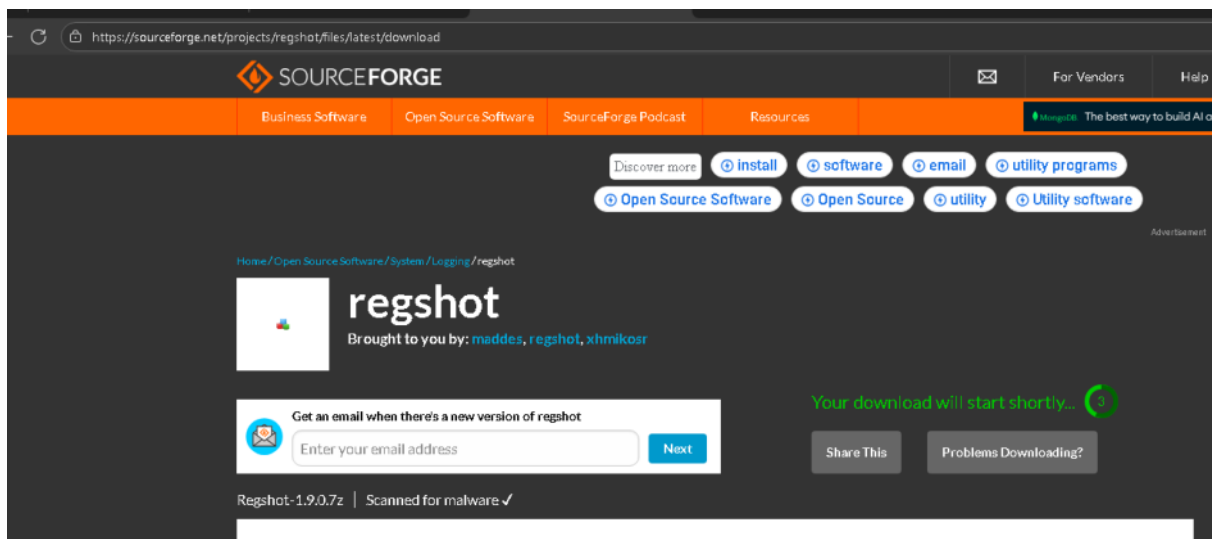
Status      Name            DisplayName
-----
Running     Sysmon64        Sysmon64

PS C:\WINDOWS\system32>
```

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

Figura 70

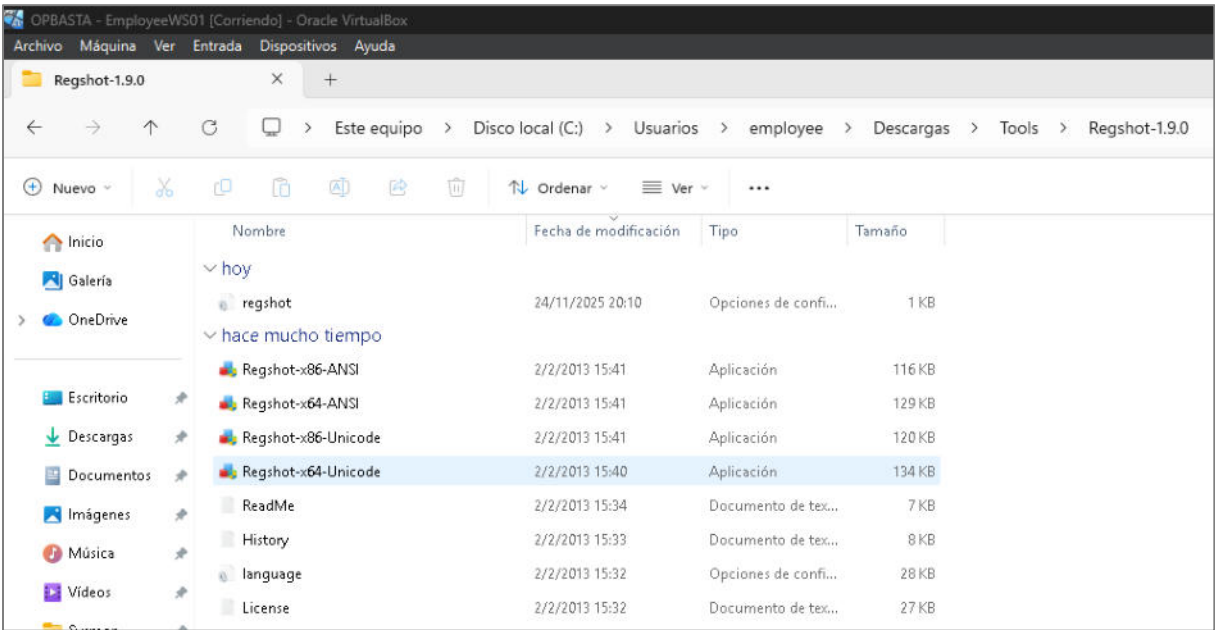
Descarga de herramienta Regshot



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

Figura 71

Vista del archivo Regshot descomprimido

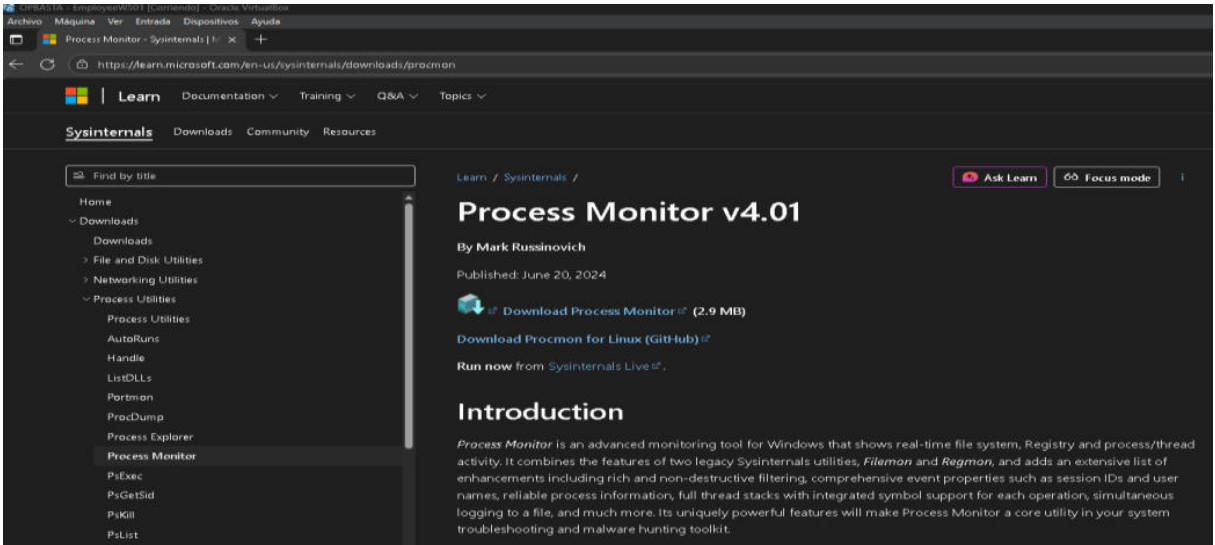


Como tercera herramienta, se descargó Procmon desde su página oficial:

<https://learn.microsoft.com/en-us/sysinternals/downloads/procmon> para tener visibilidad a nivel de procesos del OS.

Figura 72

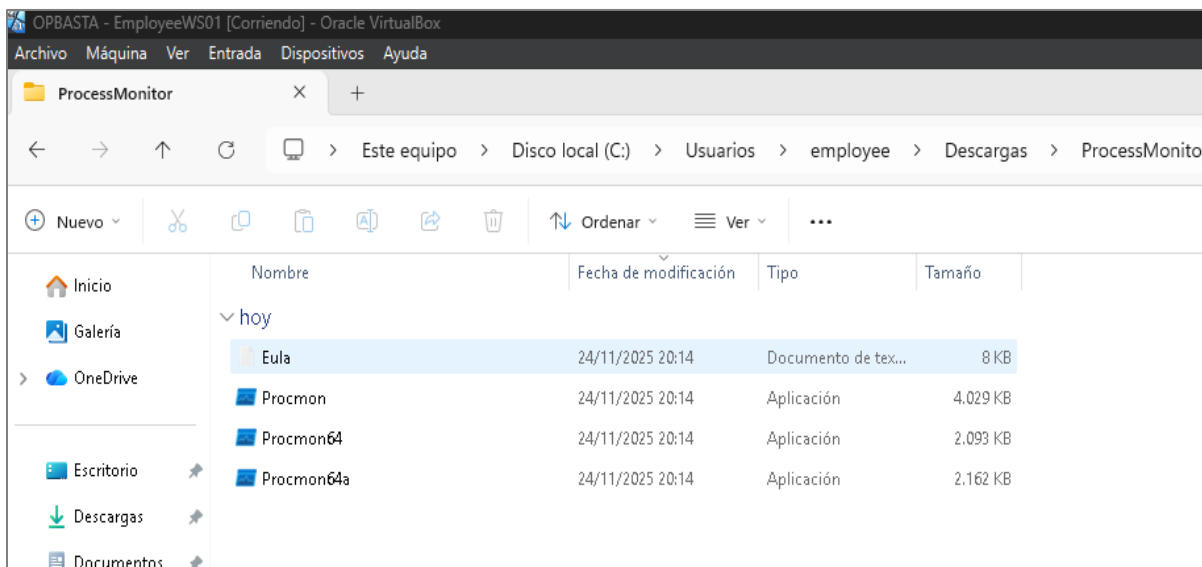
Descarga de herramienta Procmon



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

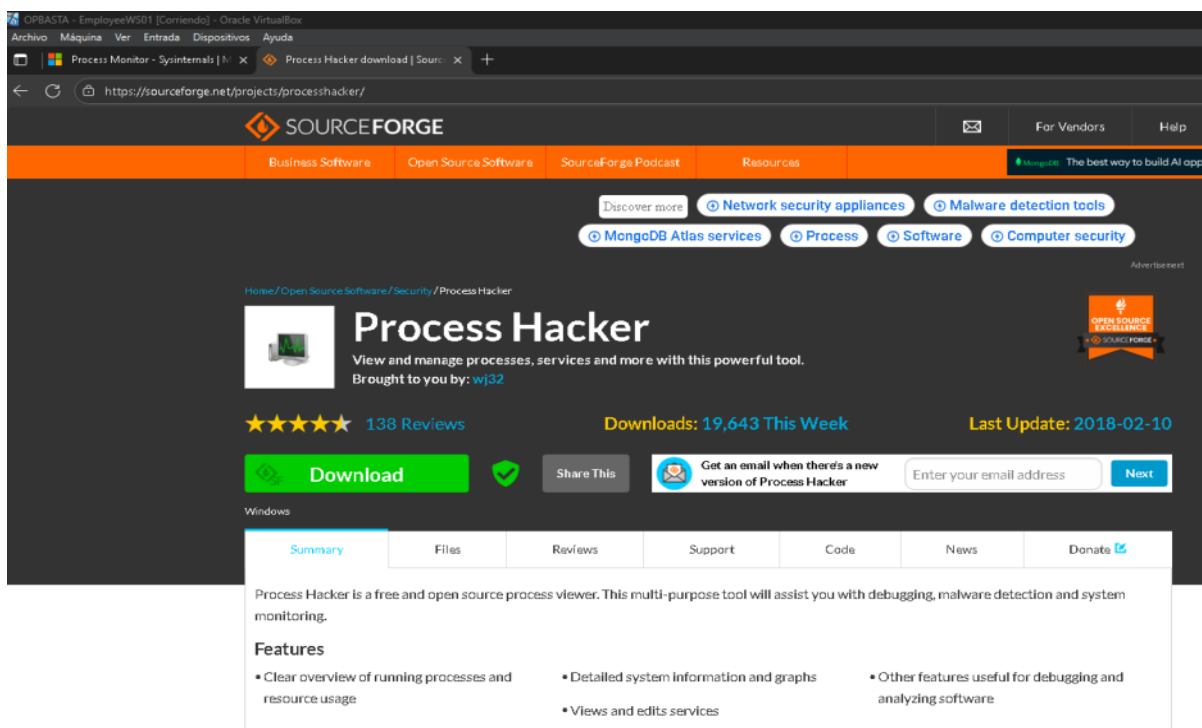
Figura 73

Vista del archivo Procmon descomprimido

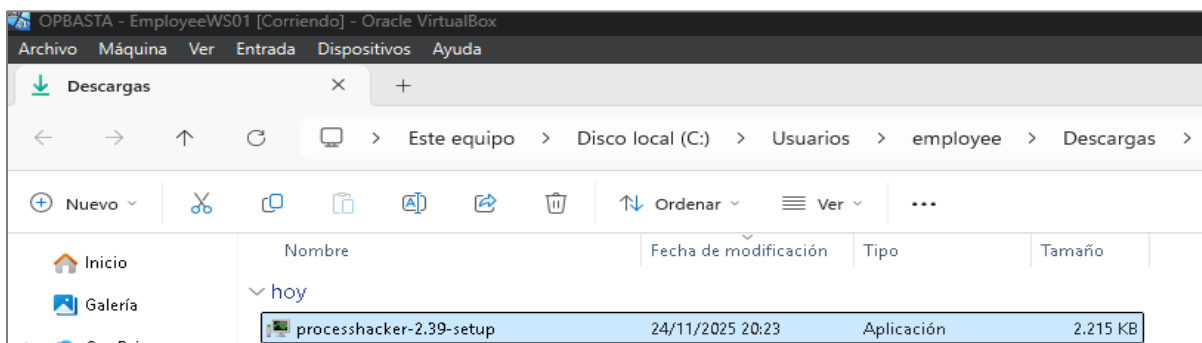


Como cuarta herramienta, se descargó Process Hacker desde su página oficial:

<https://sourceforge.net/projects/processhacker/> debido a que ayuda al análisis de procesos en tiempo real distinguiendo entre procesos padres e hijos, como también los cambios con su origen.

Figura 74*Descarga de herramienta Process Hacker*

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

Figura 75*Ejecución del archivo descargado processhacker-2.39-setup*

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

Figura 76

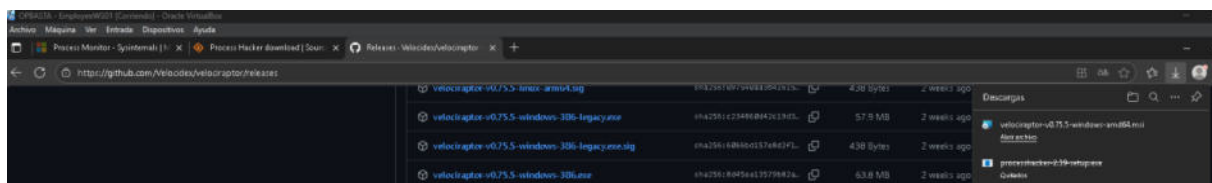
Instalación de herramienta Process Hacker



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

Figura 77

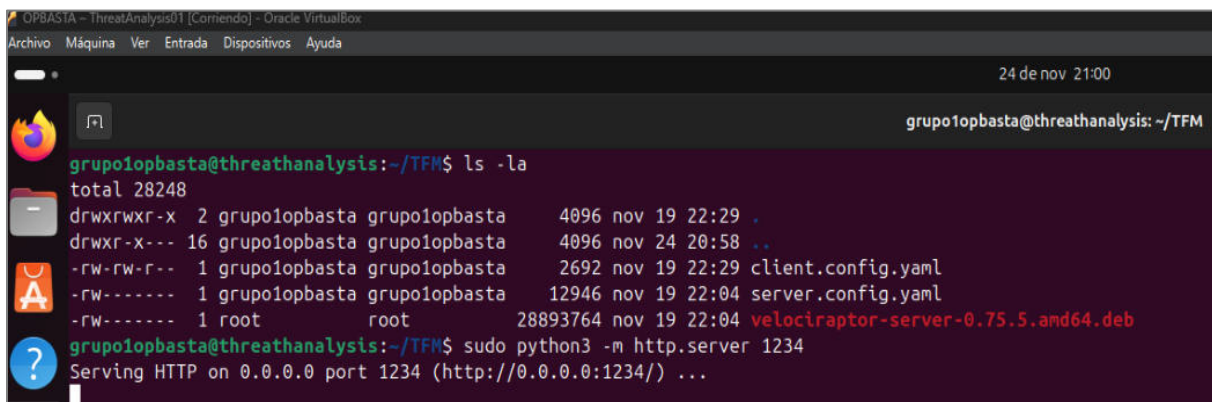
Descarga de Velociraptor para Windows



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

Figura 78

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



```
OPBASTA - ThreatAnalysis01 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
24 de nov 21:00
grupo1opbasta@threatanalysis: ~/TFM
grupo1opbasta@threatanalysis:~/TFM$ ls -la
total 28248
drwxrwxr-x  2 grupo1opbasta grupo1opbasta  4096 nov 19 22:29 .
drwxr-x--- 16 grupo1opbasta grupo1opbasta  4096 nov 24 20:58 ..
-rw-rw-r--  1 grupo1opbasta grupo1opbasta  2692 nov 19 22:29 client.config.yaml
-rw-----  1 grupo1opbasta grupo1opbasta 12946 nov 19 22:04 server.config.yaml
-rw-----  1 root          root          28893764 nov 19 22:04 velociraptor-server-0.75.5.amd64.deb
grupo1opbasta@threatanalysis:~/TFM$ sudo python3 -m http.server 1234
Serving HTTP on 0.0.0.0 port 1234 (http://0.0.0.0:1234/) ...
```

Figura 79

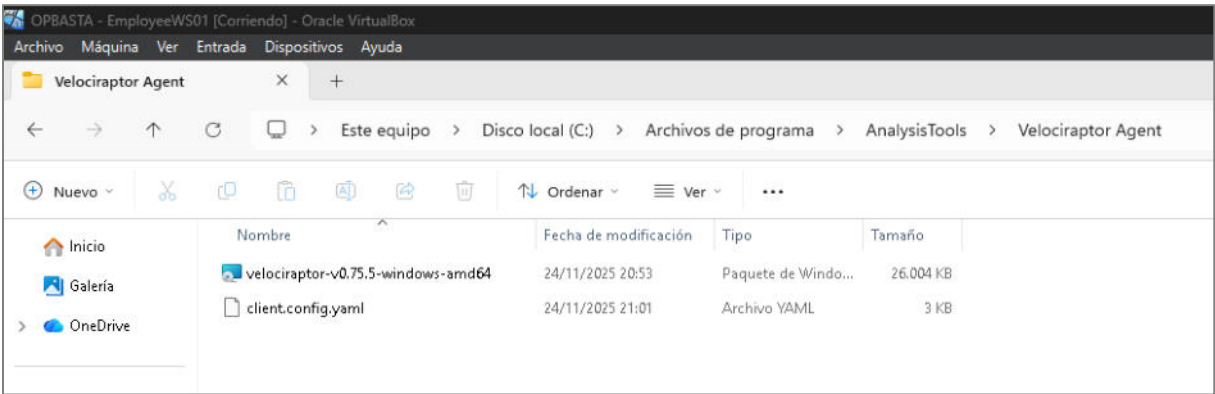
Vista del servidor con los archivos a descargar



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

Figura 80

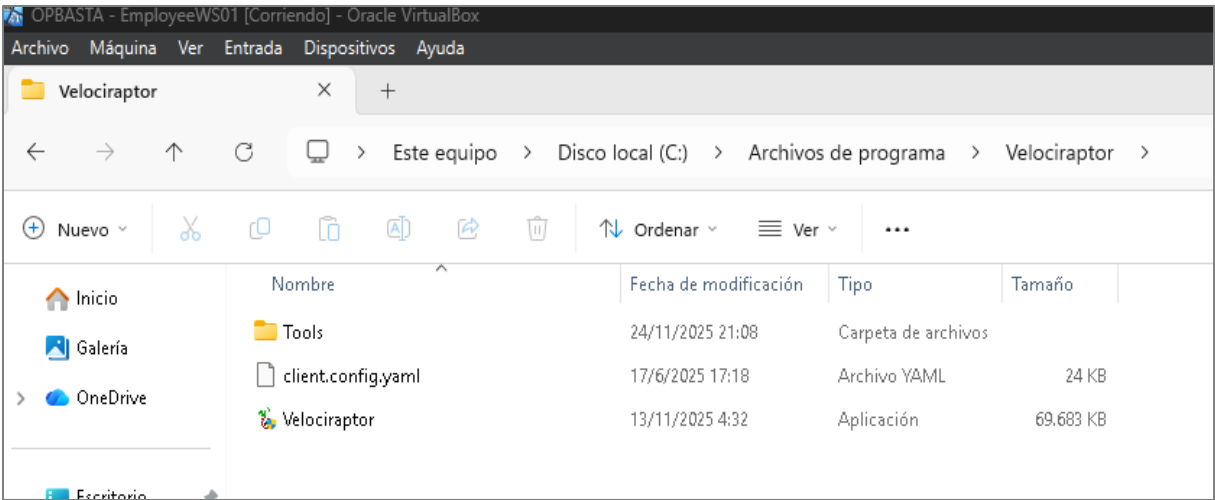
Archivos descargados desde el servidor de transferencia



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

Figura 81

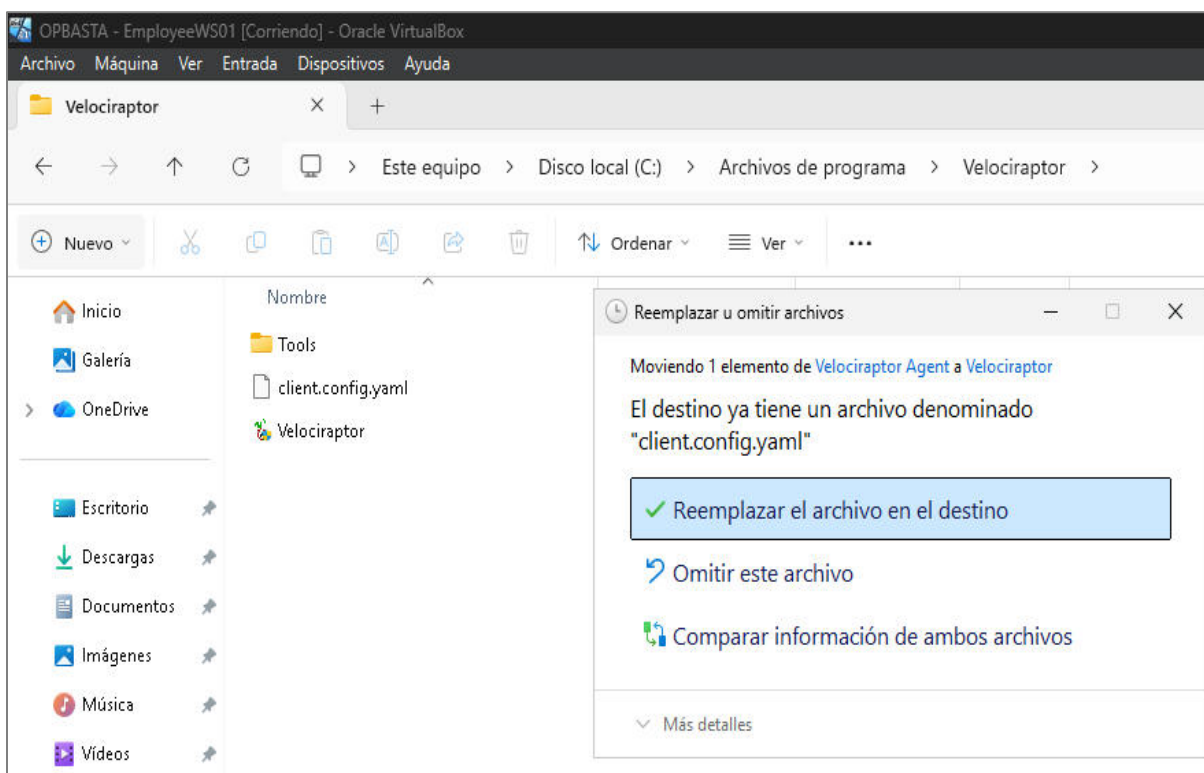
Instalación del cliente de Velociraptor



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

Figura 82

Reemplazo del archivo client.config.yaml



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

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

Figura 83

Visualización del cliente de Velociraptor desde la interfaz gráfica del servidor

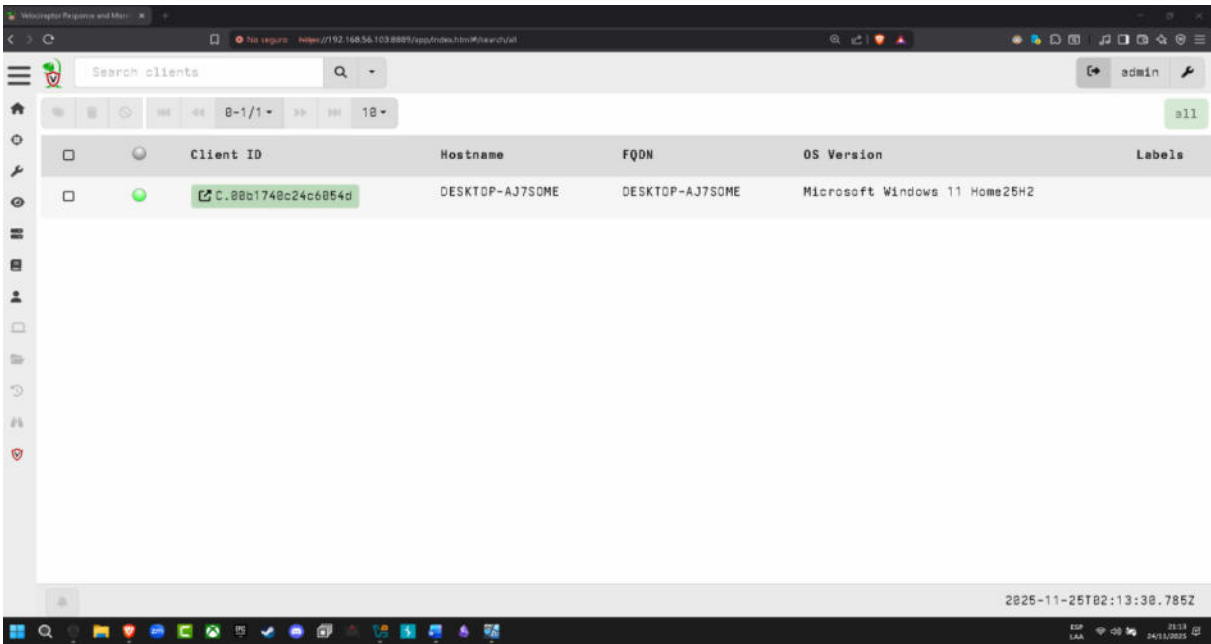
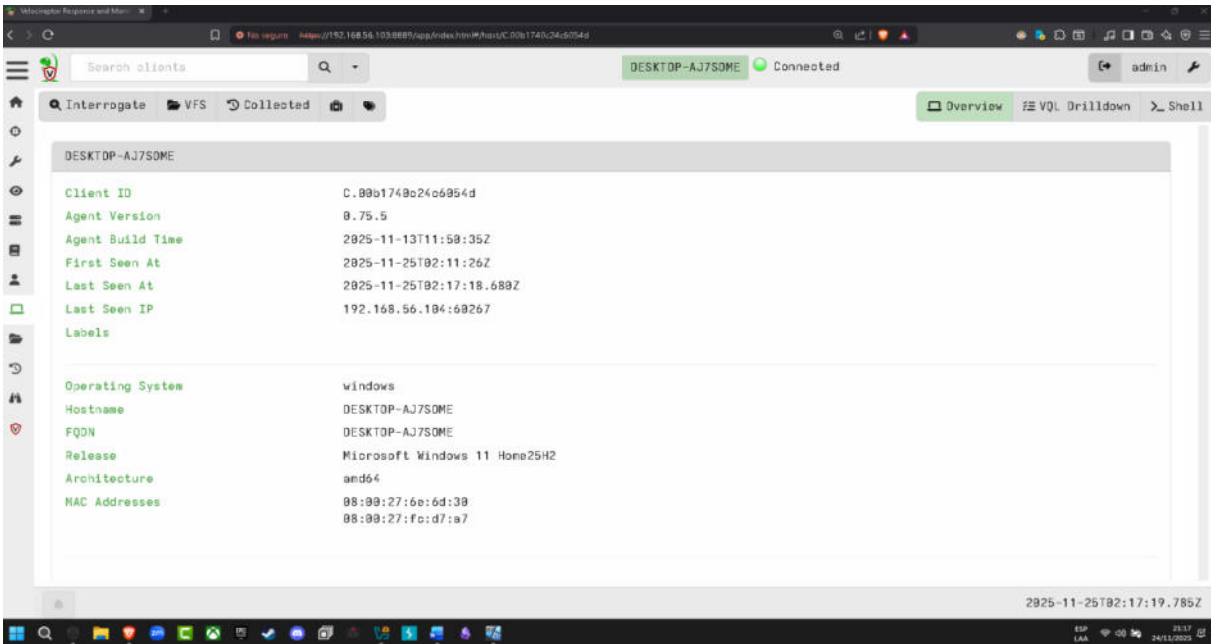


Figura 84

Reporte entregado por el cliente de Velociraptor sobre la máquina Windows

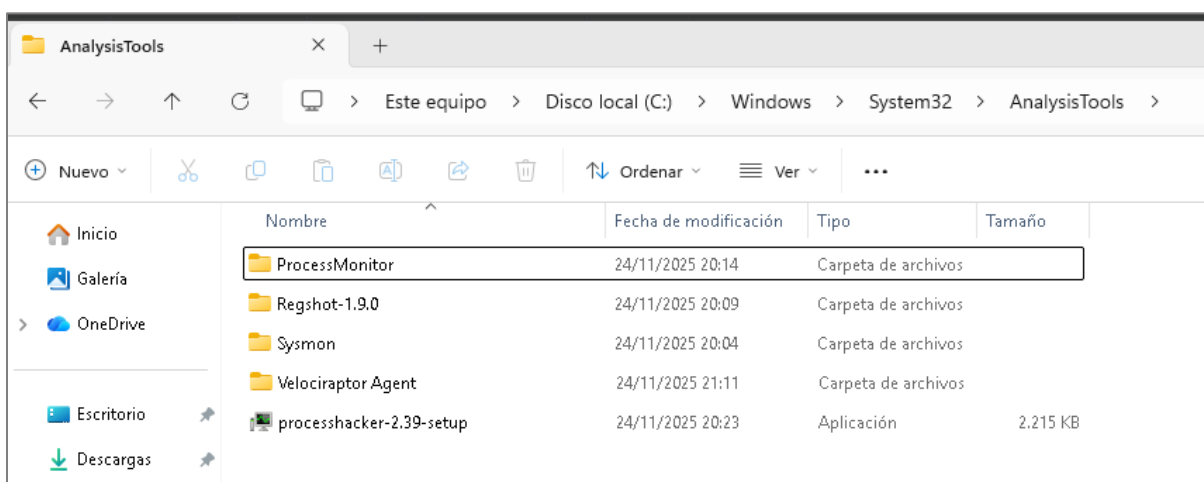


Continuando con la implementación de la máquina víctima VM OPBASTA - EmployeeHost01, una vez que se culminó con la instalación de todas las herramientas de análisis y recolección necesarias, éstas fueron reubicadas hacia el directorio “C:\Windows\System32\AnalysisTools” para evitar que el ransomware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

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

Figura 85

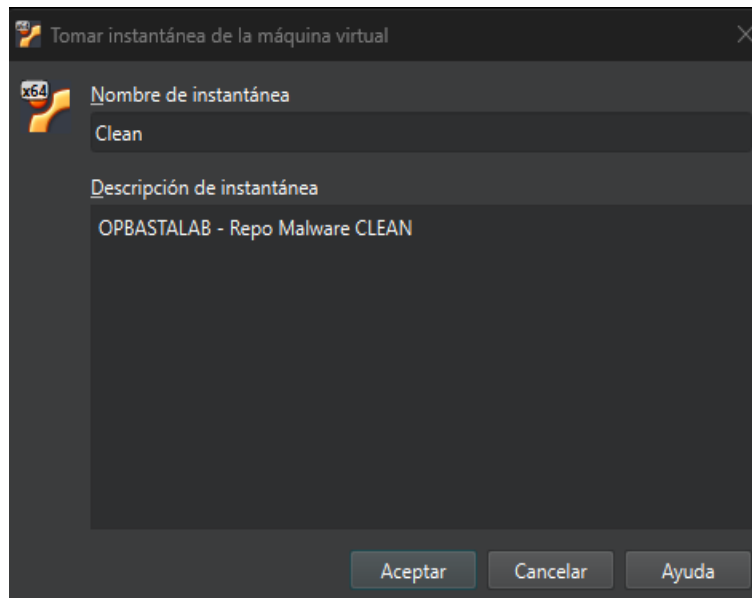
Herramientas de análisis reubicadas en AnalysisTools de System32



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

Figura 86

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

**Figura 87**

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

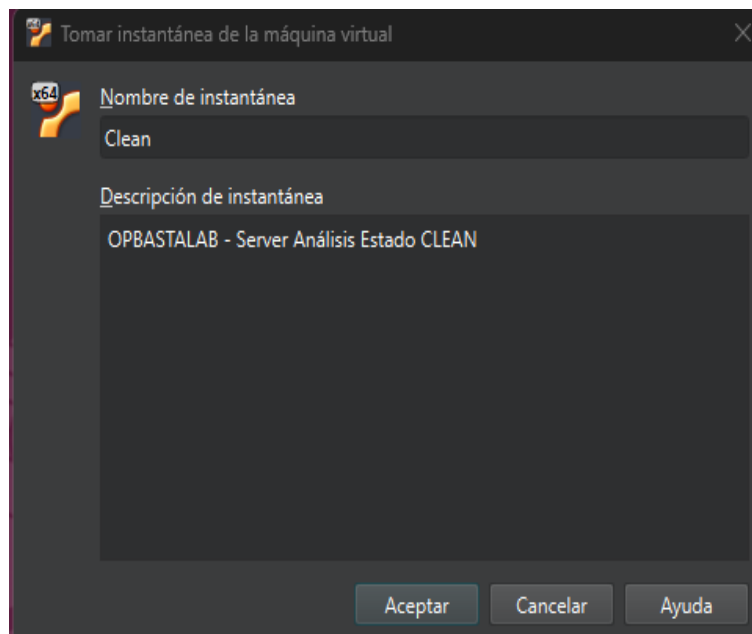
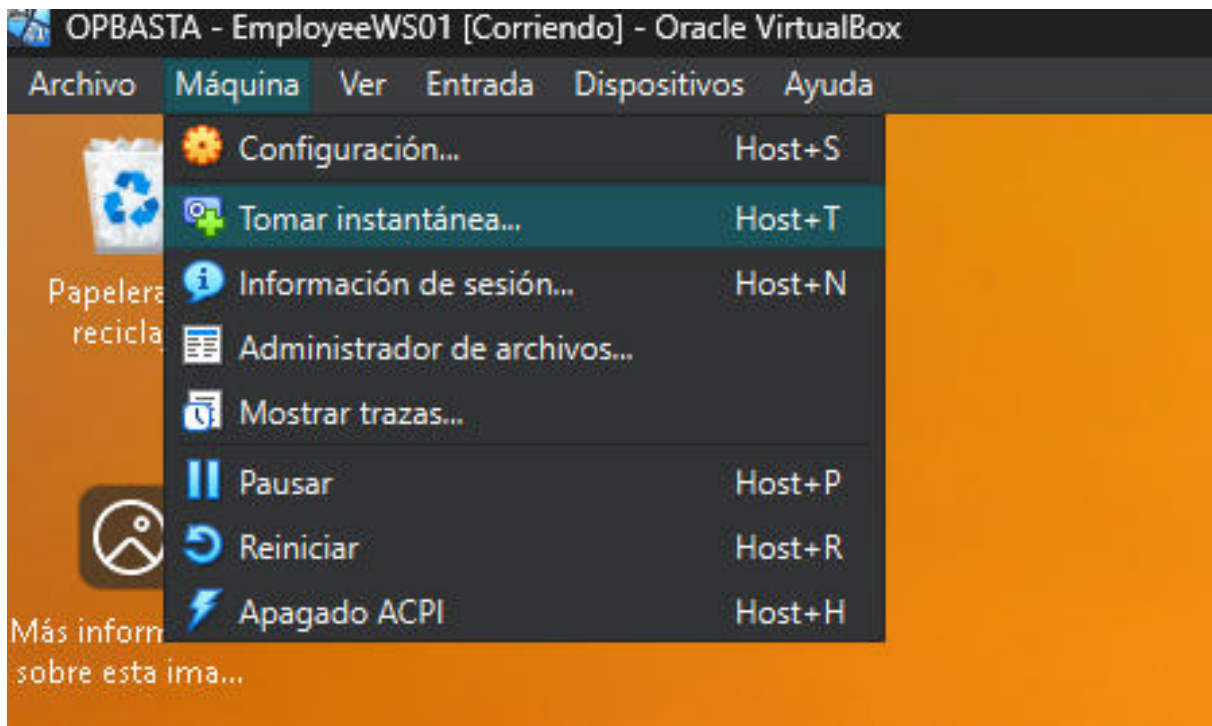
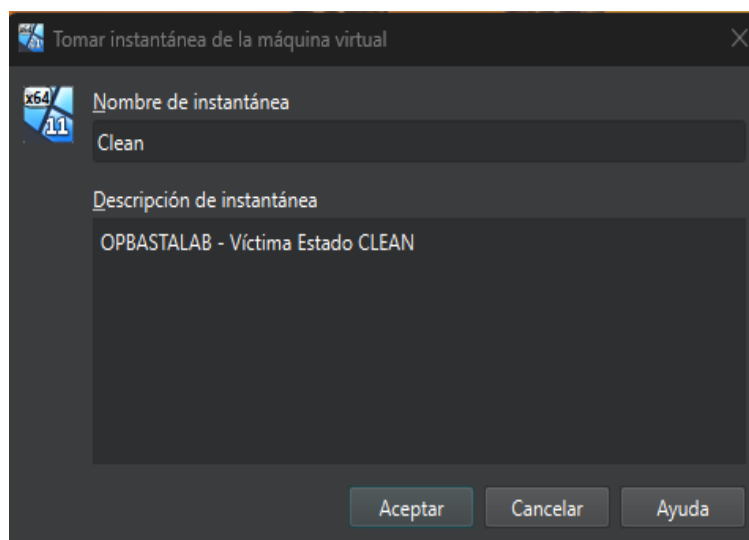


Figura 88

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

**Figura 89**

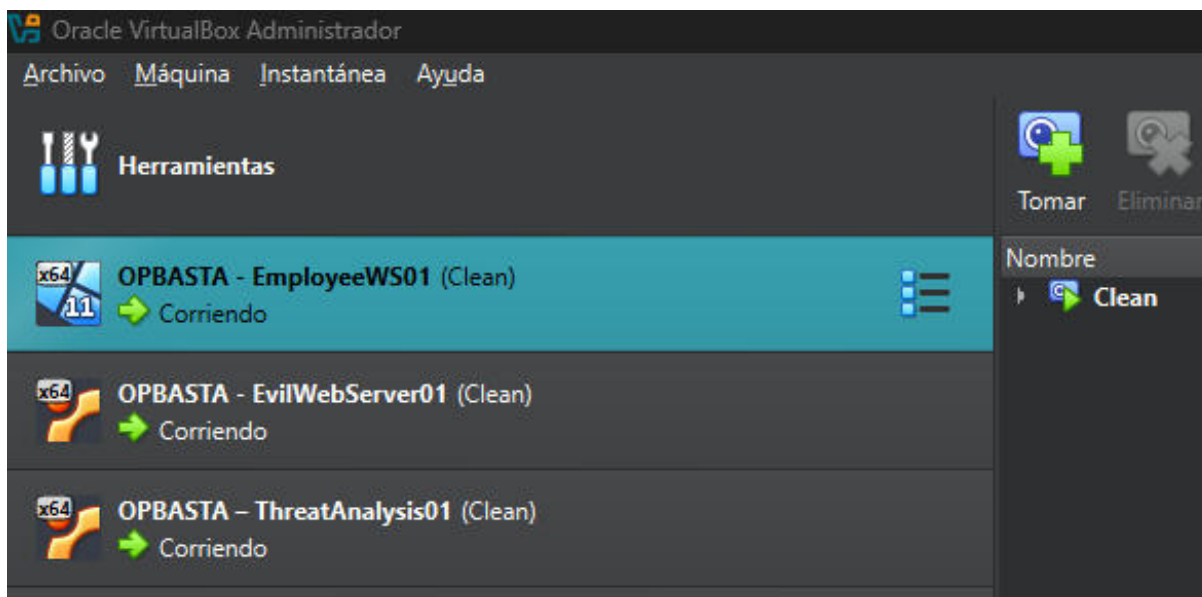
Snapshot de VM OPBASTA - EmployeeHost01 previo a la ejecución del ransomware (parte 2)



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

Figura 90

Laboratorio completo con las tres máquinas

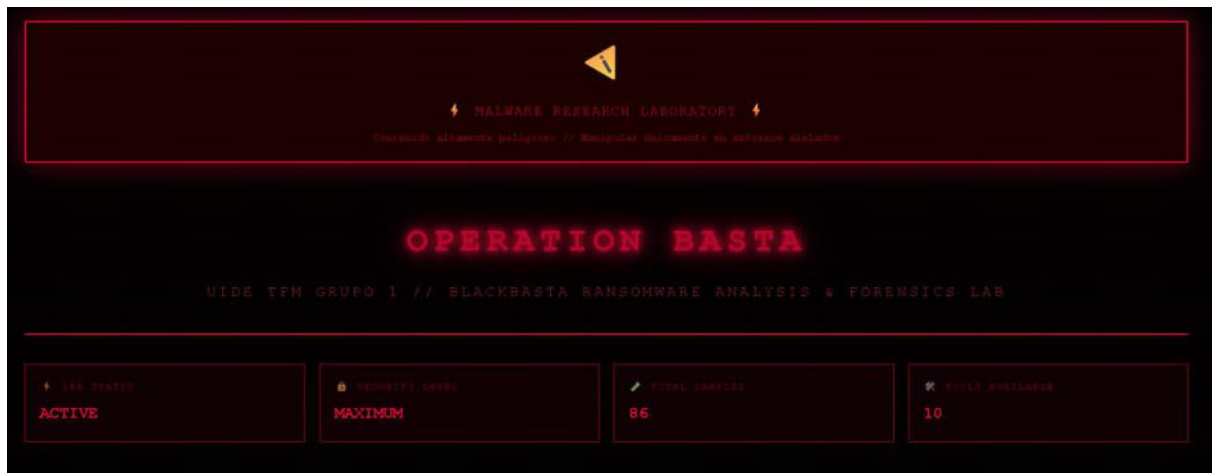


3.1.7.4. Portal “Operation Basta” - Plataforma Integrada de Distribución Segura de Muestras y Herramientas Forenses (Valor Agregado). Como parte del ecosistema desarrollado para el análisis controlado del ransomware Black Basta, se implementó un portal web interno denominado “Operation Basta”. El objetivo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

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

Figura 91

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



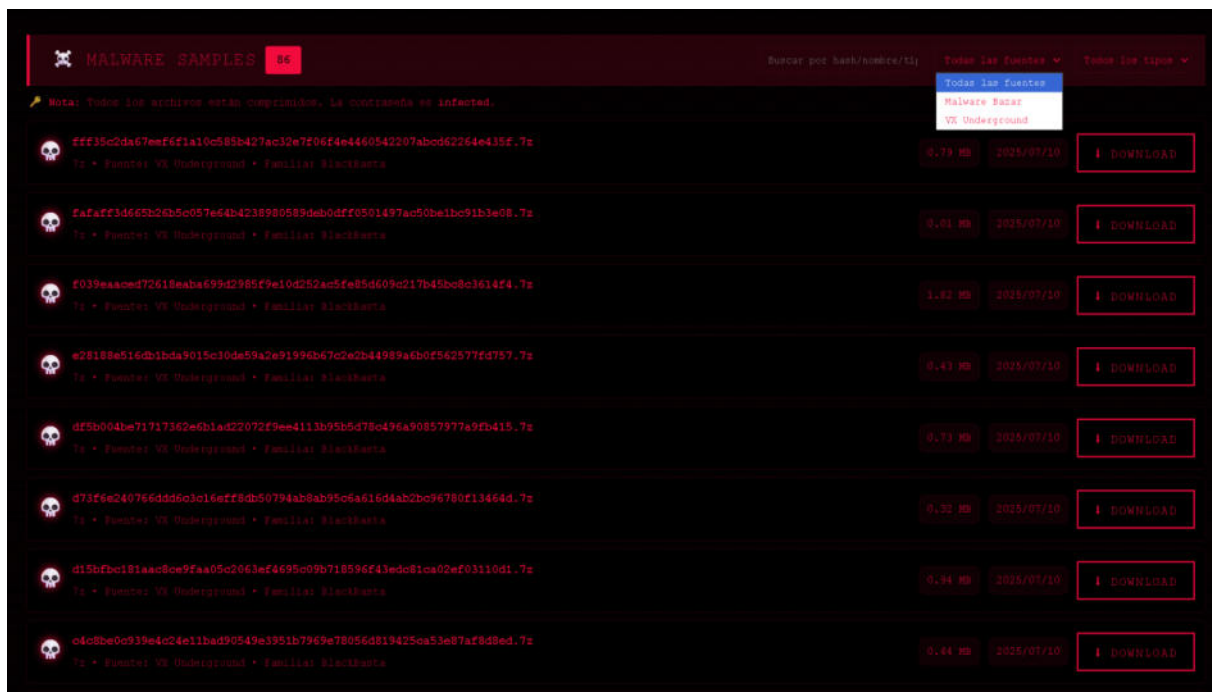
Este servidor web no cumple únicamente un rol de conveniencia; sino que, la finalidad es garantizar que todo el proceso de análisis y el flujo de trabajo se mantengan 100% offline y aislados, evitando dependencias de Internet y reforzando la seguridad operativa del laboratorio. Este almacenamiento local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

La plataforma desplegada organiza los recursos en tres secciones fundamentales:

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

Figura 92

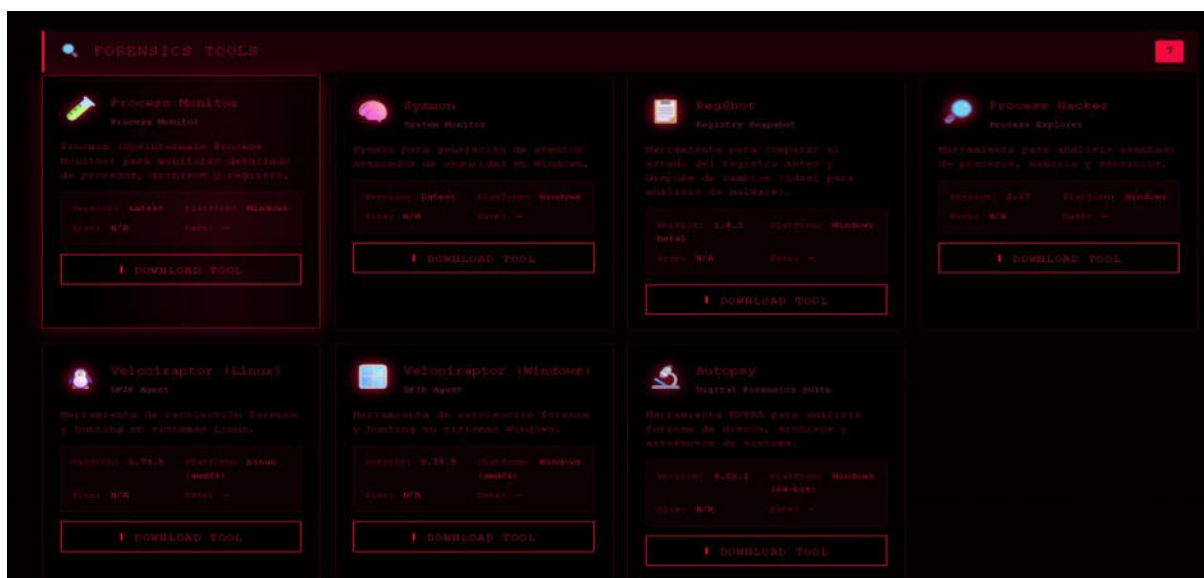
Portal “Operation Basta”- Lista de muestras de malware disponibles a descargar



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

Figura 93

Portal “Operation Basta”- Sección de herramientas forenses disponibles

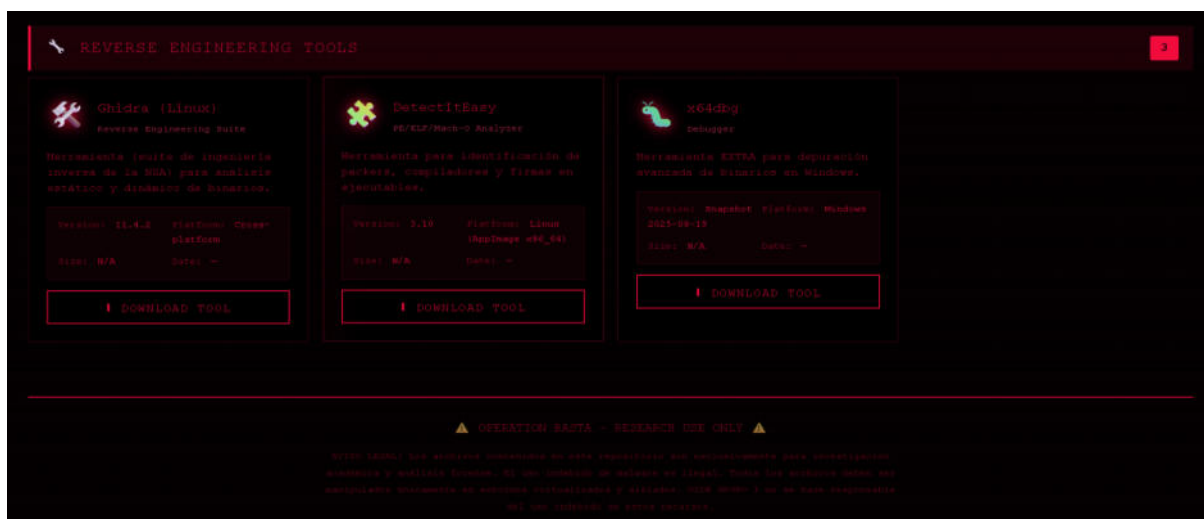


- **Herramientas de Ingeniería Inversa:** Adicionalmente, se integraron

herramientas críticas como Ghidra, Detect It Easy y x64dbg. El objetivo es proporcionar un entorno completo para análisis estático, dinámico y de empaquetamiento. Todas las herramientas están listas para su uso sin necesidad de conexiones externas.

Figura 94

Portal “Operation Basta”- Sección de herramientas de ingeniería inversa disponibles



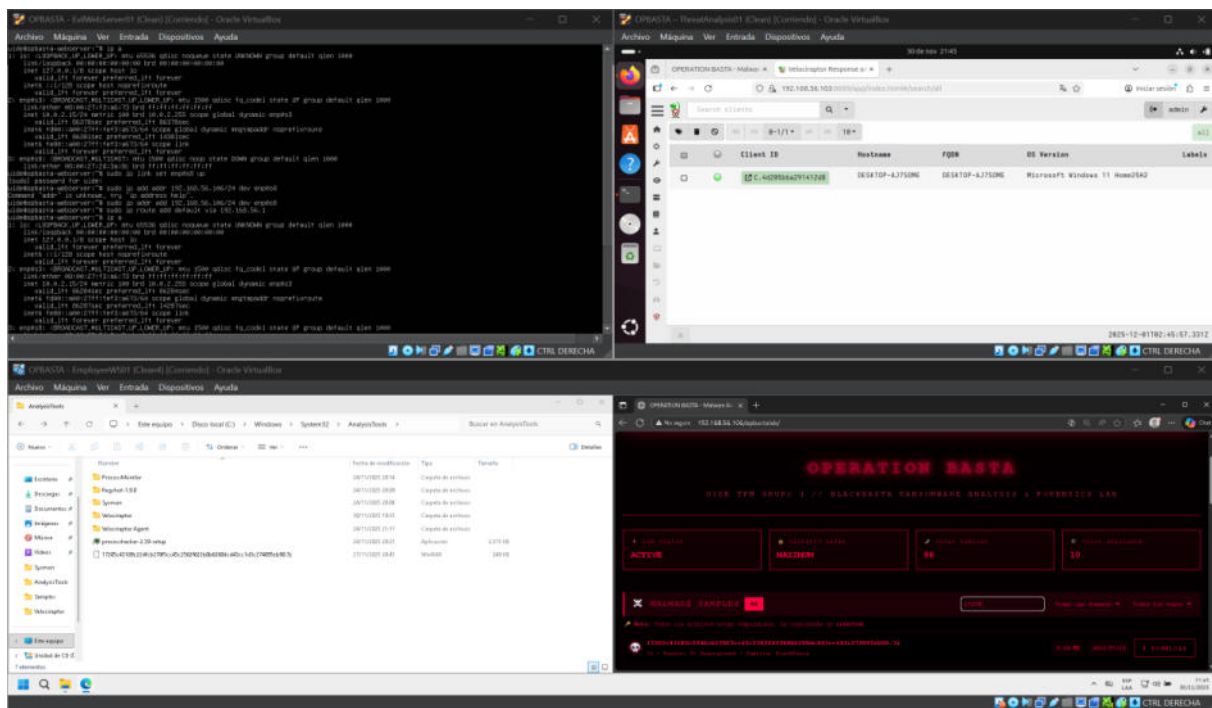
Como se pudo observar en las imágenes anteriores, el diseño visual del portal sigue una estética tipo “Malware Research Lab”, lo cual refuerza su identidad como plataforma interna de análisis avanzado. Además, el portal incluye advertencias explícitas sobre los riesgos y obligaciones legales del manejo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

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

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

Figura 95

Vista del laboratorio completo y operativo



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

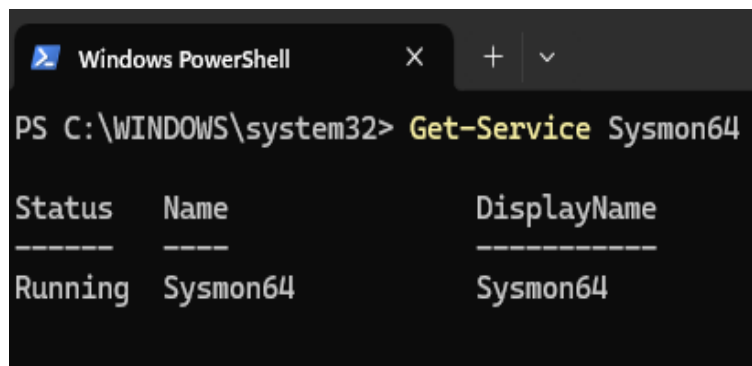
3.2. Ejecución del Ransomware Black Basta en VM OPBASTA - EmployeeHost01

3.2.1. Preparación de VM OPBASTA - EmployeeHost01 Antes de la Ejecución del Ransomware. En primer lugar, se verificó que Sysmon esté ejecutándose correctamente, para esto desde la ventana de PowerShell con permisos de administrador se ingresó el siguiente comando:

```
Get-Service Sysmon64
```

Figura 96

Verificación de que Sysmon se encuentra activo



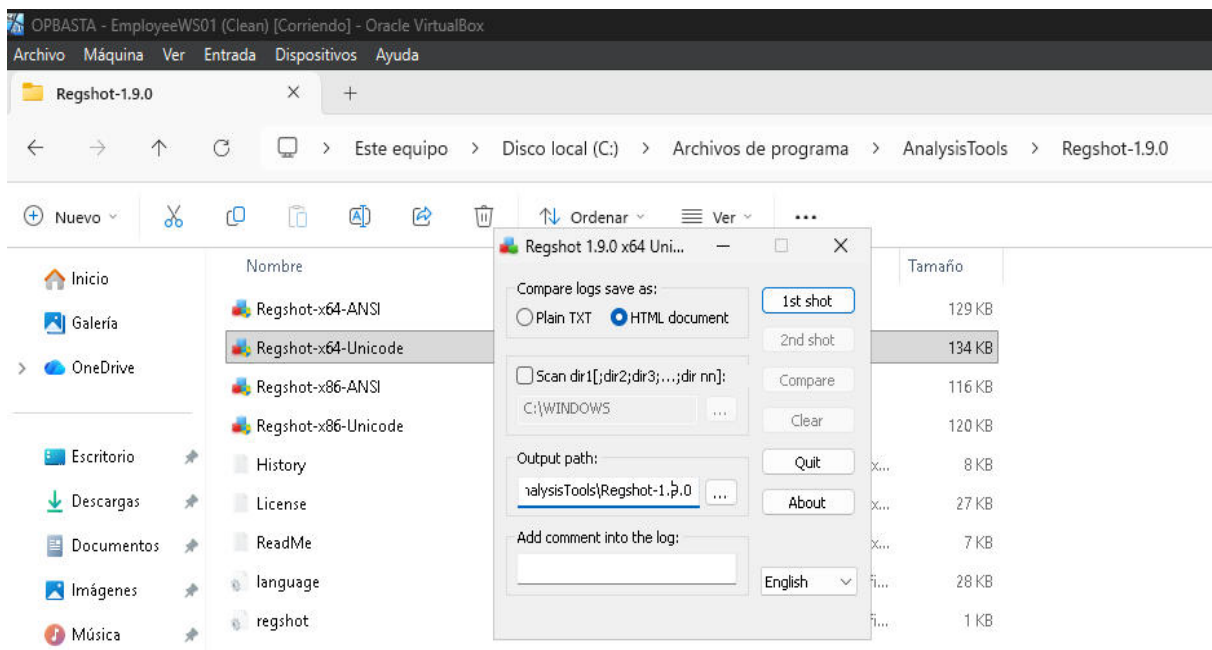
```
Windows PowerShell
PS C:\WINDOWS\system32> Get-Service Sysmon64
```

Status	Name	DisplayName
Running	Sysmon64	Sysmon64

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

Figura 97

Primer Snapshot desde Regshot antes de la ejecución del ransomware

**Figura 98**

Primer shot de Regshot

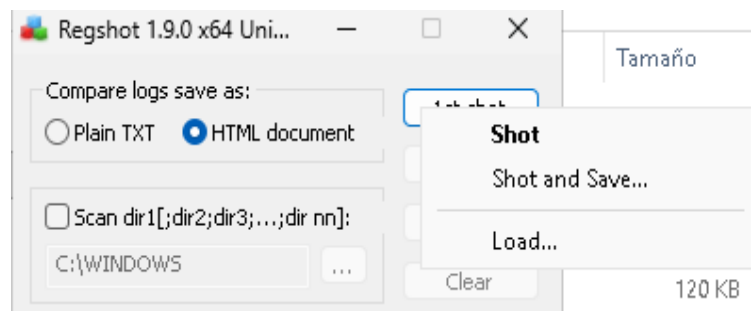
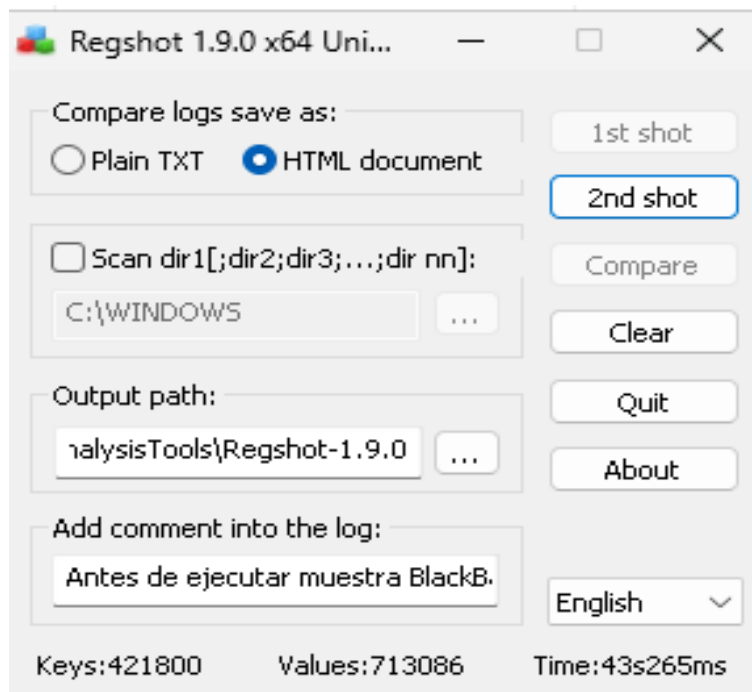


Figura 99

Finalización del primer shot Regshot



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

Figura 100

Primer shot de Regshot guardado en directorio dentro de System32

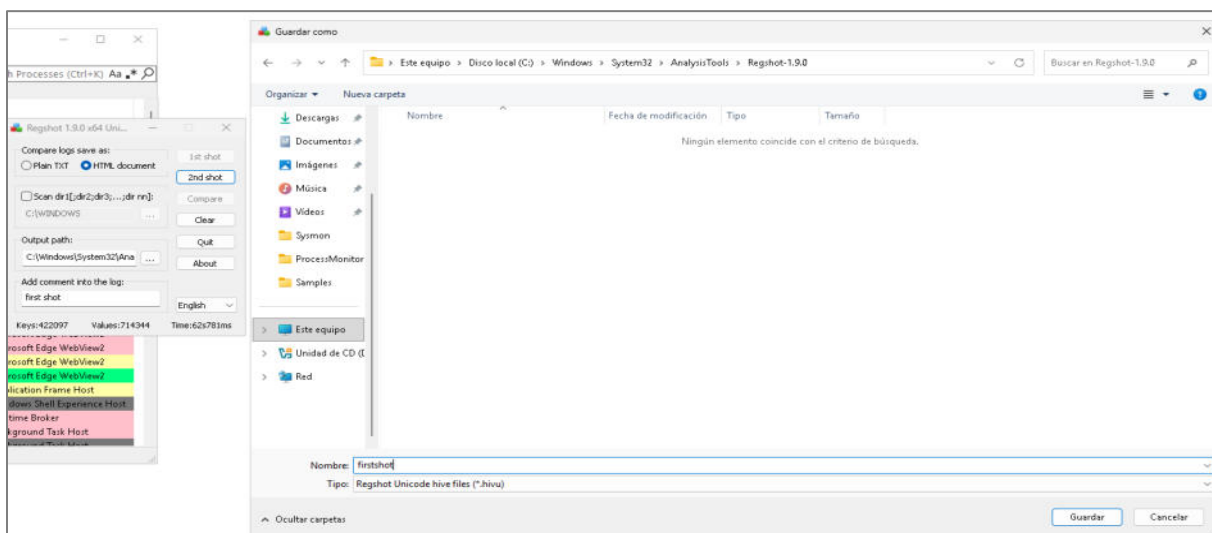
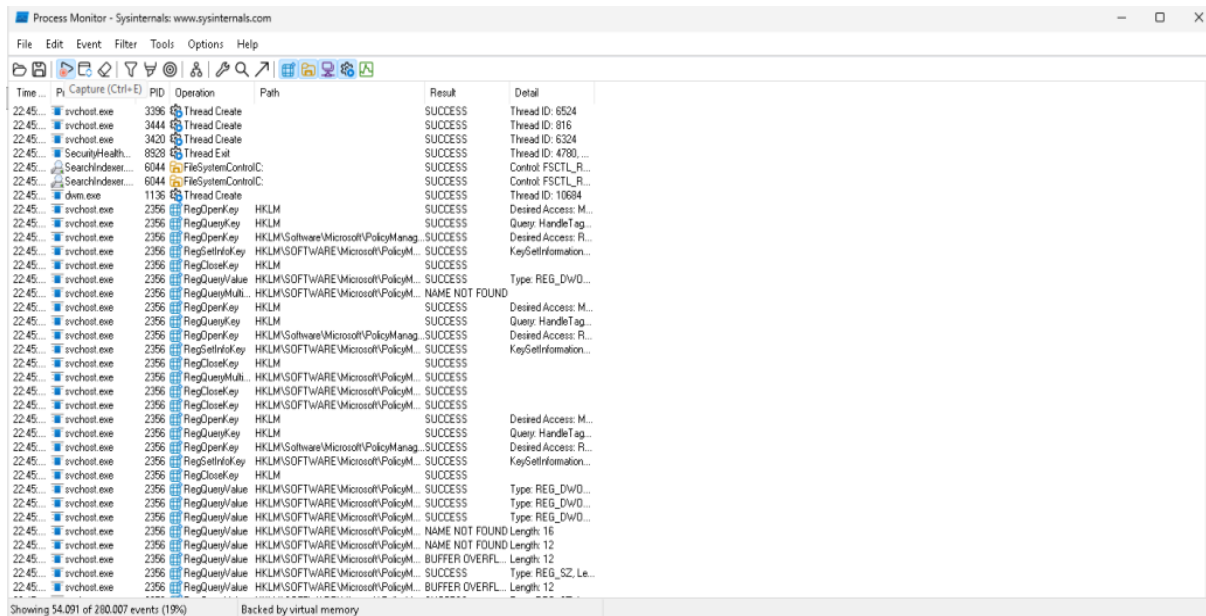


Figura 101

Ejecución de Procmon con permisos de administrador



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

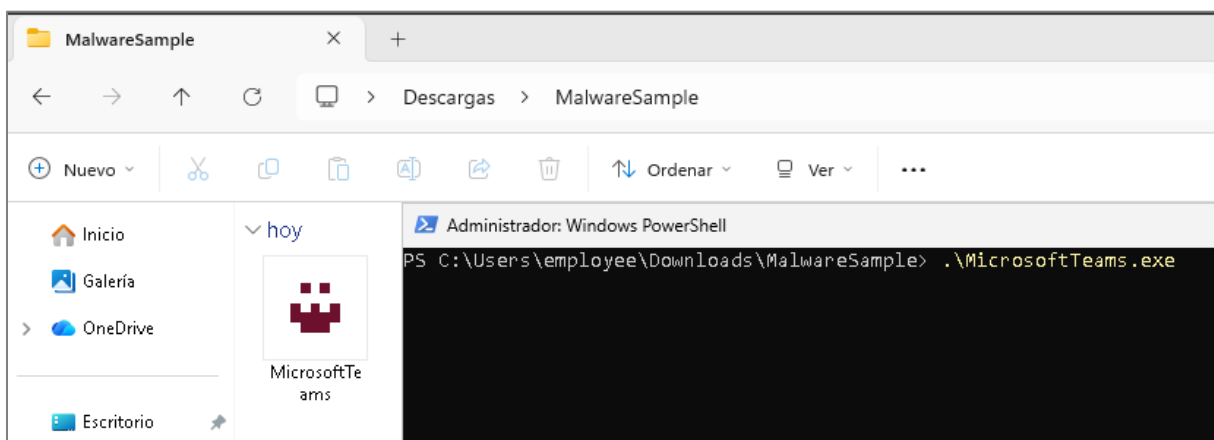
Seguidamente, se preparó la cuarta herramienta Process Hacker, también ejecutada con permisos de Administrador.

Figura 102*Ejecución de Process Hacker con permisos de administrador*

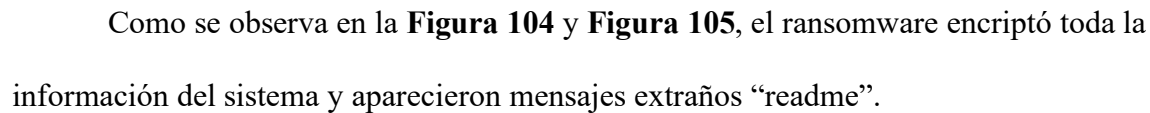
Name	PID	CPU	I/O total r...	Private b...	User name	Description
System Idle Process	0	0,00	0,00	0 KB	NT AUTHORITY\SYSTEM	
System	4	1,07	0,00	40 KB	NT AUTHORITY\SYSTEM	NT Kernel & System
Registry	128		0,00	6,45 MB	NT AUTHORITY\SYSTEM	
smss.exe	496		0,00	1,12 MB	NT AUTHORITY\SYSTEM	Windows Session Manager
Memory Compression	2128		0,00	780 KB	NT AUTHORITY\SYSTEM	
Interrupts		4,22	0,00	0		Interrupts and DPCs
csrss.exe	664		0,00	1,99 MB	NT AUTHORITY\SYSTEM	Client Server Runtime Process
wininit.exe	736		0,00	1,59 MB	NT AUTHORITY\SYSTEM	Windows Start-Up Application
services.exe	880	0,45	0,00	5,22 MB	NT AUTHORITY\SYSTEM	Services and Controller app
svchost.exe	400		0,00	12,2 MB	NT AUTHORITY\SYSTEM	Host Process for Windows Ser...
StartMenuExperi...	6092		0,00	48,12 MB	DESKTOP-A... \employee	Windows Start Experience Host
SearchHost.exe	6100		0,00	54,21 MB	DESKTOP-A... \employee	
msedgewebvie...	6652		0,00	38,05 MB	DESKTOP-A... \employee	Microsoft Edge WebView2
msedgewe...	6704		0,00	2,12 MB	DESKTOP-A... \employee	Microsoft Edge WebView2
msedgewe...	4344		0,00	14,98 MB	DESKTOP-A... \employee	Microsoft Edge WebView2
msedgewe...	6444		0,00	12,45 MB	DESKTOP-A... \employee	Microsoft Edge WebView2
msedgewe...	6384		0,00	8,69 MB	DESKTOP-A... \employee	Microsoft Edge WebView2
msedgewe...	6336		0,00	84,23 MB	DESKTOP-A... \employee	Microsoft Edge WebView2
ApplicationFrame...	8716		0,00	10,47 MB	DESKTOP-A... \employee	Application Frame Host
ShellExperienceH...	3084		0,00	17,55 MB	DESKTOP-A... \employee	Windows Shell Experience Host
RuntimeBroker.exe	528		0,00	3,06 MB	DESKTOP-A... \employee	Runtime Broker
backgroundTask...	10208		0,00	23,88 MB	DESKTOP-A... \employee	Background Task Host
RuntimeBroker.exe	1464		0,00	3,98 MB	DESKTOP-A... \employee	Runtime Broker
RuntimeBroker.exe	6872		0,00	5,26 MB	DESKTOP-A... \employee	Runtime Broker
WindowsPackage...	10392		0,00	7,34 MB	DESKTOP-A... \employee	
FileCoAuth.exe	6024		0,00	11,24 MB	DESKTOP-A... \employee	Microsoft OneDriveFile Co-Au...
unsecapp.exe	10084		0,00	1,35 MB	NT AUTHORITY\SYSTEM	Sink to receive asynchronous ...
SystemSettings.exe	7692		0,00	66,83 MB	DESKTOP-A... \employee	Settings
backgroundTask...	12248		0,00	3,86 MB	DESKTOP-A... \employee	Background Task Host
dlh.exe	11164		0,00	3,07 MB	DESKTOP-A... \employee	COM Surrogate

3.2.2. Ejecución del Ransomware Black Basta. Se ejecutó el ransomware como

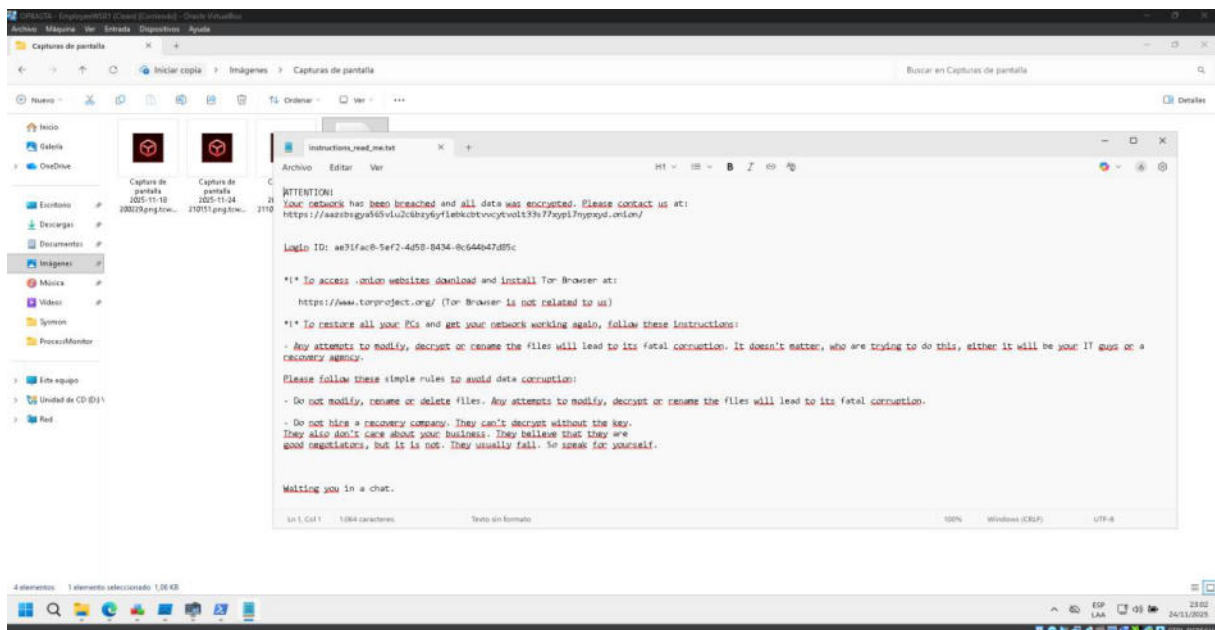
Administrador desde la consola de PowerShell.

Figura 103*Ejecución de la muestra de Black Basta desde PowerShell*

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



Archivos encriptados por Black Basta y nota de readme



3.3. Recolección de Evidencias Tras la Ejecución del Ransomware

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

Figura 106

Segundo shot posterior a la ejecución de Black Basta

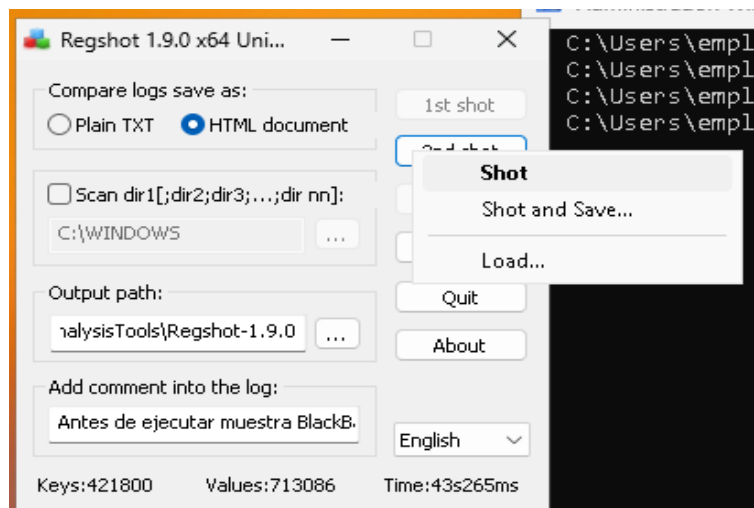
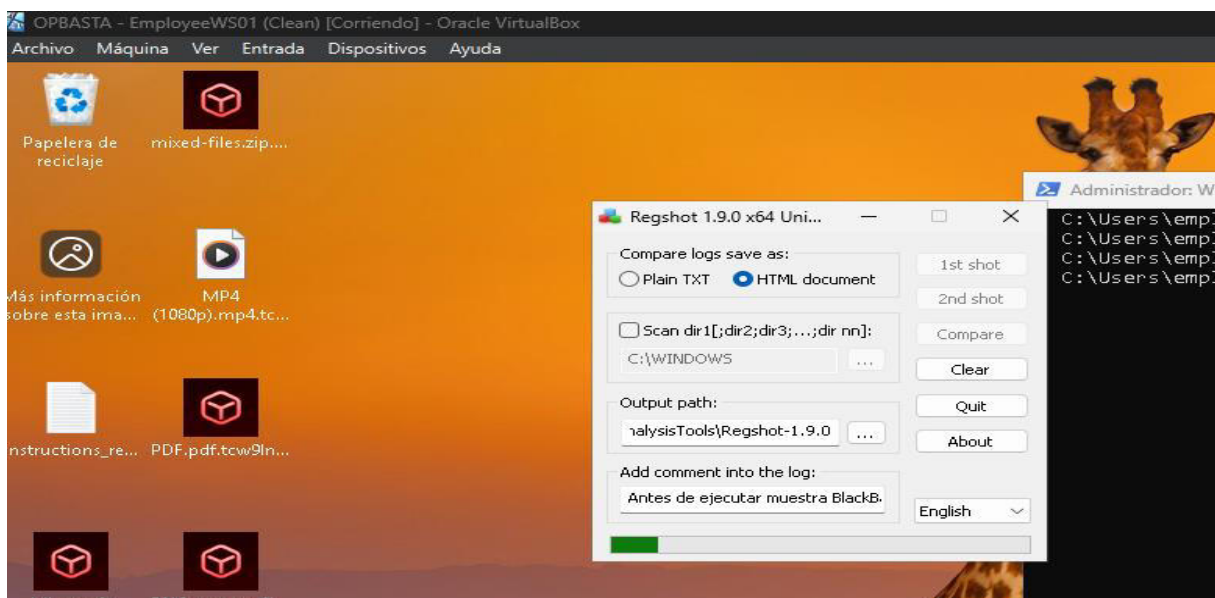


Figura 107

Proceso de comparación de primer y segundo shot obtenidos



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

Figura 108

Registros de Procmon posterior a la ejecución del ransomware

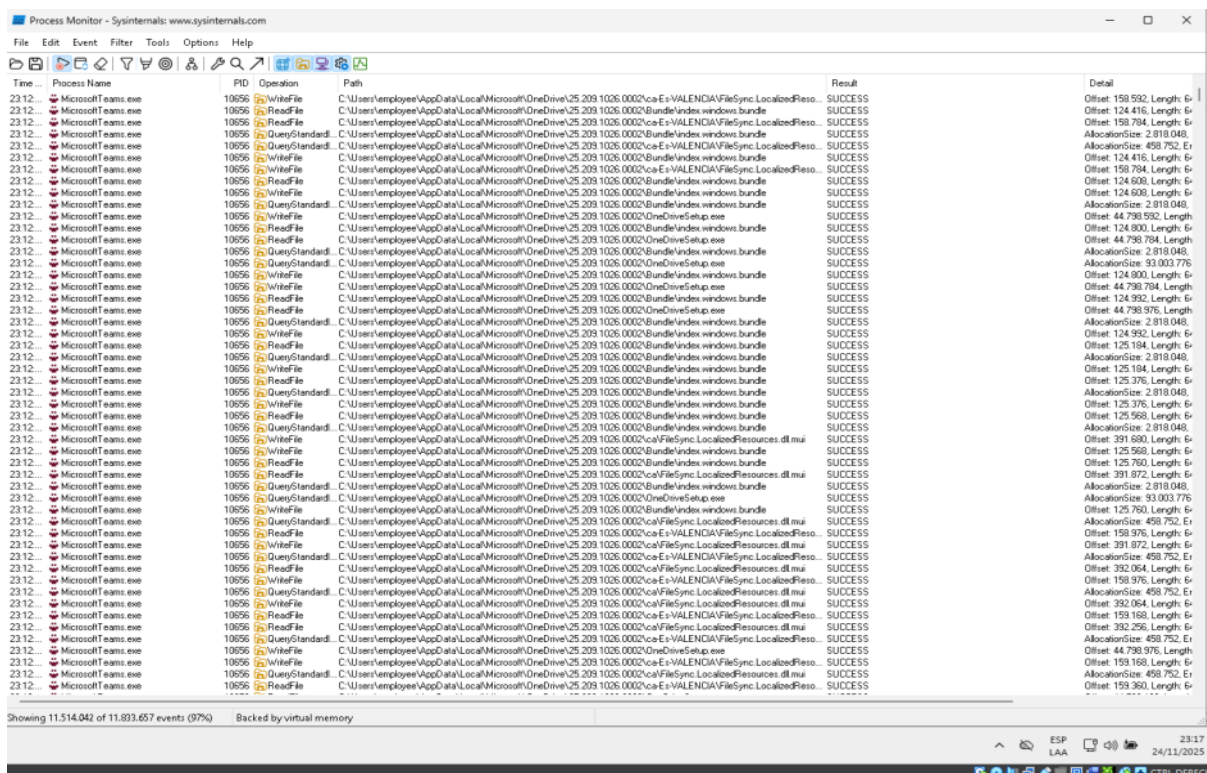
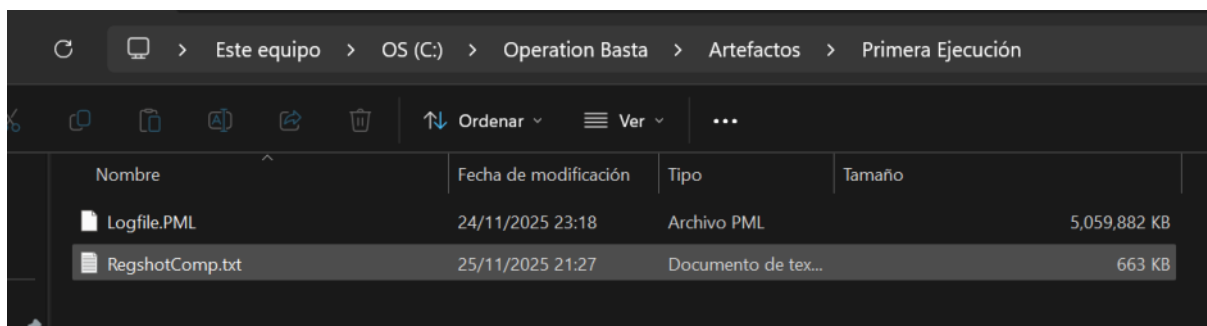


Figura 109

Archivos obtenidos de las herramientas Regshot y Procmon

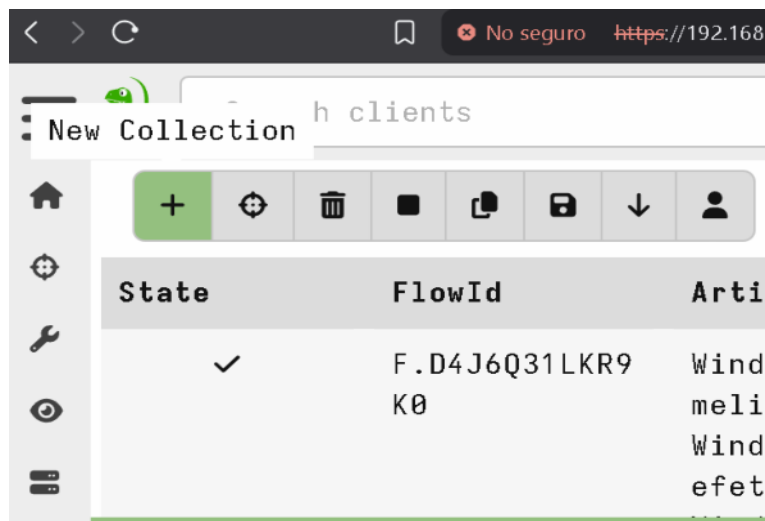


A través de la GUI del servidor de Velociraptor, instalado en VM OPBASTA -

ThreatAnalysis01, se ubicó al cliente y se dio clic en “Collected” y luego en “New Collection”, para empezar a recolectar toda información capturada por el agente de Velociraptor.

Figura 110

Selección del Collector en la interfaz gráfica de Velociraptor

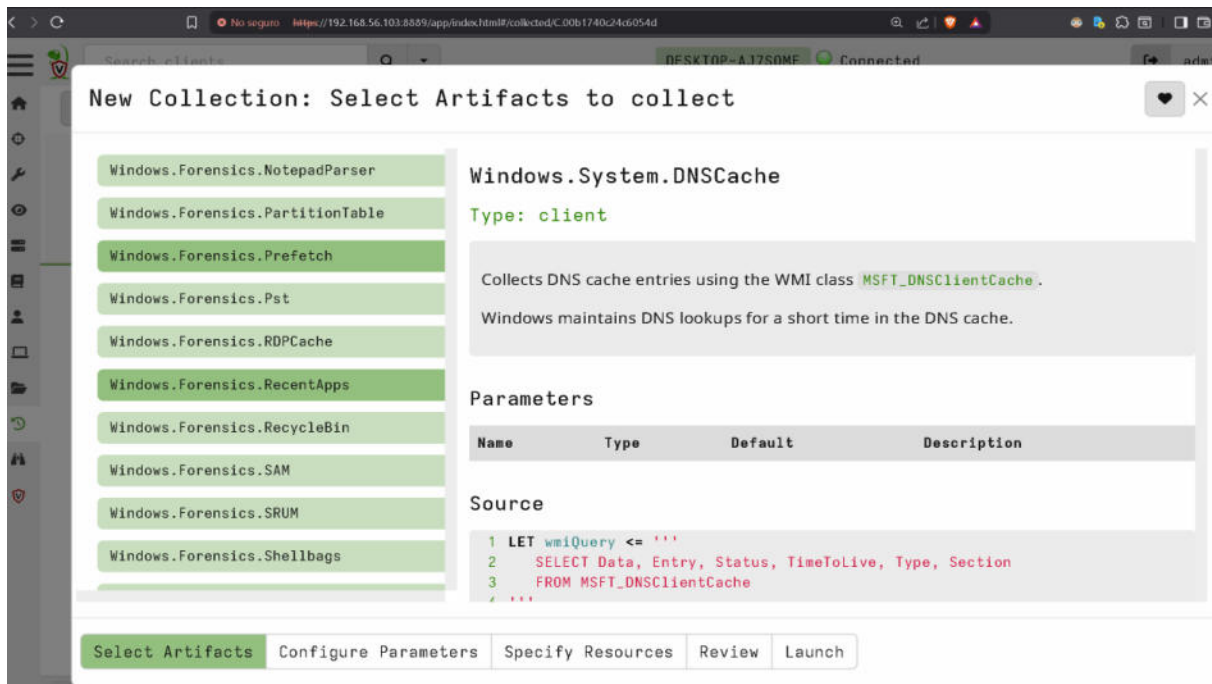


Se seleccionaron los siguientes artefactos a recolectar, los cuales sirven para analizar el comportamiento del Ransomware:

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

Figura 111

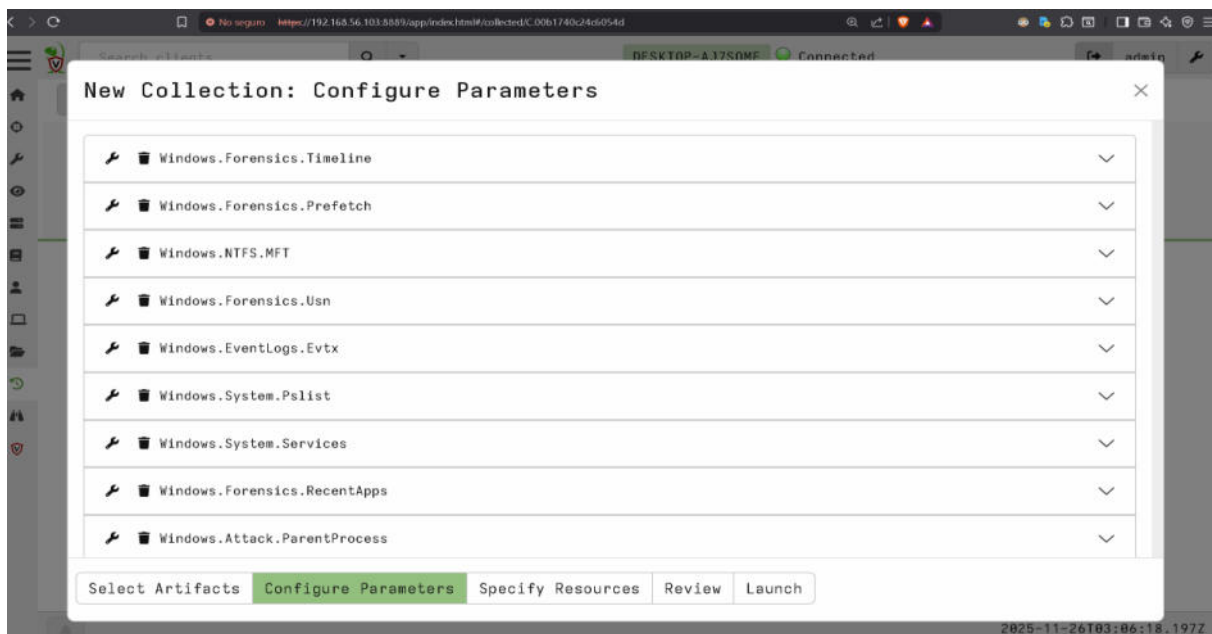
Selección de artefactos que serán recolectados



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

Figura 112

Configuración de los parámetros del colector



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

Figura 113

Configuración de Specify Resources del colector

New Collection: Specify Resources

CPU Limit Percent: 100%

I/Os/Sec: Unlimited

Max Execution Time in Seconds: 600s per artifact

Max Idle Time in Seconds: If set collection will be terminated after this many seconds with no progress.

Max Rows: 1,000,000 rows

Max Logs: 100000

Max MB uploaded: 1 Gb

Trace Frequency Seconds: To enable tracing, specify trace update frequency in seconds

Urgent: ☐ Skip queues and run query urgently

Select Artifacts | Configure Parameters | **Specify Resources** | Review | Launch

2025-11-26T03:06:55.196Z

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

Figura 114

Revisión de los artefactos capturados por el colector

State	FlowId	Artifacts	Created	Last Active	Creator	Mb	Rows
✓	F.D4J6Q31LKR9K0	Windows.Forensics.Timeline Windows.Forensics.PreFetch Windows.NTFS.MFT Windows.Forensics.Union Windows.System.Polish	2025-11-26T02:59:24.970Z	2025-11-26T03:02:06.164Z	admin	0 b	780949

Capítulo 4

4. Análisis de Resultados

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

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

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

4.1. Pruebas de Concepto

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

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

- **Muestra:**

17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe

- **Lenguaje: C**

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

Figura 115

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

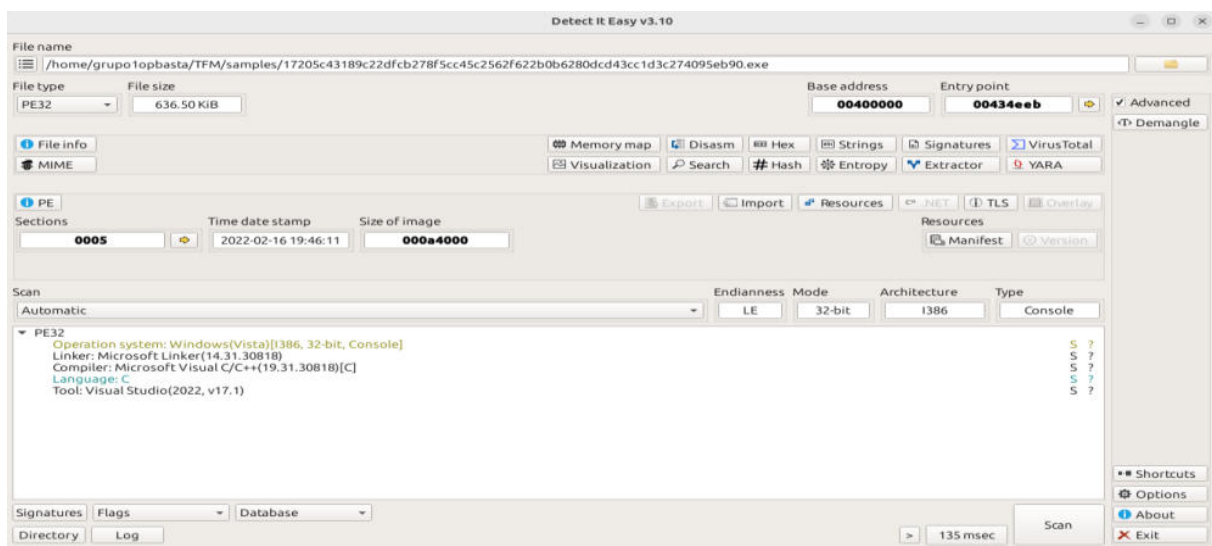


Figura 116

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

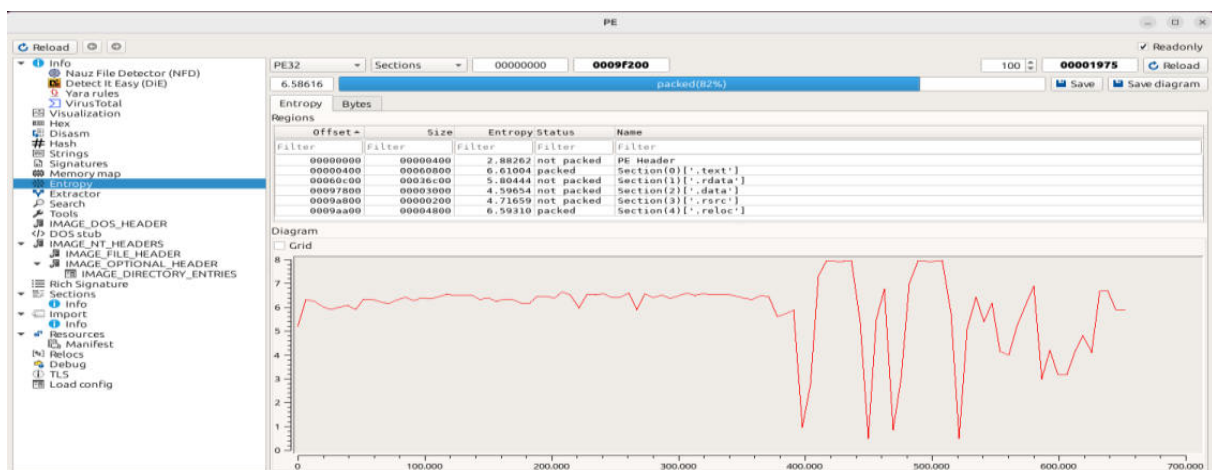
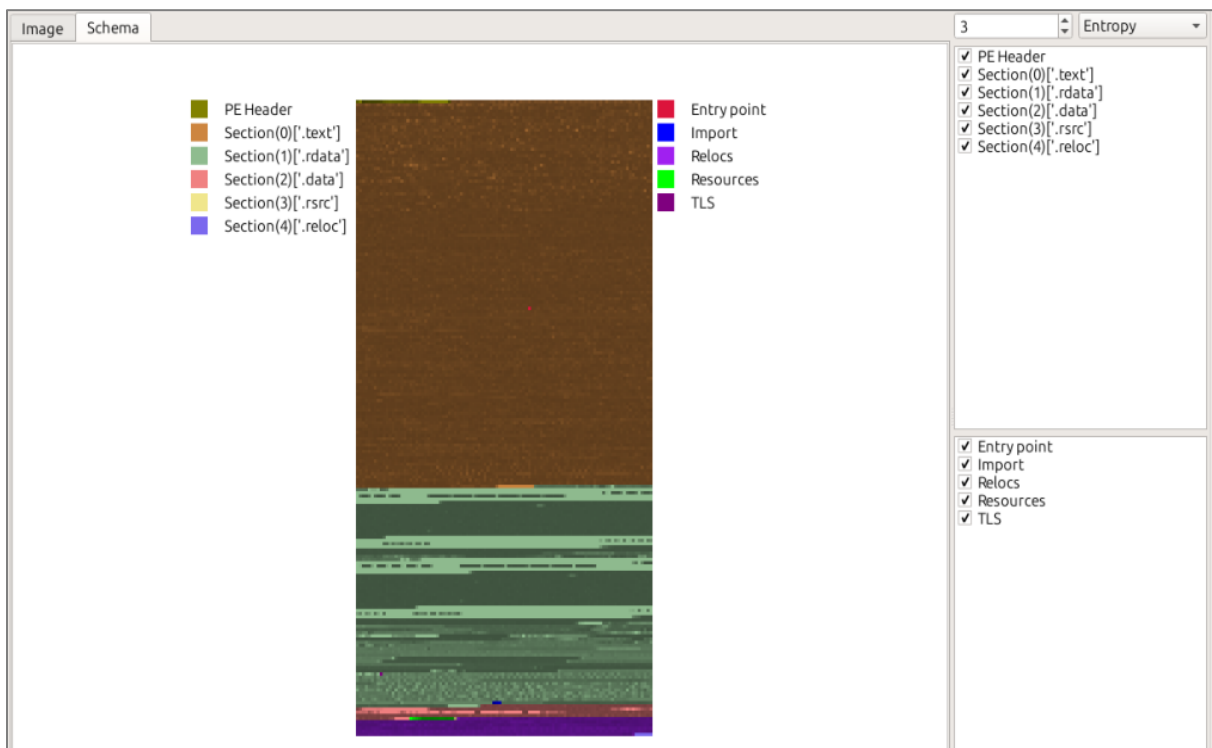


Figura 117

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



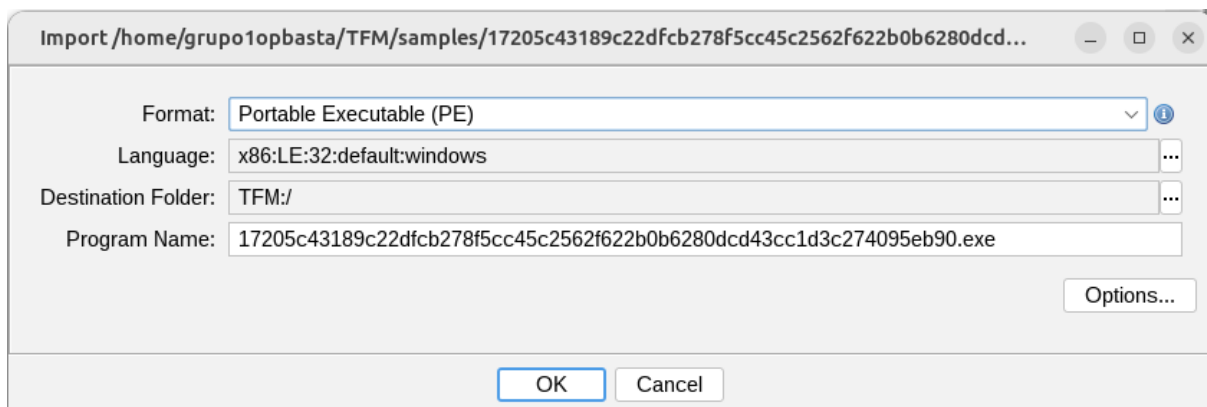
Como resultado del análisis estático preliminar con Detect It Easy, se revela un binario PE32 compilado con MSVC, que presenta un nivel de entropía del 82%, lo cual evidencia un alto grado de empaquetamiento u ofuscación. Las secciones .text, .data y .rdata presentan patrones homogéneos que corresponden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

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

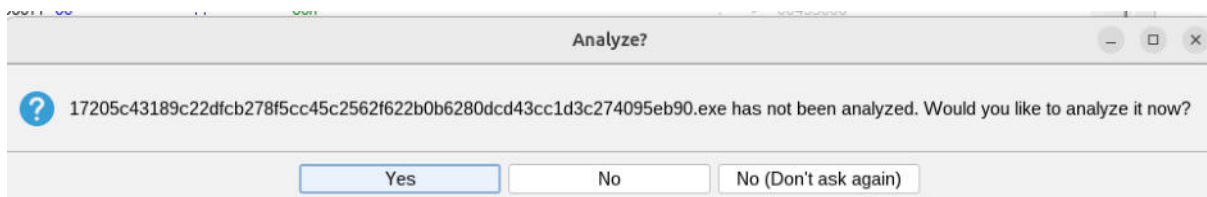
4.1.1.2. Revisión del Ransomware con la Herramienta Ghidra. Adicionalmente, se importó la muestra del ransomware en Ghidra y se ejecutó el análisis predeterminado.

Figura 118

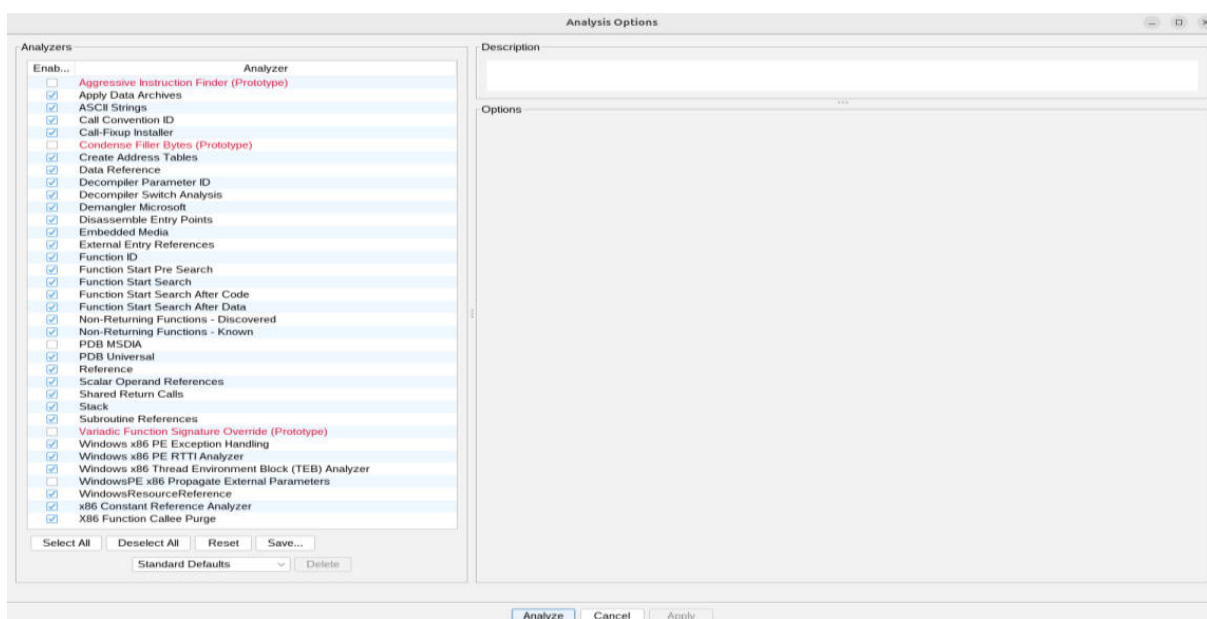
Análisis de la muestra de Black Basta en la herramienta Ghidra

**Figura 119**

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

**Figura 120**

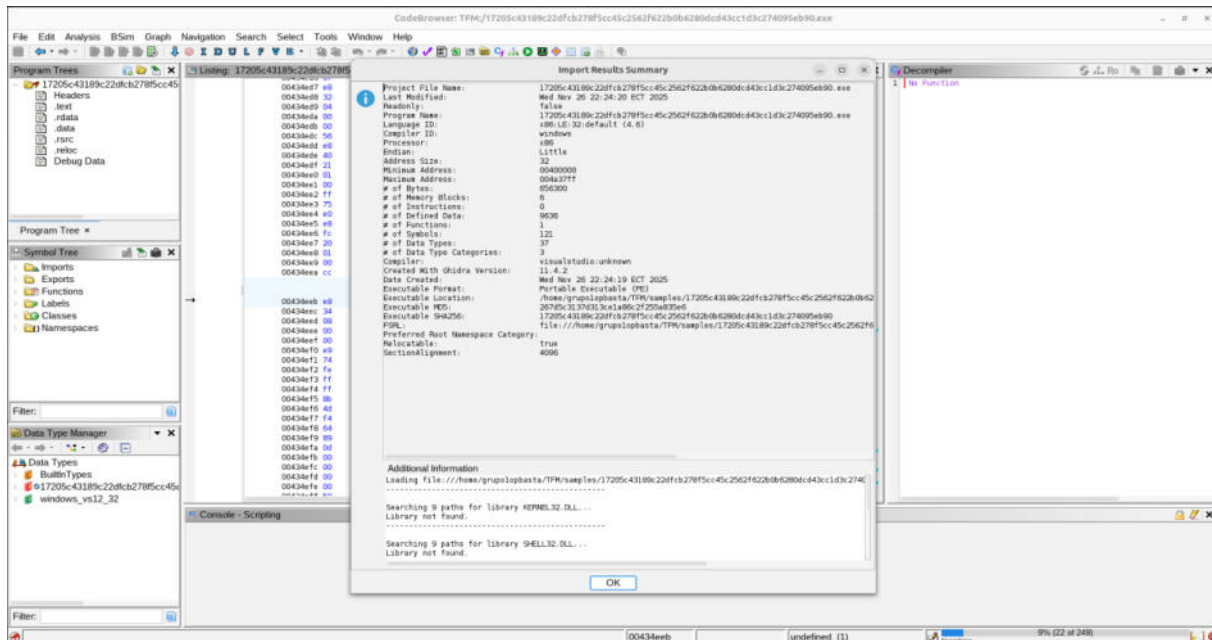
Inicio del análisis de la muestra de Black Basta en la herramienta Ghidra



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

Figura 121

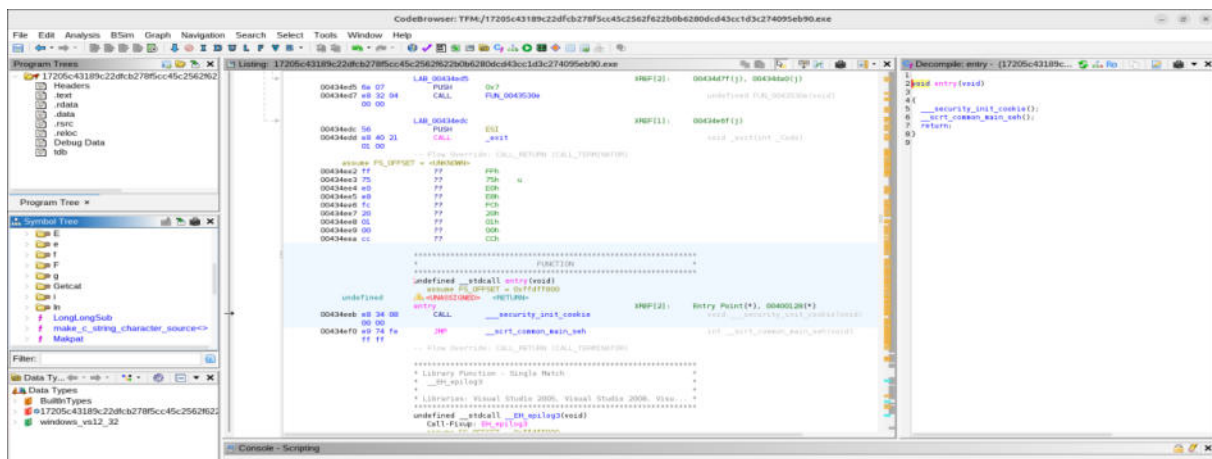
Finalización del análisis de la muestra de Black Basta en la herramienta Ghidra



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

Figura 122

Código descompilado de Black Basta en la herramienta Ghidra



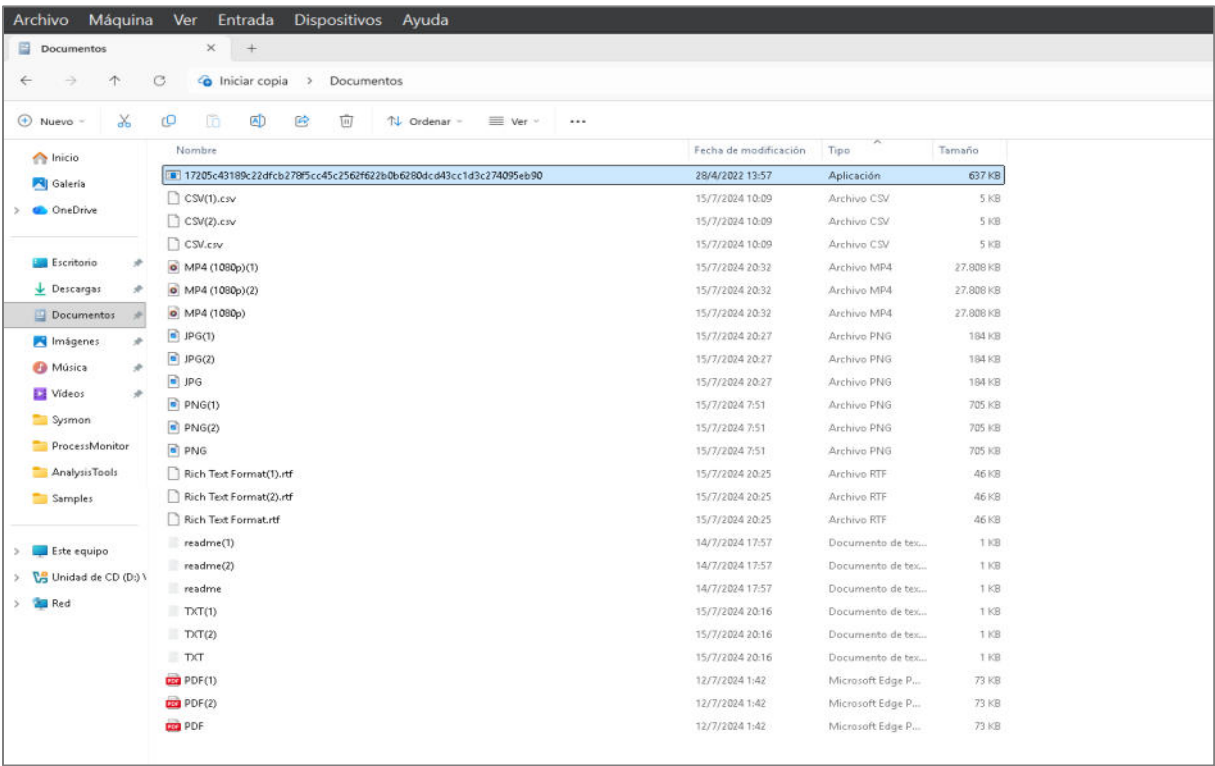
4.1.2. Análisis Dinámico. A fin de entender el funcionamiento del ransomware Black Basta observando su comportamiento en tiempo real, se ejecutó un análisis dinámico utilizando herramientas de análisis forense como Sysmon, Procmon, Process Hacker, entre otras, para recolección de artefactos útiles. A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

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

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

Figura 123

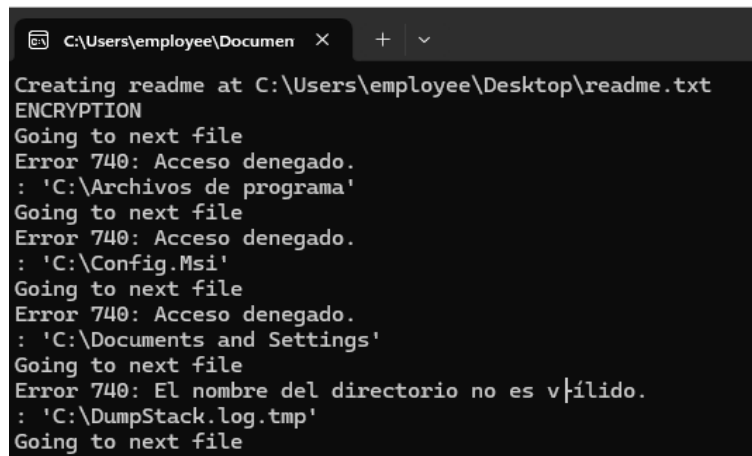
Estado inicial de los archivos antes de la ejecución de Black Basta



Nombre	Fecha de modificación	Tipo	Tamaño
17205c43189c22dfcb278f5cc45c2562f622b0b6280dc43cc1d3c274095eb90	28/4/2022 13:57	Aplicación	637 KB
CSV(1).csv	15/7/2024 10:09	Archivo CSV	5 KB
CSV(2).csv	15/7/2024 10:09	Archivo CSV	5 KB
CSV.csv	15/7/2024 10:09	Archivo CSV	5 KB
MP4 (1080p)(1)	15/7/2024 20:32	Archivo MP4	27.808 KB
MP4 (1080p)(2)	15/7/2024 20:32	Archivo MP4	27.808 KB
MP4 (1080p)	15/7/2024 20:32	Archivo MP4	27.808 KB
JPG(1)	15/7/2024 20:27	Archivo PNG	184 KB
JPG(2)	15/7/2024 20:27	Archivo PNG	184 KB
JPG	15/7/2024 20:27	Archivo PNG	184 KB
PNG(1)	15/7/2024 7:51	Archivo PNG	705 KB
PNG(2)	15/7/2024 7:51	Archivo PNG	705 KB
PNG	15/7/2024 7:51	Archivo PNG	705 KB
Rich Text Format(1).rtf	15/7/2024 20:25	Archivo RTF	46 KB
Rich Text Format(2).rtf	15/7/2024 20:25	Archivo RTF	46 KB
Rich Text Format.rtf	15/7/2024 20:25	Archivo RTF	46 KB
readme(1)	14/7/2024 17:57	Documento de tes...	1 KB
readme(2)	14/7/2024 17:57	Documento de tes...	1 KB
readme	14/7/2024 17:57	Documento de tes...	1 KB
TXT(1)	15/7/2024 20:16	Documento de tes...	1 KB
TXT(2)	15/7/2024 20:16	Documento de tes...	1 KB
TXT	15/7/2024 20:16	Documento de tes...	1 KB
PDF(1)	12/7/2024 1:42	Microsoft Edge P...	73 KB
PDF(2)	12/7/2024 1:42	Microsoft Edge P...	73 KB
PDF	12/7/2024 1:42	Microsoft Edge P...	73 KB

Figura 124

Revisión de archivo readme.txt generado por Black Basta

**Figura 125**

Encriptación de los archivos realizada por Black Basta

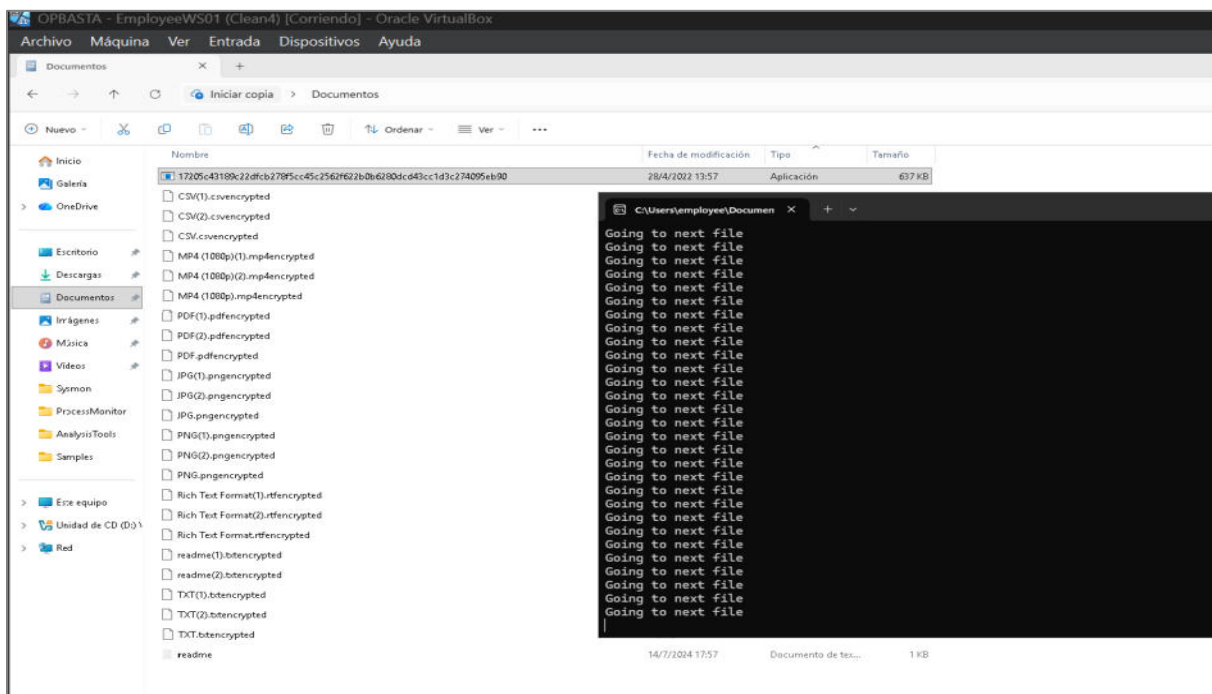
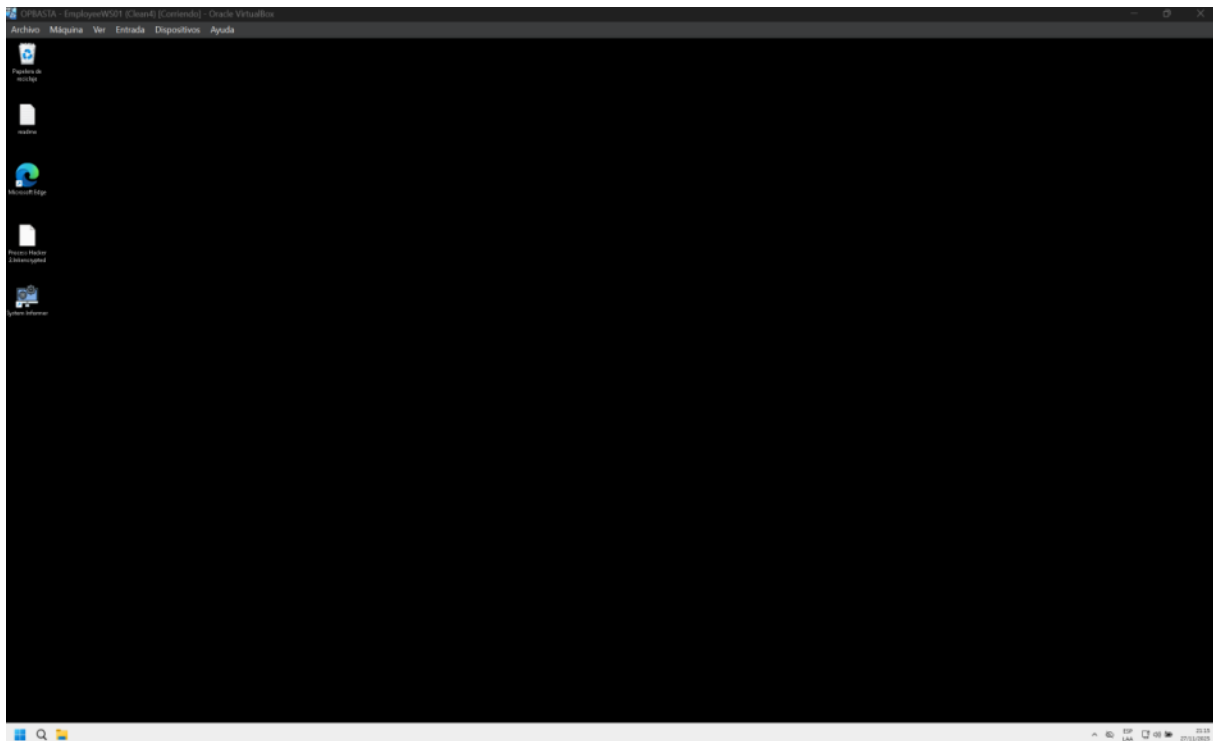


Figura 128

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



- **Creación de archivo readme.txt con contenido:** Se observó el archivo “readme.txt” en el escritorio, el cual contiene exactamente el mismo mensaje que se leyó en una parte de las funciones del código que proporcionó Ghidra.

Figura 129

Creación de archivo readme.txt en el escritorio

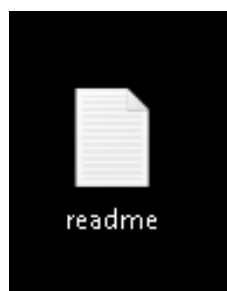


Figura 130

Revisión del archivo readme.txt generado por Black Basta



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

Figura 131

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

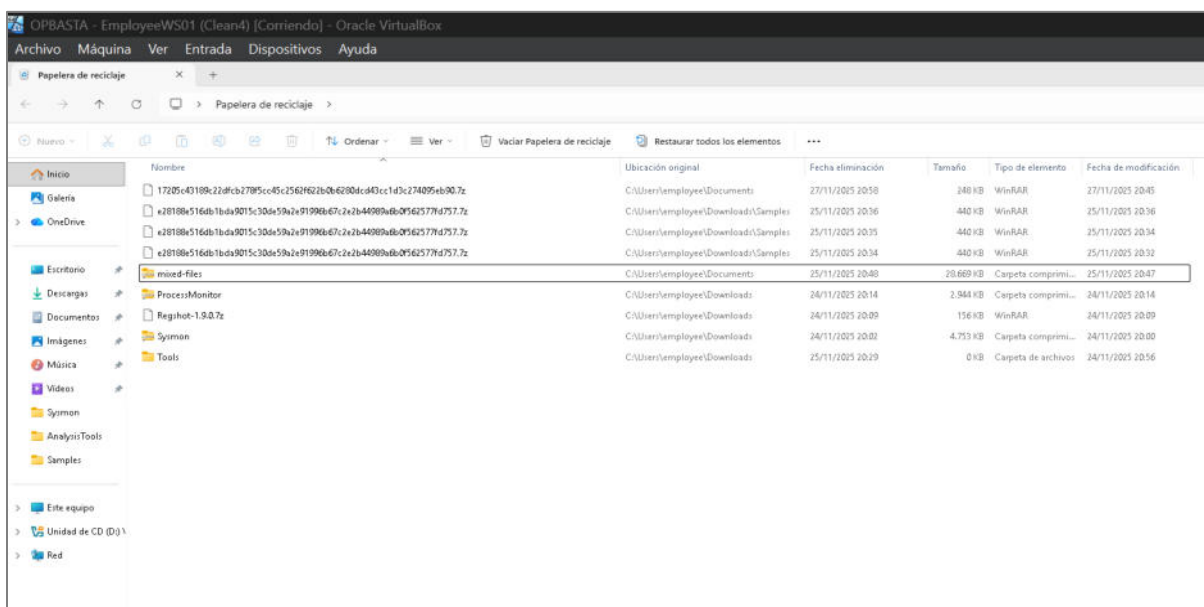


Figura 132

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

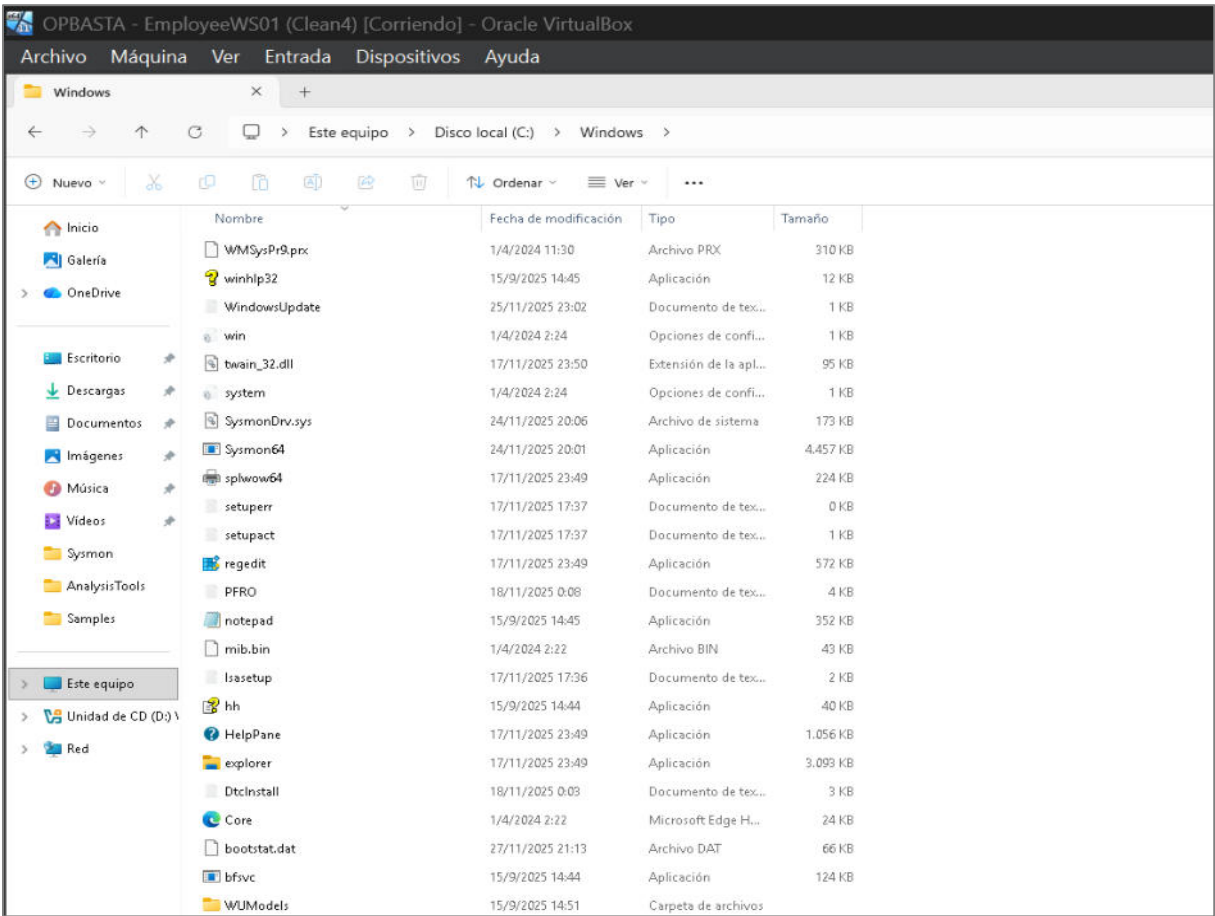
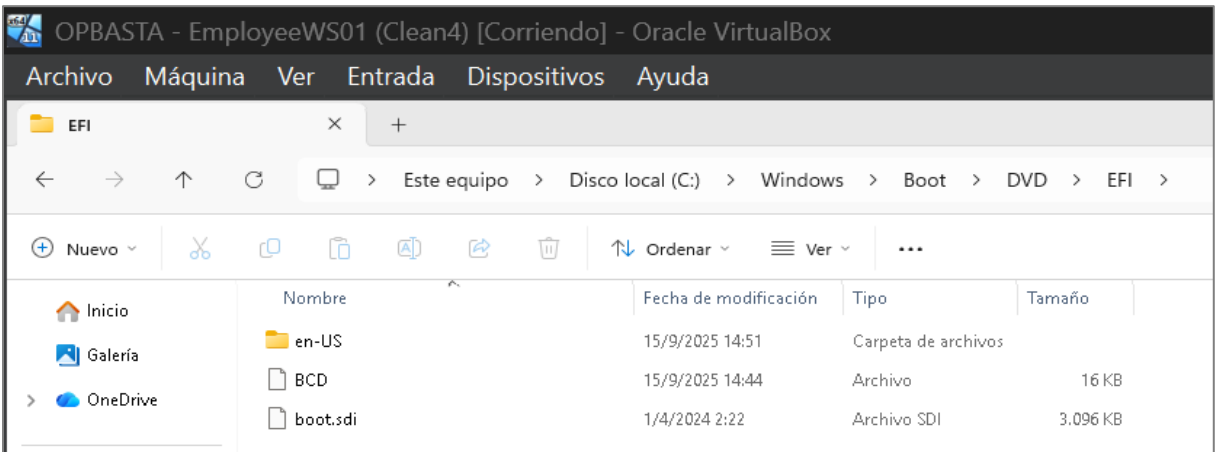


Figura 133

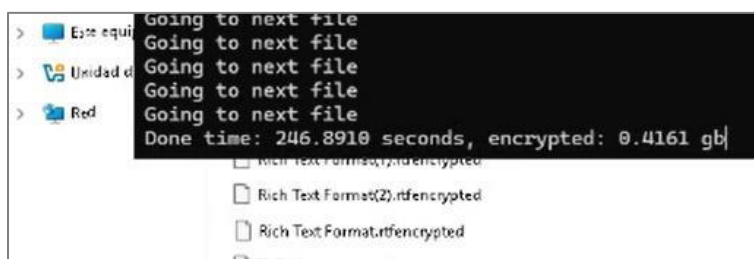
Revisión de los archivos que se encuentran en las subcarpetas de Windows



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

Figura 134

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

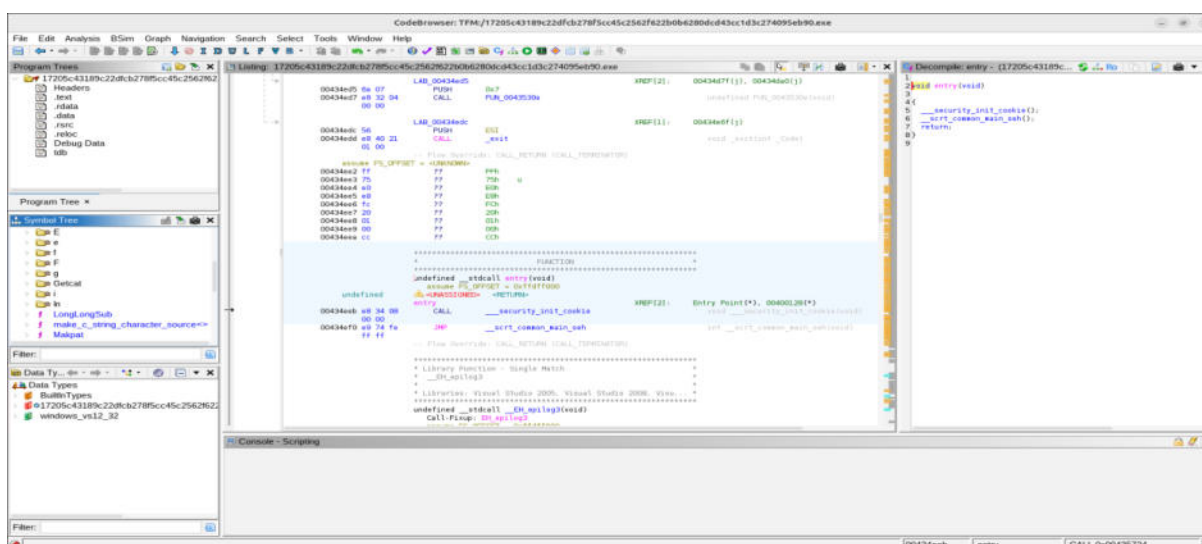


4.2. Interpretación de Resultados

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

Figura 135

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



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

- Inicialización criptográfica y preparación de claves.
- Enumeración completa del sistema de archivos por volúmenes.
- Creación de múltiples hilos de trabajo para paralelizar el cifrado.
- Procesamiento por cada hilo: renombrado de archivo, generación de claves

aleatorias por archivo, cifrado del material criptográfico con una clave pública interna (RSA-like), escritura de metadatos y blob cifrado en el archivo, cifrado del contenido.

- Alteraciones del entorno gráfico del usuario (Wallpaper).
- Finalización, limpieza de memoria y terminación controlada.

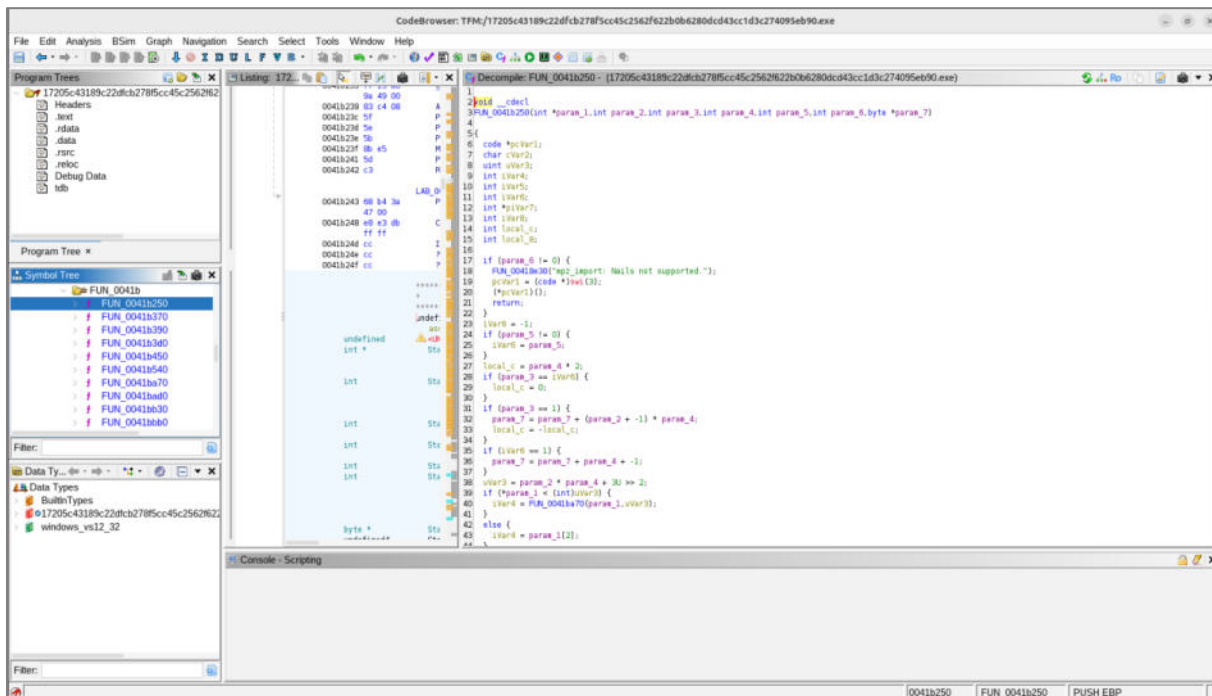
Este pipeline confirma que el atacante diseñó la muestra para ser lo más rápida posible, con mínima dependencia del entorno, y con técnicas que dificultan la recuperación de información.

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

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

Figura 136

Revisión de código en Decompiler: FUN_0041b250 en Ghidra



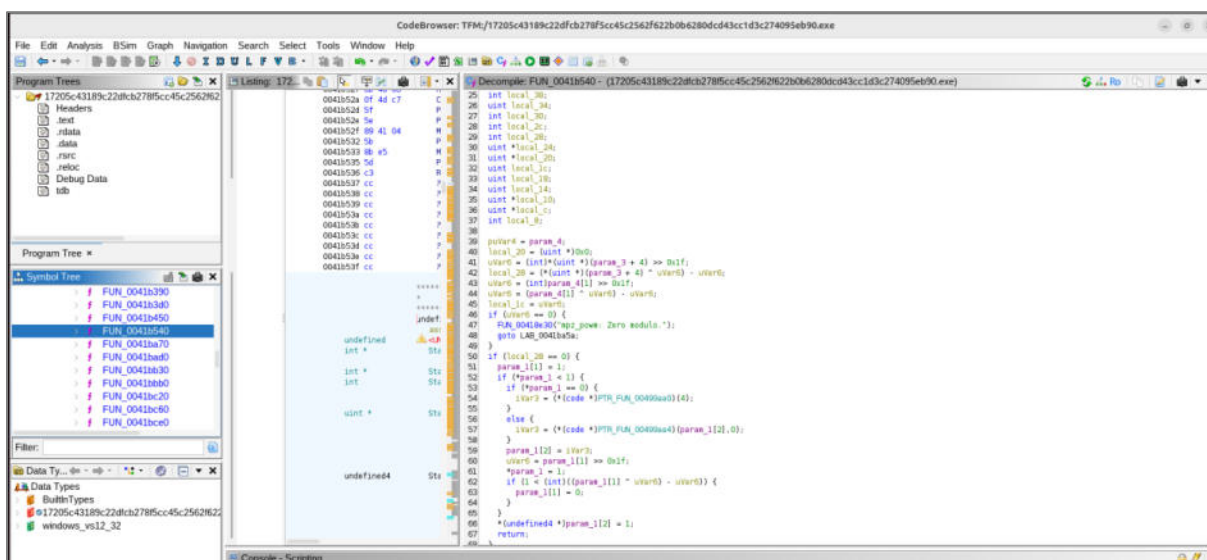
- **FUN_0041b540 > mpz_powm.-** Implementa exponenciación modular:

$\text{res} = \text{base}^{\text{exponente}} \bmod \text{modulo}$

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

Figura 137

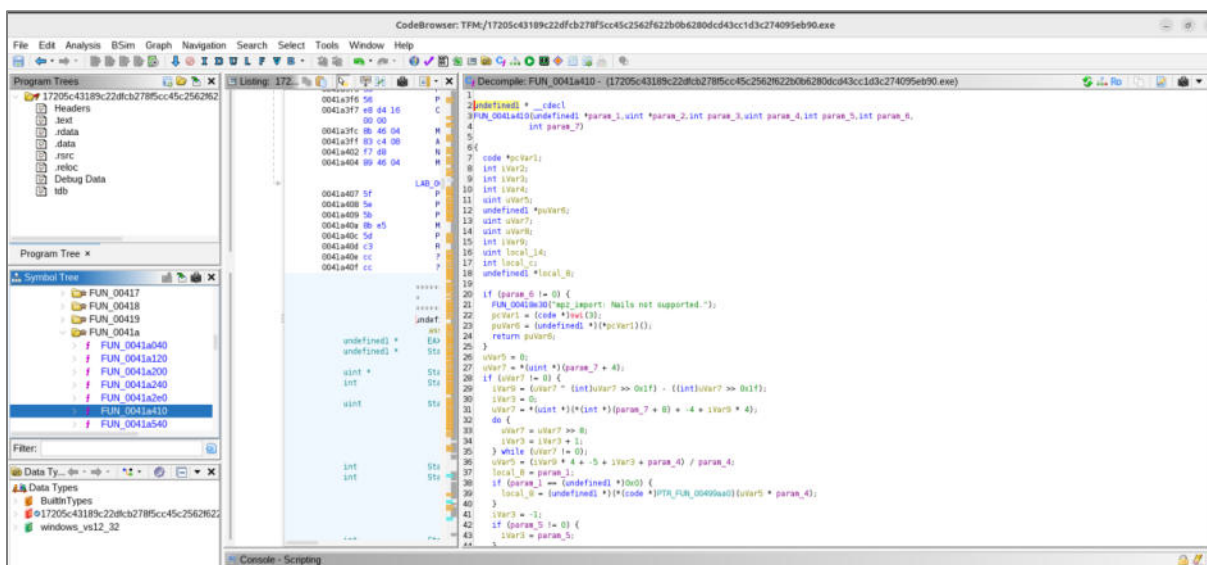
Revisión de código en Decompiler: FUN_0041b540 en Ghidra



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

Figura 138

Revisión de código en Decompiler: FUN_0041a410 en Ghidra



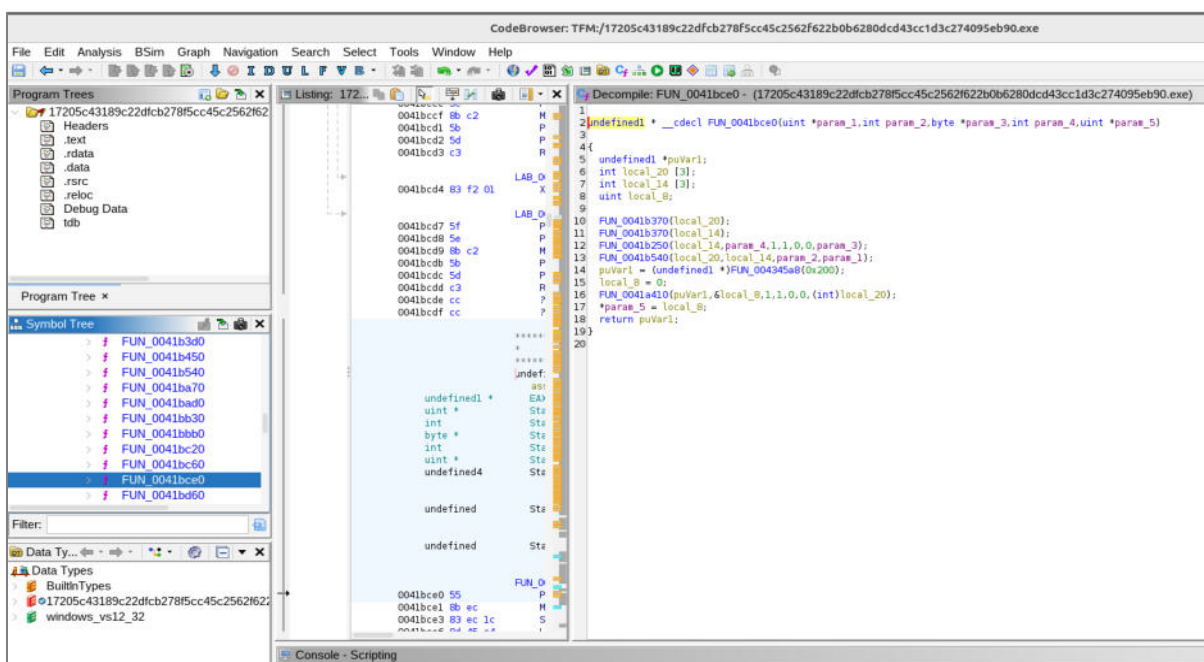
4.2.1.3. Integración Criptográfica y Cifrado Híbrido. El ransomware integra estas funciones a través de **FUN_0041bce0** y de otras funciones de nivel superior para implementar un esquema híbrido:

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

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

Figura 139

Revisión de código en Decompiler: FUN_0041bce0 en Ghidra



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

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

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

Figura 140

Revisión de código en Decompiler: FUN_0040d160 en Ghidra

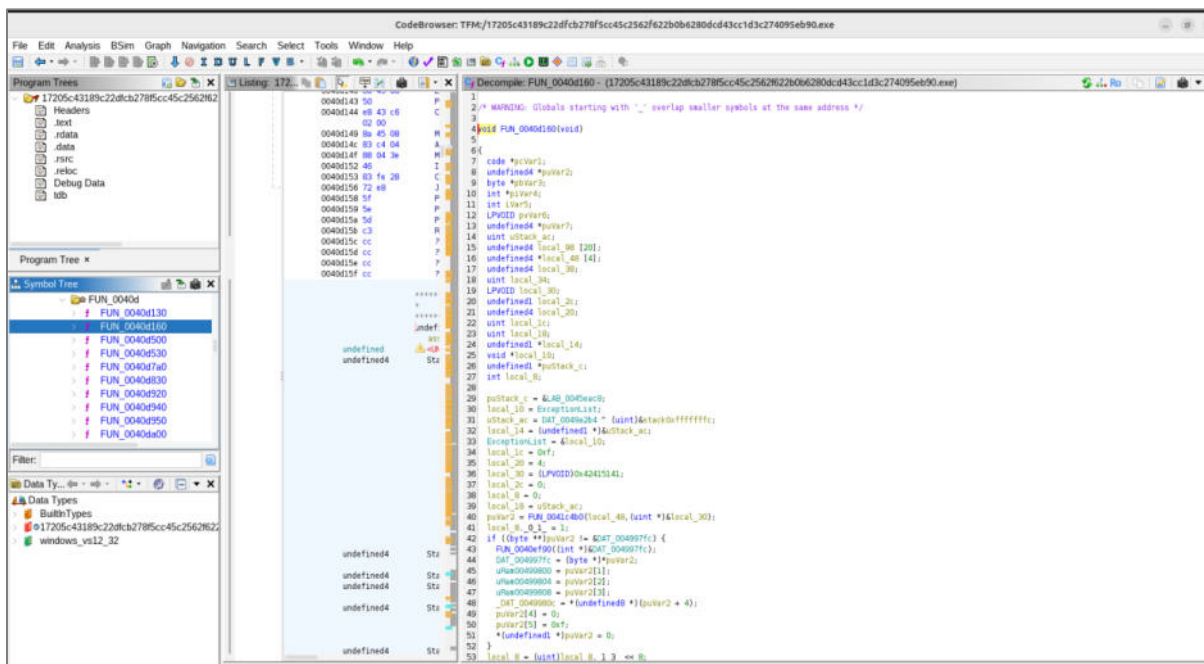


Figura 141

Análisis de código en Decompiler: FUN_0040d160 en Ghidra



4.2.1.5. Enumeración de Volúmenes y Rutas en Windows. El ransomware implementa

un sistema de descubrimiento completo del sistema de archivos usando APIs estándar:

- FindFirstVolumeW, FindNextVolumeW
- GetVolumePathNamesForVolumeNameW
- GetVolumeInformationW

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

Figura 142

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

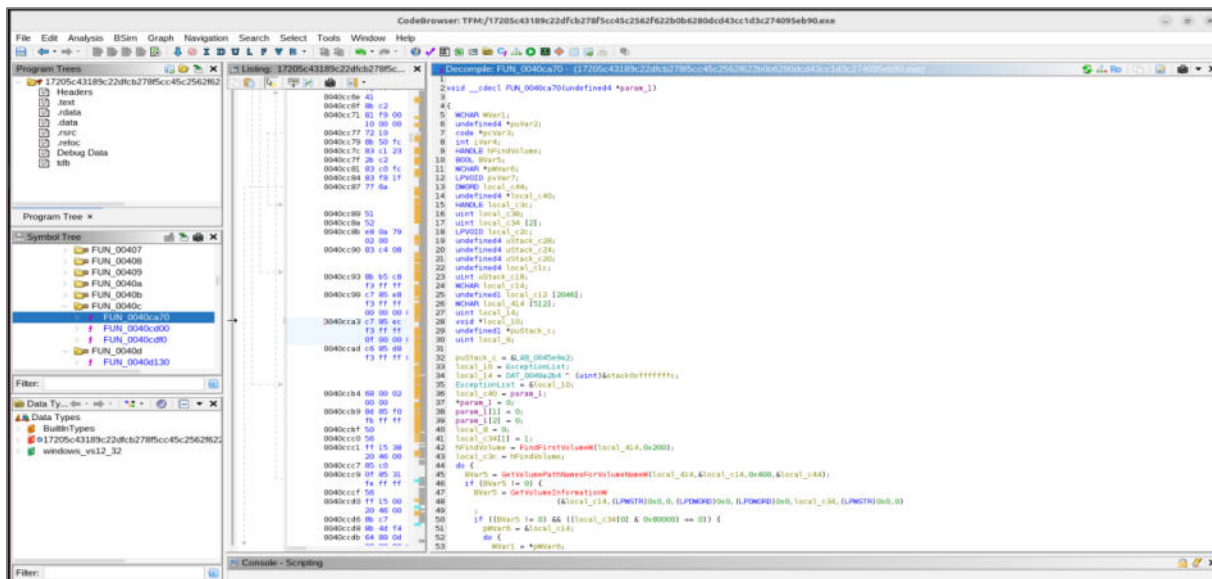


Figura 143

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

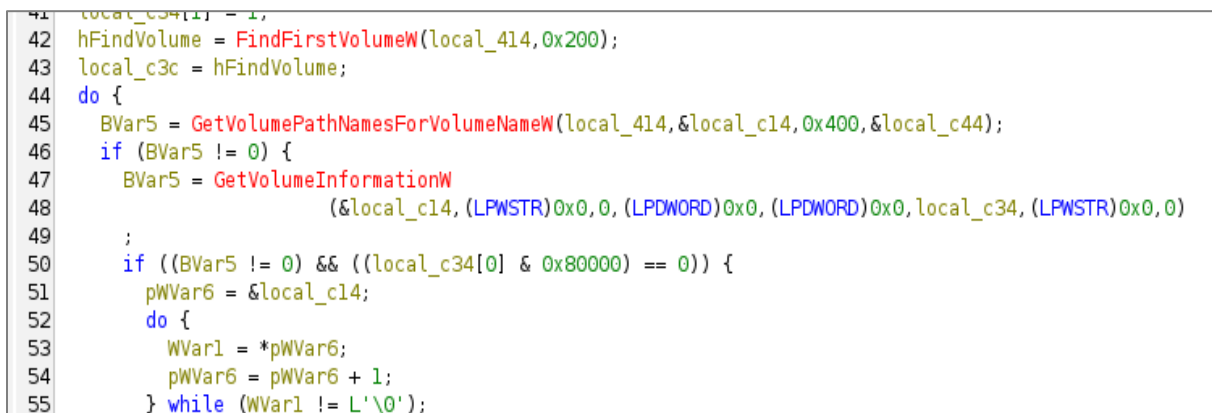


Figura 144

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

```

83     local_8 = local_8 & 0xffffffff00;
84     if (0xf < uStack_c18) {
85         pvVar7 = local_c2c;
86         if (0xfff < uStack_c18 + 1) {
87             pvVar7 = *(LPVOID *)((int)local_c2c + -4);
88             if (0x1f < (uint)((int)local_c2c + (-4 - (int)pvVar7))) {
89                 FUN_004395bf();
90                 pcVar3 = (code *)swi(3);
91                 (*pcVar3)();
92                 return;
93             }
94         }
95         FUN_0043459a(pvVar7);
96     }
97     local_c1c = 0;
98     uStack_c18 = 0xf;
99     local_c2c = (LPVOID)((uint)local_c2c & 0xffffffff00);
100    hFindVolume = local_c3c;
101 }
102 }
103 BVar5 = FindNextVolumeW(hFindVolume, local_414, 0x200);
104 } while (BVar5 != 0);
105 FindVolumeClose(hFindVolume);
106 ExceptionList = local_10;
107 FUN_00434a10(local_14 ^ (uint)&stack0xffffffffc);
108 return;
109 }
110

```

4.2.1.6. Planificación Multihilo y Cola de Trabajo. La función FUN_0040b840

coordina el cifrado paralelo realizando las siguientes tareas:

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

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

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

Figura 145

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

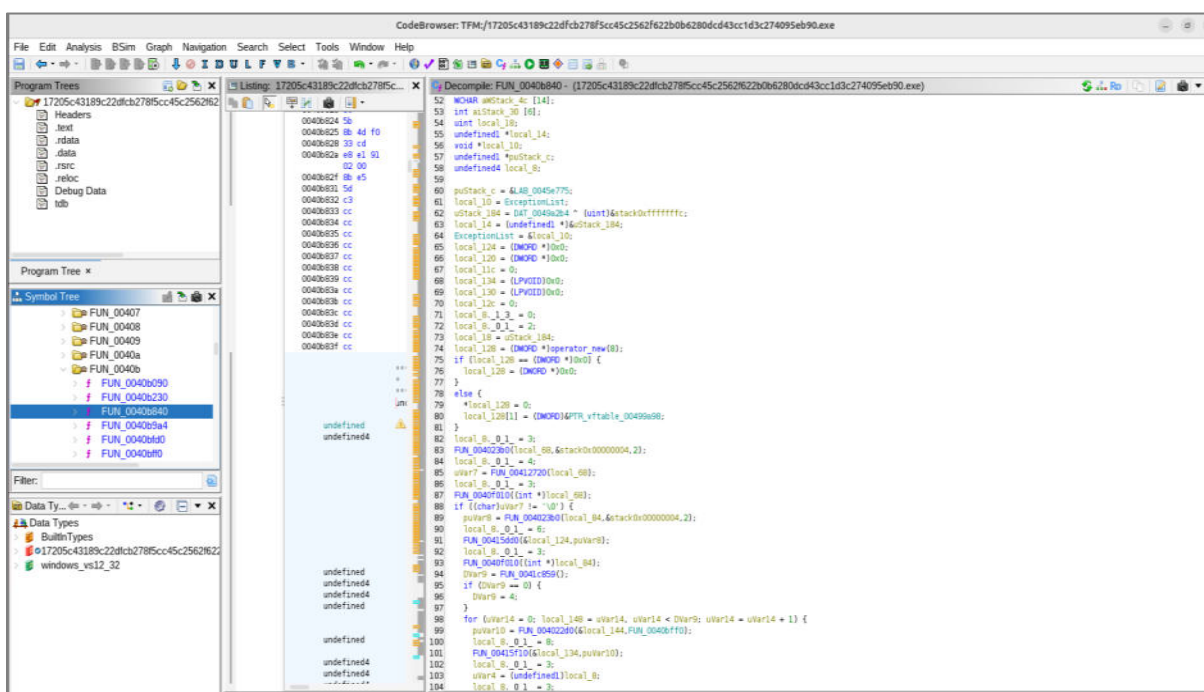


Figura 146

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

```

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

```

Figura 147

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

```

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

```

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

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

Figura 148

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

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

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

Figura 149

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

```

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

```

Figura 150

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

```

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

```

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

Figura 151

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

```

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

```

Figura 152

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

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

Figura 153

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

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

Adicionalmente, antes de cifrar valida que el origen existe:

Figura 154

Fragmento del código de la función FUN_0040bff0 para validación de origen

```
225     uVar8 = FUN_00412720(local_e8);
226     if ((char)uVar8 == '\0') {
227         FUN_004187b0(local_e8, local_19c);
228         local_8 = 0x23;
229         FUN_00418d70(0x473374);
230         local_8 = 0x21;
231         FUN_0040f010(local_1);
232     }
233     bVar4 = FUN_0040d500(1);
234     if (bVar4) {
235         FUN_004092d0(local_9);
236         puVar5 = (uint *)FUN_004092d0(local_9);
237         local_8 = 0x24;
238         FUN_00415060(local_3);
239         local_8 = 0x21;
240         FUN_0040f010(local_2);
241         if (local_358 == 0)
242             FUN_0040d500(local_1);
```

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

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

Figura 155

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

```

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

```

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

Figura 156

Fragmento del código de la función FUN_0040d500

```

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

```

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

Figura 157

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

```

244         else {
245             FUN_0040d130((undefined1 *)&local_48);
246             local_68 = local_48;
247             uStack_64 = uStack_44;
248             uStack_60 = uStack_40;
249             uStack_5c = uStack_3c;

```

Figura 158

Fragmento del código de la función FUN_0040d130

```

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

```

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

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

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

Figura 159

Fragmento del código de la función FUN_0040bfff0 donde llama a la función FUN_00415060

```

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

```

Figura 160

Fragmento del código de la función FUN_00415060

```

Decompile: FUN_00415060 - (17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe)
1
2 void __thiscall FUN_00415060(void *this,uint *param_1,uint param_2)
3
4 {
5     int iVar1;
6     code *pcVar2;
7     wchar_t ***pppwVar3;
8     basic_filebuf<> *pbVar4;
9     uint uVar5;
10    undefined4 *puVar6;
11    char *pcVar7;
12    int local_4c [5];
13    undefined4 local_38 [2];
14    undefined4 local_30;
15    wchar_t **local_2c [5];
16    uint local_18;
17    uint local_14;
18    void *local_10;
19    undefined1 *puStack_c;
20    undefined4 local_8;
21
22    puStack_c = &LAB_0045f79e;
23    local_10 = ExceptionList;
24    local_14 = DAT_0049a2b4 ^ (uint)&stack0xffffffffc;
25    ExceptionList = &local_10;
26    local_30 = 0;
27    local_8 = 0;
28    FUN_00405fd0(local_2c,param_1);
29    pppwVar3 = local_2c;
30    if (7 < local_18) {
31        pppwVar3 = (wchar_t ***)local_2c[0];
32    }
33    local_30 = 1;
34    pbVar4 = FUN_00414f70((void *)((int)this + 0x18),(wchar_t *)pppwVar3,param_2,0x40);
35    /* WARNING: Load size is inaccurate */
36    iVar1 = *(int *)(*this + 4);
37    if (pbVar4 == (basic_filebuf<> *)0x0) {
38        uVar5 = ((uint)((int *)iVar1 + 0x38 + (int)this) == 0) * 4 + 2 |
39            *(uint *)(iVar1 + 0xc + (int)this) & 0x17;
40        *(uint *)(iVar1 + 0xc + (int)this) = uVar5;
41        uVar5 = *(uint *)(iVar1 + 0x10 + (int)this) & uVar5;
42        if (uVar5 != 0) {
43            if ((uVar5 & 4) == 0) {
44                pcVar7 = "ios_base::failbit set";
45                if ((uVar5 & 2) == 0) {
46                    pcVar7 = "ios_base::eofbit set";
47                }
48            }
49            else {
50                pcVar7 = "ios_base::badbit set";
51            }
52            puVar6 = (undefined4 *)FUN_00414980(local_38,1);
53            goto LAB_0041518a;

```

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

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

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

Figura 161

Fragmento del código de la función FUN_0040ae10

```

35 FUN_00405fd0(local_54, (uint *)&DAT_0049985c);
36 local_8 = 0;
37 pvParam = local_54;
38 if (7 < local_40) {
39     pvParam = (undefined4 ***)local_54[0];
40 }
41 SystemParametersInfoW(0x14, 0, pvParam, 1);

```

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

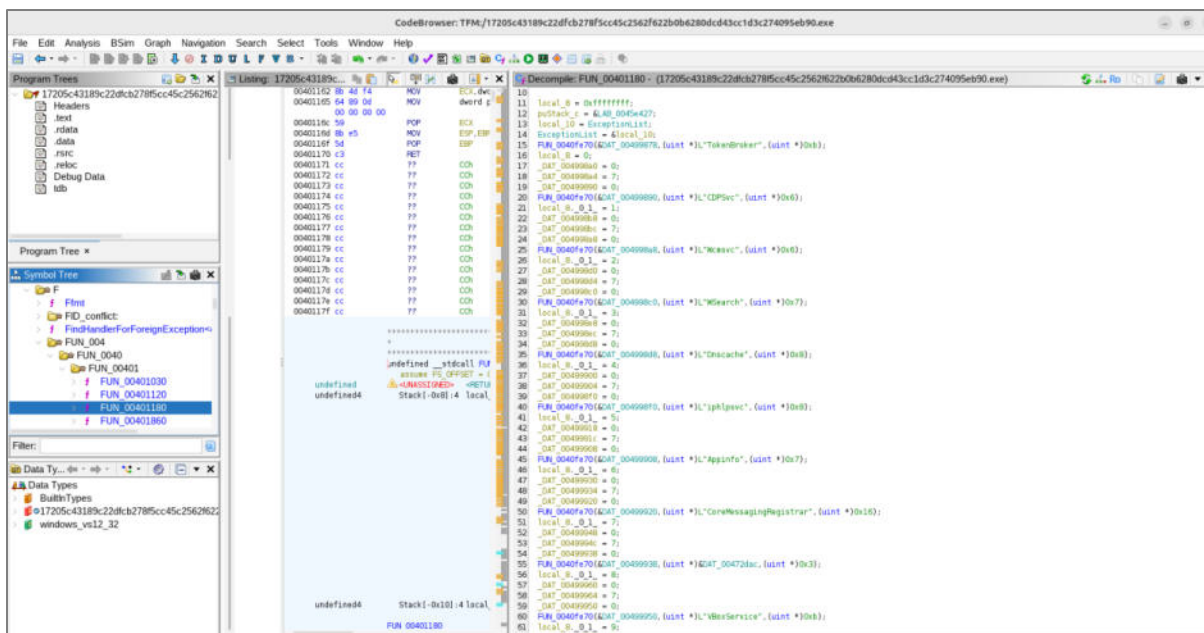
- TokenBroker,
- CDPSvc,
- WSearch,
- Dnscache,
- Appinfo,

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

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

Figura 162

Revisión de código en Decompile: FUN 00401180 en Ghidra



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

- Llama a SHGetSpecialFolderPathW con csidl = 0. Eso corresponde al

CSIDL_DESKTOP (virtual Desktop, y en muchos samples se usa para conseguir la ruta de escritorio del usuario).

- FUN_00406140 parece construir un std::wstring con “\\readme.txt”.
- FUN_00402c90 concatena local_224 (ruta del escritorio) con “\\readme.txt”, se

construye la ruta completa C:\Users\<user>\Desktop\readme.txt

Figura 163

Fragmento del código de la función FUN 0040b090

```

29 SHGetSpecialFolderPath((HWND)0x0,local_224,0,0);
30 puVar1 = FUN_00406140(local_23c,(uint *)L"\\readme.txt");
31 local_8._0_1_ = 1;
32 FUN_00402c90((uint *)local_254,(uint *)local_224,puVar1);
33 local_8._0_1_ = 3;
34 FUN_0040f010((int *)local_23c);
35 FUN_00418d70(0x473484);
36 FUN_00405c50(local_318,local_254,0x22,0x40,1);
37 local_8._0_1_ = 4;
38 if (local_2b4 != 0) {
39     FUN_00405f90(local_23c,(uint *)s_All_of_your_files_are_currently_e_00499008);

```

Figura 164

Detalle del código de la función FUN 0040b090

```

0040b15a 74 61 JZ LAB_0040b1bd
0040b15c = "All of your files are currently encrypted by no_name_software.\r\n\r\nThese files cannot be
0040b15e recovered by any means without contacting our team directly.\r\n\r\nDONT TRY TO RECOVER your data
0040b15f by yourselves. Any attempt to recover your data (including the usage of the additional recovery
0040b161 software) can damage your files. However,\r\n\r\nif you want to try - we recommend choosing the data
0040b163 of the lowest value.\r\n\r\nDONT TRY TO IGNORE us. We've downloaded a pack of your internal data and
0040b165 are ready to publish it on our news website if you do not respond.\r\n\r\nSo it will be better for both
0040b167 sides if you contact us as soon as possible.\r\n\r\nDONT TRY TO CONTACT feds or any recovery
0040b169 companies.\r\n\r\nWe have our informants in these structures, so any of your complaints will be
0040b16b immediately directed to us.\r\n\r\nSo if you will hire any recovery company for negotiations or send
0040b16d requests to the police/FBI/investigators, we will consider this as a hostile intent and initiate the
0040b16f publication of whole compromised data immediately.\r\n\r\nDONT move or rename your files. These
0040b171 parameters can be used for encryption/decryption process.\r\n\r\n\r\nTo prove that we REALLY can get
0040b173 your data back - we offer you to decrypt two random files completely free of charge.\r\n\r\n\r\nYou can
0040b175 contact our team directly for further instructions through our website.\r\n\r\n\r\nTOR VERSION
0040b177 \r\n\r\nYou should download and install TOR browser first
0040b179 https://torproject.org/\r\n\r\n\r\nhttps://aazbsbgay565vlu2c6bz7y6tfebkbctvtyvotl33x77t7mypyxd.onio
0040b17b n:80\r\n\r\n\r\nYour company id for log in: c98fa42b-3233-45df-bd7c-42529c44cb70\r\n\r\nYour company key: 3
0040b17d of any of your dc through comma. Example: 'YDC1, DC2, DC3'. You can type less if you have no
0040b17f enough\r\n\r\n\r\nYOU SHOULD BE AWARE\r\n\r\nWe will speak only with an authorized person. It can be the
0040b181 CEO, top management, etc. \r\n\r\nIn case you are not such a person - DONT CONTACT US! Your decisions
0040b183 and action can result in serious harm to your company! \r\n\r\nInform your supervisors and sta...
0040b185
0040b1a3 84 88 ec LEA ECK=>local_318, [EBP + 0xffffcc]
0040b1a5 fc ff fi
0040b1a9 e8 c2 51 CALL FUN_00410370
0040b1ab 00 00
0040b1ae c4 65 fc 04 MOV byte ptr [EBP + local_8], 0x4

```

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

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

Figura 165

Fragmento del código de la función FUN_0040ae10

```
108 local_30 = uVar4;
109 FUN_00434c40(uVar9,uVar5);
110 FUN_00418d10(0x4737c4);
111 local_8 = 0xffffffff;
112 FUN_0040ee40(&local_8);
113 ExceptionList = local_8;
114 return;
115 }
116
```

	Hex	Decimal
dword	4737C4h	4667332
sdword	4737C4h	4667332
wchar16[]	u"G样例"	
Address	004737c4	
Symbol	s_Done_time:_.4f_seconds,_encrypt_004737c4	

Figura 166

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

```
= "Done time: %.4f seconds, encrypted: %.4f gb"
return;
}
pvVar7 = ope
if (pvVar7 =
pvVar8 = fun
```

```
= "Done time: %.4f seconds, encrypted: %.4f gb"
```

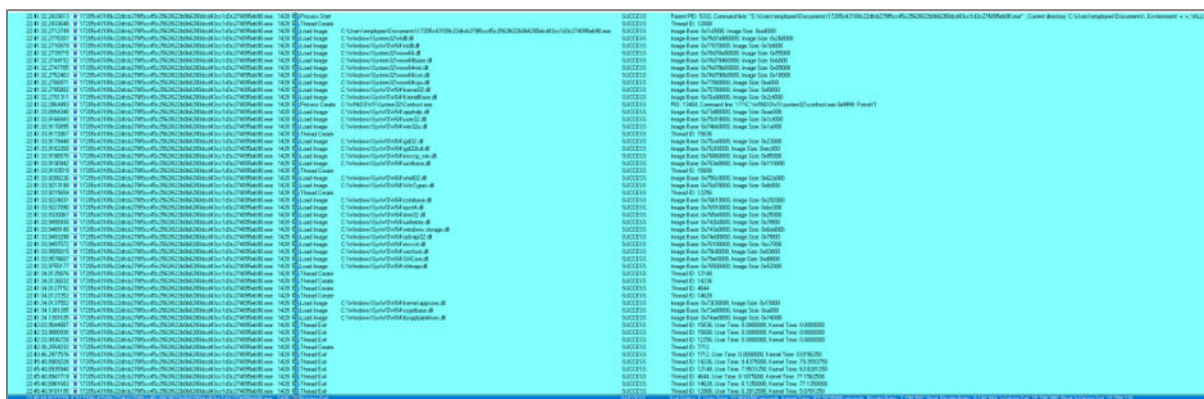
4.2.2. Análisis Dinámico. A continuación, se describirán los puntos clave del funcionamiento de Black Basta, de acuerdo con lo observado durante la ejecución del ransomware y la información recopilada de los artefactos forenses.

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

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

Figura 167

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



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

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

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

A grandes rasgos se puede decir que el ransomware:

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

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

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

Figura 168

Carga de imágenes y creación de threads

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

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

Figura 169

Llamado al módulo de Desktop Wallpaper

22:41:33.9476004	17205c43189c22db278f5cc45c256262b6b6200dc03cc1d3c274095eb90.exe	1420	RegOpenKey	HKCU\Control Panel\Desktop	SUCCESS	Desired Access: Write
22:41:33.9476004	17205c43189c22db278f5cc45c256262b6b6200dc03cc1d3c274095eb90.exe	1420	RegOpenKeyS	HKCU\Control Panel\Desktop	SUCCESS	
22:41:33.9476004	17205c43189c22db278f5cc45c256262b6b6200dc03cc1d3c274095eb90.exe	1420	RegSetValueEx	HKCU\Control Panel\Desktop\Wallpaper	SUCCESS	Type: REG_SZ, Length: 2, Data:
22:41:33.9476004	17205c43189c22db278f5cc45c256262b6b6200dc03cc1d3c274095eb90.exe	1420	RegOpenKey	HKCU\Control Panel\Desktop	SUCCESS	
22:41:33.9489188	17205c43189c22db278f5cc45c256262b6b6200dc03cc1d3c274095eb90.exe	1420	Load Image	C:\Windows\System32\Windows.storage.dll	SUCCESS	Image Base: 0x743e0000, Image Size: 0x6dd000
22:41:33.9489188	17205c43189c22db278f5cc45c256262b6b6200dc03cc1d3c274095eb90.exe	1420	Load Image	C:\Windows\System32\Windows.storage.dll	SUCCESS	Image Base: 0x743e0000, Image Size: 0x6dd000

Claves leídas:

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

Y una escritura directa:

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

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

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

- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{B4BFCC3A-...}

- HKCU\Control Panel\Desktop\UserPreferencesMask, etc.

Figura 170

Procesos de lectura sobre claves de Explorer y del perfil

22:41:33.9645075	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM	SUCCESS	Query Name
22:41:33.9646006	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions	SUCCESS	Desired Access: Read
22:41:33.9646075	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions	SUCCESS	KeyPathName: Data KeyPathName: Information, Length: 0
22:41:33.9646075	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions	SUCCESS	Query: Handle: 0, Handle: 0, 0x00
22:41:33.9646075	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Desired Access: Read
22:41:33.9647010	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions	SUCCESS	
22:41:33.9647097	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions	SUCCESS	Type: REG_DWORD, Length: 4, Data: 4
22:41:33.9647168	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_SZ, Length: 16, Data: Desktop
22:41:33.9647260	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 90
22:41:33.9647270	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 90
22:41:33.9647467	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 144
22:41:33.9647607	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_SZ, Length: 16, Data: Desktop
22:41:33.9647657	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 144
22:41:33.9647620	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 144
22:41:33.9647672	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_EXPAND_SZ, Length: 84, Data: @SystemRoot\System32\Winlogon.dll; 2780
22:41:33.9648034	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_EXPAND_SZ, Length: 80, Data: %SystemRoot%\System32\Winlogon.dll; 183
22:41:33.9648141	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 144
22:41:33.9648231	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 144
22:41:33.9648218	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 144
22:41:33.9648148	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 16
22:41:33.9648100	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
22:41:33.9648128	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
22:41:33.9648112	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 16
22:41:33.9648294	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
22:41:33.9648368	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 16
22:41:33.9648440	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 16
22:41:33.9648410	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
22:41:33.9648617	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 90
22:41:33.9648662	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	NAME NOT FOUND	Length: 90
22:41:33.9595777	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	Load Image	C:\Windows\System32\Winlogon.dll	SUCCESS	Image Base: 0x75500000, Image Size: 0x2000
22:41:33.9607253	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Query: Handle: 0, Handle: 0, 0x00
22:41:33.9607305	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	Desired Access: Read
22:41:33.9607368	1720543109-228a627895c45-59626226a200a081c1d5-274095e00	1428	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\B4BFCC3A-0	SUCCESS	

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

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

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

4.2.2.5. Comprobaciones de Estado del Sistema SafeBoot, SmartLocker y Session

Manager. En la captura de Procmon aparecen muchas consultas hacia claves del sistema bajo el proceso de la muestra del ransomware, estas son:

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

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

Manager

Adicionalmente, claves bajo HKLM\SOFTWARE\Microsoft\Windows

NT\CurrentVersion\AppCompatFlags\Sdb y valores tipo sysmain.sdb

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

- RegOpenKey HKLM\SOFTWARE\Microsoft\Windows

NT\CurrentVersion\AppCompatFlags\Sdb\SMARTLOCKER

- RegQueryValue ... \SMARTLOCKER\ENABLED
- RegQueryValue ... \SMARTLOCKER\DISABLED

Figura 171

Consultas de claves del sistema por parte del ransomware

[illegible]

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

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

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

Comprobación de llamado a Image File Executions Options

[illegible]

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

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

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

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

Figura 173

Análisis realizado con la herramienta Procmon



Figura 174

Total de eventos registrados por Procmon

22:41:31,0087716	Explorer.EXE	5332	RegCloseKey	HKCR\CLSID\{603D3800-BD81-11d0-...	SUCCESS	
22:41:31,0113871	Explorer.EXE	5332	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Name
22:41:31,0114442	Explorer.EXE	5332	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
22:41:31,0114589	Explorer.EXE	5332	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
22:41:31,0114773	Explorer.EXE	5332	RegOpenKey	HKCU\Software\Classes\Applications\...	NAME NOT FOUND	Desired Access: R...
22:41:31,0115215	Explorer.EXE	5332	RegOpenKey	HKCR\Applications\Procmon64.exe	NAME NOT FOUND	Desired Access: R...
22:41:31,0115578	Explorer.EXE	5332	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Name
Showing 443.818 of 22.411.855 events (1.%)		Backed by virtual memory				

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

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

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

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

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

- `QueryDirectory`
- `CreateFile`
- `ReadFile`
- `CloseFile`

Se observa actividad en directorios de usuario:

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

Se observa actividad en directorios del sistema:

- `C:\Windows\System32`

- C:\Program Files
- C:\ProgramData

Se observa actividad en directorios críticos del sistema de archivos:

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

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

Figura 177

Creación y cierre de archivos en unidades de disco detectadas

274095eb90.exe	1428	QueryNameInfo...	C:\
274095eb90.exe	1428	QueryAttributel...	C:\
274095eb90.exe	1428	CloseFile	C:\
274095eb90.exe	1428	CreateFile	D:\
274095eb90.exe	1428	QueryNameInfo...	D:\
274095eb90.exe	1428	QueryAttributel...	D:\
274095eb90.exe	1428	CloseFile	D:\
274095eb90.exe	1428	CreateFile	C:\
274095eb90.exe	1428	QueryNetwork...	C:\
274095eb90.exe	1428	CloseFile	C:\

Figura 178

Registro de búsqueda del ransomware de archivos que cifrar

1428	CreateFile	C:\ProgramData	SUCCESS	DesiredAccess: ReadAttributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenResult: Opened
1428	QueryNetwork...	C:\ProgramData	SUCCESS	CreationTime: 1/4/2024 2:26:06, LastAccessTime: 27/11/2025 22:41:32, LastWriteTime: 27/11/2025 20:47:54, ChangeTime: 27/11/2025 20:47:54, AllocationSize: 4096, EndOfFile: 4096, FileAttributes: HDNCI
1428	CloseFile	C:\ProgramData	SUCCESS	
1428	CreateFile	C:\Recovery	SUCCESS	DesiredAccess: ReadAttributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenResult: Opened
1428	QueryNetwork...	C:\Recovery	SUCCESS	CreationTime: 17/11/2025 17:40:01, LastAccessTime: 17/11/2025 23:52:43, LastWriteTime: 17/11/2025 23:52:43, ChangeTime: 17/11/2025 23:52:43, AllocationSize: 0, EndOfFile: 0, FileAttributes: HDNCI
1428	CloseFile	C:\Recovery	SUCCESS	
1428	CreateFile	C:\usafle.apo	SHARING VIOLAT...	DesiredAccess: ReadAttributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a
1428	CreateFile	C:\System Volume Information	SUCCESS	DesiredAccess: ReadAttributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenResult: Opened
1428	QueryNetwork...	C:\System Volume Information	SUCCESS	CreationTime: 17/11/2025 17:36:53, LastAccessTime: 25/11/2025 23:10:40, LastWriteTime: 25/11/2025 23:10:40, ChangeTime: 25/11/2025 23:10:40, AllocationSize: 12288, FileAttributes: HSD
1428	CloseFile	C:\System Volume Information	SUCCESS	
1428	CreateFile	C:\Users	SUCCESS	DesiredAccess: ReadAttributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenResult: Opened
1428	QueryNetwork...	C:\Users	SUCCESS	CreationTime: 1/4/2024 2:21:16, LastAccessTime: 27/11/2025 22:41:34, LastWriteTime: 18/11/2025 0:15:50, ChangeTime: 18/11/2025 0:15:50, AllocationSize: 4096, EndOfFile: 4096, FileAttributes: RD
1428	CloseFile	C:\Users	SUCCESS	
1428	QueryDirectory	C:\	NO MORE FILES	FileInformationClass: FileBothDirectoryInformation
1428	CloseFile	C:\	SUCCESS	

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

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

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

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

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

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

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

Registros de procesos de encriptación de archivos

[illegible]

Registros de procesos de encriptación de archivos

```

1 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
2 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
3 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
4 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
5 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
6 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
7 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
8 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
9 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
10 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
11 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
12 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
13 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
14 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
15 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
16 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
17 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
18 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
19 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
20 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
21 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
22 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
23 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
24 ReadFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe;encrypted
25 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\25_209_1026_0002\OneDriveSetup.exe;encrypted
26 WriteFile C:\Users\employee\AppData\Local\Microsoft\OneDrive\25_209_1026_0002\OneDriveSetup.exe;encrypted
27 QueryStandardInformationFile C:\Users\employee\AppData\Local\Microsoft\Edge\UserData\Component_crx_cache\bbc93cbca8636743d1fbd32be3bdc8ef802323895b4d136c26b9cbcf40a6a4e;encrypted
28 WriteFile C:\Users\employee\AppData\Local\Microsoft\Edge\UserData\Component_crx_cache\bbc93cbca8636743d1fbd32be3bdc8ef802323895b4d136c26b9cbcf40a6a4e;encrypted
29 ReadFile C:\Users\employee\AppData\Local\Microsoft\Edge\UserData\Component_crx_cache\bbc93cbca8636743d1fbd32be3bdc8ef802323895b4d136c26b9cbcf40a6a4e;encrypted

```

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

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

En primer lugar, un evento ID 1 muestra la ejecución de WinRAR por parte del usuario “employee”, seguidamente, se registra un evento ID 11 (regla Downloads) donde WinRAR crea un archivo con nombre tipo hash en la carpeta Downloads. Instantes después, un segundo evento ID 1 evidencia la ejecución explícita del binario

17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe desde el directorio de descargas, con explorer.exe como proceso padre, no muestra información de producto o compañía y en el nombre se observa la coincidencia exacta con su hash SHA-256, características típicas de muestras de malware utilizadas en laboratorios o campañas delictivas.

Figura 181

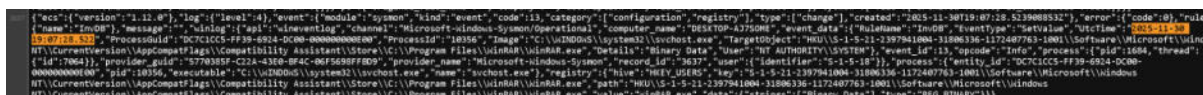
Registros de eventos en Sysmon (parte 1)



```
{
  "version": "1.1.0",
  "log": {
    "level": 4,
    "event": {
      "module": "sysmon",
      "kind": "event",
      "code": 11,
      "category": [
        "file"
      ],
      "type": [
        "creation"
      ],
      "created": "2023-11-30T10:07:16.3300000Z",
      "error": {
        "code": 0,
        "rule": "Rule1"
      },
      "name": "Downloads",
      "message": "winlog",
      "channel": "Microsoft-Windows-Sysmon/Operational",
      "computer_name": "DESKTOP-A3750M",
      "event_data": {
        "RuleName": "Downloads",
        "UtcTime": "2023-11-30T10:07:16.330Z",
        "ProcessGuid": "DC7C1C55-95EC-692C-2907-000000000000",
        "ProcessId": "13408",
        "Image": "C:\\Program Files\\WinRAR\\WinRAR.exe",
        "TargetFilename": "C:\\Users\\Employee\\Downloads\\17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe",
        "CreationUtcTime": "2023-11-30T10:07:16.330Z",
        "User": "DESKTOP-A3750M\\Employee",
        "EventId": 11,
        "OpCode": "Info",
        "Process": {
          "pid": 1684,
          "thread": {
            "id": 7064
          },
          "provider_guid": "5770835F-C224-4368-B94C-00F5589F7809",
          "provider_name": "Microsoft-Windows-Sysmon",
          "record_id": "3636",
          "user": {
            "identifier": "S-1-5-18"},
          "file": {
            "path": "C:\\Users\\Employee\\Downloads\\17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe",
            "directory": "C:\\Users\\Employee\\Downloads",
            "name": "17205c43189c22dfcb278f5cc45c2562f622b0b6280dcd43cc1d3c274095eb90.exe",
            "signature": {
              "subject_name": "",
              "status": "",
              "signed": false,
              "valid": false,
              "process": {
                "entry_id": "DC7C1C55-95EC-692C-2907-000000000000",
                "pid": 13408,
                "executable": "C:\\Program Files\\WinRAR\\WinRAR.exe",
                "name": "WinRAR.exe",
                "hash": {},
                "pe": {
                  "original_file_name": "",
                  "company": "",
                  "description": "",
                  "file_version": "",
                  "product": ""},
                "sysmon": {
                  "file": {
                    "archived": false,
                    "is_executable": false}}}}}}}}}
```

Figura 182

Registros de eventos en Sysmon (parte 2)



```
{
  "version": "1.1.0",
  "log": {
    "level": 4,
    "event": {
      "module": "sysmon",
      "kind": "event",
      "code": 1,
      "category": [
        "configuration",
        "registry"
      ],
      "type": [
        "change"
      ],
      "created": "2023-11-30T10:07:16.3300000Z",
      "error": {
        "code": 0,
        "rule": "Rule1"
      },
      "name": "Registry",
      "message": "winlog",
      "channel": "Microsoft-Windows-Sysmon/Operational",
      "computer_name": "DESKTOP-A3750M",
      "event_data": {
        "RuleName": "Registry",
        "UtcTime": "2023-11-30T10:07:16.330Z",
        "ProcessGuid": "DC7C1C55-95EC-692C-2907-000000000000",
        "ProcessId": "10356",
        "Image": "C:\\WINDOWS\\system32\\svchost.exe",
        "TargetObject": "HKEY\\S-1-5-21-3397941004-31806336-1172407763-1001\\Software\\Microsoft\\Windows NT\\CurrentVersion\\AppCompatFlags\\Compatibility Assistant\\Store\\C:\\Program Files\\WinRAR\\WinRAR.exe",
        "Details": "Binary Data",
        "User": "NT AUTHORITY\\SYSTEM",
        "EventId": 1,
        "OpCode": "Info",
        "Process": {
          "pid": 1684,
          "thread": {
            "id": 7064
          },
          "provider_guid": "5770835F-C224-4368-B94C-00F5589F7809",
          "provider_name": "Microsoft-Windows-Sysmon",
          "record_id": "3637",
          "user": {
            "identifier": "S-1-5-18"},
          "process": {
            "entry_id": "DC7C1C55-95EC-692C-2907-000000000000",
            "pid": 10356,
            "executable": "C:\\WINDOWS\\system32\\svchost.exe",
            "name": "svchost.exe",
            "registry": {
              "hive": "HKEY_USERS",
              "key": "S-1-5-21-3397941004-31806336-1172407763-1001\\Software\\Microsoft\\Windows NT\\CurrentVersion\\AppCompatFlags\\Compatibility Assistant\\Store\\C:\\Program Files\\WinRAR\\WinRAR.exe",
              "path": "HKEY\\S-1-5-21-3397941004-31806336-1172407763-1001\\Software\\Microsoft\\Windows NT\\CurrentVersion\\AppCompatFlags\\Compatibility Assistant\\Store\\C:\\Program Files\\WinRAR\\WinRAR.exe",
              "value": "WinRAR.exe",
              "data": {
                "strings": [
                  "Binary Data"
                ],
                "type": "REG_BINARY"}}}}}}}
```

Registros de eventos en Sysmon por ejecución del Binario

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

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

Registros de eventos ID 13 en Sysmon por ejecución del Binario

Figura 185

Registro de evento ID 5 en Sysmon por terminación de proceso

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

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

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

Figura 186

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

```

11:03.1895518Z", "AccessedTime": "2025-11-18T05:10:37.3027208Z", "Type": "Win32", "Name": "FontCache-5-1-5-21_
205-11-18T04:40:15.1242291Z", "AccessedTime": "2025-11-18T04:40:15.1242291Z", "Type": "Win32", "Name": "FontC
", "Resident": true, "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 130, "Id": 3, "Inode": 132-130-3, "Size": 340, "Name": "", "Resident": true}], "Hardlinks": [{"Users\\Employee\\AppData\\Local\\Microsoft\\Edge\\User Data\\Default
65Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102260, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 134-16-0, "Size": 72, "Name
700723Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102409, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 135-16-0, "Size": 72,
ParentSequenceNumber": 13}, {"Times": {"CreateTime": "2025-11-19T01:09:50.815052Z", "FileModifiedTime": "2025-11-28T01:45:16.3122155Z", "MFTModifiedTime": "2025-11-28T01:45:16.3220852Z", "AccessedTime": "2025-11
04", "Typeid": 169, "Id": 3, "Inode": 137-169-3, "Size": 8192, "Name": "$I30", "Resident": false, "Type": "INDEX_ROOT", "Typeid": 176, "Id": 4, "Inode": 137-176-4, "Size": 8, "Name": "$I30", "Resident": true}], "Hardlinks": [
, "ParentEntryNumber": 132, "ParentSequenceNumber": 3}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 138-16-0, "Size": 72, "Name": "", "Resident": true, "Type": "FILE_NAME", "Typeid
18T04:40:15.165972Z", "Type": "Win32", "Name": "5-1-5-21-2397941004-3106336-1172407763-1000", "ParentEntryNumber": 138, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16
22Z", "AccessedTime": "2025-11-18T04:40:15.165972Z", "Type": "Win32", "Name": "SystemAppData", "ParentEntryNumber": 139, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16,
"$FILE_NAME", "Typeid": 48, "Id": 2, "Inode": 141-48-2, "Size": 170, "Name": "", "Resident": true, "Type": "INDEX_ROOT", "Typeid": 144, "Id": 1, "Inode": 141-144-1, "Size": 144, "Name": "$I30", "Resident": true}], "Hard
w), "Type": "FILE_NAME", "Typeid": 48, "Id": 3, "Inode": 142-48-3, "Size": 76, "Name": "", "Resident": true, "Type": "INDEX_ROOT", "Typeid": 144, "Id": 1, "Inode": 142-144-1, "Size": 16, "Name": "$I30", "Resident": tr
24.7657148Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102290, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 143-16-0, "Size
2", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102307, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 144-16-0, "Size": 72, "Name
NAME", "Typeid": 48, "Id": 2, "Inode": 141-48-2, "Size": 170, "Name": "", "Resident": true, "Type": "INDEX_ROOT", "Typeid": 144, "Id": 1, "Inode": 141-144-1, "Size": 16, "Name": "$I30", "Resident": true, "Type": "SI
ModifiedTime": "2025-11-18T05:04:26.2870535Z", "MFTModifiedTime": "2025-11-18T05:04:26.2870535Z", "AccessedTime": "2025-11-18T05:04:26.2870535Z", "Type": "Win32", "Name": "a226d7d9d6691e_com15.dat", "Parent
55Z", "FileModifiedTime": "2025-11-18T05:04:26.2870535Z", "MFTModifiedTime": "2025-11-18T05:04:26.2870535Z", "AccessedTime": "2025-11-18T05:04:26.2870535Z", "Type": "Win32", "Name": "a226d7d9d6691e_com15.dat,
", "FileModifiedTime": "2025-11-18T05:04:26.2870535Z", "MFTModifiedTime": "2025-11-18T05:04:26.2870535Z", "AccessedTime": "2025-11-18T05:04:26.2870535Z", "Type": "Win32", "Name": "a226d7d9d6691e_com15.dat, LOC
38929Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102353, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 150-16-0, "Size": 72,
-36T04:04:26.769981Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102321, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 151-16
", "Win32", "Name": "7. defaultuser0.cdp", "ParentEntryNumber": 116, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 153-16-0, "Size": 72, "Name": "", "Residow
104", "Type": "2025-11-28T01:45:39.479072Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102270, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0,
44-24.7064525Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102280, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 155-16-0, "Size
44-Store\\Operational.evtx", "ParentEntryNumber": 6861, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 156-16-0, "Size": 72, "Name": "", "Resident": true
74-Windows-CloudStore\\Operational.evtx", "ParentEntryNumber": 6861, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 157-16-0, "Size": 72, "Name": "", "Re
rosoft\\Windows-CloudStore\\Initialization.evtx", "ParentEntryNumber": 6861, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 158-16-0, "Size": 72, "Name
32.02.955294Z", "Type": "Win32", "Name": "f.000150\\encrypted", "ParentEntryNumber": 102264, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 159-16-0, "Si
4408Z", "AccessedTime": "2025-11-26T04:04:24.936807Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 102300, "ParentSequenceNumber": 3}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Type
45Setup.exe\\encrypted", "ParentEntryNumber": 190761, "ParentSequenceNumber": 5}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 161-16-0, "Size": 72, "Name": "", "Resident": true, "Type
j", "Type": "LOG.old\\encrypted", "ParentEntryNumber": 102128, "Id": 1, "Size": 328, "Name": "", "Resident": true, "Hardlinks": [{"Users\\Employee\\AppData\\Local\\Microsoft\\Edge\\User Data\\Default\\Extension State\\I
Name": "LOG.old\\encrypted", "ParentEntryNumber": 102454, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 163-16-0, "Size": 72, "Name": "", "Resident": true, "Type
32", "AccessedTime": "2025-11-26T04:04:26.4613341Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 103470, "ParentSequenceNumber": 3}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid
Size": 88, "Name": "", "Resident": true, "Type": "SDATA", "Typeid": 128, "Id": 1, "Inode": 165-128-1, "Size": 4, "Name": "", "Resident": true, "Hardlinks": [{"Users\\Employee\\AppData\\Local\\Microsoft\\Edge\\User Da
2025-11-19T01:09:44.3769268Z", "MFTModifiedTime": "2025-11-19T01:09:44.3769268Z", "AccessedTime": "2025-11-25T01:38:01.1800060Z", "Type": "Win32", "Name": "messages.json\\encrypted", "ParentEntryNumber": 20783
167-40-2", "Size": 70, "Name": "", "Resident": true, "Type": "INDEX_ROOT", "Typeid": 144, "Id": 1, "Inode": 167-144-1, "Size": 16, "Name": "$I30", "Resident": true, "Type": "INDEX_INFORMATION", "Typeid": 169, "Id": 3,
32.02.46.4739254Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 103066, "ParentSequenceNumber": 3}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 168-16-0, "Size
104:04:26.4844999Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 103497, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 169-16-0,
73Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 103502, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 170-16-0, "Size": 72, "Name
72", "AccessedTime": "2025-11-26T04:04:27.556807Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 95498, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid
edTime": "2025-11-28T04:04:27.5743897Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 95498, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0,
01:45:44.200007Z", "AccessedTime": "2025-11-26T04:04:27.686623Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 95505, "ParentSequenceNumber": 2}], "Attributes": [{"Type": "STANDARD_INFORM
34-998866Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 1260, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 174-16-0, "Size": 72,
38T01:46:23.763716Z", "Type": "Win32", "Name": "LOG.old\\encrypted", "ParentEntryNumber": 100850, "ParentSequenceNumber": 5}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 175-16-0
857", "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 176-16-0, "Size": 72, "Name": "", "Resident": true, "Type": "FILE_NAME", "Typeid": 48, "Id": 3, "Inode
ParentEntryNumber": 5557, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 176-16-0, "Size": 72, "Name": "", "Resident": true, "Type": "FILE_NAME", "Typeid
51-25T00:57:59.7623796Z", "Type": "Win32", "Name": "UpdateSessionOrchestration.c85a35a-ab81-4429-974f-099989cdfd54.1.etl", "ParentEntryNumber": 3636, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "S
20355Z.38d.5.etl", "ParentEntryNumber": 144017, "ParentSequenceNumber": 1}], "Attributes": [{"Type": "STANDARD_INFORMATION", "Typeid": 16, "Id": 0, "Inode": 178-16-0, "Size": 72, "Name": "", "Resident": true, "Type

```

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

Además, el análisis de las marcas temporales mostró que:

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

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

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

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

Figura 187

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

```

{
  "Name": "SecurityHealth", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\SecurityHealth", "Details": "H:\\system32\\SecurityHealth\\stray.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "WinTray", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\WinTray", "Details": "H:\\system32\\WinTray.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
  "Name": "EdgeCleanup", "OSPath": "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\EdgeCleanup", "Details": "H:\\Program Files (x86)\\Microsoft\\Edge\\Application\\Application\\msedge.exe", "Enabled": "disabled", "Upload": "", "Source": "Windows.Sys.StartupItems"}
}

```

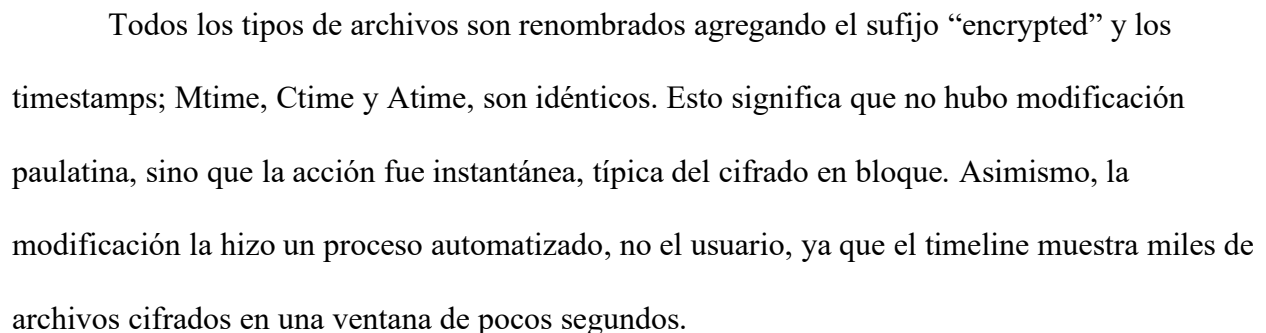
En este caso, el artefacto devuelve 14 entradas, todas correspondientes a software legítimo del sistema, OneDrive, Microsoft Edge o VirtualBox. Es importante destacar que NO se observa ninguna persistencia del ransomware Black Basta. Esto es consistente con el comportamiento típico de Black Basta, en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

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

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

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

Registro de eventos en artefacto Generic.Forensic.Timeline

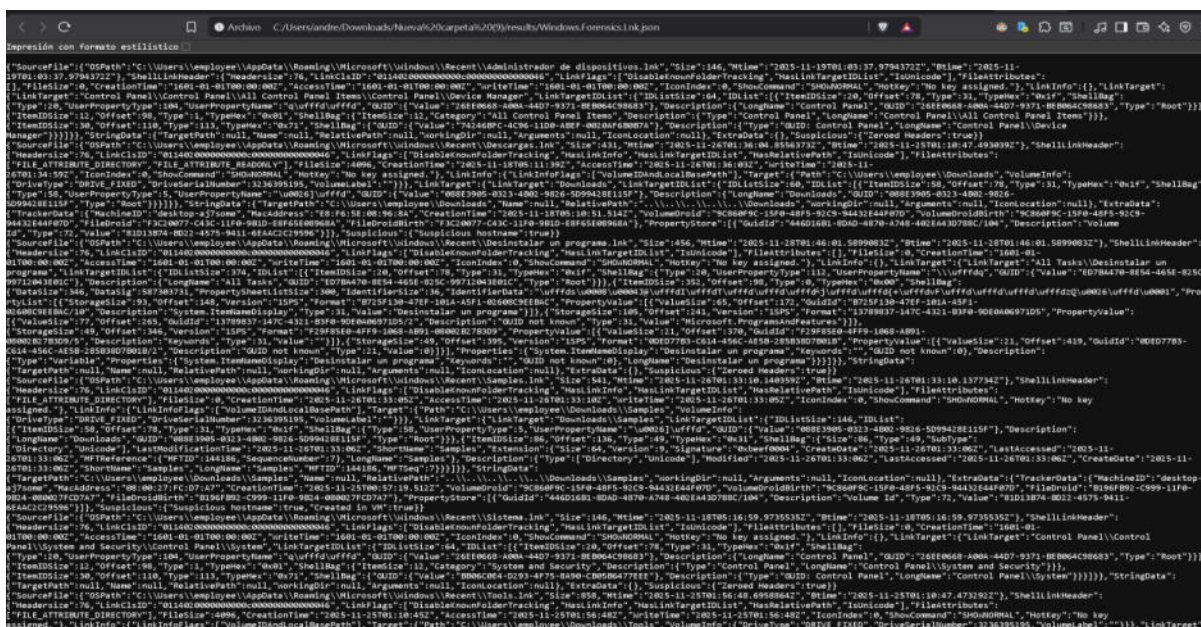


- .txt
- .pdf
- .png
- .jpg
- .mp4
- Bases de datos internas (.jtx, .jcp)

- accesos directos para provocar pérdida de funcionalidad en los equipos, daños visibles al usuario e incremento del impacto psicológico.

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

Registro de eventos en artefacto Windows.Forensics.Lnk



4.2.4. Identificación de Evidencias Adicionales y Verificación de Integridad.

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

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

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

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

Tabla 4

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

N.º	Archivo	Directorio	SHA-256
1	Microsoft-Windows-Sysmon_Operational.evtx	evtx_2025111 4_150238	0e29c1def196bfd88db689ec2bca2b24 02602949d8167ff290d9c5320bf6833a 7d3b57437574e16da3b80457214ce05
	Microsoft-Windows-PowerShell_Operational.evtx	evtx_2025111 4_150238	5a193a320372c57ecb19f40811421e22 5
3	Windows PowerShell.evtx	evtx_2025111	5ced1290d59f396aad798726230f09bb
		4_150238	b9651c196b5624b42fbef7a60a39b5f0

4	Security.evtx	evtx_2025111	5fc0ece1ac2469b3bccd548d021c4eca4
		4_150238	54934ec458084a7ff0d286eebcdcb28
5	System.evtx	evtx_2025111	67546bc73f7848bb2137db24531f0a80
		4_150238	37d7af55cbd3cb5e5293d792e85b887c
6	Microsoft-Windows- Windows Firewall With Advanced	evtx_2025111	f3bb934218d975b77b3daeebec8b1461
	Security_Firewall.evtx	4_150238	e0553f69025eabbb9c56479c09a85631
7	Microsoft-Windows- SmbClient_Security.evtx	evtx_2025111	b790c872a389511da32f190adf66f301
		4_150238	283c86a7acbbbfce154c70858fae70d
8 8	Microsoft-Windows- SMBServer_Security.evtx	evtx_2025111	b790c872a389511da32f190adf66f301
		4_150238	283c86a7acbbbfce154c70858fae70d

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

Tabla 5

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

N.º	Archivo	Directorio	SHA-256
1	Microsoft-Windows-	evtx_202511	dd15aa581814191bbb0b9fa8de366cb5680
	Sysmon_Operational.evtx	14_154547	74b4704ccd081666e0d330773a088

	Microsoft-Windows-		
2	PowerShell_Operational.evtx	evtx_20251114_154547	7025cb4bc9d57389eefdbe68359ff4937bb6ceb825e3925cb16051194e3ebf4d
3	Windows PowerShell.evtx	evtx_20251114_154547	09764b1332ff4e77e16a8ee987fda6bf69b68c268a112a282e81e12198c16de5
4	Security.evtx	evtx_20251114_154547	d576ce79722b7b8b8b9f77a7185d8db0836699e3d267a9158c9a08de13656578
5	System.evtx	evtx_20251114_154547	f9ba6bdf7f24d37210e329279bfb84415cd624612b23edbbf69f5609611afc3c
	Microsoft-Windows-		
6	Windows Firewall With Advanced Security_Firewall.evtx	evtx_20251114_154547	f3bb934218d975b77b3daeebec8b1461e0553f69025eabbb9c56479c09a85631
7	Microsoft-Windows-SmbClient_Security.evtx	evtx_20251114_154547	4f930eafdc7b9ecfbf0b76bc054b3802826508d132869a6eb00dc1b49a466298
8	Microsoft-Windows-SMBServer_Security.evtx	evtx_20251114_154547	b790c872a389511da32f190adf66f301283c86a7acbbbfce154c70858fae70d

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

Tabla 6*Lista de Hash SHA-256 de Archivos PML de Otras evidencias recopiladas*

N.º	Archivo	Tipo	SHA-256
1	procmon_20251114_121	Procmon Log	2cf136ca7e4ec8fa98ab9ee729de470662fa7
	545-1.pml		b4690ec45368470d4763911636d
2	procmon_20251114_121	Procmon Log	527129616af9b5b0a848550095fe774a016
	545.pml		09989b07082c554809e0fcd644315
3	winpmem_20251114_12	Volcado de	296cf640111a015b6ae57a3f5d60e67c9ec8
	3536.raw	Memoria	a1e2390ad7c2086e36c3e7d4cf79

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

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

- Determinar los procesos creados, modificados o finalizados.
- Detectar la ejecución de scripts o comandos (CMD / PowerShell).
- Identificar accesos a recursos del sistema.
- Verificar creación de servicios, modificaciones de firewall y cambios en

políticas.

- Identificar actividad relacionada con Sysmon (si está presente).
- Correlacionar acciones típicas de Black Basta, como carga del

payload, desactivación de mecanismos de seguridad, enumeración de sistema, movimientos laterales (si aplicara), ejecución del módulo de cifrado.

4.2.5.1. *Microsoft-Windows-Sysmon/Operational.evtx*. Tras revisar el archivo

En la **Figura 190**, se muestra información obtenida a raíz del análisis de Microsoft-Sysmon/Operational.evtx con hash:

Figura 190

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

The screenshot shows the Windows Event Viewer interface. The top bar indicates the log is 'Microsoft-Windows-Sysmon_Operational.evtx'. The 'Event' column in the table below is highlighted with a red box. A red arrow points from this box to the 'Description' pane at the bottom, which provides details for Event ID 11. The description states that the component 'Microsoft-Windows-Sysmon' could not be found and lists the following information included with the event: 'DefaultUserModified', '2025-11-14 20:02:40.939', '(f525b92f-8ade-6917-9c03-000000000900)', '4248', 'C:\Users\Isra\Downloads\n1\blackbasta.exe', and 'C:\Users\Default\AppData\Roaming\Microsoft\Windows\SendTo\readme.txt'.

Type	Date	Time	Event	Source	Category	User	Computer
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:41	11	Microsoft-Windows	(11)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:39	1	Microsoft-Windows	(1)	\SYSTEM	DESKTOP-6GNRB4I
Information	14/11/2025	15:02:38	1	Microsoft-Windows	(1)	\SYSTEM	DESKTOP-6GNRB4I

Description

The description for Event ID (11) in Source (Microsoft-Windows-Sysmon) could not be found. Either the component that raises this event is not installed on the computer or the installation is corrupted.You can install or repair the component or try to change Description.

The following information was included with the event:

DefaultUserModified
 2025-11-14 20:02:40.939
 (f525b92f-8ade-6917-9c03-000000000900)
 4248
 C:\Users\Isra\Downloads\n1\blackbasta.exe
 C:\Users\Default\AppData\Roaming\Microsoft\Windows\SendTo\readme.txt
 2025-11-14 20:02:40.939
 DESKTOP-6GNRB4I\Isra

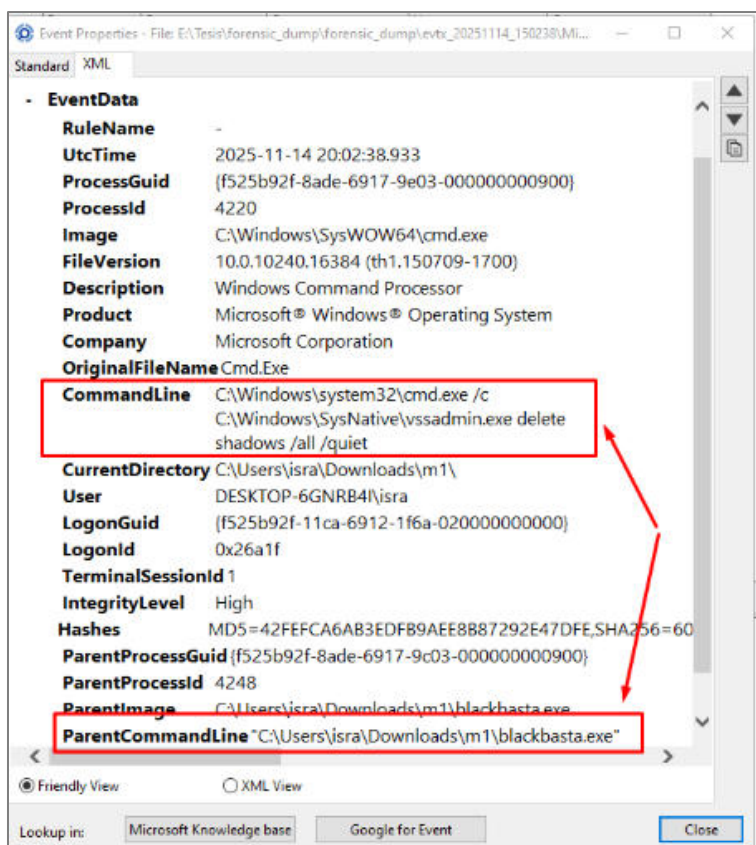
Sysmon registró la ejecución secuencial de los siguientes procesos:

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

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

Figura 191

Revisión del detalle de los eventos generados



Los registros incluyen la siguiente información:

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

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

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

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

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

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

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

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

Tabla 7

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

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

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

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

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

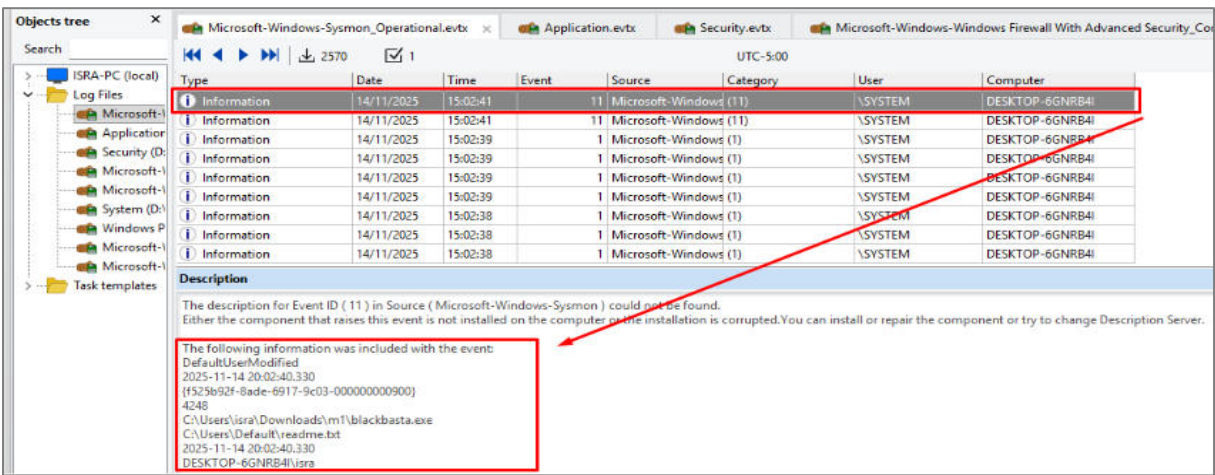
Los eventos reflejan los siguientes parámetros:

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

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

Figura 192

Registros de la creación de notas de rescate obtenidas de Sysmon



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

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

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

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

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

Tabla 8

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

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

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

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

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

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

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

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

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

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

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

A continuación, en la **Figura 193**, se muestra una vista general del archivo

Figura 193

Análisis de archivo procmon 20251114.pml filtrado por blackbasta.exe

Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:15:55.8613292	blackbasta.exe	2900	Process Start		SUCCESS	Parent PID: 2296, Command line: "C:\Users\user\Downloads\1\blackbasta.exe", Current directory: C:\Users\user\Downloads\1
12:15:55.861412	blackbasta.exe	2900	Thread Create		SUCCESS	Thread ID: 5492
12:15:55.861904	blackbasta.exe	2900	Load Image	C:\Users\user\Downloads\1\blackbasta.exe	SUCCESS	Image Base: 0x400000, Image Size: 0x40000
12:15:55.862434	blackbasta.exe	2900	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x7630000, Image Size: 0x3c2000
12:15:55.863310	blackbasta.exe	2900	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x7630000, Image Size: 0x3c2000
12:15:55.863376	blackbasta.exe	2900	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x7630000, Image Size: 0x3c2000
12:15:55.863484	blackbasta.exe	2900	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x7630000, Image Size: 0x3c2000
12:15:55.865495	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.865524	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.865554	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.865762	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.865846	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866176	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866204	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866240	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866268	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866296	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866324	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866352	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866380	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866408	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866436	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866464	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866492	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866520	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866548	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866576	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866604	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866632	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866660	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866688	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866716	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866744	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866772	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866800	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866828	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866856	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866884	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866912	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000
12:15:55.866940	blackbasta.exe	2900	Process Start		REPARSE	Image Base: 0x400000, Image Size: 0x40000

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

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

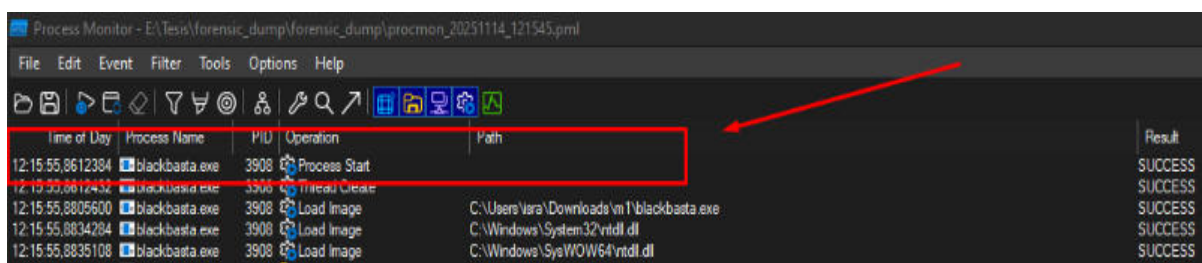
Esta estrategia de respuesta rápida justifica las marcas de tiempo observadas en el registro `procmon_20251114_121545.pml` analizado en Procmon, el cual captura los primeros **1.7 segundos críticos** de la infección (de 12:15:55 a 12:15:57), momento en el cual se detuvo la captura para garantizar su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

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

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

Figura 194

Análisis de archivo `procmon_20251114_121545.pml` filtrado por Process Start



Time of Day	Process Name	PID	Operation	Path	Result
12:15:55.8612384	blackbasta.exe	3908	Process Start		SUCCESS
12:15:55.8612432	blackbasta.exe	3908	Thread Create		SUCCESS
12:15:55.8805600	blackbasta.exe	3908	Load Image	C:\Users\isra\Downloads\m1\blackbasta.exe	SUCCESS
12:15:55.8834284	blackbasta.exe	3908	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS
12:15:55.8835108	blackbasta.exe	3908	Load Image	C:\Windows\System32\WOW64\ntdll.dll	SUCCESS

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

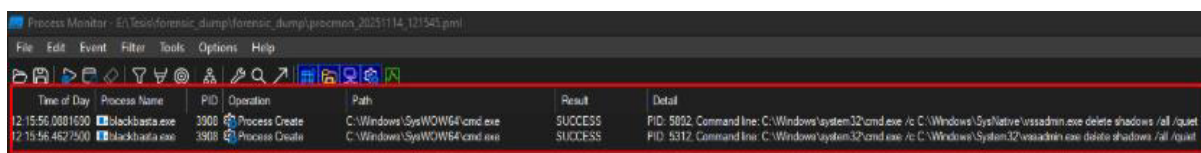
- Fase 2: Evasión de Defensa y Destrucción de Respaldos (T+0.2s).- A 200** milisegundos de iniciarse la ejecución, el malware inició su rutina de evasión. Con el propósito de localizar esta actividad, se utilizó el filtro “*Detail contains vssadmin*” para búsqueda de palabras clave en los argumentos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

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

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

Figura 195

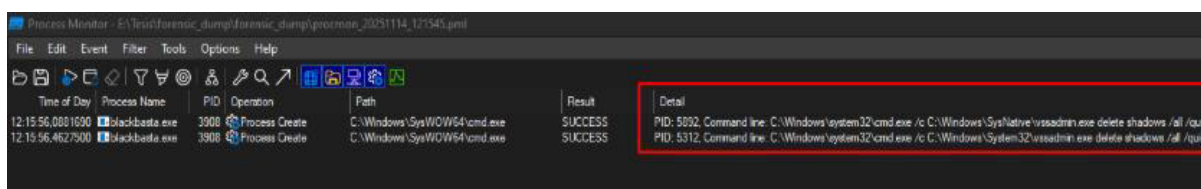
Análisis de archivo procmon_20251114_121545.pml filtrado por Detail contains vssadmin



Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:15:56.0881690	blackbasta.exe	3908	Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 5802, Command line: C:\Windows\system32\cmd.exe /c C:\Windows\SysNative\vssadmin.exe delete shadows /all /quiet
12:15:56.4627500	blackbasta.exe	3908	Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 5312, Command line: C:\Windows\system32\cmd.exe /c C:\Windows\System32\vssadmin.exe delete shadows /all /quiet

Figura 196

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



Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:15:56.0881690	blackbasta.exe	3908	Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 5802, Command line: C:\Windows\system32\cmd.exe /c C:\Windows\SysNative\vssadmin.exe delete shadows /all /quiet
12:15:56.4627500	blackbasta.exe	3908	Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 5312, Command line: C:\Windows\system32\cmd.exe /c C:\Windows\System32\vssadmin.exe delete shadows /all /quiet

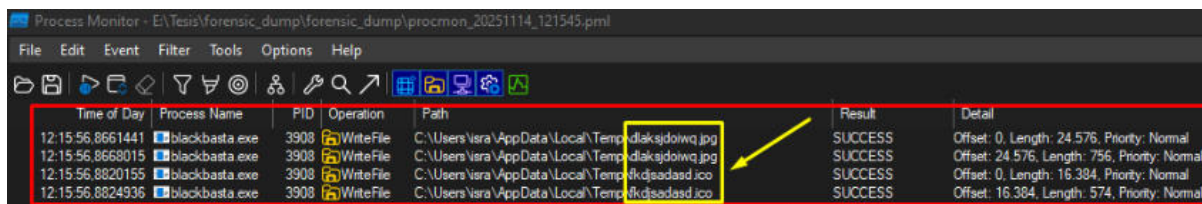
En la **Figura 196**, se evidencia la sofisticación del malware al usar rutas SysNative para saltar la redirección de archivos de 32 bits.

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

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

Figura 197

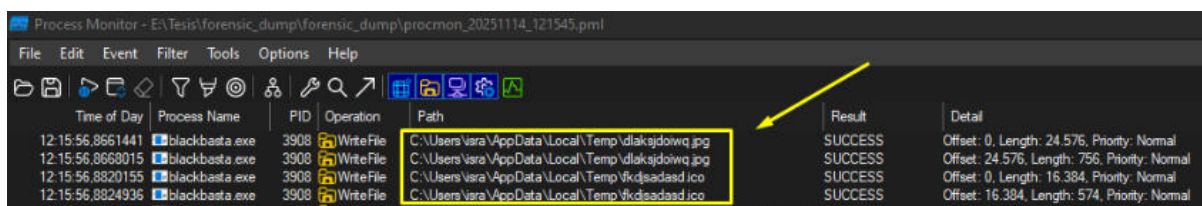
Análisis de archivo procmon_20251114_121545.pml filtrado por Temp y por WriteFile



Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:15:56.8661441	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\dlaksjdowiwq.jpg	SUCCESS	Offset: 0, Length: 24,576, Priority: Normal
12:15:56.8668015	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\dlaksjdowiwq.jpg	SUCCESS	Offset: 24,576, Length: 756, Priority: Normal
12:15:56.8820155	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\fkdsadasd.ico	SUCCESS	Offset: 0, Length: 16,384, Priority: Normal
12:15:56.8824936	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\fkdsadasd.ico	SUCCESS	Offset: 16,384, Length: 574, Priority: Normal

Figura 198

Análisis de archivo procmon_20251114_121545.pml filtrado por Temp



Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:15:56.8661441	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\dlaksjdowiwq.jpg	SUCCESS	Offset: 0, Length: 24,576, Priority: Normal
12:15:56.8668015	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\dlaksjdowiwq.jpg	SUCCESS	Offset: 24,576, Length: 756, Priority: Normal
12:15:56.8820155	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\fkdsadasd.ico	SUCCESS	Offset: 0, Length: 16,384, Priority: Normal
12:15:56.8824936	blackbasta.exe	3908	WriteFile	C:\Users\vara\AppData\Local\Temp\fkdsadasd.ico	SUCCESS	Offset: 16,384, Length: 574, Priority: Normal

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

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

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

Análisis de resultados del archivo procmon 20251114 121545.pml tras creación de readme.txt

[illegible]

Figura 200

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

Time of Day	Process Name	PID	Operation	Path	Result	Detail
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	ACCESS DENIED	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	REPARSE	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a, OpenResult: unknown
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	ACCESS DENIED	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0, OpenResult: Created
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Name: 'C:\Users\PEACOME\...' CreationTime: 14/11/2025 12:15:57, LastAccessTime: 14/11/2025 12:15:57, LastWriteTime: 14/11/2025 12:15:57, ChangeTime: 14/11/2025 12:15:57, FileAttributes: A
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	AllocationSize: 0, EndOfFile: 0, NumberOfLinks: 1, DeletePending: False, Directory: False
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Offset: 0, Length: 4091, Priority: Normal
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Offset: 0, Length: 2
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0, OpenResult: Created
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Name: 'C:\Users\PEACOME\...' CreationTime: 14/11/2025 12:15:57, LastAccessTime: 14/11/2025 12:15:57, LastWriteTime: 14/11/2025 12:15:57, ChangeTime: 14/11/2025 12:15:57, FileAttributes: A
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	AllocationSize: 0, EndOfFile: 0, NumberOfLinks: 1, DeletePending: False, Directory: False
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Offset: 0, Length: 4091, Priority: Normal
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Offset: 0, Length: 2
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	PATH NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0, OpenResult: Created
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Name: 'C:\Users\PEACOME\...' CreationTime: 14/11/2025 12:15:57, LastAccessTime: 14/11/2025 12:15:57, LastWriteTime: 14/11/2025 12:15:57, ChangeTime: 14/11/2025 12:15:57, FileAttributes: A
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	AllocationSize: 0, EndOfFile: 0, NumberOfLinks: 1, DeletePending: False, Directory: False
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Offset: 0, Length: 4091, Priority: Normal
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	SUCCESS	Offset: 0, Length: 2
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	ACCESS DENIED	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	ACCESS DENIED	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0
12/15/2025 12:15:57	blackbasta.exe	3508	WriteFile	C:\Users\PEACOME\...	NAME NOT FOUND	Desired Access: Generic Read/Write, Delete, Disposition: Open, Options: Attributes: N, ShareMode: None, AllocationSize: n/a

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

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

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

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

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

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

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

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

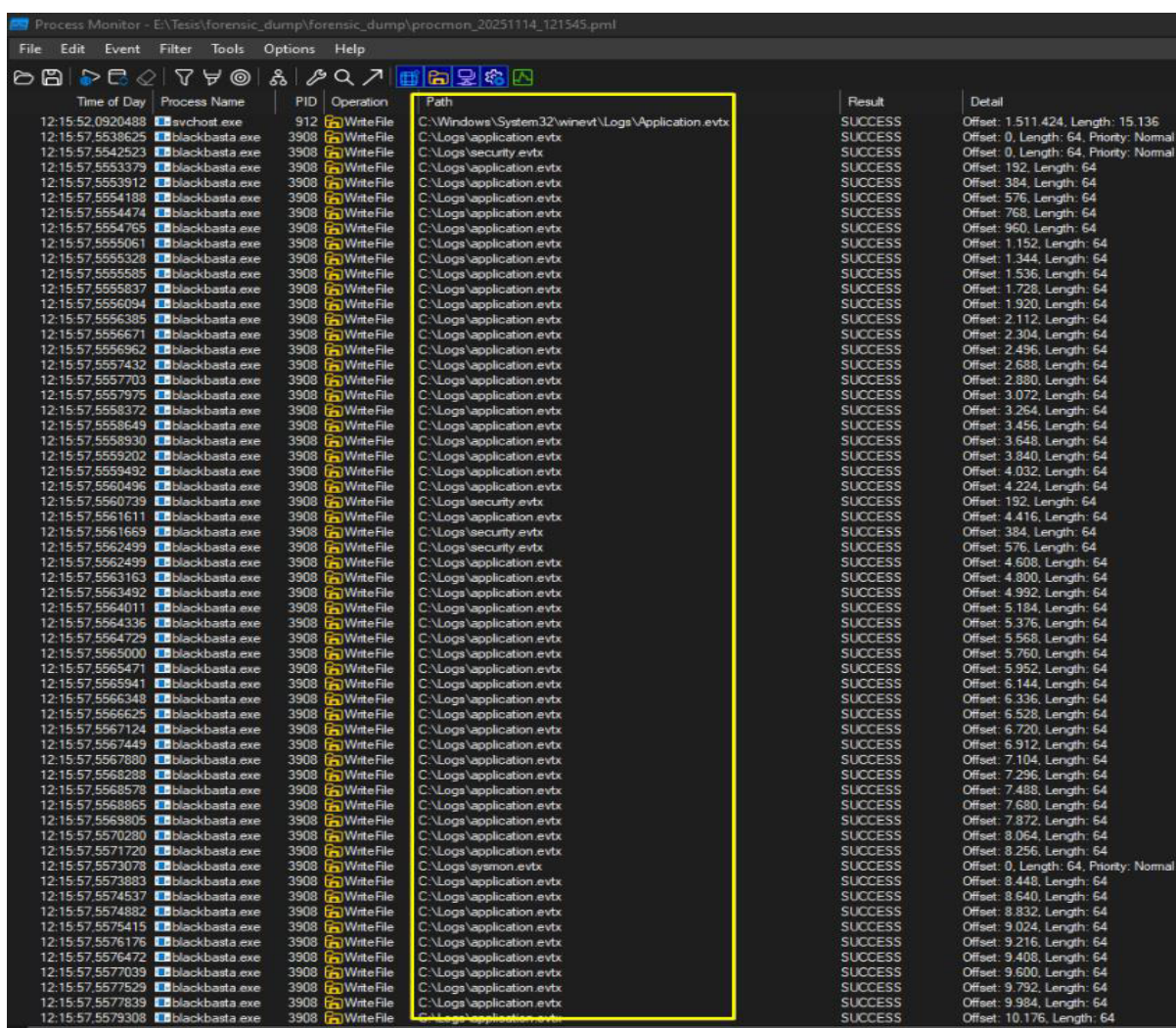
- **Fase 5: Inicio del Cifrado y Actividad Anti-Forenses (T+1.6s).**- El punto clave de la investigación fue localizar el “Tiempo Cero” del cifrado. Se esperaba que el cifrado iniciara con los documentos de usuario; sin embargo, el malware empezó atacando los registros de auditoría. Para esta fase, se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

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

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

Figura 201

Análisis del archivo procmon_20251114_121545.pml por corrupción de evidencias



Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:15:52.0920488	svchost.exe	912	WriteFile	C:\Windows\System32\winevt\Logs\Application.evtx	SUCCESS	Offset: 1,511,424, Length: 15,136
12:15:57.5538625	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 0, Length: 64, Priority: Normal
12:15:57.5542523	blackbasta.exe	3908	WriteFile	C:\Logs\security.evtx	SUCCESS	Offset: 0, Length: 64, Priority: Normal
12:15:57.5553379	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 192, Length: 64
12:15:57.5553912	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 384, Length: 64
12:15:57.5554188	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 576, Length: 64
12:15:57.5554474	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 768, Length: 64
12:15:57.5554765	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 960, Length: 64
12:15:57.5555061	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 1,152, Length: 64
12:15:57.5555328	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 1,344, Length: 64
12:15:57.5555585	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 1,536, Length: 64
12:15:57.5555837	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 1,728, Length: 64
12:15:57.5556094	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 1,920, Length: 64
12:15:57.5556385	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 2,112, Length: 64
12:15:57.5556671	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 2,304, Length: 64
12:15:57.5556962	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 2,496, Length: 64
12:15:57.5557432	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 2,688, Length: 64
12:15:57.5557703	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 2,880, Length: 64
12:15:57.5557975	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 3,072, Length: 64
12:15:57.5558372	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 3,264, Length: 64
12:15:57.5558649	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 3,456, Length: 64
12:15:57.5558930	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 3,648, Length: 64
12:15:57.5559202	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 3,840, Length: 64
12:15:57.5559492	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 4,032, Length: 64
12:15:57.5559746	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 4,224, Length: 64
12:15:57.5560739	blackbasta.exe	3908	WriteFile	C:\Logs\security.evtx	SUCCESS	Offset: 192, Length: 64
12:15:57.5561611	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 4,416, Length: 64
12:15:57.5561663	blackbasta.exe	3908	WriteFile	C:\Logs\security.evtx	SUCCESS	Offset: 384, Length: 64
12:15:57.5562499	blackbasta.exe	3908	WriteFile	C:\Logs\security.evtx	SUCCESS	Offset: 576, Length: 64
12:15:57.5562499	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 4,608, Length: 64
12:15:57.5563163	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 4,800, Length: 64
12:15:57.5563492	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 4,992, Length: 64
12:15:57.5564011	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 5,184, Length: 64
12:15:57.5564336	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 5,376, Length: 64
12:15:57.5564729	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 5,568, Length: 64
12:15:57.5565000	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 5,760, Length: 64
12:15:57.5565471	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 5,952, Length: 64
12:15:57.5565941	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 6,144, Length: 64
12:15:57.5566348	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 6,336, Length: 64
12:15:57.5566625	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 6,528, Length: 64
12:15:57.5567124	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 6,720, Length: 64
12:15:57.5567449	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 6,912, Length: 64
12:15:57.5567880	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 7,104, Length: 64
12:15:57.5568289	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 7,296, Length: 64
12:15:57.5568578	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 7,488, Length: 64
12:15:57.5568865	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 7,680, Length: 64
12:15:57.5569805	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 7,872, Length: 64
12:15:57.5570280	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 8,064, Length: 64
12:15:57.5571720	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 8,256, Length: 64
12:15:57.5573078	blackbasta.exe	3908	WriteFile	C:\Logs\sysmon.evtx	SUCCESS	Offset: 0, Length: 64, Priority: Normal
12:15:57.5573883	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 8,448, Length: 64
12:15:57.5574537	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 8,640, Length: 64
12:15:57.5574882	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 8,832, Length: 64
12:15:57.5575415	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 9,024, Length: 64
12:15:57.5576176	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 9,216, Length: 64
12:15:57.5576472	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 9,408, Length: 64
12:15:57.5577039	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 9,600, Length: 64
12:15:57.5577529	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 9,792, Length: 64
12:15:57.5577839	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 9,984, Length: 64
12:15:57.5579308	blackbasta.exe	3908	WriteFile	C:\Logs\application.evtx	SUCCESS	Offset: 10,176, Length: 64

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

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

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

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

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

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

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

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

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

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

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

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

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

Tabla 9

Alineación del comportamiento observado con el marco MITRE ATT&CK

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

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

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

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

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

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

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

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

lograr esto, se utilizó el plugin *windows.pslist*, el cual recorre la lista doblemente enlazada

EPROCESS del kernel para enumerar los procesos que se encontraban activos en el instante

exacto de la suspensión. Con el objetivo de reconocer el identificador de proceso (PID) asignado

al ransomware, se ejecutó el siguiente comando:

```
python vol.py -f "D:\Tesis\forensic_dump\MEM 1200ms.raw" windows.pslist
```

Figura 202

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

```

PS D:\Tesis\volatility3> python vol.py -f "D:\Tesis\Forensic_dump\MEM_1200ms.raw" windows.pslist
volatility 3 framework 2.21.0
Progress: 100.00
PDB scanning finished
Offset(V) Threads Handles SessionId Wow64 CreateTime ExitTime File output
PID PPID ImageFileName
4 0 System 0xe0004ac61840 112 - N/A False 2025-11-24 14:44:58.000000 UTC N/A Disabled
252 4 smss.exe 0xe0004c416040 2 - N/A False 2025-11-24 14:44:58.000000 UTC N/A Disabled
348 340 csrss.exe 0xe0004c94a6c0 9 - 0 False 2025-11-24 14:45:12.000000 UTC N/A Disabled
432 340 wininit.exe 0xe0004ad0b080 1 - 0 False 2025-11-24 14:45:13.000000 UTC N/A Disabled
440 424 csrss.exe 0xe0004c8c4500 10 - 1 False 2025-11-24 14:45:13.000000 UTC N/A Disabled
500 424 winlogon.exe 0xe0004cb71080 5 - 1 False 2025-11-24 14:45:13.000000 UTC N/A Disabled
548 432 services.exe 0xe0004cb99080 4 - 0 False 2025-11-24 14:45:13.000000 UTC N/A Disabled
568 432 lsass.exe 0xe0004cb9e080 6 - 0 False 2025-11-24 14:45:14.000000 UTC N/A Disabled
640 548 svchost.exe 0xe0004cc11500 21 - 0 False 2025-11-24 14:45:15.000000 UTC N/A Disabled
696 548 svchost.exe 0xe0004cc29840 12 - 0 False 2025-11-24 14:45:15.000000 UTC N/A Disabled
800 500 dwm.exe 0xe0004cc78080 8 - 1 False 2025-11-24 14:45:15.000000 UTC N/A Disabled
876 548 svchost.exe 0xe0004ccaa080 65 - 0 False 2025-11-24 14:45:15.000000 UTC N/A Disabled
912 548 svchost.exe 0xe0004ccc6840 20 - 0 False 2025-11-24 14:45:15.000000 UTC N/A Disabled
924 548 svchost.exe 0xe0004ccc7840 13 - 0 False 2025-11-24 14:45:15.000000 UTC N/A Disabled
1016 548 svchost.exe 0xe0004ce00840 12 - 0 False 2025-11-24 14:45:16.000000 UTC N/A Disabled
1112 548 svchost.exe 0xe0004ce1a840 26 - 0 False 2025-11-24 14:45:16.000000 UTC N/A Disabled
1076 548 svchost.exe 0xe0004ce83840 18 - 0 False 2025-11-24 14:45:16.000000 UTC N/A Disabled
1252 548 spoolsv.exe 0xe0004cf30800 9 - 0 False 2025-11-24 14:45:17.000000 UTC N/A Disabled
1320 548 svchost.exe 0xe0004cf644c0 22 - 0 False 2025-11-24 14:45:17.000000 UTC N/A Disabled
1552 548 svchost.exe 0xe0004d010840 3 - 0 False 2025-11-24 14:45:18.000000 UTC N/A Disabled
1672 548 svchost.exe 0xe0004d08b1c0 10 - 0 False 2025-11-24 14:45:18.000000 UTC N/A Disabled
1704 548 svchost.exe 0xe0004d0ab080 3 - 0 False 2025-11-24 14:45:18.000000 UTC N/A Disabled
1724 548 svchost.exe 0xe0004d0c0840 8 - 0 False 2025-11-24 14:45:18.000000 UTC N/A Disabled
1748 548 VGAuthService.exe 0xe0004d06b080 2 - 0 False 2025-11-24 14:45:18.000000 UTC N/A Disabled
1756 548 MsMpEng.exe 0xe0004d0fc5c0 6 - 0 False 2025-11-24 14:45:18.000000 UTC N/A Disabled

```

Figura 203

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

4728	548	svchost.exe	0xe0004cdc0880	5	-	0	False	2025-11-24	17:26:41.000000	UTC	N/A	Disabled		
2768	100	SearchProtocolHost.exe	0xe0004c876800	8	-	1	False	2025-11-24	17:26:41.000000	UTC	N/A	Disabled		
2956	2948	blackbasta.exe	0xe0004d90d840	11	-	1	True	2025-11-24	17:26:46.000000	UTC	N/A	Disabled		
4476	956	conhost.exe	0xe0004c4cf8000	0	-	1	False	2025-11-24	17:26:46.000000	UTC	2025-11-24	17:26:46.000000	UTC	Disabled
3456	4140	net.exe	0xe0004ca93840	0	-	1	False	2025-11-24	17:26:47.000000	UTC	2025-11-24	17:26:47.000000	UTC	Disabled
1198	104	SearchFilterHost.exe	0xe0004cd0e840	7	-	0	False	2025-11-24	17:26:48.000000	UTC	N/A	Disabled		
360	4140	net.exe	0xe0004ca9d840	0	-	1	False	2025-11-24	17:26:48.000000	UTC	2025-11-24	17:26:51.000000	UTC	Disabled
2376	4140	sysdump.exe	0xe0004cbcb840	3	-	1	False	2025-11-24	17:26:51.000000	UTC	N/A	Disabled		
4744	640	backgroundTask	0xe0004d6a5000	0	-	1	False	2025-11-24	17:26:41.000000	UTC	2025-11-24	17:26:41.000000	UTC	Disabled

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

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

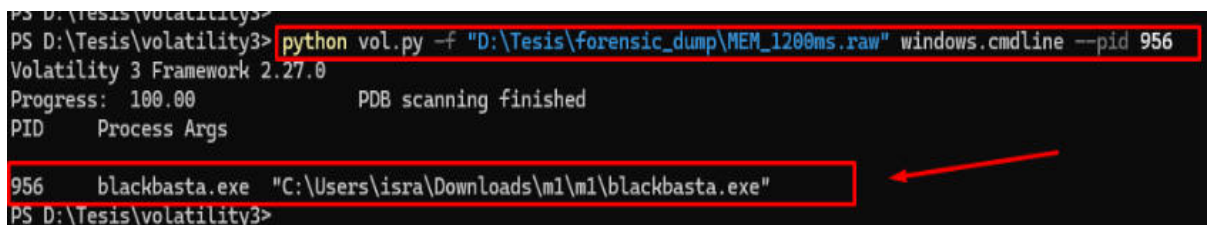
4.2.9.2. Ejecución de Comandos Para Descartar Falsos Positivos. Previo a realizar el análisis detallado, se ejecutaron plugin adicionales para corroborar que el PID 956 encontrado no se tratara de un falso positivo, por ejemplo, un proceso legítimo con inyección.

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

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

Figura 204

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



```
PS D:\Tesis\volatility3> python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw" windows.cmdline --pid 956  
Volatility 3 Framework 2.27.0  
Progress: 100.00          PDB scanning finished  
PID    Process Args  
956    blackbasta.exe "C:\Users\isra\Downloads\m1\m1\blackbasta.exe"  
PS D:\Tesis\volatility3>
```

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

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

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

Figura 205

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

```
PS D:\Tesis\volatility3> python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw" windows.netscan
```

Volatility 3 Framework 2.27.0
Progress: 100.00 PDB scanning finished

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

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

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

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

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

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

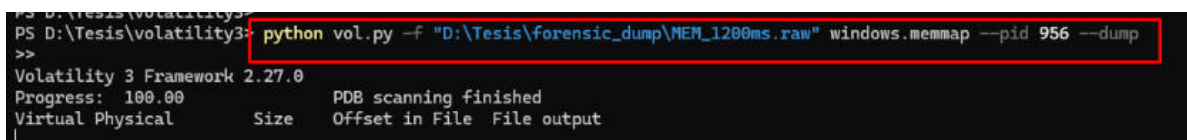
4.2.9.4. Extracción Forense del Espacio de Memoria. Dado que el análisis preliminar de inyecciones de código mediante el plugin *windows.malfind* resultó negativo, indicando que Black Basta opera en esta fase como un ejecutable tradicional y no mediante inyección dispersa, se determinó que la evidencia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

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

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

Figura 206

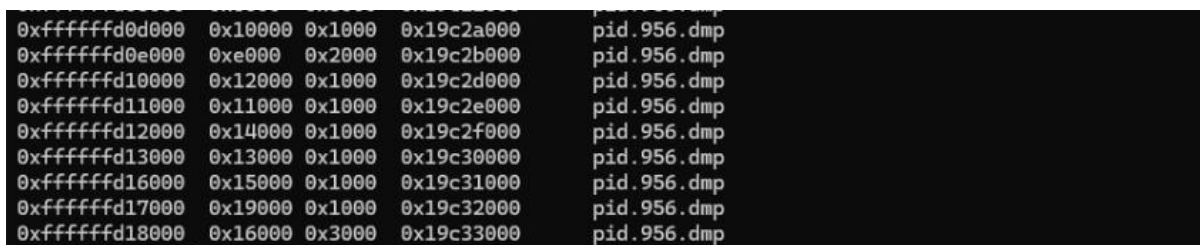
Extracción del espacio de memoria mediante el plugin Windows.memmap



```
PS D:\Tesis\volatility3> python vol.py -f "D:\Tesis\forensic_dump\MEM_1200ms.raw" windows.memmap --pid 956 --dump
>>
Volatility 3 Framework 2.27.0
Progress: 100.00 PDB scanning finished
Virtual Physical      Size  Offset in File  File output
```

Figura 207

Generación del archivo pid.956.dmp



```
0xffffffffd0d000 0x10000 0x1000 0x19c2a000 pid.956.dmp
0xffffffffd0e000 0xe000 0x2000 0x19c2b000 pid.956.dmp
0xffffffffd10000 0x12000 0x1000 0x19c2d000 pid.956.dmp
0xffffffffd11000 0x11000 0x1000 0x19c2e000 pid.956.dmp
0xffffffffd12000 0x14000 0x1000 0x19c2f000 pid.956.dmp
0xffffffffd13000 0x13000 0x1000 0x19c30000 pid.956.dmp
0xffffffffd16000 0x15000 0x1000 0x19c31000 pid.956.dmp
0xffffffffd17000 0x19000 0x1000 0x19c32000 pid.956.dmp
0xffffffffd18000 0x16000 0x3000 0x19c33000 pid.956.dmp
```

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

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

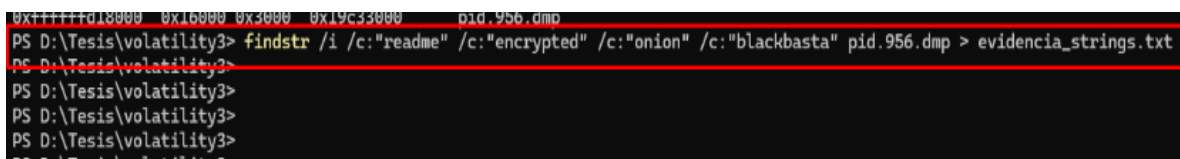
- **Análisis Estático de Contenido.-** Para interpretar la información contenida en el volcado de memoria cruda (pid.956.dmp), se empleó una metodología de “Strings Scavenging”. Esta técnica permite identificar y extraer cadenas legibles (ASCII y Unicode) incrustadas entre las estructuras binarias del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

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

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

Figura 208

Procesamiento del archivo de volcado mediante el uso del comando findstr



```
0x00000000 0x00000000 0x00000000 0x00000000 pid.956.dmp
PS D:\Tesis\volatility3> findstr /i /c:"readme" /c:"encrypted" /c:"onion" /c:"blackbasta" pid.956.dmp > evidencia_strings.txt
PS D:\Tesis\volatility3>
PS D:\Tesis\volatility3>
PS D:\Tesis\volatility3>
PS D:\Tesis\volatility3>
```

Como primer hallazgo crítico, de la cadena recuperada (Unicode), se visualiza la ruta completa del ejecutable en su espacio de memoria, confirmando su origen exacto:

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

Figura 209

Hallazgo de la ruta del ejecutable dentro del archivo analizado

```

1  C:\Users\isra\Downloads\m1\m1\blackbasta.exe
2  <SHIMENGSTATE PID="956" FILENAME="C:\Users\isra\Downloads\m1\m1\blackbasta.exe" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance http://www.w3.org/2001/XMLSchema-instance"
4  >
5  <data>
6  <path>C:\Users\isra\Downloads\m1\m1\blackbasta.exe
7  </path>
8  <url>https://aazsbagya565vlu2c6bzy6yfielkcbtvvctvolt33a77xyp17nypxyd.onion:80/
9  </url>
10 <message>Your data are stolen and encrypted
11 </message>
12 </data>
13 </SHIMENGSTATE>
14 </xml>
15
16 https://aazsbagya565vlu2c6bzy6yfielkcbtvvctvolt33a77xyp17nypxyd.onion:80/

```

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

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

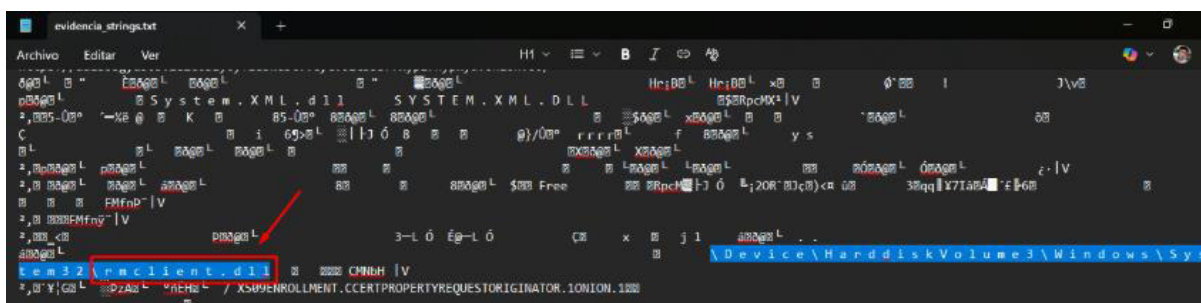
Finalmente, se logra correlación entre la evidencia física y volátil, es decir, entre el archivo hash en disco y el proceso en RAM, cerrando la cadena de custodia técnica del incidente.

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

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

Figura 210

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



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

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

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

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

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

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

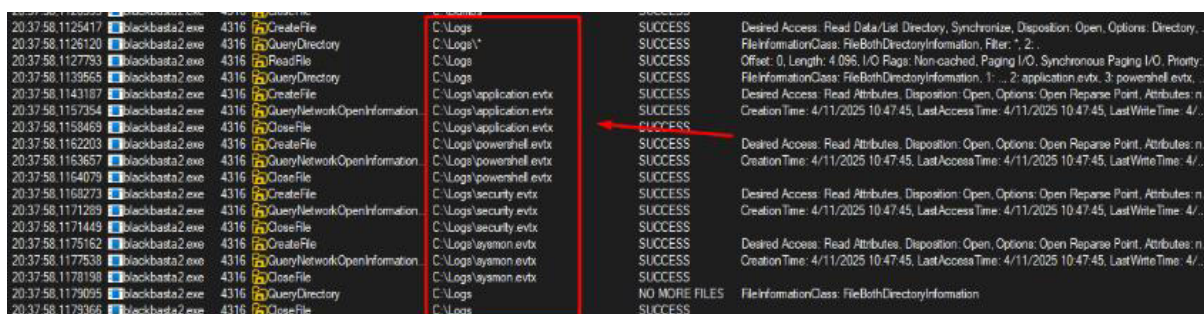
4.2.10.1 Análisis de archivo PML. Para validar la persistencia de las tácticas obtenidas, se analiza el registro de actividad procmon_20251127_203753.pml.

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

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

Figura 211

Identificación de accesos a directorio C:\Logs



Time	Process	Operation	Path	Result	Details
20:37:58.1125417	blackbasta2.exe	CreateFile	C:\Logs	SUCCESS	Desired Access: Read Data/List Directory, Synchronize, Disposition: Open, Options: Directory, ...
20:37:58.1126120	blackbasta2.exe	QueryDirectory	C:\Logs*	SUCCESS	FileInformationClass: FileBothDirectoryInformation, Filter: *, 2, ...
20:37:58.1127793	blackbasta2.exe	ReadFile	C:\Logs	SUCCESS	Offset: 0, Length: 4,096, I/O Flags: Non-cached, Paging I/O, Synchronous Paging I/O, Priority...
20:37:58.1139565	blackbasta2.exe	QueryDirectory	C:\Logs	SUCCESS	FileInformationClass: FileBothDirectoryInformation, 1, ... 2: application evtx, 3: powershell evtx, ...
20:37:58.1143187	blackbasta2.exe	CreateFile	C:\Logs\application evtx	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n...
20:37:58.1157354	blackbasta2.exe	QueryNetworkOpenInformation	C:\Logs\application evtx	SUCCESS	Creation Time: 4/11/2025 10:47:45, LastAccessTime: 4/11/2025 10:47:45, LastWriteTime: 4/...
20:37:58.1158469	blackbasta2.exe	CreateFile	C:\Logs\powercat.exe	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n...
20:37:58.1163657	blackbasta2.exe	QueryNetworkOpenInformation	C:\Logs\powercat.exe	SUCCESS	Creation Time: 4/11/2025 10:47:45, LastAccessTime: 4/11/2025 10:47:45, LastWriteTime: 4/...
20:37:58.1164079	blackbasta2.exe	CreateFile	C:\Logs\powercat.exe	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n...
20:37:58.1168273	blackbasta2.exe	QueryNetworkOpenInformation	C:\Logs\powercat.exe	SUCCESS	Creation Time: 4/11/2025 10:47:45, LastAccessTime: 4/11/2025 10:47:45, LastWriteTime: 4/...
20:37:58.1171269	blackbasta2.exe	CreateFile	C:\Logs\security evtx	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n...
20:37:58.1171449	blackbasta2.exe	QueryNetworkOpenInformation	C:\Logs\security evtx	SUCCESS	Creation Time: 4/11/2025 10:47:45, LastAccessTime: 4/11/2025 10:47:45, LastWriteTime: 4/...
20:37:58.1175162	blackbasta2.exe	CreateFile	C:\Logs\ayamon evtx	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n...
20:37:58.1177638	blackbasta2.exe	QueryNetworkOpenInformation	C:\Logs\ayamon evtx	SUCCESS	Creation Time: 4/11/2025 10:47:45, LastAccessTime: 4/11/2025 10:47:45, LastWriteTime: 4/...
20:37:58.1178198	blackbasta2.exe	CreateFile	C:\Logs\ayamon evtx	SUCCESS	Creation Time: 4/11/2025 10:47:45, LastAccessTime: 4/11/2025 10:47:45, LastWriteTime: 4/...
20:37:58.1179095	blackbasta2.exe	QueryDirectory	C:\Logs	NO MORE FILES	FileInformationClass: FileBothDirectoryInformation
20:37:58.1179366	blackbasta2.exe	CloseFile	C:\Logs	SUCCESS	

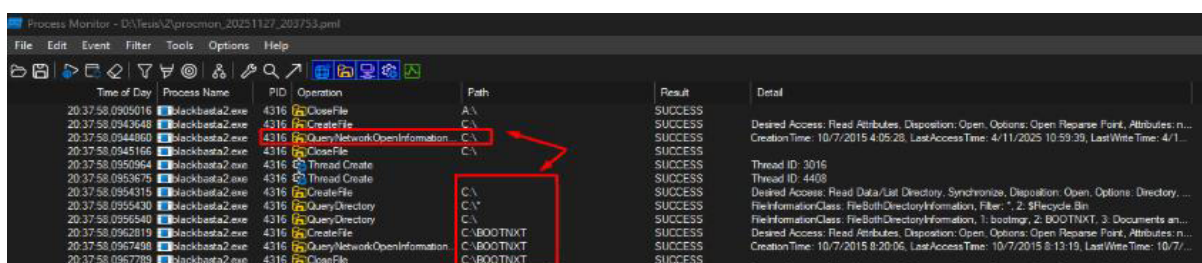
Unos milisegundos tras su ejecución, a las 20:37:58, se observa el acceso al directorio C:\Logs y la interacción con los archivos de registro del sistema: C:\Logs\application.evtx; C:\Logs\powershell.evtx; C:\Logs\security.evtx.

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

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

Figura 212

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



Time of Day	Process Name	PID	Operation	Path	Result	Detail
20:37:58.0905016	blackbasta2.exe	4316	CloseFile	A:\	SUCCESS	
20:37:58.0943648	blackbasta2.exe	4316	CreateFile	C:\	SUCCESS	
20:37:58.0944860	blackbasta2.exe	4316	QueryNetworkOpenInformation	C:\	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n...
20:37:58.0945166	blackbasta2.exe	4316	CloseFile	C:\	SUCCESS	CreationTime: 10/7/2015 4:05:28, LastAccessTime: 4/11/2025 10:59:39, LastWriteTime: 4/1...
20:37:58.0950964	blackbasta2.exe	4316	Thread Create		SUCCESS	Thread ID: 3016
20:37:58.0953675	blackbasta2.exe	4316	Thread Create		SUCCESS	Thread ID: 4408
20:37:58.0954315	blackbasta2.exe	4316	CreateFile	C:\	SUCCESS	Desired Access: Read Data/List Directory, Synchronize, Disposition: Open, Options: Directory...
20:37:58.0955430	blackbasta2.exe	4316	QueryDirectory	C:\	SUCCESS	FileInformationClass: FileBothDirectoryInformation, Filter: *, 2, \$Recycle Bin
20:37:58.0956540	blackbasta2.exe	4316	QueryDirectory	C:\	SUCCESS	FileInformationClass: FileBothDirectoryInformation, 1, bootmgr, 2, BOOTNXT, 3, Documents an...
20:37:58.0962819	blackbasta2.exe	4316	CreateFile	C:\BOOTNXT	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n...
20:37:58.0967498	blackbasta2.exe	4316	QueryNetworkOpenInformation	C:\BOOTNXT	SUCCESS	CreationTime: 10/7/2015 8:20:06, LastAccessTime: 10/7/2015 8:13:19, LastWriteTime: 10/7/...
20:37:58.0967789	blackbasta2.exe	4316	CloseFile	C:\BOOTNXT	SUCCESS	

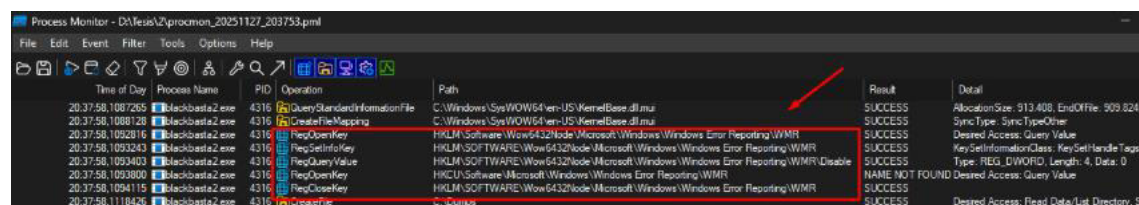
El proceso blackbasta2.exe PID: 4316, interactúa con la clave de registro HKLM\Software\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR.

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

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

Figura 213

Identificación de interacciones con Windows Error Reporting

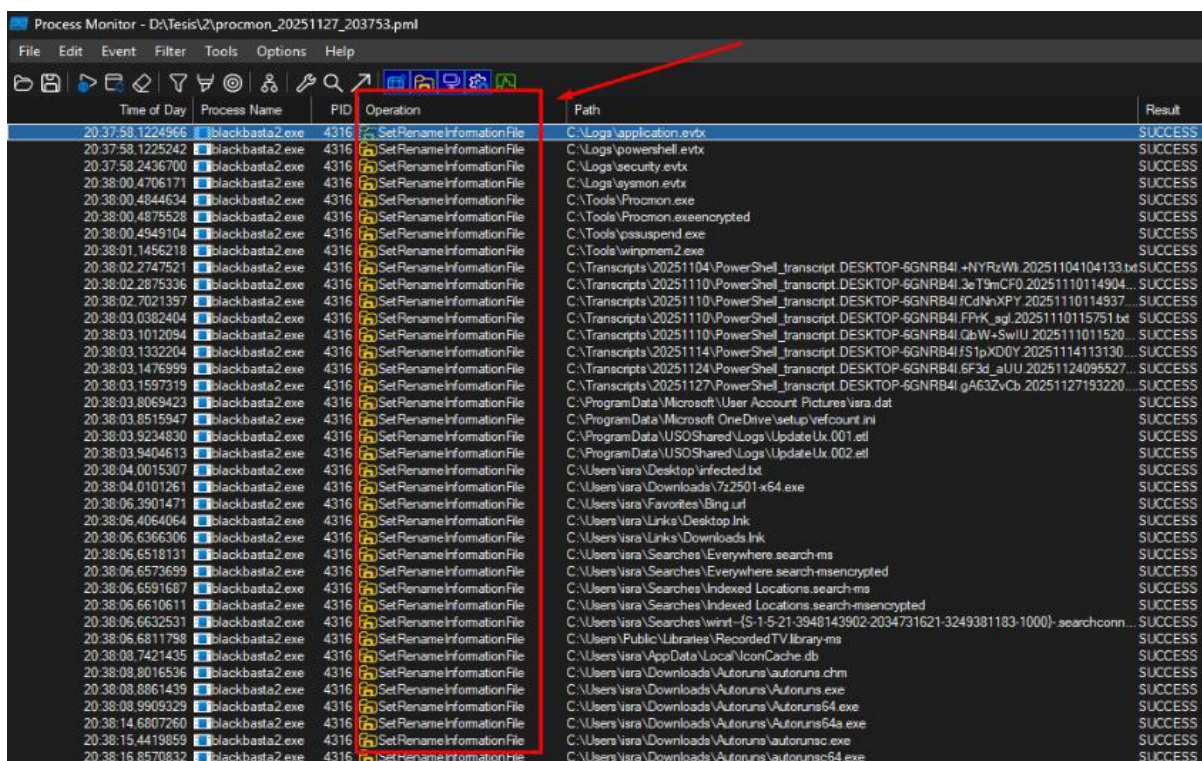


Time of Day	Process Name	PID	Operation	Path	Result	Detail
20:37:58.1087255	blackbasta2.exe	4316	RegOpenKey	C:\Windows\System32\wininit\KernelBase.dll.mui	SUCCESS	AllocationSize: 913,408, EndOfFile: 909,824
20:37:58.1088128	blackbasta2.exe	4316	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR	SUCCESS	Sync Type: Sync TypeOther
20:37:58.1092816	blackbasta2.exe	4316	RegSetInfoKey	HKLM\Software\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR	SUCCESS	Desired Access: Query Value
20:37:58.1093243	blackbasta2.exe	4316	RegQueryValue	HKLM\Software\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR\Disable	SUCCESS	KeySetInformationClass: KeySetHandleTaget
20:37:58.1093403	blackbasta2.exe	4316	RegOpenKey	HKLM\Software\Microsoft\Windows\Windows Error Reporting\WMR	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
20:37:58.1093800	blackbasta2.exe	4316	RegCloseKey	HKLM\Software\Microsoft\Windows\Windows Error Reporting\WMR	SUCCESS	NAME NOT FOUND Desired Access: Query Value
20:37:58.1094115	blackbasta2.exe	4316	RegCloseKey	HKLM\Software\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR	SUCCESS	Desired Access: Read Data/List Directory, S
20:37:58.1110426	blackbasta2.exe	4316	Process Exit	C:\Program	SUCCESS	

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

Figura 214

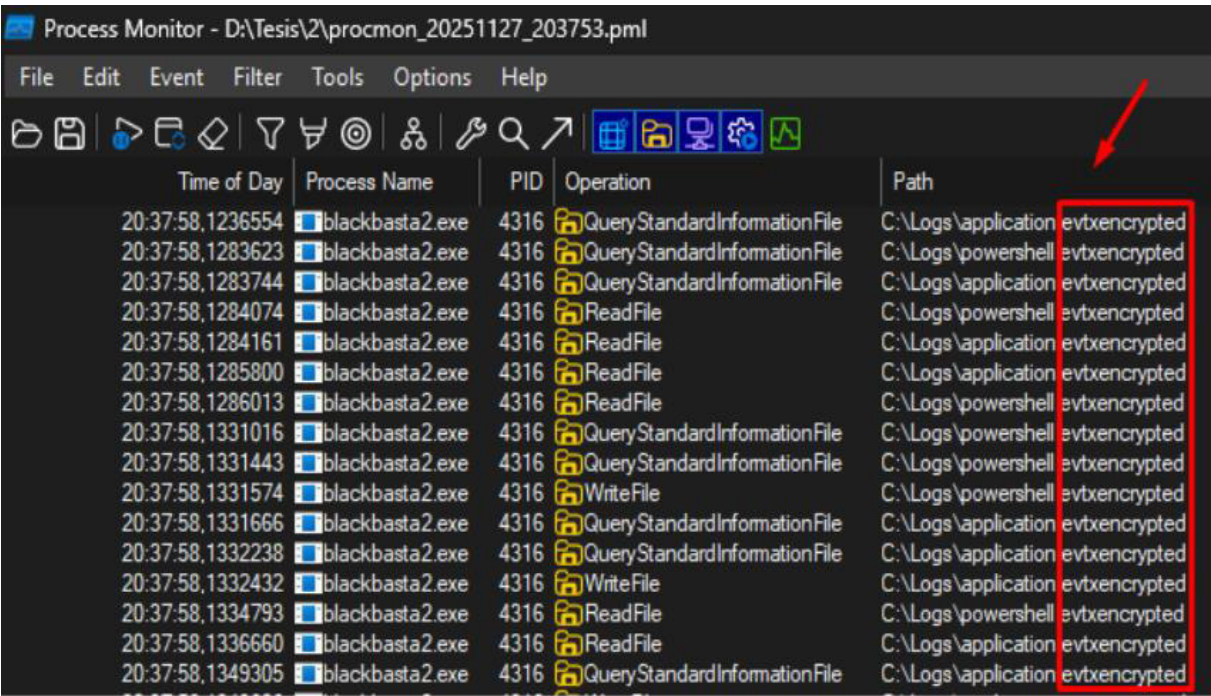
Identificación de operaciones SetRenameInformationFile



Time of Day	Process Name	PID	Operation	Path	Result
20:37:58.1224966	blackbasta2.exe	4316	SetRenameInformationFile	C:\Logs\application.evtx	SUCCESS
20:37:58.1225242	blackbasta2.exe	4316	SetRenameInformationFile	C:\Logs\powershell.evtx	SUCCESS
20:37:58.2436700	blackbasta2.exe	4316	SetRenameInformationFile	C:\Logs\security.evtx	SUCCESS
20:38:00.4706171	blackbasta2.exe	4316	SetRenameInformationFile	C:\Logs\sysmon.evtx	SUCCESS
20:38:00.4844634	blackbasta2.exe	4316	SetRenameInformationFile	C:\Tools\Procmon.exe	SUCCESS
20:38:00.4875528	blackbasta2.exe	4316	SetRenameInformationFile	C:\Tools\Procmon.exe.encrypted	SUCCESS
20:38:00.4949104	blackbasta2.exe	4316	SetRenameInformationFile	C:\Tools\Procmon.exe.encrypted	SUCCESS
20:38:01.1456218	blackbasta2.exe	4316	SetRenameInformationFile	C:\Tools\winpmem2.exe	SUCCESS
20:38:02.2747521	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\202511104_PowerShell_transcript_DESKTOP-6G9NRB4I_+NYRzWl_20251104104133.txt	SUCCESS
20:38:02.2875336	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\202511110_PowerShell_transcript_DESKTOP-6G9NRB4I_3eT9mCF0_20251101114904...	SUCCESS
20:38:02.3082404	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\202511110_PowerShell_transcript_DESKTOP-6G9NRB4I_FcDnXPY_20251101114937...	SUCCESS
20:38:03.0382404	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\202511110_PowerShell_transcript_DESKTOP-6G9NRB4I_FcDnXPY_20251101114937...	SUCCESS
20:38:03.1012094	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\202511110_PowerShell_transcript_DESKTOP-6G9NRB4I_6bWwSwIU_20251101115751.txt	SUCCESS
20:38:03.1332204	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\202511114_PowerShell_transcript_DESKTOP-6G9NRB4I_5pXDOY_20251114113130...	SUCCESS
20:38:03.1476999	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\202511124_PowerShell_transcript_DESKTOP-6G9NRB4I_6F3d_aUU_20251124095527...	SUCCESS
20:38:03.1597319	blackbasta2.exe	4316	SetRenameInformationFile	C:\Transcripts\20251127_PowerShell_transcript_DESKTOP-6G9NRB4I_gA63ZvCb_20251127193220...	SUCCESS
20:38:03.8069423	blackbasta2.exe	4316	SetRenameInformationFile	C:\ProgramData\Microsoft\User Account Pictures\vara.dat	SUCCESS
20:38:03.8515947	blackbasta2.exe	4316	SetRenameInformationFile	C:\ProgramData\Microsoft\OneDrive\setup\velcount.ini	SUCCESS
20:38:03.9234830	blackbasta2.exe	4316	SetRenameInformationFile	C:\ProgramData\USOShared\Logs\UpdateUx_001.etl	SUCCESS
20:38:03.9404613	blackbasta2.exe	4316	SetRenameInformationFile	C:\ProgramData\USOShared\Logs\UpdateUx_002.etl	SUCCESS
20:38:04.0015307	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Desktop\infected.txt	SUCCESS
20:38:04.0101261	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Downloads\7z2501-x64.exe	SUCCESS
20:38:06.3901471	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Favorites\Bing.url	SUCCESS
20:38:06.4064064	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Links\Desktop.lnk	SUCCESS
20:38:06.6366306	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Links\Downloads.lnk	SUCCESS
20:38:06.6518131	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Searches\Everywhere_search.ms	SUCCESS
20:38:06.6573699	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Searches\Everywhere_search-ms.encrypted	SUCCESS
20:38:06.6591687	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Searches\Indexed Locations_search.ms	SUCCESS
20:38:06.6610611	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Searches\Indexed Locations_search-ms.encrypted	SUCCESS
20:38:06.6632531	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Searches\winrt-(S-1-5-21-3948143902-2034731621-3249381183-1000)-searchconn...	SUCCESS
20:38:06.6811798	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\Public\Libraries\RecordedTV library.ms	SUCCESS
20:38:08.7421435	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\AppData\Local\IconCache.db	SUCCESS
20:38:08.8016536	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Downloads\Autoruns\autoruns.chm	SUCCESS
20:38:08.8861439	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Downloads\Autoruns\Autoruns.exe	SUCCESS
20:38:08.9909329	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Downloads\Autoruns\Autoruns64.exe	SUCCESS
20:38:14.6807260	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Downloads\Autoruns\Autoruns64a.exe	SUCCESS
20:38:15.4419859	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Downloads\Autoruns\Autoruns64a.exe	SUCCESS
20:38:16.8570832	blackbasta2.exe	4316	SetRenameInformationFile	C:\Users\vara\Downloads\Autoruns\Autoruns64a.exe	SUCCESS

Figura 215

Identificación de operaciones QueryStandardInformationFile



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

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

Tabla 10

Matriz de Diferencias y Similitudes entre Variantes de Black Basta

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

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

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

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

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

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

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

Figura 216

Identificación de procesos activos a través del plugin windows.pslist

```

336 conhost.exe \??\C:\Windows\system32\conhost.exe 0x4
1684 OneDrive.Sync. "C:\Users\isra\AppData\Local\Microsoft\OneDrive\25.194.1005.0003\OneDrive.Sync.Service.
4540 taskeng.exe taskeng.exe {B831586F-656A-47EA-A9D0-638B2CC1683D}
604 notepad.exe "C:\Windows\System32\notepad.exe" "C:\Users\isra\Downloads\m1\wimpmem.ps1"
1520 SearchProtocol "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe12_Global\
7483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot
ata\Temp\usgthrsvc" "DownLevelDaemon"
1377 SearchFilterHo "C:\Windows\system32\SearchFilterHost.exe" 8 616 620 628 8192 624
2652 blackb "C:\Users\isra\Downloads\m1\m2\blackbasta2.exe"
852 conhost.exe \??\C:\Windows\system32\conhost.exe 0x4
3952 net.exe -
4936 dllhost.exe -
1580 net.exe -
  
```

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

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

Figura 217

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

```

PS D:\Tesis\volatility3> python vol.py -f "D:\Tesis\2\NEM_1200ms_205208.raw" windows.dllexport --pid 2652
Volatility 3 Framework 2.27.0
Progress: 100.00
  
```

PID	Process Base	Size	Name	Path	LoadCount	LoadTime	File output
2652	blackb	0xba0000	blackbasta2.exe	C:\Users\isra\Downloads\m1\m2\blackbasta2.exe	-1	2025-11-28 01:52:04.000000 UTC	Disabled
2652	blackb	0x7ffc2dcd0000	ntdll.dll	C:\Windows\SYSTEM32\ntdll.dll	-1	2025-11-28 01:52:04.000000 UTC	Disabled
2652	blackb	0x5c760000	wow64.dll	C:\Windows\system32\wow64.dll	-1	2025-11-28 01:52:04.000000 UTC	Disabled
2652	blackb	0x5c7c0000	wow64win.dll	C:\Windows\system32\wow64win.dll	6	2025-11-28 01:52:04.000000 UTC	Disabled
2652	blackb	0x5c7b0000	wow64cpu.dll	C:\Windows\system32\wow64cpu.dll	6	2025-11-28 01:52:04.000000 UTC	Disabled
2652	blackb	0xba0000	blackbasta2.exe	C:\Users\isra\Downloads\m1\m2\blackbasta2.exe	-	-	Disabled
2652	blackb	0x77800000	ntdll.dll	C:\Windows\SYSTEM32\ntdll.dll	-	-	Disabled
2652	blackb	0x76d10000	KERNEL32.DLL	C:\Windows\SYSTEM32\KERNEL32.DLL	-	-	Disabled
2652	blackb	0x74cc0000	KERNELBASE.dll	C:\Windows\SYSTEM32\KERNELBASE.dll	-	-	Disabled
2652	blackb	0x74040000	apphelp.dll	C:\Windows\system32\apphelp.dll	-	N/A	Disabled
2652	blackb	0x74930000	USER32.dll	C:\Windows\SYSTEM32\USER32.dll	-	-	Disabled
2652	blackb	0x755a0000	GDI32.dll	C:\Windows\SYSTEM32\GDI32.dll	-	-	Disabled
2652	blackb	0x756f0000	SHELL32.dll	C:\Windows\SYSTEM32\SHELL32.dll	-	-	Disabled
2652	blackb	0x77120000	msvcrt.dll	C:\Windows\SYSTEM32\msvcrt.dll	-	-	Disabled
2652	blackb	0x750c0000	windows.storage.dll	C:\Windows\SYSTEM32\windows.storage.dll	-	-	Disabled
2652	blackb	0x74a70000	combase.dll	C:\Windows\SYSTEM32\combase.dll	-	-	Disabled
2652	blackb	0x76ab0000	RPCRT4.dll	C:\Windows\SYSTEM32\RPCRT4.dll	-	-	Disabled
2652	blackb	0x74910000	SspiCli.dll	C:\Windows\SYSTEM32\SspiCli.dll	-	-	Disabled
2652	blackb	0x74900000	CRYPTBASE.dll	C:\Windows\SYSTEM32\CRYPTBASE.dll	-	-	Disabled
2652	blackb	0x748a0000	bcryptPrimitives.dll	C:\Windows\SYSTEM32\bcryptPrimitives.dll	-	-	Disabled
2652	blackb	0x77230000	sechost.dll	C:\Windows\SYSTEM32\sechost.dll	-	-	Disabled
2652	blackb	0x76ed0000	advapi32.dll	C:\Windows\SYSTEM32\advapi32.dll	-	N/A	Disabled
2652	blackb	0x771e0000	shlwapi.dll	C:\Windows\SYSTEM32\shlwapi.dll	-	-	Disabled
2652	blackb	0x777f0000	kernel.appcore.dll	C:\Windows\SYSTEM32\kernel.appcore.dll	-	-	Disabled
2652	blackb	0x74c30000	shcore.dll	C:\Windows\SYSTEM32\shcore.dll	-	-	Disabled
2652	blackb	0x77400000	powrprof.dll	C:\Windows\SYSTEM32\powrprof.dll	-	-	Disabled
2652	blackb	0x74f40000	profapi.dll	C:\Windows\SYSTEM32\profapi.dll	-	-	Disabled
2652	blackb	0x76bf0000	IMM32.DLL	C:\Windows\SYSTEM32\IMM32.DLL	-	-	Disabled

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

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

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

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

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

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

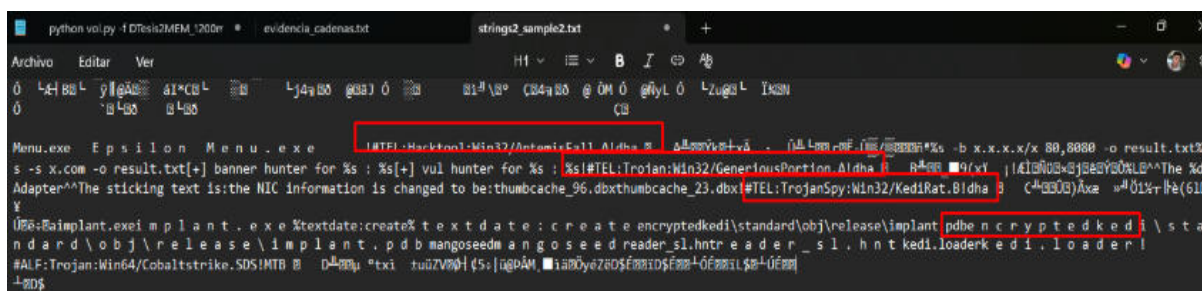
El volcado de memoria analizado incluye cadenas de texto que coinciden con firmas de amenazas de Microsoft; lo cual sugiere que el proceso de *Windows Defender "MsMpEng.exe"* escaneó estos componentes en memoria, aunque también existe la posibilidad de que el propio malware contiene estas cadenas para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

Algunos identificadores encontrados son:

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

Figura 218

Identificadores encontrados en strings



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

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

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

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

4.3. Perfil Mínimo de Black Basta

4.3.1. Información General de Black Basta.

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

4.3.2. Comportamiento Observado.

4.3.2.1. Procesos. Se destacan los siguientes:

- **Proceso principal:** blackbasta.exe PID 956
- **Procesos Secundarios funcionando en simultaneo:** taskhostw.exe,

SecurityHealthHost.exe, dmclient.exe, Microsoft Edge y Windows Terminal (No generados necesariamente por el ransomware).

- **Directorios comprometidos:** C:\Users, C:\ProgramData, C:\Temp, C:\Users\Default

- **Total de eventos generados:** 22.411.855
- **Eventos de File System Activity:** 21.968.037
- **Eventos de Network Activity, Process and Threats activity y Registry**

Activity: 443.818.

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

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

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

4.3.2.3. Comportamiento del Cifrado. Se destacan los siguientes:

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

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

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

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

4.3.4. Actividades Registradas. Se utiliza la herramienta Procmon.

Lectura de claves:

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

Escritura directa:

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

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

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

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

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

- QueryDirectory
- CreateFile
- ReadFile
- CloseFile

Llamados a directorios de Usuario:

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

Llamados a directorios del sistema:

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

Llamados a directorios críticos del sistema de archivos:

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

Accesos a:

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

- Archivos .db, .ini, .log

4.3.5. Artefactos Forenses.

4.3.5.1. Lista de Artefactos Analizados.

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

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

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

Herramienta utilizada: Sysmon

Artefacto: Microsoft-Windows-Sysmon-Operational

Hash: 0e29c1def196bfd88db689ec2bca2b2402602949d8167ff290d9c5320bf6833a

Eventos Generados:

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

Ejecución de comandos:

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

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

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

Ejecución de comandos:

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

C:\Users\Default\readme.txt

Event ID 5: Terminación de procesos.

Event ID 13: Claves e inventario de aplicaciones.

- **Actividad en archivos PML.**

Herramienta utilizada: Procmon

Artefacto: procmon_20251114_121545.pml

Hash: 2cf136ca7e4ec8fa98ab9ee729de470662fa7b4690ec45368470d4763911636d

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

Peticiones de eliminar los Volume Shadow Copies:

C:\Windows\System32\vssadmin.exe

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

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

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

Tiempo de Inicio: 12:15:57.515

Tiempo de finalización: 12:20:27.933

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

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

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

- **Actividad en memoria.**

Herramienta utilizada: winpmem

Herramienta de análisis: Volatility 3

Artefacto: winpmem_20251114_123536.raw

Hash: 296cf640111a015b6ae57a3f5d60e67c9ec8a1e2390ad7c2086e36c3e7d4cf79

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- **Regla propuesta (Formato SIGMA)(YAML):**

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

```

logsource:

  product: windows

  category: file_event

detection:

  selection:

    TargetFilename|endswith: '.evtx'

  filter:

    Image|endswith: '\\svchost.exe'

  condition: selection and not filter

level: critical

tags:

  - attack.defense_evasion

  - attack.t1070.001

```

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

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

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

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

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

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

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

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

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

- **Regla YARA propuesta:**

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

```

// Artefactos de Evasión y Bloqueo

$dll_ref = "rmclient.dll" ascii wide nocase

$cmd_shadow = "delete shadows /all /quiet" ascii wide


// Indicadores de Cifrado (Variables pero predictivos)

$ext1 = ".basta" wide

$ext2 = "encrypted" wide


condition:

// Detección de nota + (técnica de evasión O extensión
conocida)

$note_text and ($onion_url or $dll_ref or $cmd_shadow or any of
($ext*))
}

```

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

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

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

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

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

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

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

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

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

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

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

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

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

4.4 Falsos Positivos

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

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

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

Capítulo 5

5. Conclusiones y Recomendaciones

5.1. Conclusiones de Análisis Estático

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

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

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

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

5.2. Conclusiones de Análisis Dinámico

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

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

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

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

5.3. *Análisis de Artefactos*

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

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

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

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

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

5.4. Perfil de Detección Mínimo Black Basta

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

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

5.5. Recomendaciones Generales

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

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

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

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

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

Referencias

- AUTOPSY*. (2025). Obtenido de AUTOPSY: <https://www.autopsy.com/>
- Brandt A, M. P. (2020). *Maze Attackers Adopt Ragnar Locker Virtual Machine Technique*.
- Brinkmann, M. (6 de Febrero de 2016). *g hacks.net*. Obtenido de <https://www.ghacks.net/2016/02/06/detect-it-easy/>
- Brumm, C. (4 de Agosto de 2021). *My learnings on Microsoft Defender for Endpoint and Exclusions*. Obtenido de Medium: <https://medium.com/codex/my-learnings-on-microsoft-defender-for-endpoint-and-exclusions-ddacf2fdd047>
- CISA, C. a. (08 de Noviembre de 2024). *#StopRansomware: Black Basta*. Obtenido de <https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-131a>
- Command and Scripting Interpreter: Windows Command Shell (T1059.003)*. MITRE ATT&CK. (2024). Obtenido de MITRE Corporation: <https://attack.mitre.org/techniques/T1059/003/>
- Crawford, D. (28 de Diciembre de 2023). *What is ChaCha20?* Obtenido de <https://protonvpn.com/blog/chacha20?srsId=AfmBOooB-pz3LxQ-SiMpY1bXkwV2Er7GktCNq2WEg7uRxVzWxurSElez>
- Cybersecurity & Infrastructure Security Agency (CISA), F. B.-S.-I. (2024). *#StopRansomware: Black Basta*. Obtenido de <https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-131a>
- Finkel. (2007). *Advanced Windows Debugging*. Addison-Wesley.
- Fox, N. (6 de Abril de 2023). *Varonis*. Obtenido de <https://www.varonis.com/blog/how-to-use-ghidra>
- Franco, D. (30 de Noviembre de 2022). *Digital Forensic Analysis of Ransomwares for Identification and Binary Extraction of Cryptographic Keys*. Obtenido de

- <https://www.heraldopenaccess.us/openaccess/digital-forensic-analysis-of-ransomwares-for-identification-and-binary-extraction-of-cryptographic-keys>
g0tmi1k. (18 de Junio de 2025). *KALI*. Obtenido de KALI:
<https://www.kali.org/docs/introduction/what-is-kali-linux/>
- Gobierno de Canarias*. (s.f.). Obtenido de Gobierno de Canarias:
<https://www3.gobiernodecanarias.org/medusa/ecoescuela/seguridad/ciudadania-y-seguridad-tic/principios-legales/software-libre/ubuntu-linux/>
- Hendrikx, M. (20 de Octubre de 2014). *How To Geek*. Obtenido de How To Geek:
<https://www.howtogeek.com/198679/how-to-use-regshot-to-monitor-your-registry/>
- Ibáñez, J. T. (2025). *Iníciate en Análisis Forense Digital*.
- INCIBE*. (Junio de 2025). Obtenido de INCIBE:
<https://www.incibe.es/ciudadania/herramientas/wireshark>
- INCIBE. (20 de Marzo de 2025). *Claves forenses en Windows: artefactos esenciales para la investigación digital*. Obtenido de <https://www.incibe.es/incibe-cert/blog/claves-forenses-en-windows-artefactos-esenciales-para-la-investigacion-digital>
- Jamil Ispahany, M. R. (2024). iCNN-LSTM: A batch-based incremental ransomware detection system using Sysmon. *Cryptography and Security (cs.CR)*.
- Jim Hodsworth, M. K. (5 de septiembre de 2024). *¿Qué es el ransomware como servicio (RaaS)?* Obtenido de <https://www.ibm.com/mx-es/think/topics/ransomware-as-a-service>
- KaliTools*. (30 de Agosto de 2021). Obtenido de <https://en.kali.tools/?p=1644>
- Kosinski, M. (4 de junio de 2024). *¿Qué es el ransomware?* Obtenido de <https://www.ibm.com/es-es/think/topics/ransomware>

Learn Microsoft. (07 de Julio de 2025). Obtenido de Learn Microsoft:

<https://learn.microsoft.com/es-es/powershell/scripting/overview?view=powershell-7.5>

Paliwal, A. (6 de Mayo de 2025). *What Is Black Basta Ransomware and How to Mitigate Attack*.

Obtenido de <https://blog.qualys.com/vulnerabilities-threat-research/2024/09/19/black-basta-ransomware-what-you-need-to-know>

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

Russinovich, M. (6 de Febrero de 2024). *Learn Microsoft*. Obtenido de Learn Microsoft:

<https://learn.microsoft.com/es-es/sysinternals/downloads/autoruns>

Russinovich, M. (20 de junio de 2024). *Learn Microsoft*. Obtenido de Learn Microsoft:

<https://learn.microsoft.com/es-es/sysinternals/downloads/procmon>

Russinovich, M., & Garnier, T. (23 de Julio de 2024). *Learn Microsoft*. Obtenido de Learn

Microsoft: <https://learn.microsoft.com/es-es/sysinternals/downloads/sysmon>

Shklar, L., & Rich, R. (s.f.). *Web Application Architecture: Principles, Protocols and Practices*.

softonic. (s.f.). Obtenido de softonic: <https://virtualbox.softonic.com/>

SOTHIS. (2025). Obtenido de SOTHIS: <https://www.sothis.tech/adquisicion-de-memoria-ram-2/>

Velociraptor. (2025). Obtenido de Velociraptor: <https://docs.velociraptor.app/docs/overview/>

VOLATILITY. (s.f.). Obtenido de VOLATILITY: <https://volatilityfoundation.org/>

YARA. (2022). Obtenido de YARA: <https://yara.readthedocs.io/en/latest/>

Apéndices

Información del laboratorio OPBASTA-LAB:

https://drive.google.com/drive/folders/16hvzXp-WBmYF14Kf-ic4rI0WjCPWcZ_r?usp=sharing