

# PEC2

---

ANÁLISIS AVANZADO DE MALWARE

Montesinos Guzmán Wilma Alejandra

## Tabla de contenido

1. Introducción.....	1
2. Análisis estático .....	1
2.1. Inventariado.....	1
2.2. File .....	3
2.3. Du .....	4
2.4. Clamscan .....	4
2.5. VirusTotal.....	4
2.6. Binwalk.....	5
2.7. Rabin2.....	6
2.7. Exiftool .....	10
2.8. Entroper .....	11
2.9. Floss .....	12
2.9. Xxd .....	13
2.10. Yara .....	13
2.11. PEiD. ....	14
2.12. CFF explorer.....	15
2.13. PE detective .....	22
3. Análisis dinámico .....	22
3.1. Process monitor .....	22
3.2. Process explorer.....	28
4. Conclusiones .....	29

# 1. Introducción

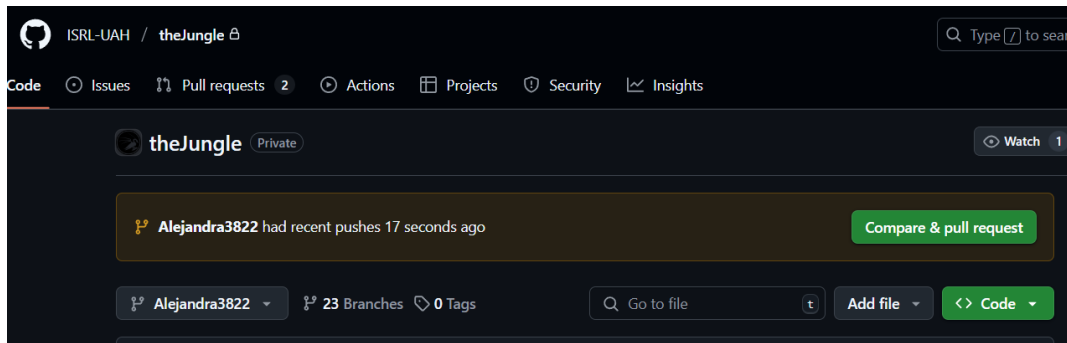
Para esta práctica utilizaremos dos máquinas virtuales: una con la distribución de Linux llamada "Remnux" y otra con Windows 10 equipada con "Flare". Ambas estarán configuradas dentro de la misma red interna, donde la máquina "Remnux" actuará como puerta de enlace (gateway) de "Flare". Además, solo la máquina Remnux tendrá acceso a internet, ya que necesita comunicarse con GitHub para interactuar con la rama Alejandra3822, donde subiremos todo nuestro análisis. En este trabajo, analizaremos la muestra de malware Dexter, identificada mediante el hash `SHA-256: 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92`.

Dexter es un malware diseñado específicamente para el robo de datos de tarjetas de crédito, atacando principalmente sistemas de puntos de venta (POS). Este malware intercepta la información sensible durante el procesamiento de las transacciones, capturando los datos de las tarjetas antes de su encriptación, lo que permite a los atacantes acceder a información confidencial para realizar fraudes. Durante el análisis, investigaremos el comportamiento del malware mediante técnicas de análisis estático y dinámico, evaluando su funcionalidad y los posibles indicadores de compromiso (IOCs). El objetivo es comprender a profundidad su impacto y desarrollar estrategias para mitigar futuros ataques.

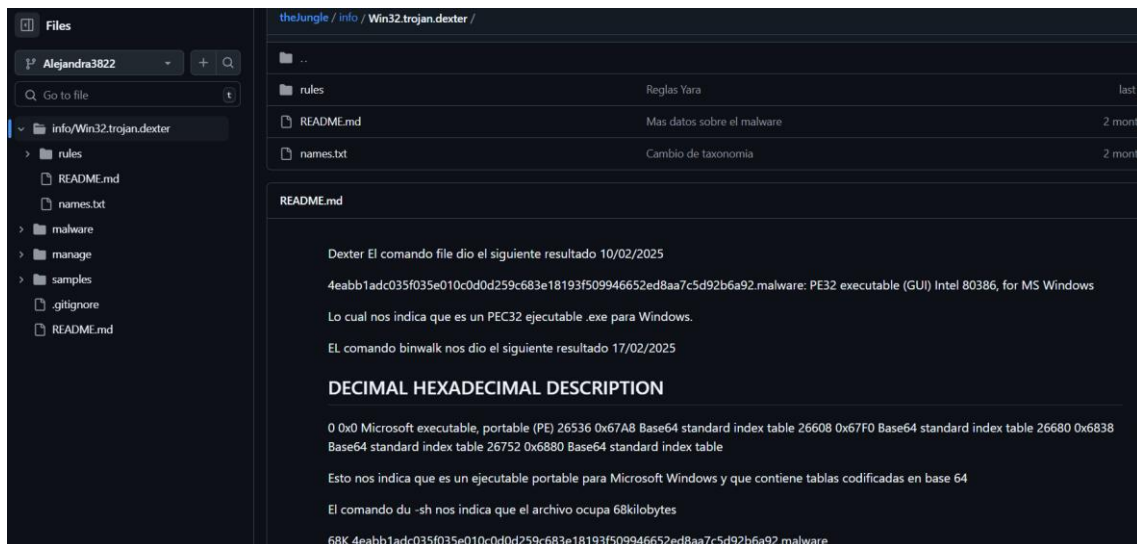
## 2. Análisis estático

### 2.1. Inventariado

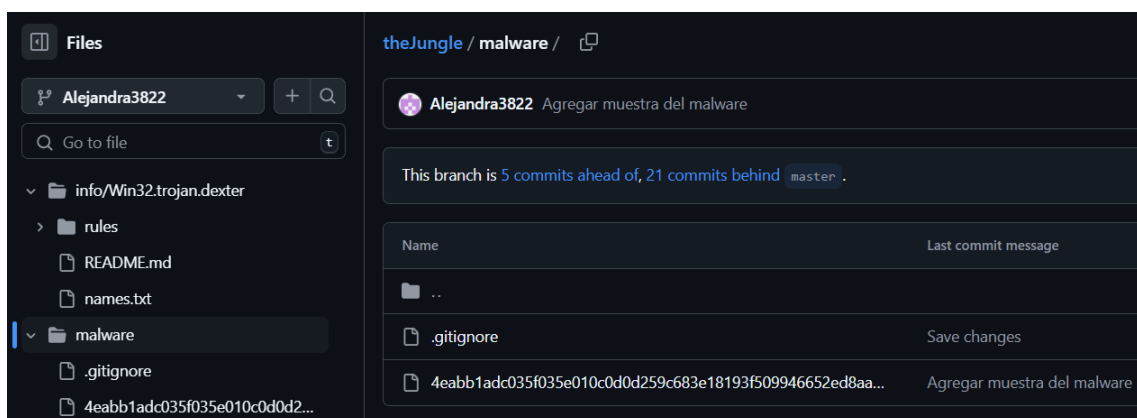
Se creó una rama en theJungle con el nombre Alejandra3822.



Se cuenta con la carpeta “info” donde se fueron registrando los hallazgos sobre la muestra según las clases de Análisis Avanzado de Malware y una carpeta llamada rules para las reglas de yara.



En la carpeta “malware” se tiene la muestra descomprimida donde el nombre es el correspondiente hash SHA-256 de la misma con la extensión “malware”.



Finalmente en “samples” esta el punto zip de la muestra obtenida del repositorio theZoo, la contraseña para extraer el archivo y los correspondientes resúmenes hash en diferentes algoritmos.

```
remnux@remnux:~/theJungle/samples/Win32.trojan.dexter$ ls -l
total 72
-rw-rw-r-- 1 remnux remnux 33 Apr 4 18:24 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.md5sum
-rw-rw-r-- 1 remnux remnux 115 Apr 4 11:38 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.sha1sum
-rw-rw-r-- 1 remnux remnux 139 Apr 4 11:38 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.sha256sum
-rw-rw-r-- 1 remnux remnux 221 Apr 4 11:38 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.ssdeep
-rw-rw-r-- 1 remnux remnux 10 Apr 4 11:38 Dexter.pass
-rw-rw-r-- 1 remnux remnux 50216 Feb 2 18:42 Dexter.zip
```

En la última fila podemos observar ssdeep, una herramienta que se diferencia de los algoritmos tradicionales, ya que está diseñada para medir la similitud entre dos archivos. Esto es particularmente útil en casos donde se analizan variantes de malware, ya que permite identificar archivos que comparten características comunes aunque no sean idénticos. A diferencia de otros algoritmos de hashing, como MD5 o SHA-256, que solo detectan coincidencias exactas, ssdeep se enfoca en encontrar relaciones parciales o patrones similares, facilitando la detección de variantes y evoluciones dentro de una misma familia de malware.

Hash	Valor
<b>MD5</b>	140d24af0c2b3a18529df12dfbc5f6de
<b>SHA-1</b>	e8db5ad2b7ffede3e41b9c3adb24f3232d764931
<b>SHA-256</b>	4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92
<b>SSDEEP</b>	1536:LP1Jm9HYIzX7Bm8JYCNE1dekdwxiin+tw9t6tEMY4Wq:LPb+HYmtHqCWdekyiPwgEMY4

## 2.2. File

Mediante el comando file podemos identificar el tipo de archivo, este comando ignora la extensión del archivo y analiza los bytes iniciales para compararlos con una base de datos interna de números mágicos. En este caso, nos indica que es un archivo PE (portable executable) diseñado para sistemas de 32 bits del sistema operativo Windows y que posiblemente el archivo tiene una interfaz gráfica GUI para interactuar con el usuario.

```
remnux@remnux:~/theJungle/malware$ file 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware: PE32 executable (GUI) Intel 80386, for MS Windows
```

## 2.3. Du

Con el comando `du -sh` podemos observar el tamaño del archivo, en este caso es 68KB. No podemos deducir mucho con esto porque no tenemos otras muestras para comparar, pero podemos decir que es un tamaño bastante pequeño lo cual podría deberse a que contiene código compactado que se expande en memoria al ejecutarse o también para ocultarse debido a que algunas herramientas de seguridad priorizan el análisis de malware de archivos más grandes.

```
remnux@remnux:~/theJungle/malware$ du -sh 4eabblad035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
68K    4eabblad035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
```

## 2.4. Clamscan

Ahora ejecutamos el antivirus open source basado en firmas “Clamscan” para verificar si detectará este tipo de malware. Al analizar la muestra el antivirus detectó que el archivo contiene un malware correspondiente con la firma Win.Malware.Dexter-9654223-0, lo que sugiere que pertenece a la familia Dexter, un tipo de malware conocido por atacar sistemas POS (Point of sale).

```
remnux@remnux:~/theJungle/malware$ clamscan 4eabblad035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
LibClamAV Warning: *****
LibClamAV Warning: *** The virus database is older than 7 days! ***
LibClamAV Warning: *** Please update it as soon as possible. ***
LibClamAV Warning: *****
/home/remnux/theJungle/malware/4eabblad035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware: Win.Malware.Dexter-9654223-0 FOUND

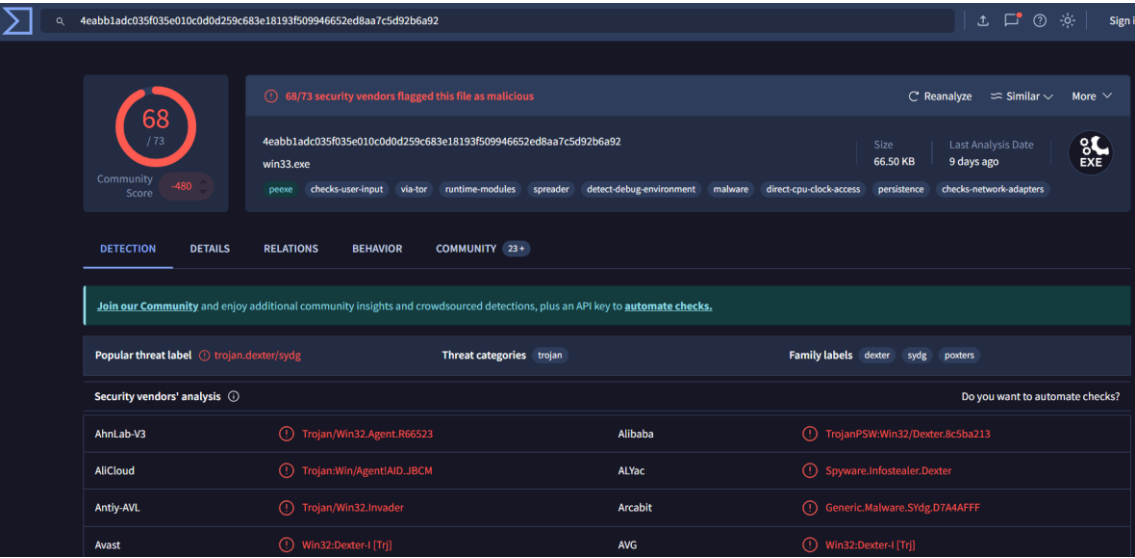
----- SCAN SUMMARY -----
Known viruses: 8704004
Engine version: 0.103.12
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.06 MB
Data read: 0.06 MB (ratio 1.00:1)
Time: 34.900 sec (0 m 34 s)
Start Date: 2025:02:24 06:48:43
End Date: 2025:02:24 06:49:18
```

## 2.5. VirusTotal

El análisis de virus total lo ha identificado como malicioso por 68 de 73 proveedores de seguridad, clasificándolo como un troyano relacionado con la familia Dexter. Los antivirus que no lo detectaron son: acronis, CMC y zoner. Se puede destacar que el

tamaño de la muestra que nos indica virus total es 66.50 KB lo cual es inferior a la que nosotros descargamos de theZoo que es de 68KB según nuestro análisis.

Es importante señalar que, en virus total, lo ideal es utilizar el hash del archivo para realizar un análisis, ya que subir el archivo directamente puede ser riesgoso porque virus total almacena los archivos enviados, y los usuarios con versiones premium podrían obtener acceso a ellos, lo que incrementa el peligro de que el malware se propague.



## 2.6. Binwalk

Al correr el comando binwalk podemos identificar, extraer y analizar datos incrustados dentro de archivos binarios, sin ejecutarlos. La salida del análisis realizado con el comando binwalk lo ha identificado como un ejecutable portable de Microsoft (PE), lo que confirma su diseño para sistemas Windows. Además, podemos observar que contiene varias tablas estándar de índice en base64 ubicadas en distintas secciones del archivo. Estas tablas podrían encontrarse en cualquier archivo de este tipo, pero en este contexto podemos formular la teoría de que podrían ser utilizadas para la ofuscación, almacenamiento de payloads ocultos o comunicación con servidores remotos.

```
remnux@remnux:~/theJungle/malware$ binwalk 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Microsoft executable, portable (PE)
26536	0x67A8	Base64 standard index table
26608	0x67F0	Base64 standard index table
26680	0x6838	Base64 standard index table
26752	0x6880	Base64 standard index table

## 2.7. Rabin2

Rabin2 se utiliza para la inspección, análisis y extracción de información de archivos binarios, como anteriormente vimos que el archivo contenía tablas en base64 con el comando binwalk, ahora configuramos una variable de entorno RABIN2\_DEBASE64=0 para evitar que rabin2 intente decodificar automáticamente las cadenas en este formato.

El primer filtro que realizamos es para identificar las IPs utilizando expresiones regulares e identificamos la IP 151.248.115.107 que aparecía como indicador de compromiso (IOC) en el reporte de la amenaza que analizamos para la primera tarea.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zz 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}.'
```

Aplicamos otro filtro para buscar algunos indicadores de compromiso (IOC) que observamos en el reporte de Microsoft, de los cuales se puede destacar que contramos:

- SecureDll.dll: Identificado en dos ubicaciones diferentes dentro de la sección .data del archivo. Este componente es conocido por su funcionalidad de keylogger.
- Strokes.log: Otro archivo que aparece en la sección .data y está relacionado con datos cifrados que el malware utiliza para registrar y almacenar información sensible, como pulsaciones de teclas.
- Tmp.log: Similar al archivo anterior, asociado con datos cifrados que podrían contener información adicional recopilada por el malware.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zzz 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep -E "javaplugin.exe|strokes.log|SecureDll.dll|tmp.log"
```

Filtrando cadenas relacionadas con las configuraciones del navegador y políticas de seguridad, podemos identificar lo siguiente:



- **LowRiskFileTypes:** Está asociada a una configuración que reduce los niveles de seguridad del sistema, permitiendo la ejecución de archivos de alto riesgo como extensiones .exe, .bat, .reg y .vbs. Esto facilita la ejecución de malware o scripts peligrosos sin la intervención del usuario.
- **HelperSolutions:** Indica posibles claves de registro o rutas que el malware utiliza para almacenar configuraciones maliciosas o datos relacionados con su operación.
- **Internet Settings\Zones\0:** Hace referencia a configuraciones de zonas de seguridad en el navegador web, donde el malware podría haber cambiado valores para disminuir restricciones de seguridad.
- **Policies:** Podría estar relacionada con políticas del sistema que el malware altera para garantizar su persistencia o facilitar sus actividades maliciosas.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zzz 4eabblad035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep -E "LowRiskFileTypes|HelperSolutions|Internet Settings|Policies"
209 0x00006500 0x00407100 33 34 (.data) ascii Software\\HelperSolutions Software
259 0x00006a00 0x00407600 33 34 (.data) ascii Software\\HelperSolutions Software
318 0x00006eb8 0x00407ab8 33 34 (.data) ascii Software\\HelperSolutions Software
365 0x000072d8 0x00407ed8 33 34 (.data) ascii Software\\HelperSolutions Software
416 0x000076e0 0x004082e0 63 64 (.data) ascii Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Associations
417 0x00007720 0x00408320 16 17 (.data) ascii LowRiskFileTypes
418 0x00007738 0x00408338 67 68 (.data) ascii Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings\\Zones\\
\\0
420 0x00007788 0x00408388 67 68 (.data) ascii Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings\\Zones\\
\\0
427 0x00007940 0x00408540 33 34 (.data) ascii Software\\HelperSolutions Software
468 0x00007ca0 0x004088a0 33 34 (.data) ascii Software\\HelperSolutions Software
511 0x00008048 0x00408c48 33 34 (.data) ascii Software\\HelperSolutions Software
594 0x000086a8 0x004092a8 33 34 (.data) ascii Software\\HelperSolutions Software
```

Los mutex (mutual exclusions) son utilizados por malware como Dexter para evitar que más de una instancia del programa malicioso se ejecute en el sistema simultáneamente. Esto asegura la estabilidad y el control del malware, evitando conflictos internos o duplicación de procesos maliciosos. Si ahora filtramos la cadena ".Mutex" para que nos entregue todo lo relacionado con este string, obtenemos:

- Cadenas como "CreateMutexA" y "UpdateMutex" que en Windows están relacionadas con la creación y actualización de mutex. Esto confirma que el malware interactúa directamente con componentes del sistema operativo para establecer sus mecanismos de exclusión y persistencia.
- Una referencia a "WindowsServiceStabilityMutex" que indica que el malware crea un identificador único como marcador de infección, este es un indicador de compromiso (IOC) característico de Dexter.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zz 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep '.Mutex'
144 0x00005fc4 0x00406bc4 12 13 .text ascii CreateMutexA
228 0x0000669c 0x0040729c 12 13 .data ascii UpdateMutex:
278 0x00006b9c 0x0040779c 12 13 .data ascii UpdateMutex:
337 0x00007054 0x00407c54 12 13 .data ascii UpdateMutex:
384 0x00007474 0x00408074 12 13 .data ascii UpdateMutex:
446 0x00007adc 0x004086dc 12 13 .data ascii UpdateMutex:
487 0x00007e3c 0x00408a3c 12 13 .data ascii UpdateMutex:
530 0x000081e4 0x00408de4 12 13 .data ascii UpdateMutex:
552 0x000082f8 0x00408ef8 28 29 .data ascii WindowsServiceStabilityMutex
613 0x00008844 0x00409444 12 13 .data ascii UpdateMutex:
```

El informe de Microsoft sobre el malware Dexter menciona específicamente que este utiliza rutas como `%APPDATA%\Java Security Plugin\javaplugin.exe` para instalarse y garantizar su persistencia en el sistema. Esto coincide con la cadena `javaplugin` encontrada con `rabin2`.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zz 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep 'java'
410 0x000075e4 0x004081e4 10 22 .data utf16le javaplugin
```

Al buscar con que ejecutables interactúa podemos notar una característica muy distintiva de Dexter ya que aparecen cadenas relacionadas con `iexplore.exe`, `firefox.exe`, `chrome.exe`, esto confirma el interés del malware en interceptar datos directamente desde navegadores populares. Sobre todo, por múltiples líneas donde hace referencia a procesos de Internet Explorer y este es un indicador de compromiso que se destaca también en el informe de Microsoft.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zz 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep '.exe'
212 0x0000658c 0x0040718c 12 13 .data ascii wmiprvse.exe
213 0x0000659c 0x0040719c 11 12 .data ascii LogonUI.exe
214 0x000065a8 0x004071a8 11 12 .data ascii svchost.exe
215 0x000065b4 0x004071b4 12 13 .data ascii iexplore.exe
216 0x000065c4 0x004071c4 12 13 .data ascii explorer.exe
218 0x000065dc 0x004071dc 8 9 .data ascii smss.exe
219 0x000065e8 0x004071e8 9 10 .data ascii csrss.exe
220 0x000065f4 0x004071f4 12 13 .data ascii winlogon.exe
221 0x00006604 0x00407204 9 10 .data ascii lsass.exe
222 0x00006610 0x00407210 11 12 .data ascii spoolsv.exe
223 0x0000661c 0x0040721c 7 8 .data ascii alg.exe
224 0x00006624 0x00407224 11 12 .data ascii wuauclt.exe
225 0x00006630 0x00407230 11 12 .data ascii firefox.exe
226 0x0000663c 0x0040723c 10 11 .data ascii chrome.exe
227 0x00006648 0x00407248 10 11 .data ascii devenv.exe
262 0x00006a8c 0x0040768c 12 13 .data ascii wmiprvse.exe
263 0x00006a9c 0x0040769c 11 12 .data ascii LogonUI.exe
264 0x00006aa8 0x004076a8 11 12 .data ascii svchost.exe
265 0x00006ab4 0x004076b4 12 13 .data ascii iexplore.exe
266 0x00006ac4 0x004076c4 12 13 .data ascii explorer.exe
```

En siguiente ilustración podemos observar algo que nos llama la atención, tenemos una cadena `%s\\%s%s.exe` que parece estar destinada a la generación dinámica de nombres o rutas. Esto podría ser un intento del malware para crear nombres dinámicos y que de esta forma sea más difícil detectarlo.

```

527 0x00008178 0x00408d78 11 12 .data ascii firefox.exe
528 0x00008184 0x00408d84 10 11 .data ascii chrome.exe
529 0x00008190 0x00408d90 10 11 .data ascii devenv.exe
561 0x000083c0 0x00408fc0 31 64 .data utf16le \Internet Explorer\iexplore.exe
562 0x00008400 0x00409000 31 64 .data utf16le \Internet Explorer\iexplore.exe
597 0x00008734 0x00409334 12 13 .data ascii wmiprvse.exe
598 0x00008744 0x00409344 11 12 .data ascii LogonUI.exe
599 0x00008750 0x00409350 11 12 .data ascii svchost.exe
600 0x0000875c 0x0040935c 12 13 .data ascii iexplore.exe
601 0x0000876c 0x0040936c 12 13 .data ascii explorer.exe
603 0x00008784 0x00409384 8 9 .data ascii smss.exe
604 0x00008790 0x00409390 9 10 .data ascii csrss.exe
605 0x0000879c 0x0040939c 12 13 .data ascii winlogon.exe
606 0x000087ac 0x004093ac 9 10 .data ascii lsass.exe
607 0x000087b8 0x004093b8 11 12 .data ascii spoolsv.exe
608 0x000087c4 0x004093c4 7 8 .data ascii alg.exe
609 0x000087cc 0x004093cc 11 12 .data ascii wuaclt.exe
610 0x000087d8 0x004093d8 11 12 .data ascii firefox.exe
611 0x000087e4 0x004093e4 10 11 .data ascii chrome.exe
612 0x000087f0 0x004093f0 10 11 .data ascii devenv.exe
632 0x00008914 0x00409514 12 26 .data utf16le %s\s\s.exe
634 0x0000893c 0x0040953c 12 26 .data utf16le %s\s\s.exe

```

Entre las dll que utiliza podemos destacar las siguientes:

- Kernel32.dll proporciona funciones básicas del sistema, como manejo de archivos, memoria y procesos. En este caso de malware podría por ejemplo utilizarlo para crear y gestionar procesos maliciosos.
- Advapi32.dll ofrece funciones avanzadas de seguridad y manipulación de claves del registro. Según los detalles del informe algunos usos maliciosos serían modificar claves del registro para persistencia por ejemplo ajustes en HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.
- Wininet.dll y urlmon.dll son librerías relacionadas con la comunicación de red y el manejo de HTTP/FTP. Dexter podría utilizarlas para enviar la información robada y/o para descargar payloads adicionales desde servidores remotos.
- Ws2\_32.dll contiene funciones para manejo de sockets y comunicación de red. Usada por Dexter para establecer conexiones con servidores remotos, facilitando la transferencia de datos robados.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zz 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep 'dll'
159 0x000060bc 0x00406cbc 12 13 .text ascii KERNEL32.dll
170 0x00006178 0x00406d78 10 11 .text ascii USER32.dll
184 0x0000627c 0x00406e7c 12 13 .text ascii ADVAPI32.dll
186 0x0000629e 0x00406e9e 11 12 .text ascii SHELL32.dll
188 0x000062ba 0x00406eba 9 10 .text ascii ole32.dll
191 0x000062da 0x00406eda 11 12 .text ascii SHLWAPI.dll
200 0x00006386 0x00406f86 11 12 .text ascii WININET.dll
202 0x000063aa 0x00406faa 10 11 .text ascii urlmon.dll
203 0x000063b6 0x00406fb6 10 11 .text ascii WS2_32.dll
205 0x000063d2 0x00406fd2 10 11 .text ascii RPCRT4.dll
300 0x00006cbc 0x004078bc 12 13 .data ascii kernel32.dll
553 0x00008318 0x00408f18 9 10 .data ascii nt.dll
560 0x000083ac 0x00408fac 12 13 .data ascii kernel32.dll
563 0x00008440 0x00409040 10 11 .data ascii user32.dll
564 0x0000844c 0x0040904c 12 13 .data ascii advapi32.dll
565 0x0000845c 0x0040905c 11 12 .data ascii shell32.dll
566 0x00008468 0x00409068 10 11 .data ascii urlmon.dll
567 0x00008474 0x00409074 11 12 .data ascii wininet.dll
568 0x00008480 0x00409080 9 10 .data ascii gdi32.dll
569 0x0000848c 0x0040908c 10 11 .data ascii rpcrt4.dll
570 0x00008498 0x00409098 10 11 .data ascii ws2_32.dll
572 0x000084b8 0x004090b8 10 11 .data ascii user32.dll
576 0x000084f0 0x004090f0 13 14 .data ascii SecureDll.dll
577 0x00008500 0x00409100 13 14 .data ascii SecureDll.dll
984 0x0000e3ae 0x004109ae 4 5 .rsrc ascii .dll
```

Finalmente, con rabin2 identificamos las variables que son mencionadas por el informe de Microsoft, las cuales son utilizadas para reducir la configuración de seguridad del navegador web al crear claves de registro mediante estas.

```
remnux@remnux:~/theJungle/malware$ RABIN2_DEBASE64=0 rabin2 -zz 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware | grep 'val'
239 0x00006700 0x00407300 5 6 .data ascii &val=
289 0x00006c00 0x00407800 5 6 .data ascii &val=
348 0x000070b8 0x00407cb8 5 6 .data ascii &val=
395 0x000074d8 0x004080d8 5 6 .data ascii &val=
457 0x00007b40 0x00408740 5 6 .data ascii &val=
498 0x00007ea0 0x00408aa0 5 6 .data ascii &val=
541 0x00008248 0x00408e48 5 6 .data ascii &val=
582 0x00008534 0x00409134 4 5 .data ascii val1
583 0x0000853c 0x0040913c 4 5 .data ascii val2
624 0x000088a8 0x004094a8 5 6 .data ascii &val=
1153 0x0000f193 0x00411793 5 6 .rsrc ascii &val1
```

## 2.7. Exiftool

La herramienta exiftool nos sirve para leer los metadatos de un archivo, aquí podemos destacar:

- Time stamp nos indica que la muestra fue compilada el 2013-08-28.
- Confirma que es un portable ejecutable (PE) de 32 bits para Windows y que cuenta con un subsistema que lo hace compatible con entornos gráficos (GUI)

```

remnux@remnux:~/theJungle/malware$ exiftool 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
ExifTool Version Number      : 12.76
File Name                    : 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
Directory                   : .
File Size                    : 68 kB
File Modification Date/Time  : 2025:02:17 09:22:45-05:00
File Access Date/Time       : 2025:02:23 18:38:31-05:00
File Inode Change Date/Time  : 2025:02:17 09:23:15-05:00
File Permissions             : -rw-rw-r--
File Type                    : Win32 EXE
File Type Extension          : exe
MIME Type                    : application/octet-stream
Machine Type                 : Intel 386 or later, and compatibles
Time Stamp                   : 2013:08:28 12:22:09-04:00
Image File Characteristics   : Executable, 32-bit
PE Type                      : PE32
Linker Version                : 8.0
Code Size                    : 24576
Initialized Data Size         : 42496
Uninitialized Data Size       : 0
Entry Point                   : 0x3af0
OS Version                   : 4.0
Image Version                 : 0.0
Subsystem Version             : 4.0
Subsystem                     : Windows GUI
remnux@remnux:~/theJungle/malware$

```

## 2.8. Entroper

Clonamos del repositorio la herramienta entroper que nos permitirá graficar la entropía de un fichero.

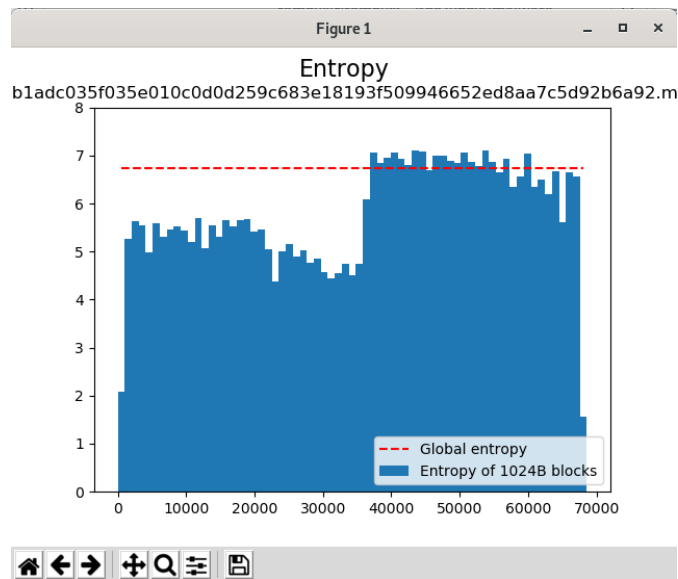
```

remnux@remnux:~/theJungle/malware$ git clone https://github.com/junquera/entroper
Cloning into 'entroper'...
remote: Enumerating objects: 38, done.
remote: Total 38 (delta 0), reused 0 (delta 0), pack-reused 38 (from 1)
Unpacking objects: 100% (38/38), 24.76 KiB | 563.00 KiB/s, done.
remnux@remnux:~/theJungle/malware$ python3 entroper/entroper.py 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
67 1024 6.7446951217436375

```

La siguiente gráfica de entropía revela información clave sobre la estructura y las características del archivo analizado:

- La baja entropía inicial sugiere datos estructurados, probablemente encabezados o configuraciones iniciales en texto plano.
- La alta entropía en la segunda parte del archivo indica la presencia de datos cifrados o codificados por ejemplo en base64.



## 2.9. Floss

Floss permite identificar cadenas que podrían ser indicadores de compromiso (IOCs) o revelar información sobre el comportamiento del archivo malicioso. En este caso nos llama la atención:

- Estas cadenas update, checkin, scanin, uninstall podrían estar relacionadas con comandos o configuraciones internas del malware. Un dato importante es que una vez que Dexter tiene la información que necesita se auto desinstala del sistema para evitar ser detectado.
- Volvemos a identificar %s=%s, %s%s%d, %s\\%s\\%s.exe que parecen ser estructuras dinámicas sugieren que el malware construye rutas o nombres de archivos en tiempo de ejecución.

```
remnux@remnux:~/theJungle/malware$ floss 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware > flossDexter
```

```
FLOSS DECODED STRINGS (12)

/v4jJ0A=
jpdpx
jpdpx
jpdpx
/uQrQA==
update-
checkin:
scanin:
uninstall
%s%s
%s%s%d
%s\\%s
%s\\%s\\%s.exe
```

## 2.9. Xxd

El comando xxd se utiliza para realizar un volcado hexadecimal de un archivo, permitiendo analizar su contenido en formato hexadecimal y ASCII. Es especialmente útil en análisis forense y de malware, ya que revela detalles sobre la estructura interna de los archivos. En este caso, identificamos el número mágico del archivo ejecutable .exe, que es 4D 5A, correspondiente a la firma "MZ".

```
remnux@remnux:~/theJungle/malware$ xxd 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ.....
00000010: b800 0000 0000 0000 4000 0000 0000 0000  .....@.....
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000030: 0000 0000 0000 0000 0000 0000 d800 0000  .....
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  .....!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program canno
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS
00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000  mode....$.....
00000080: 22c7 480d 66a6 265e 66a6 265e 66a6 265e  ".H.f.&^f.&^f.&^
00000090: a5a9 7b5e 73a6 265e 66a6 275e 15a6 265e  ..{^s.&^f.'^..&^
000000a0: 4160 4b5e 6fa6 265e 4160 5a5e 67a6 265e  A^K^o.&^A`Z^g.&^
000000b0: 4160 5e5e 67a6 265e 5269 6368 66a6 265e  A`^g.&^Richf.&^
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000d0: 0000 0000 0000 0000 5045 0000 4c01 0400  .....PE..L...
000000e0: b123 1e52 0000 0000 0000 0000 e000 0201  .#.R.....
000000f0: 0b01 0800 0060 0000 00a6 0000 0000 0000  .....
00000100: f03a 0000 0010 0000 0070 0000 0000 4000  .....p....@.
00000110: 0010 0000 0002 0000 0400 0000 0000 0000  .....
00000120: 0400 0000 0000 0000 0040 0100 0004 0000  .....@.....
```

## 2.10. Yara

Creamos una regla de YARA personalizada para detectar la IP 151.248.115.107, que previamente identificamos utilizando rabin2 durante el análisis del archivo malicioso. Aunque esta regla puede servir para localizar instancias específicas relacionadas con esta IP, cabe destacar que los atacantes suelen cambiar o rotar direcciones IP rápidamente, lo que reduce la efectividad de este tipo de detección.

```
remnux@remnux:~/theJungle/malware$ yara ~/theJungle/info/Win32.trojan.dexter/rules/rule_dexter_ip.yar
r 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
DetectSpecificIP 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
```

```
remnux@remnux:~/theJungle/info/Win32.trojan.dexter/rules$ cat rule_dexter_ip.yar
rule DetectSpecificIP {
  strings:
    $ip_address = "151.248.115.107"
  condition:
    $ip_address
}
```

En cambio, strokes es un indicador de compromiso (IOC) clave usado por el malware para registrar pulsaciones de teclas de la víctima, por lo que en esta ocasión se creó una regla de yara que detecta esta cadena específica y verifica si el archivo es un ejecutable mediante el número mágico 4D 5A.

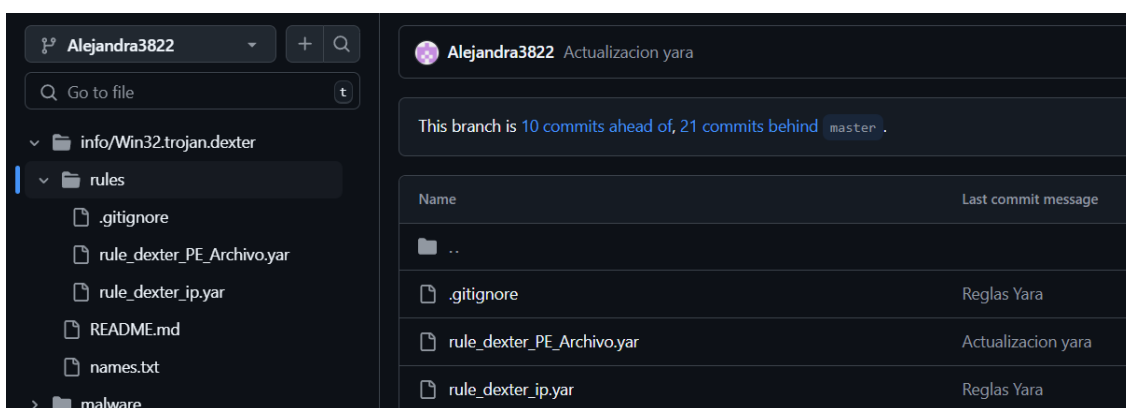


```
remnux@remnux:~/theJungle/malware$ yara -/theJungle/info/Win32.trojan.dexter/rules/rule_dexter_PE_Archivo.yar 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
PE_StrokesLog_Base64 4eabb1adc035f035e010c0d0d259c683e18193f509946652ed8aa7c5d92b6a92.malware
```

```
GNU nano 4.8
rule PE_StrokesLog_Base64
{
  strings:
    $mz_header = { 4D 5A }
    $strokes = "strokes"

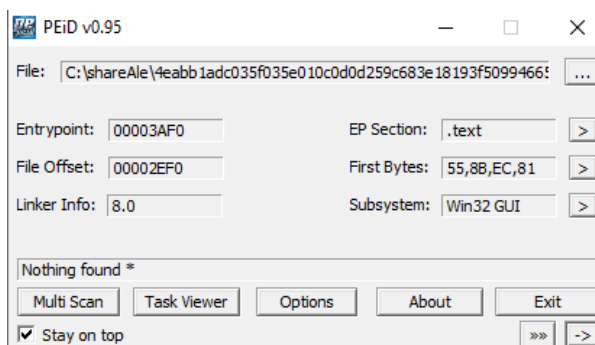
  condition:
    $mz_header and $strokes
}
```

Esto se encuentra inventariado en el github dentro de la carpeta info.



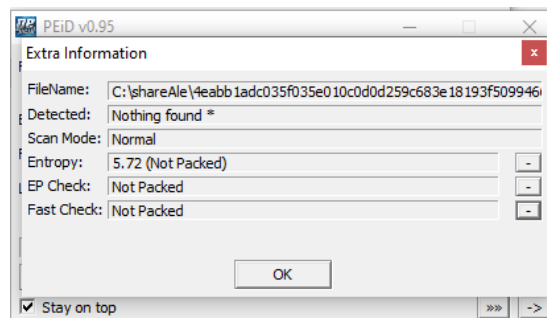
## 2.11. PEiD.

Se identificó que el punto de entrada (Entrypoint) se encuentra en la dirección 00003AF0, dentro de la sección .text, lo que indica el lugar donde comienza la ejecución del código y que el subsistema es Win32 GUI. Sin embargo, PEiD no detectó un packer, mostrando el resultado "Nothing found".



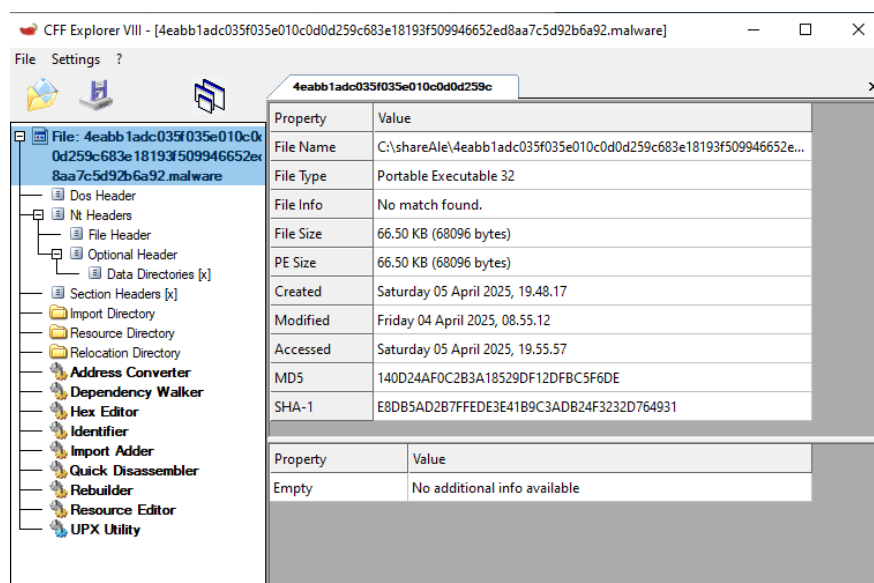


En la ventana de información extra podemos observar que la entropía es 5.72, lo que indicaría que el archivo no está empaquetado.

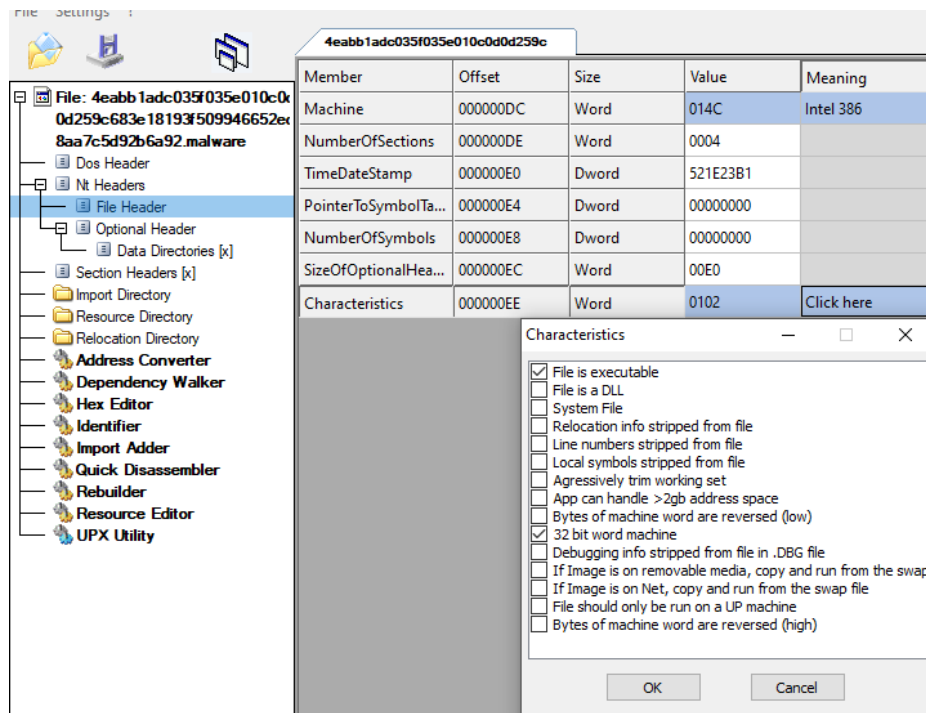


## 2.12. CFF explorer

Hasta ahora trabajamos en la máquina Remnux, ahora pasamos a Flare con la herramienta CFF explorer que nos sirve para analizar archivos portables ejecutables (PE). Aquí podemos observar que nos indica que el tamaño del archivo es de 66.5 KB y como tipo lo identifica como exe de 32 bits.

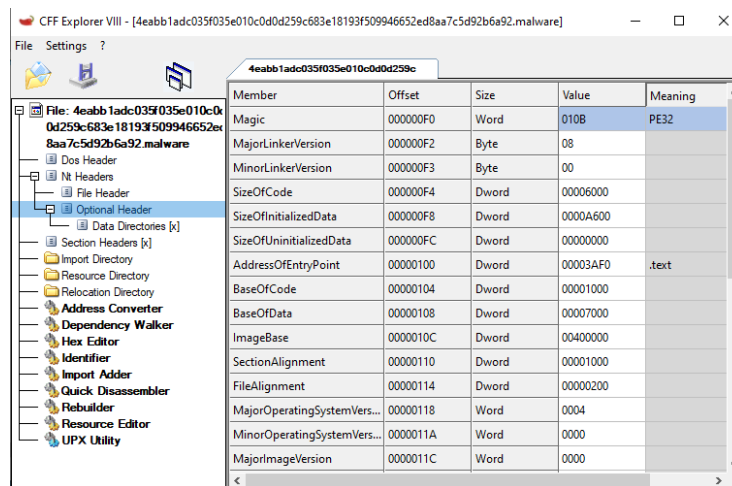


Al igual que antes podemos observar el time stamp pero esta vez esta en hexadecimal y también nos indica que el fichero tiene 4 secciones.



En las cabeceras opcionales pero relevantes podemos destacar:

- ImageBase 00400000 que representa la dirección preferida donde el archivo intenta ser cargado en memoria, es estándar para archivos ejecutables de Windows.
- AddressOfEntryPoint 00003AF0 que indica la dirección relativa al inicio del código que será ejecutado. Este punto de entrada apunta a la sección .text, donde normalmente reside el código ejecutable.
- BaseOfCode 00001000 es la dirección virtual inicial de la sección que contiene el código ejecutable, confirmando su ubicación lógica dentro de la memoria.



En la sección de cabeceras podemos comparar virtual size que es el tamaño que ocupa la sección en memoria cuando se carga el archivo y raw size que es el tamaño real de la sección en el archivo en disco.

#### Sección .text:

- Virtual size: 00005FDE
- Raw size: 00006000

Esto indica que la sección ocupa un espacio similar en disco y en memoria, lo que es común para código ejecutable.

#### Sección .data:

- Virtual Size: 00003110
- Raw Size: 00000260

La discrepancia aquí es significativa, ya que se reserva más espacio en memoria que en disco, lo cual podría ser típico para secciones que contienen datos dinámicos o variables.

#### Sección .rsrc:

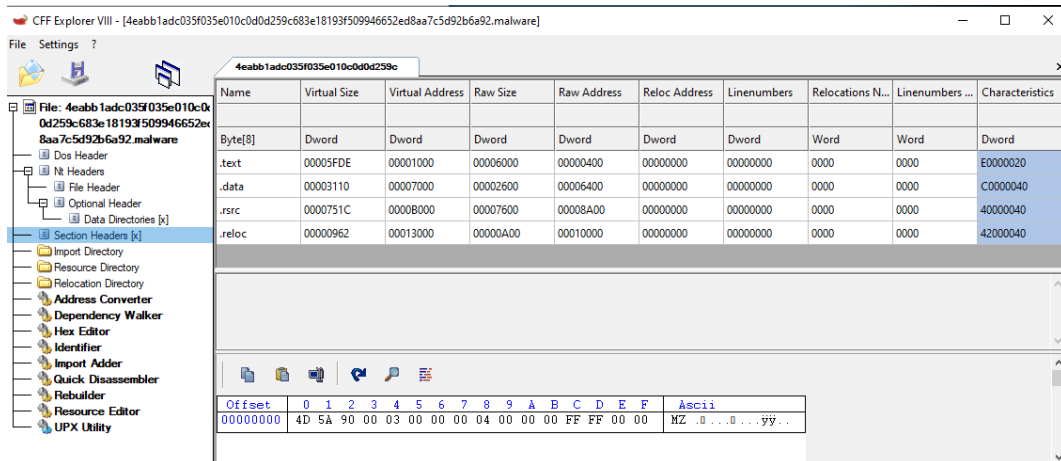
- Virtual Size: 0000751C
- Raw Size: 00007600

Los tamaños son casi iguales, lo que sugiere que los recursos están completamente empaquetados y alineados para su carga en memoria.

## Sección .reloc:

- Virtual Size: 00000962
- Raw Size: 00000A00

Esto muestra que la sección está preparada para manejar reubicaciones, aunque ocupa un tamaño pequeño.

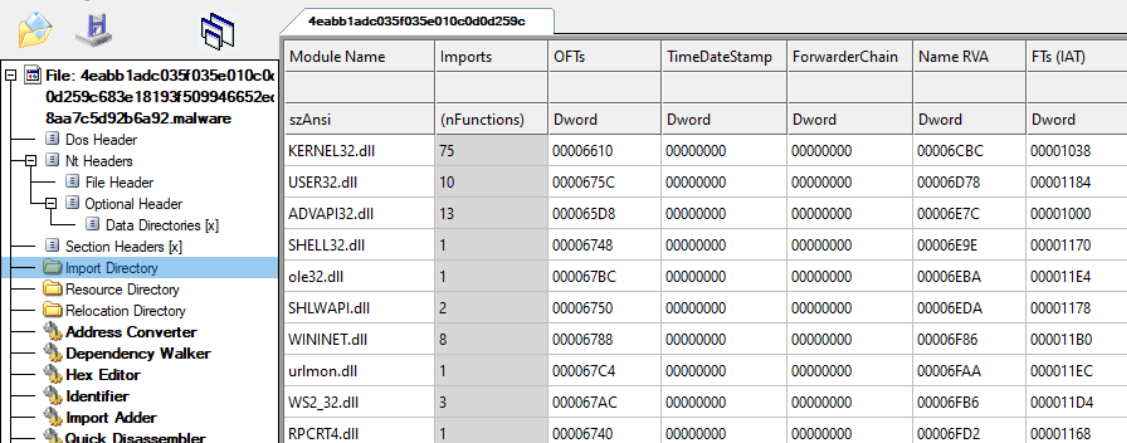


La lista de librerías importadas por el malware Dexter es la siguiente:

KERNEL32.dll Maneja funciones fundamentales del sistema operativo como la gestión de memoria, procesos, hilos (threads) y operaciones básicas de entrada/salida.

- USER32.dll es para la interfaz gráfica de usuario (GUI) ya que controla ventanas, botones y otros elementos visuales.
- ADVAPI32.dll proporciona funciones avanzadas relacionadas con la seguridad, como la gestión de cuentas de usuario, claves de registro y servicios del sistema.
- SHELL32.dll ofrece funciones para interactuar con el shell de Windows, como abrir o gestionar archivos y carpetas.
- ole32.dll se usa para trabajar con el modelo de objetos COM (component object model), que permite que las aplicaciones intercambien información entre sí.
- SHLWAPI.dll contiene utilidades para operaciones con cadenas, rutas y otros recursos de sistema.

- WININET.dll facilita el manejo de conexiones a Internet y protocolos como HTTP y FTP, crucial para las actividades de comunicación del malware.
- urlmon.dll sirve para gestionar URL y descargar contenido desde Internet.
- WS2\_32.dll maneja los sockets de red, fundamentales para las comunicaciones en red (como enviar datos al servidor C&C).
- RPCRT4.dll soporta llamadas a procedimientos remotos (RPC), esenciales para la comunicación entre procesos.



Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	75	00006610	00000000	00000000	00006CBC	00001038
USER32.dll	10	0000675C	00000000	00000000	00006D78	00001184
ADVAPI32.dll	13	000065D8	00000000	00000000	00006E7C	00001000
SHELL32.dll	1	00006748	00000000	00000000	00006E9E	00001170
ole32.dll	1	000067BC	00000000	00000000	00006EBA	000011E4
SHLWAPI.dll	2	00006750	00000000	00000000	00006EDA	00001178
WININET.dll	8	00006788	00000000	00000000	00006F86	000011B0
urlmon.dll	1	000067C4	00000000	00000000	00006FAA	000011EC
WS2_32.dll	3	000067AC	00000000	00000000	00006FB6	000011D4
RPCRT4.dll	1	00006740	00000000	00000000	00006FD2	00001168

Ahora vamos a describir las funciones relacionadas con ADVAPI32.dll:

- OpenProcessToken: Abre el token de acceso asociado con un proceso.
- AdjustTokenPrivileges: Modifica los privilegios de un token de acceso.
- RegCloseKey: Cierra una clave de registro abierta.
- RegSetValueExA: Establece un valor en el registro (ASCII).
- RegOpenKeyExA: Abre una clave de registro específica (ASCII).
- RegQueryValueExA: Recupera datos asociados con un valor del registro (ASCII).
- RegDeleteValueA: Elimina un valor de una clave de registro (ASCII).
- GetUserNameA: Obtiene el nombre del usuario actual (ASCII).
- RegCreateKeyExA: Crea o abre una clave del registro (ASCII).
- RegSetValueExW: Establece un valor en el registro (Unicode).
- RegNotifyChangeKeyValue: Detecta cambios en una clave de registro.
- RegDeleteKeyA: Elimina una clave del registro (ASCII).

Al usar funciones como RegSetValueExA, RegCreateKeyExA o RegOpenKeyExA, el malware puede modificar el registro de Windows para garantizar su persistencia, configurándose para ejecutarse automáticamente al iniciar el sistema. Además, funciones como RegNotifyChangeKeyValue le permiten monitorear modificaciones en claves críticas del registro, lo que podría ser útil para reaccionar ante cambios realizados por el usuario o software antivirus. Otras funciones como AdjustTokenPrivileges y OpenProcessToken son empleadas para elevar sus privilegios y obtener acceso a áreas restringidas del sistema, asegurando que pueda realizar acciones avanzadas sin interrupciones. Finalmente, el propio malware puede usar funciones como RegDeleteKeyA para eliminar las claves que creó, asegurando que se borren rastros en el sistema cuando decide desinstalarse, cumpliendo así con sus técnicas de evasión.

Dentro de las funciones de WININET.dll tenemos:

- HttpSendRequestA: Envía una solicitud HTTP al servidor.
- InternetCloseHandle: Cierra un identificador de conexión a Internet.
- HttpOpenRequestA: Crea una solicitud HTTP específica (GET, POST, etc.).
- InternetOpenA: Inicializa el acceso a las funciones de Internet.
- InternetGetCookieA: Obtiene cookies asociadas a una URL específica.
- InternetReadFile: Lee datos desde una conexión de Internet.
- InternetOpenUrlA: Abre una URL específica.
- InternetConnectA: Establece una conexión con un servidor remoto.

Dexter podría utilizar estas funciones para comunicarse con su servidor de comando y control (C&C). Mediante HttpSendRequestA, HttpOpenRequestA y InternetConnectA, el malware podría enviar información robada, como datos de tarjetas de crédito, a un servidor remoto. Las funciones InternetReadFile y InternetOpenUrlA le permitirían descargar y ejecutar otros componentes maliciosos desde Internet. Además, al usar InternetGetCookieA, podría recopilar cookies que contengan información de inicio de sesión o de sesión del usuario para ampliar su capacidad de robo de datos. Por último, al cerrar conexiones con InternetCloseHandle, evita dejar rastros.

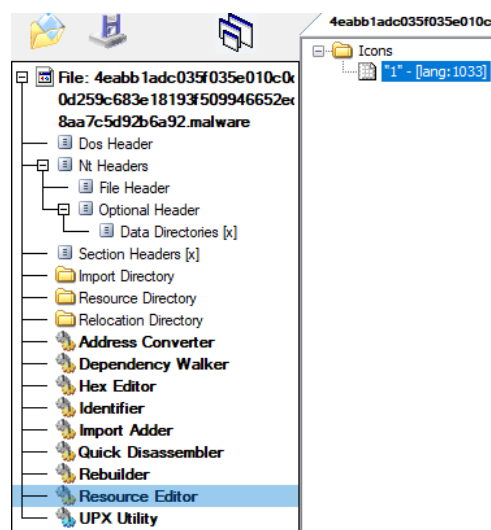
OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00006EE6	00006EE6	0059	HttpSendRequestA
00006EFA	00006EFA	0069	InternetCloseHandle
00006F10	00006F10	0055	HttpOpenRequestA
00006F38	00006F38	0092	InternetOpenA
00006F48	00006F48	0084	InternetGetCookieA
00006F5E	00006F5E	009A	InternetReadFile
00006F72	00006F72	0093	InternetOpenUrlA
00006F24	00006F24	006F	InternetConnectA

Al revisar shell32.dll solo tenemos una función:

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00006E8A	00006E8A	00B2	SHGetFolderPathW

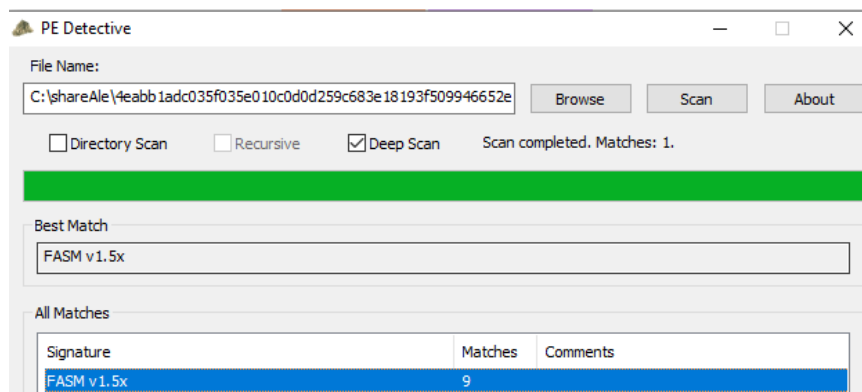
En el caso de Dexter, podría la función SHGetFolderPathW para obtener rutas de directorios especiales del sistema, como "Escritorio", "Mis Documentos" o "Archivos de programa".

Finalmente, revisamos el editor de recursos porque los malware suelen tener archivos incrustados dentro de su ejecutable principal ya que permite al malware ocultar componentes adicionales y evitar una detección inmediata. Sin embargo, en el caso del malware Dexter no se identificó esto.



## 2.13. PE detective

El análisis con PE detective indica que el archivo analizado probablemente fue creado usando FASM v1.5x (Flat Assembler), una herramienta de ensamblador que permite crear ejecutables en lenguaje de bajo nivel. Esto sugiere que el malware Dexter podría haber sido desarrollado usando ensamblador para optimizar su tamaño y para complicar su análisis al ser más difícil de interpretar para herramientas de descompilación estándar.



## 3. Análisis dinámico

### 3.1. Process monitor

Primero aplicamos un filtro para incluir el nombre del proceso, en este caso es el nombre de nuestra muestra de malware .exe que corrimos haciendo doble clic.

Column	Relation	Value	Action
<input checked="" type="checkbox"/>	Process N...	contains 4eabb1adc035f035e010c0d0d25...	Include

A continuación, analizamos lo más relevante que podemos observar desde que iniciamos el malware:

#### 1. Process Start:

- El malware inicia su proceso con el clic en el ejecutable y obtiene el Process ID (PID) 5384. Esto confirma el comienzo de su ejecución.



- Se ejecuta desde la ruta C:\shareAle\ y su proceso padre tiene el PID 4400.

## **2. Thread Create:**

- El proceso crea un nuevo hilo de ejecución dentro de sí mismo.
- La base de la imagen es 0x400000, lo que indica la dirección en memoria donde comienza la ejecución del binario principal, ya identificada en el análisis estático previo.

## **3. Load Image (ntdll.dll):**

- El malware carga ntdll.dll desde C:\Windows\System32.
- Esta biblioteca es clave para interactuar con funciones de bajo nivel del sistema operativo, como la gestión de memoria y excepciones.
- La base de la imagen está en 0x7fface230000.

## **4. Load Image (ntdll.dll - versión SysWOW64):**

- Carga otra versión de ntdll.dll, esta vez desde C:\Windows\SysWOW64.
- Esto asegura compatibilidad con sistemas Windows de 32 bits en arquitecturas de 64 bits.
- La base de la imagen está en 0x77df0000.

## **5. CreateFile (Prefetch):**

- El malware accede al archivo de prefetch ubicado en C:\Windows\Prefetch.
- Los archivos de prefetch almacenan información sobre programas ejecutados previamente para acelerar futuras ejecuciones.

## **6. QueryEAFile (Extended Attributes):**

- Consulta los atributos extendidos del archivo de prefetch.

- Esto podría estar relacionado con la verificación de permisos o metadata adicional asociada al archivo.

#### **7. QueryStandardInformationFile:**

- Obtiene información estándar del archivo de prefetch, como: tamaño del archivo, número de enlaces, fecha de creación/modificación.
- Posiblemente verifica esta información para decidir sus siguientes acciones.

#### **8. ReadFile (Prefetch):**

- Lee 4,458 bytes de información del archivo de prefetch, lo cual puede incluir datos sobre ejecuciones anteriores de otros programas.

#### **9. CloseFile (Prefetch):**

- Cierra el archivo de prefetch tras recolectar o validar información, indicando que el malware ha terminado de interactuar con él.

#### **10. RegOpenKey (HKLM\SYSTEM\CurrentControlSet\Control\Session Manager):**

- Abre la clave del registro Session Manager bajo HKLM (HKEY\_LOCAL\_MACHINE).
- Busca configuraciones relacionadas con la gestión de sesiones del sistema, tal vez para identificar configuraciones que pueda modificar para persistencia o evasión.

#### **11. RegQueryValue (HKLM\SYSTEM\CurrentControlSet\Control\Session Manager):**

- Consulta un valor en la clave mencionada.
- El malware obtiene un valor, como podría ser configuraciones predeterminadas del sistema.

**12. RegQueryValue (HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibleDeadlock):**

- Intenta consultar un subvalor relacionado con excepciones en bloqueos.
- El resultado es "NAME NOT FOUND", esto indica que el malware está probando si hay configuraciones específicas relacionadas con excepciones para decidir su comportamiento.

**13. RegOpenKey (HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap):**

- Intenta abrir la clave del registro relacionada con la administración de memoria segmentada.
- La operación fue redirigida, lo que sugiere que el acceso a esta clave está controlado o se redirige a otra ubicación interna del sistema.

**14. RegOpenKey (HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\ResourcePolicies):**

- Abre otra subclave llamada ResourcePolicies.
- Pero la clave no existe, posiblemente porque este sistema no tiene políticas de recursos configuradas.

**15. CreateFile (C:\Windows):**

- Intenta acceder al directorio principal del sistema operativo C:\Windows.
- Como resultado, el malware logra acceder con permisos de ejecución y sincronización, probablemente buscando archivos críticos del sistema.

**16. RegOpenKey (HKLM\SYSTEM\CurrentControlSet\Control\hivelist):**

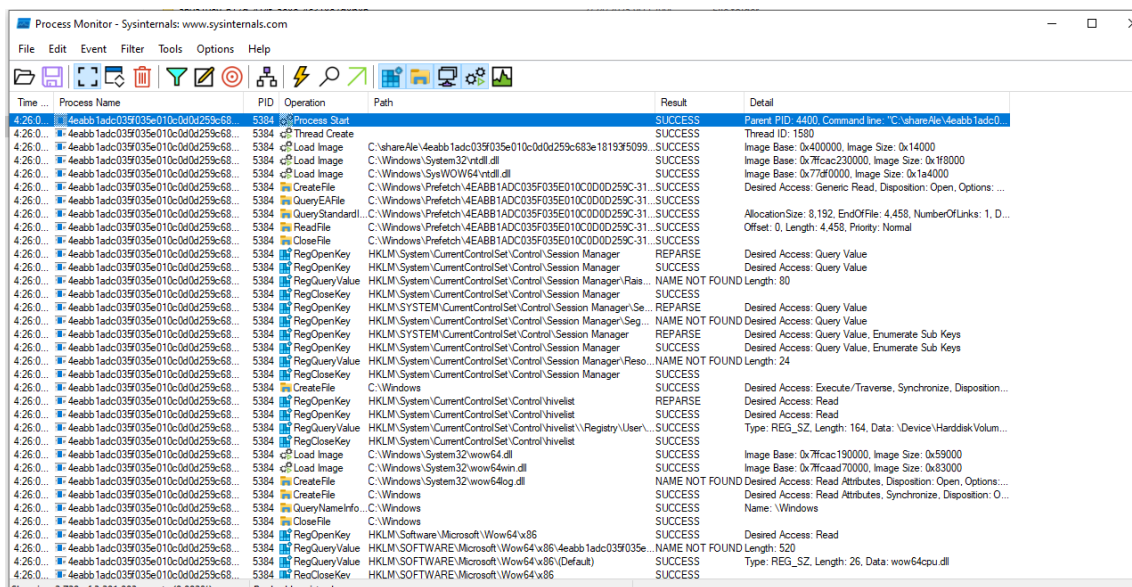
- Intenta abrir la clave hivelist, que contiene una lista de "hives" actualmente cargados en el sistema. Esto le permite al malware mapear las configuraciones actuales del sistema y los perfiles de usuario activos.

- El resultado “REPARSE” indica que el acceso inicial fue redirigido internamente.

## 17. RegQueryValue

(HKLM\SYSTEM\CurrentControlSet\Control\hivelist\Registry\User\S-1-5-21-2620994960-2257086029-795191477-1001\_Classes):

- Consulta un valor específico en el subdirectorio del registro relacionado con el SID (Security Identifier) del usuario 1001.
- Esta clave \_Classes contiene asociaciones de archivos, extensiones y configuraciones específicas para ese usuario.
- Al ser exitoso, el malware accede exitosamente al valor, recopilando datos del entorno del usuario.



Time ...	Process Name	PID	Operation	Path	Result	Detail
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	Process Start		SUCCESS	Parent PID: 4400, Command line: "C:\share\4eabb1adc0...
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	Thread Create		SUCCESS	Thread ID: 1580
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	Load Image	C:\share\4eabb1adc039f035e010c0d0d259c68\18193f5099...	SUCCESS	Image Base: 0x400000, Image Size: 0x14000
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x7f6ac230000, Image Size: 0x7f8000
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Image Base: 0x77d0000, Image Size: 0x1a4000
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	CreateFile	C:\Windows\Prefetch\4EABB1ADC039F035E010C0D0D259C-31...	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: ...
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	QueryEaFile	C:\Windows\Prefetch\4EABB1ADC039F035E010C0D0D259C-31...	SUCCESS	
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	QueryStandard...	C:\Windows\Prefetch\4EABB1ADC039F035E010C0D0D259C-31...	SUCCESS	AllocationSize: 8,192, EndOfFile: 4,458, NumberOfLinks: 1, D...
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	ReadFile	C:\Windows\Prefetch\4EABB1ADC039F035E010C0D0D259C-31...	SUCCESS	Offset: 0, Length: 4,458, Priority: Normal
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	CreateFile	C:\Windows\Prefetch\4EABB1ADC039F035E010C0D0D259C-31...	SUCCESS	
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Query Value
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Query Value
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegQueryValue	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Ra...	NAME NOT FOUND	Length: 80
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegCloseKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS	
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Se...	REPARSE	Desired Access: Query Value
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Seg...	NAME NOT FOUND	Desired Access: Query Value
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: Query Value, Enumerate Sub Keys
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: Query Value, Enumerate Sub Keys
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegQueryValue	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Reso...	NAME NOT FOUND	Length: 24
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegCloseKey	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	SUCCESS	
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	CreateFile	C:\Windows	SUCCESS	Desired Access: Execute/Traverse, Synchronize, Disposition...
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\hivelist	REPARSE	Desired Access: Read
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Control\hivelist	SUCCESS	Desired Access: Read
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegQueryValue	HKLM\SYSTEM\CurrentControlSet\Control\hivelist\Registry\User\...	SUCCESS	Type: REG_SZ, Length: 164, Data: \Device\Harddisk Volum...
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegCloseKey	HKLM\SYSTEM\CurrentControlSet\Control\hivelist	SUCCESS	
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x7f6ac190000, Image Size: 0x59000
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	Image Base: 0x7f6ac170000, Image Size: 0x3000
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	CreateFile	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: ...
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	CreateFile	C:\Windows	SUCCESS	Desired Access: Read Attributes, Synchronize, Disposition: O...
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	QueryNameInfo...	C:\Windows	SUCCESS	Name: \Windows
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	CloseFile	C:\Windows	SUCCESS	
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegOpenKey	HKLM\SOFTWARE\Microsoft\Wow64\86	SUCCESS	Desired Access: Read
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\86\4eabb1adc039f035e...	NAME NOT FOUND	Length: 520
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegQueryValue	HKLM\SOFTWARE\Microsoft\Wow64\86\4eabb1adc039f035e...	SUCCESS	Type: REG_SZ, Length: 26, Data: wow64cpu.dll
4:26:0...	4eabb1adc039f035e010c0d0d259c68	5384	RegCloseKey	HKLM\SOFTWARE\Microsoft\Wow64\86	SUCCESS	

## 1. QuerySecurityFile (iexplore.exe):

- Consulta información de seguridad del archivo iexplore.exe ubicado en C:\Program Files (x86)\Internet Explorer\.

## 2. QueryBasicInformationFile (iexplore.exe):

- Consulta información básica del archivo iexplore.exe.

### **3. CreateFile (iexplore.exe):**

- Abre el archivo iexplore.exe.

### **4. QueryStandardInformationFile (iexplore.exe):**

- Obtiene información sobre el archivo iexplore.exe, como su tamaño y enlaces.

### **5. CreateFileMapping (iexplore.exe):**

- Intenta crear un mapeo en memoria del archivo iexplore.exe.
- El resultado indica “FILE LOCKED WITH ONLY READERS”, lo que significa que el archivo está bloqueado y solo disponible para lectura. Esto podría ser un mecanismo de protección del sistema o una limitación de permisos.

### **6. QuerySecurityFile (sysmain.sdb):**

- Consulta información de seguridad del archivo sysmain.sdb ubicado en C:\Windows\appatch\.

### **7. QueryBasicInformationFile (sysmain.sdb):**

- Consulta información básica del archivo sysmain.sdb.

### **8. CreateFile (sysmain.sdb):**

- Abre el archivo sysmain.sdb, que es una base de datos del sistema utilizada para compatibilidad de aplicaciones.

### **9. QueryStandardInformationFile (sysmain.sdb):**

- Obtiene información sobre el archivo sysmain.sdb.

### **10. CreateFileMapping (sysmain.sdb):**

- Intenta crear un mapeo en memoria del archivo sysmain.sdb.
- El resultado indica “FILE LOCKED WITH ONLY READERS”, indicando que el archivo está bloqueado y no puede ser modificado, pero está disponible para lectura.

## 11. CloseFile (sysmain.sdb):

- Cierra el archivo sysmain.sdb tras realizar las operaciones necesarias, lo que confirma que el proceso ha terminado de interactuar con este archivo.

Time	Process Name	PID	Operation	Path	Result	Detail
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	QuerySecurityFile	C:\Program Files (x86)\Internet Explorer\explore.exe	SUCCESS	Information: Owner, Group, DACL, SACL, Label, Attribute, Pr...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	QueryBasicInfor...	C:\Program Files (x86)\Internet Explorer\explore.exe	SUCCESS	CreationTime: 5/5/2023 5:22:59 AM, LastAccessTime: 4/6/...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: ...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	QueryStandardI	C:\Windows\apppatch\sysmain.sdb	SUCCESS	AllocationSize: 4,075,520, EndOfFile: 4,073,452, NumberOfU...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	CreateFileMap...	C:\Windows\apppatch\sysmain.sdb	FILE LOCKED WITH ONLY READERS	SyncType: SyncTypeCreateSection, PageProtection: PAGE_...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	QueryStandardI	C:\Windows\apppatch\sysmain.sdb	SUCCESS	AllocationSize: 4,075,520, EndOfFile: 4,073,452, NumberOfU...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	CreateFileMap...	C:\Windows\apppatch\sysmain.sdb	SUCCESS	SyncType: SyncTypeOther
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	CloseFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	CloseFile	C:\Program Files (x86)\Internet Explorer\explore.exe	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\SOFTWARE\Microsoft\OLE	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	REPARSE	Desired Access: Read
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS	Desired Access: Read
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegSetInfoKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS	KeySetInformationClass: KeySetHandleTagsInformation, Len...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize\DisableM...	NAME NOT FOUND	Length: 20
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	REPARSE	Desired Access: Read
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS	Desired Access: Read
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegSetInfoKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS	KeySetInformationClass: KeySetHandleTagsInformation, Len...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize\DisableU...	NAME NOT FOUND	Length: 20
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	cp Thread Exit		SUCCESS	Thread ID: 4544, User Time: 0.0156250, Kernel Time: 0.031...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	cp Thread Exit		SUCCESS	Thread ID: 5852, User Time: 0.0000000, Kernel Time: 0.000...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	cp Thread Exit		SUCCESS	Thread ID: 4712, User Time: 0.0156250, Kernel Time: 0.000...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	cp Thread Exit		SUCCESS	Exit Status: 0, User Time: 0.0312500 seconds, Kernel Time: ...
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	cp Process Exit		SUCCESS	Desired Access: All Access
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegOpenKey	HKLM\System\CurrentControlSet\Services\lsam\State\UserSettings\S-1-5-21-262099...	SUCCESS	NAME NOT FOUND Length: 40
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegQueryValue	HKLM\System\CurrentControlSet\Services\lsam\State\UserSettings\S-1-5-21-262099...	NAME NOT FOUND	Length: 40
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\System\CurrentControlSet\Services\lsam\State\UserSettings\S-1-5-21-262099...	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	CloseFile	C:\Windows	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	CloseFile	C:\sharefile	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\System\CurrentControlSet\Control\Nls\CustomLocale	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Opt...	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	SUCCESS	
4:33:3...	deabb1ad0-039035e010c0d0d259c68...	3068	RegCloseKey	HKLM	SUCCESS	

## 3.2. Process explorer

En process monitor identificó el PID (Process ID) padre que ejecutó el malware como 4400. Este dato es crucial para determinar qué proceso originó la ejecución del archivo malicioso.

Para obtener más información sobre este PID, se realizó una búsqueda en process explorer, una herramienta avanzada para el análisis de procesos en tiempo real. Allí se verificó que el proceso con el PID 4400 corresponde al Explorador de Windows.

