***TRABAJO PRÁCTICO***

***Problemas Bancarios - lab#2***

***Materia***

Informática Forense

**Integrantes del Grupo:**

**Bernardo Napoleon Luscher**

**Turno: Noche**

**Docente: Antonio, Juan Manuel**

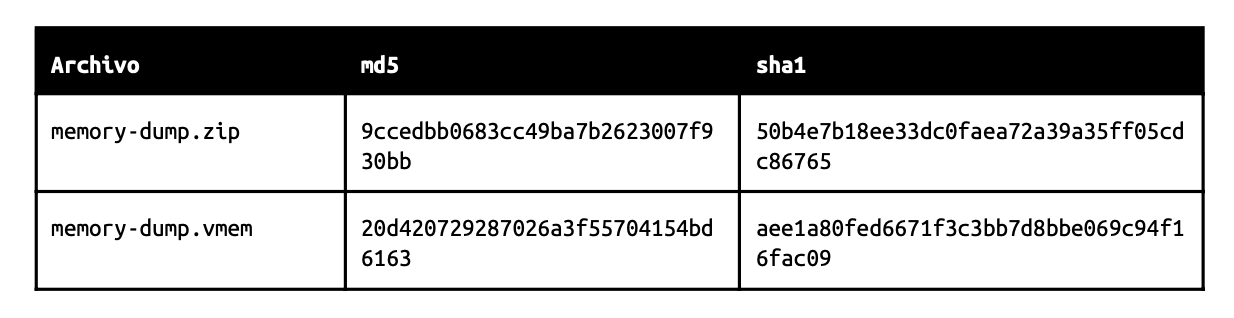
**Fecha de entrega: 13/11/2023**

**Introducción**

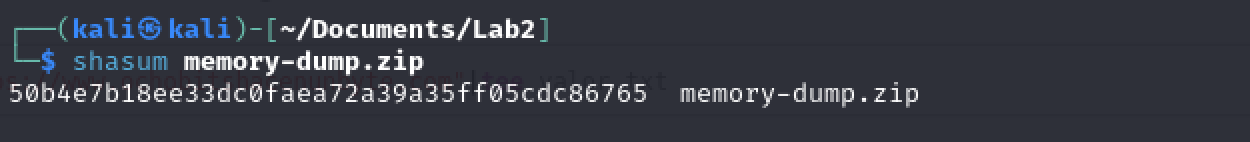
El CEO de la compañía ETOLEPSED S.A. se ha puesto en contacto con usted para realizar un análisis forense sobre un incidente que ha ocurrido recientemente. Uno de sus empleados habría recibido un correo electrónico de un compañero de trabajo con un PDF adjunto. Si bien al abrirlo no le pareció notar nada extraño, más tarde descubrió cierta actividad inusual en su cuenta bancaria. La compañía ETOLEPSED S.A. ha logrado obtener una imagen de la memoria RAM de la máquina del empleado afectado y se sospecha que podría estar infectada con algún tipo de malware. Le solicitan a usted que analice la evidencia e informe sobre cualquier actividad sospechosa encontrada.

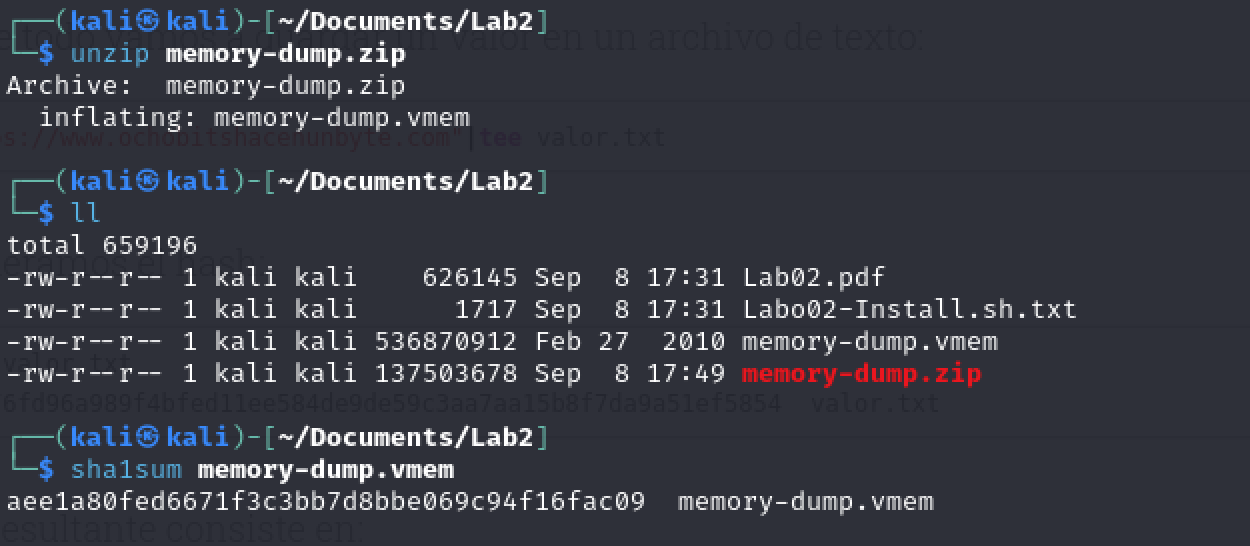
Dicha evidencia se corresponde con una captura de la memoria RAM de la computadora afectada.

1. Validar el hash de la evidencia. Continuar con las siguientes preguntas sólo en caso que la evidencia se encuentre validada.



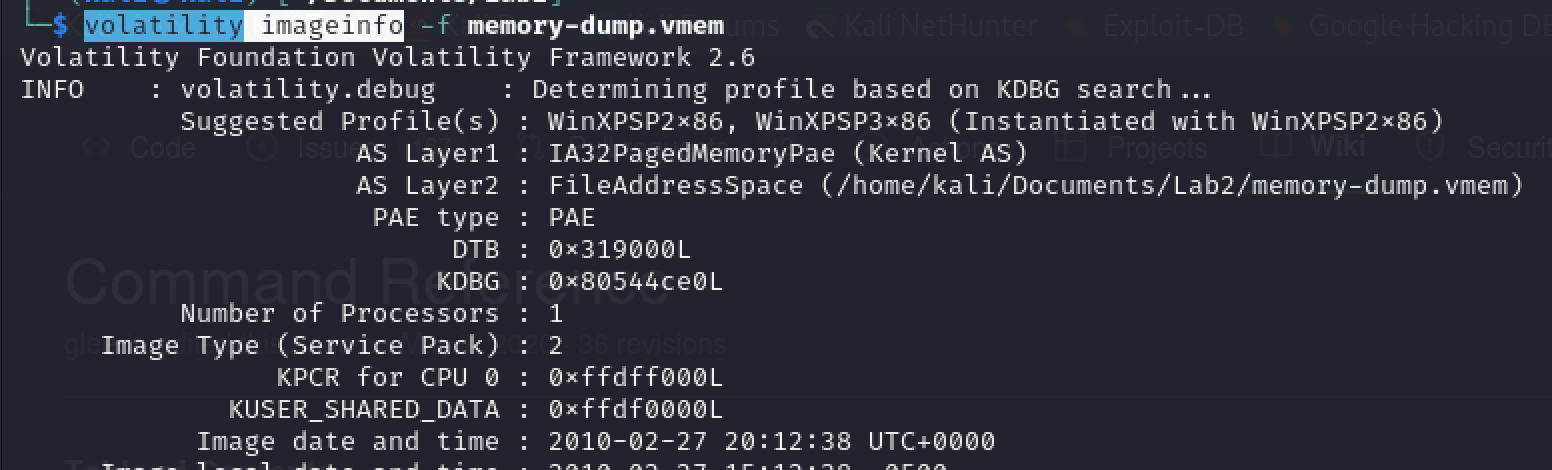
Solución: Se validó el hash de las evidencias mediante la terminal del Kali linux. El resultado fue que la evidencia descargada es consistente con la relevada. Dejemos la prueba de la evaluación a continuación:

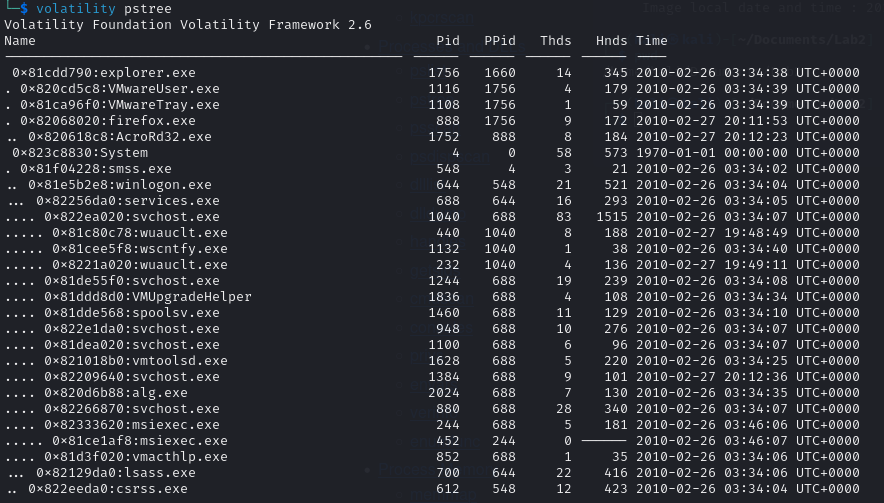




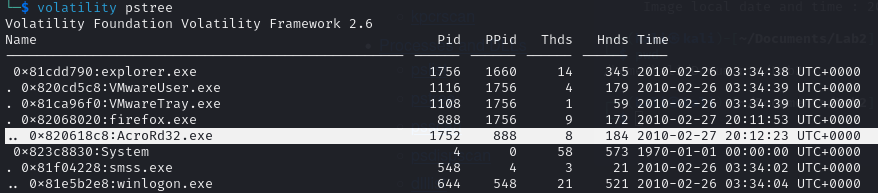
1. Enumere los procesos que se estaban ejecutando en la máquina de la víctima. Según lo expresado por el empleado, ¿Qué proceso debió haber sido el responsable del exploit inicial?

Solución: Para poder enumerar los procesos debemos:

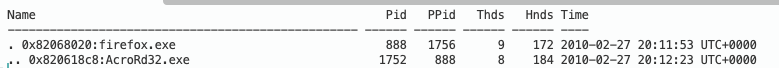
* 1. en primer lugar, determinar mediante "volatility imageinfo", al principio de un análisis de memoria forense y determinar el tipo de sistema operativo, la versión y otros parámetros esenciales que se utilizarán en análisis posteriores con Volatility.Informándonos los profiles y KDBG ("Kernel Debugger Block," que es una estructura de datos utilizada en el análisis de memoria forense para obtener información relevante sobre el kernel del sistema operativo) a ser usados.
  2. Vamos a usar el comando "volatility pstree" es un comando útil en Volatility para generar un árbol de procesos a partir de una imagen de memoria. Este árbol de procesos muestra la relación entre los procesos en ejecución en el sistema en el momento en que se capturó la imagen de memoria, cual es solicitado en el punto.



Completado la respuesta del punto 2, según manifiesta el empleado, *“habría recibido un correo electrónico de un compañero de trabajo con un PDF adjunto”*, podemos sospechar del proceso de Firefox.exe [Pid 888], quien dispara la ejecución del sub proceso de Acrobat Reader (AcroRd32.exe [Pid 1752], seleccionado en la imagen a continuación).



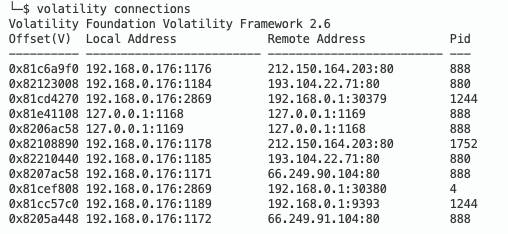
Detalle procesos sospechosos:



1. Enumere los puertos de red que estaban abiertos en la máquina de la víctima durante la infección. ¿Hay algún proceso sospechoso que tenga sockets abiertos?

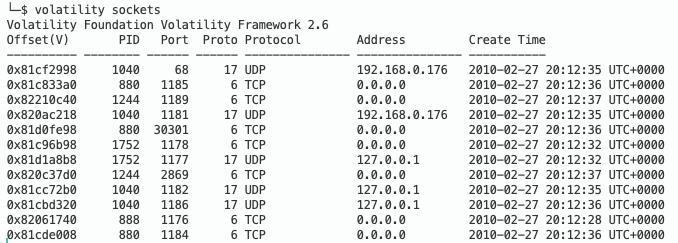
Solución: recuperamos las conexiones activas durante la captura de memoria al igual que los sockets mediante los comandos:

* volatility connections



* volatility sockets

Aplicamos una limpieza de la información de los sockets para quedarnos únicamente con las fechas y horas que corresponde al momento que se iniciaron los procesos sospechosos (después de abrirse el Acrobat Reader). Y tenemos a continuación los puertos de red abiertos.



Adicionalmente a los PID identificados [1752: Acrobat y 888: Firefox] tenemos 3 procesos sospechosos que corresponden a procesos svchost.exe y empezaron a abrir puertos con los siguientes process Id:

* 1040:
* 880
* 1244

“svchost.exe” significa "Service Host," que es un componente fundamental en el sistema operativo Windows. Su función principal es actuar como un proceso contenedor para alojar y ejecutar servicios de red de Windows en este caso. Sin embargo, en raras ocasiones, los ciberdelincuentes pueden usar nombres similares para camuflar malware. Por lo tanto, es importante asegurarse de que svchost.exe se ejecute desde la ubicación correcta en el sistema (normalmente, la carpeta C:\Windows\System32) y que no haya alteraciones sospechosas en su comportamiento.

1. ¿Existen URLs en la memoria? En caso afirmativo enumere las URLs sospechosas que encuentre en el proceso.

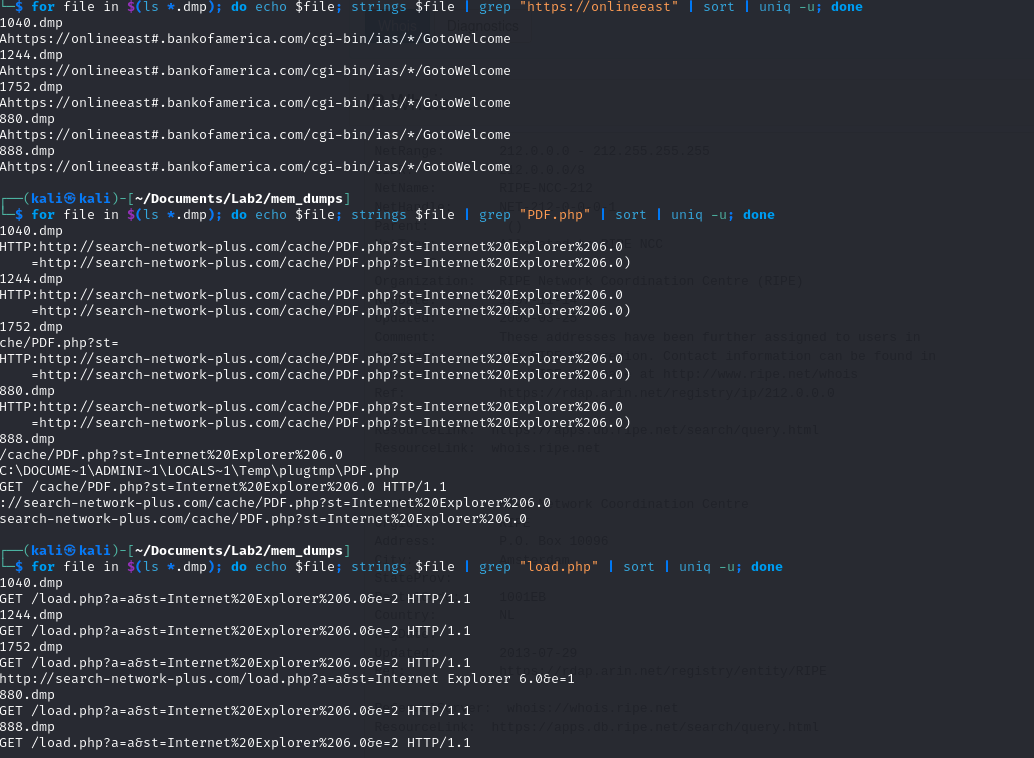
Si, existen una variedad de URLS en memoria que obtenemos con el comando “strings” que aplicamos a los dumps de memoria específicos de los procesos sospechosos.

Los siguientes URLS sospechosos se destacan:

* [https://onlineeast#.bankofamerica.com/cgi-bin/ias/\*/GotoWelcome](https://onlineeast#.bankofamerica.com/cgi-bin/ias/*/GotoWelcome)
* <http://search-network-plus.com/cache/PDF.php?st=Internet%20Explorer%206.0>
* http://search-network-plus.com/load.php?a=a&st=Internet Explorer 6.0&e=1

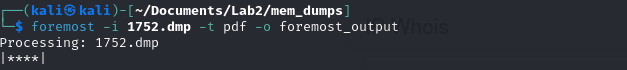
1. ¿Existen otros procesos que contengan URLs que puedan indicar actividad en un Banco? Si es así, ¿cuáles son estos procesos y cuáles son las URLs?

Observamos que las urls obtenidas en el punto 4. aparecen el dump de memoria de los demás procesos sospechosos.

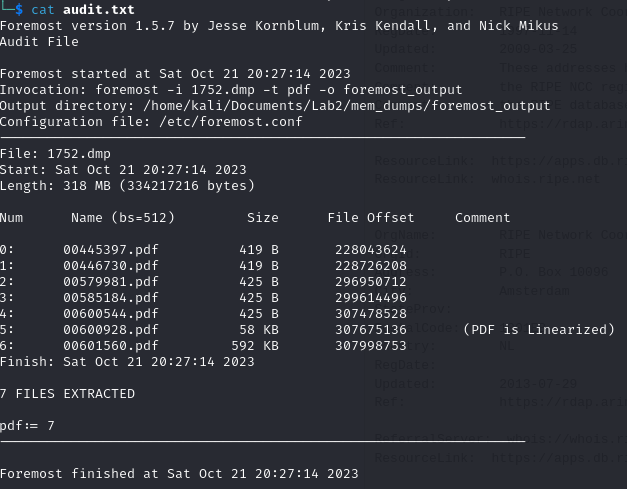


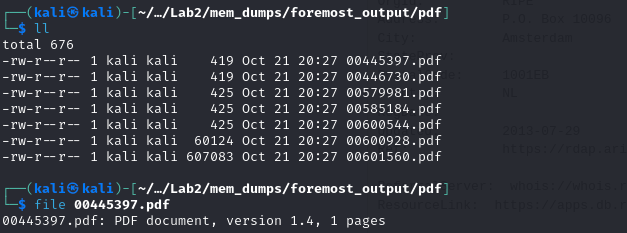
1. Verifique si el proceso sospechoso tiene archivos embebidos. En caso afirmativo intente extraerlos.

El proceso sospecho por excelencia es el AcroRd32.exe Pid [1752], utilizaremos la aplicación Foremost para realizar data carving.



Observamos que el proceso tiene archivos embebidos y se realiza la extracción mediante la aplicación. Resultados:

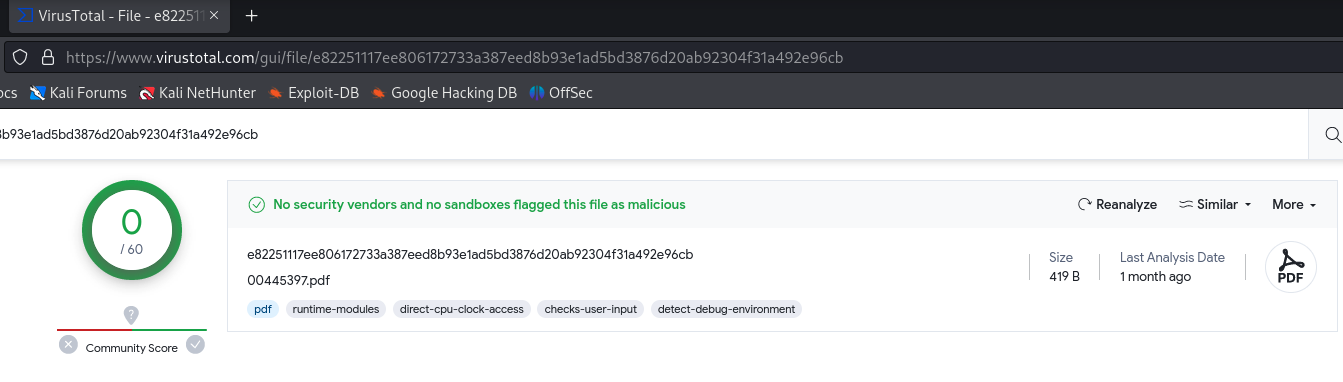




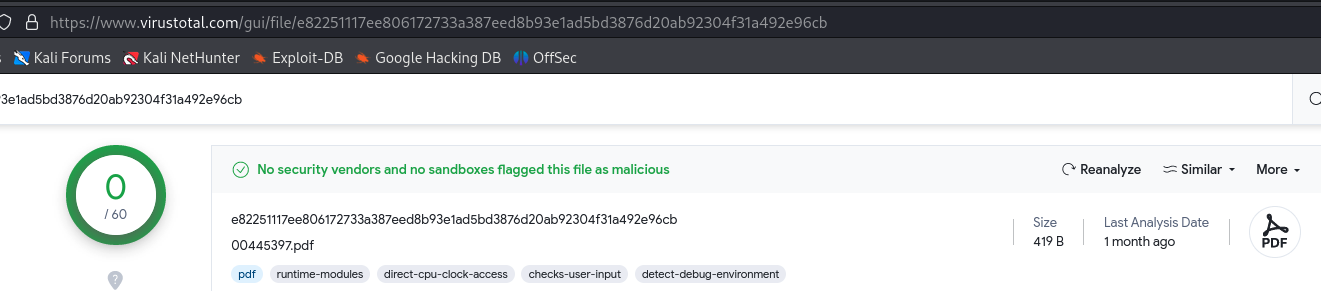
1. Si se extrajo un archivo del proceso inicial, ¿qué técnica utilizaron estos archivos para efectuar el exploit?

Solución: Revisamos uno por uno los archivos obtenidos en el punto 6, extraído del proceso de Acrobat con “Virus Total” subiéndolos a la url: https://www.virustotal.com/gui/home/upload

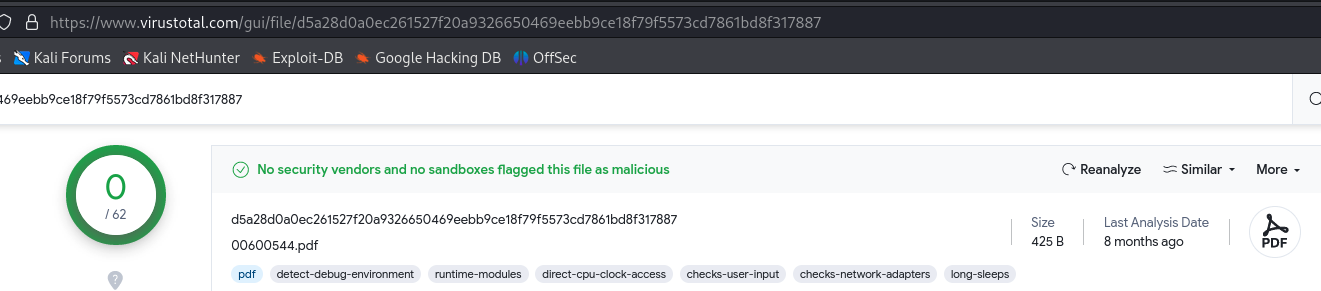
* 1. 00445397.pdf



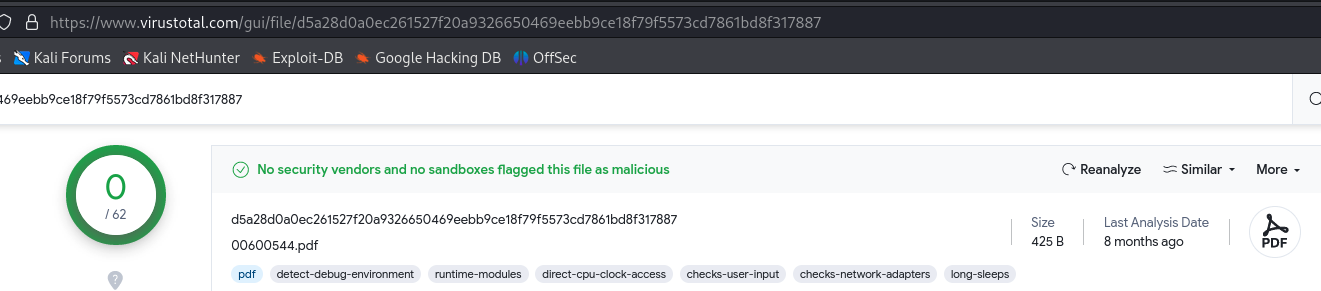
* 1. 00446730.pdf

f

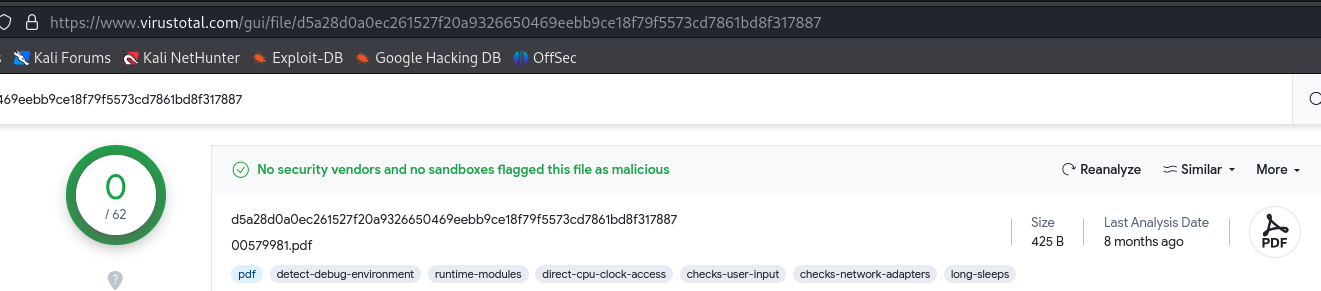
* 1. 00579981.pdf



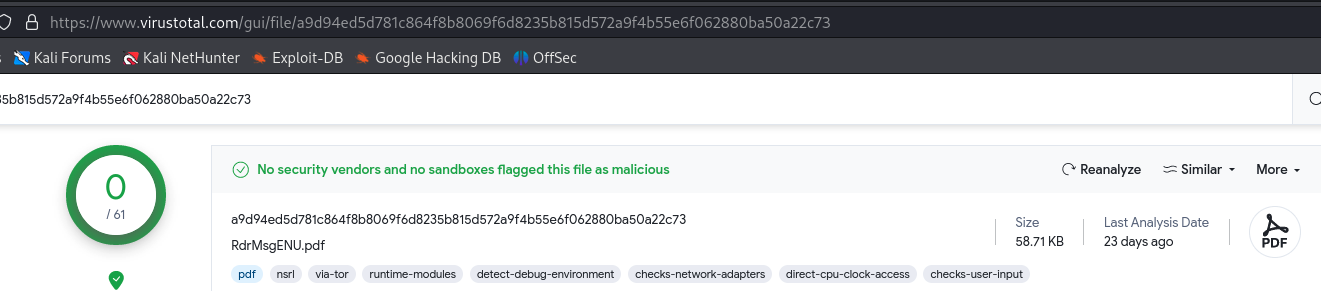
* 1. 00585184.pdf



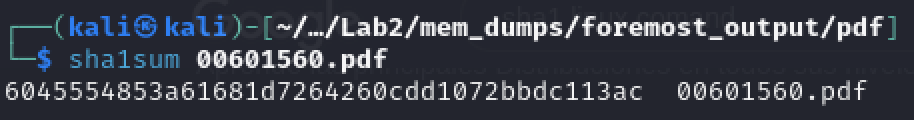
* 1. 00600544.pdf



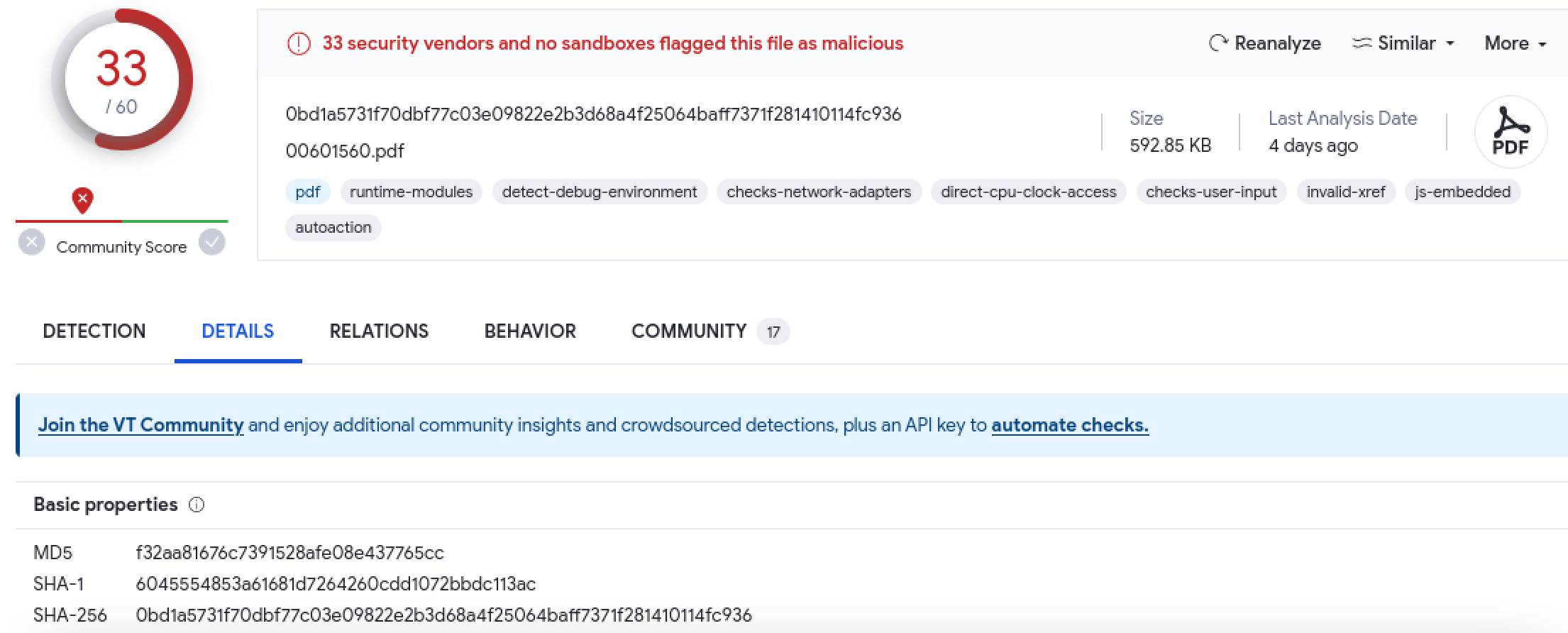
* 1. 00600928.pdf



* 1. 00601560.pdf

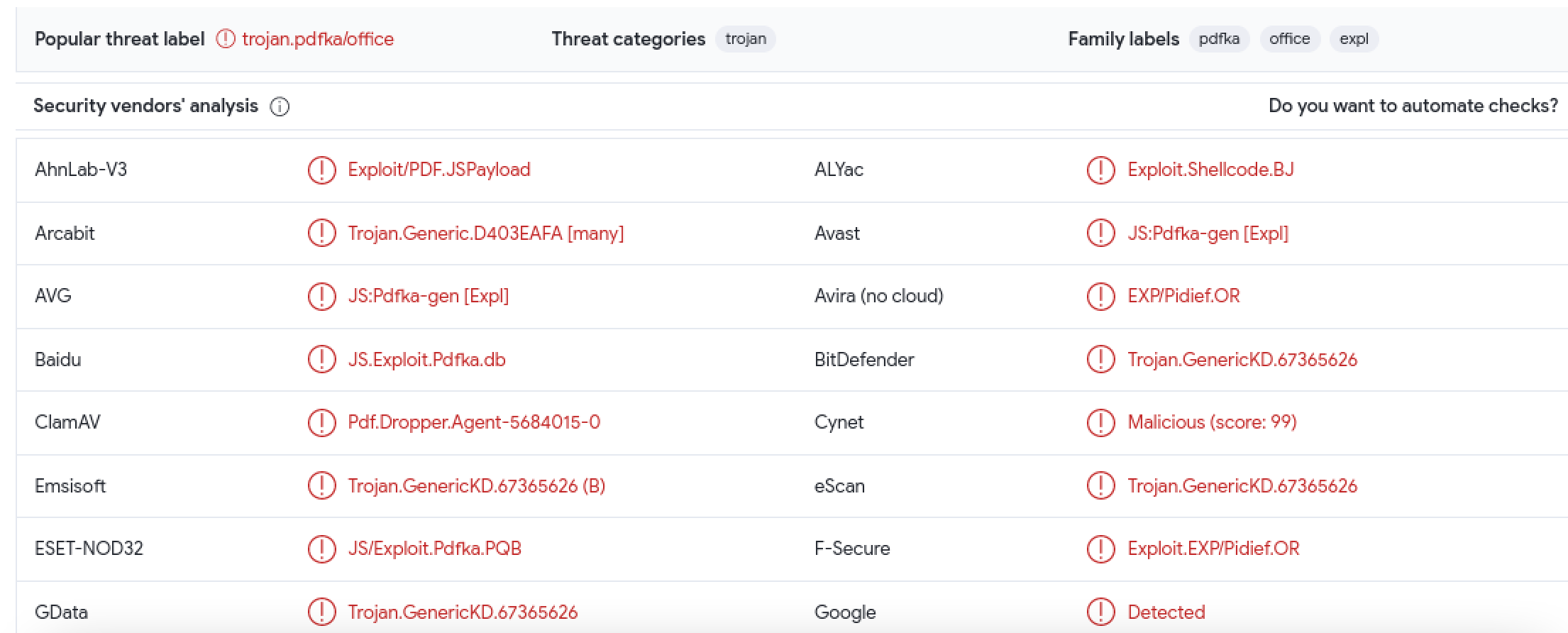


Este ultimo archivo pdf analizado por Virus Total hace match en el hash de archivo anteriormente obtenido:



Evidentemente es el archivo malicioso posiblemente del tipo:

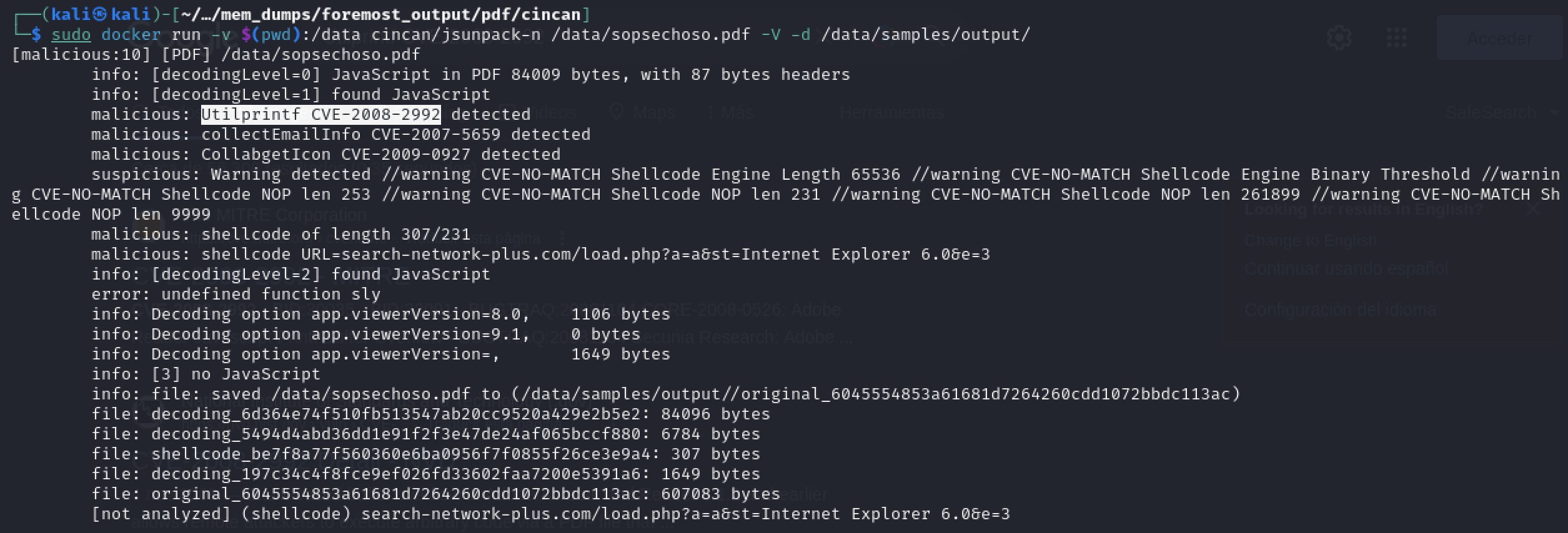
*Popular threat label > trojan.name/pdfka*



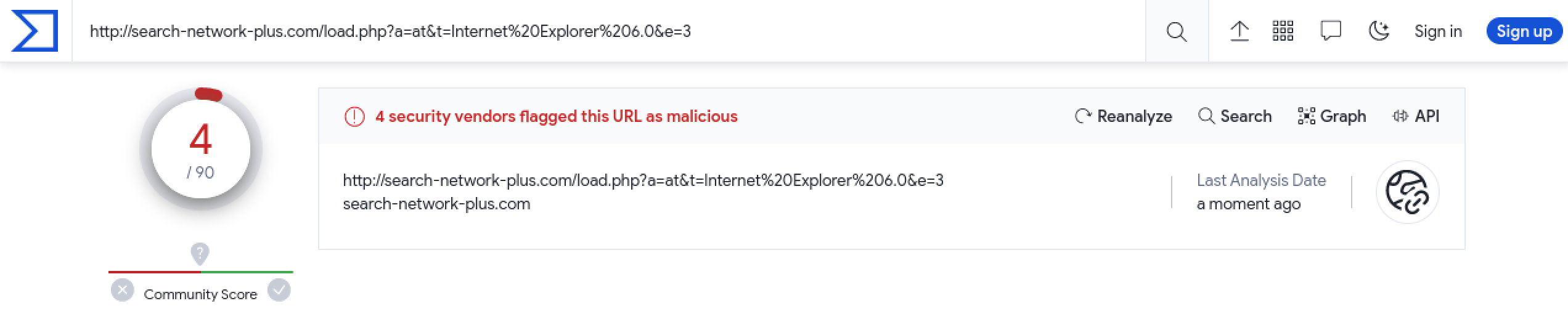
Finalmente analizando en profundidad el archivo pdf malicioso mediante la herramienta Jsunpack, concluimos que mediante un JavaScript embebido y ofuscado, luego de desofuscarlo, ejecutaba otro JavaScript que intenta y logra explotar, 3 vulnerabilidades de una versión del adobe Acrobat Reader: *Utilprintf CVE-2008-2992, collectEmailInfo CVE-2007-5659, CollabgetIcon CVE-2009-0927*, ejecutando un load.php a un sitio web (*URL=search-network-plus.com/load.php?a=a&st=Internet Explorer 6.0&e=3*), para descargarse un downloader.



Observamos a continuación la evidencia encontrada haciendo mención del JavaScript encontrado y el sito de referencia y la respuesta de Virus Total al analizar la url.



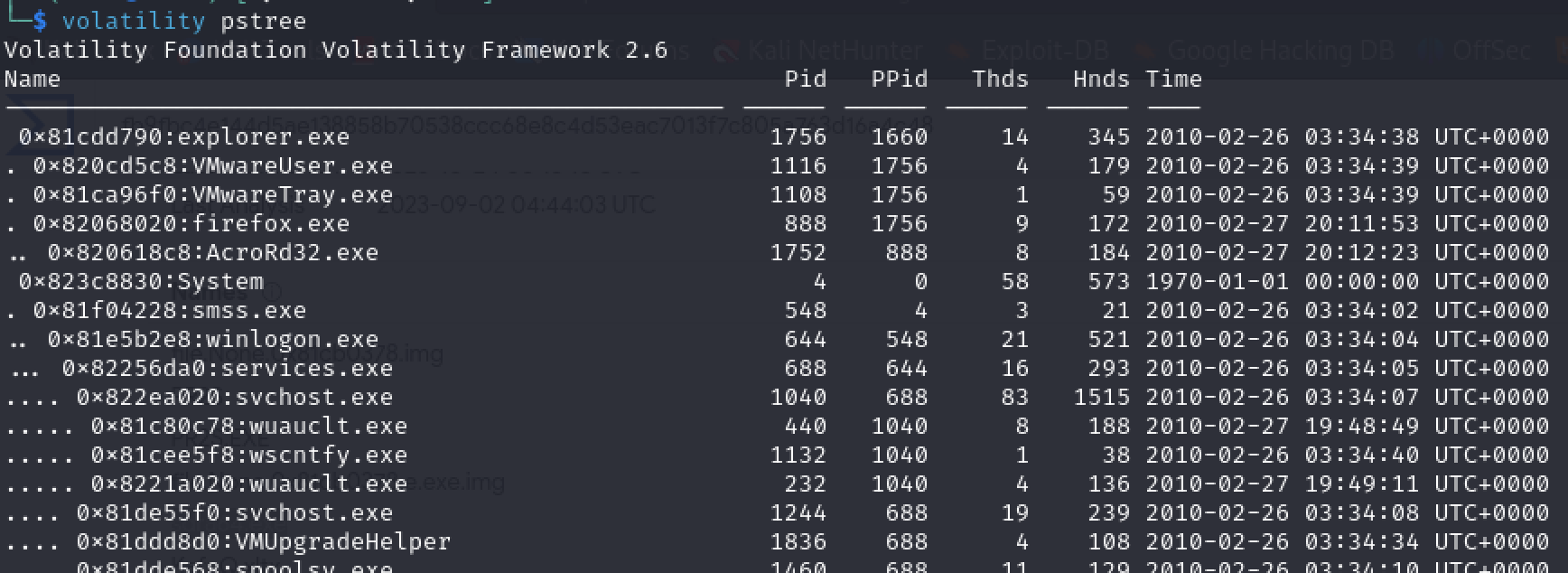




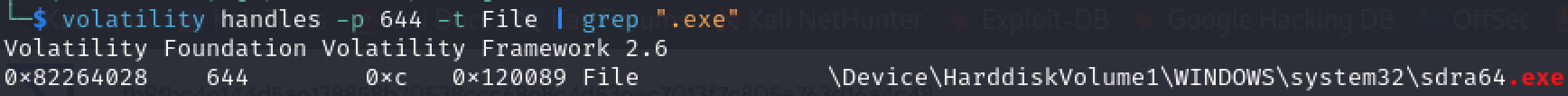


1. Enumere los archivos sospechosos que fueron cargados por cualquier proceso en la máquina de la víctima. A partir de esta información, ¿cuál fue el payload del exploit que afectó a la cuenta bancaria de la víctima?

Solución: Con el comando "volatility handles", proporciona información sobre los handles que los procesos tienen abiertos en la imagen de memoria, lo que puede ser útil para comprender cómo los procesos interactúan con recursos del sistema y para investigar la actividad del sistema en un momento específico. Si puntualmente analizamos el proceso winlogin.exe [Pid 644].

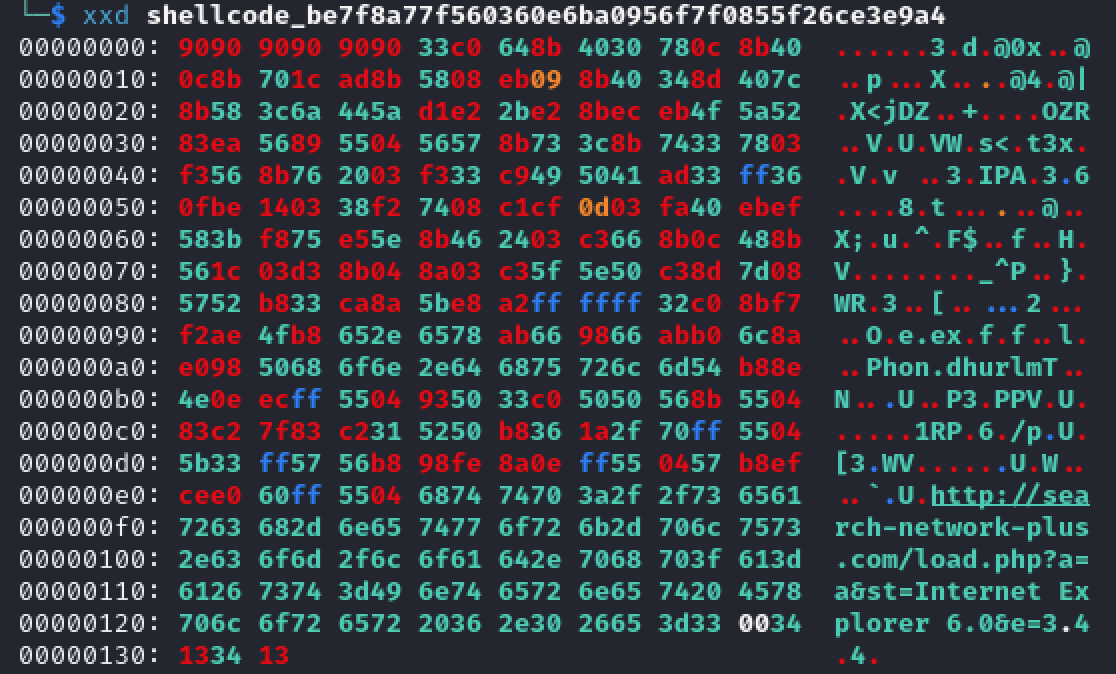




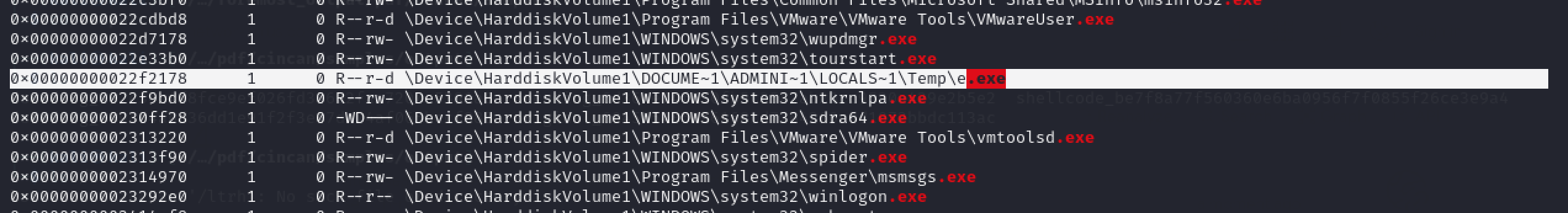


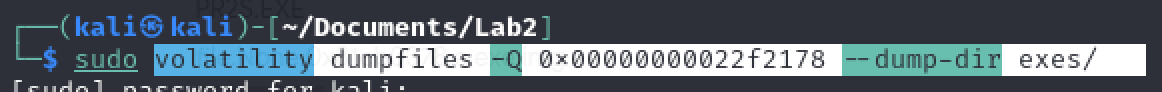
Tiene el archive sdra64.exe que resulta ser un archivo sospecho cargado en la maquina como parte de un mecanismo de persistencia.

El PAYLOAD que afecto la cuenta es el siguiente, que provoca la descarga del malaware:

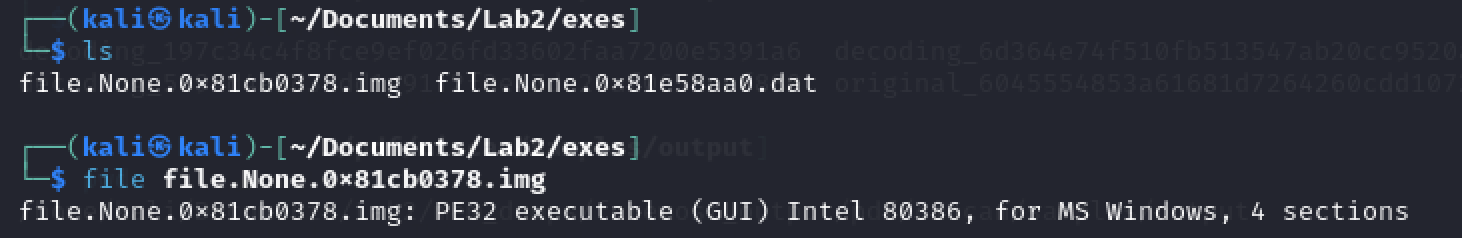


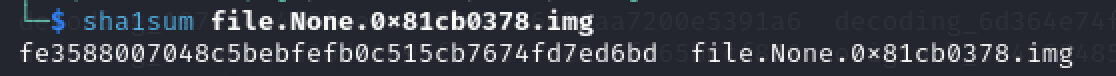
1. Si se pueden extraer archivos sospechosos de algún proceso inyectado, ¿algún producto antivirus detecta el ejecutable sospechoso? ¿Cuál es el resultado general de los productos antivirus?

Solución: Si realizamos un escaneo de archivo .exe de la memoria, nos encontramos con el siguiente archivo sospechoso: *Device\HarddiskVolume1\DOCUME~1\ADMINI~1\LOCALS~1\Temp\e.exe*

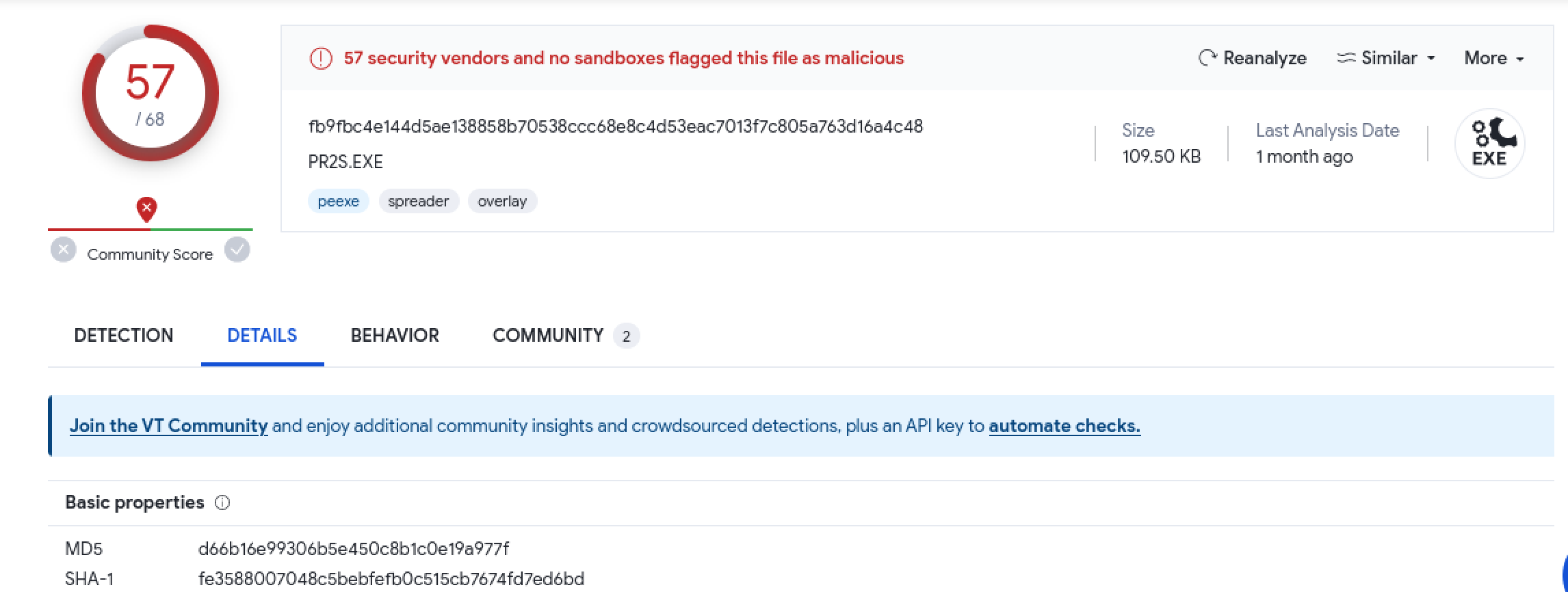
Haciendo un dumpfiles con Volatitlity del archivo, mediande su dirección física terminando en f2178,

nos damos cuenta que ese archivo extraído es un ejecutable:

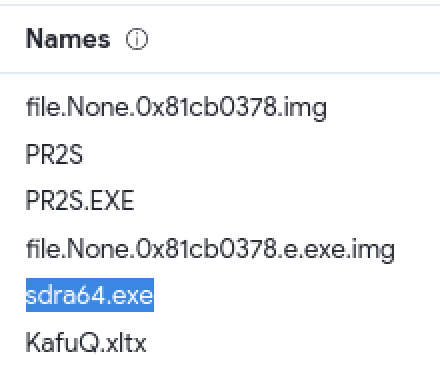


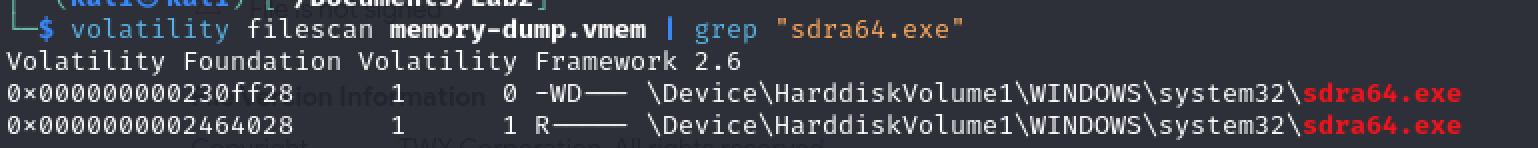


Analizando el archivo con “Virus Total” descubrimos que es malicioso:



Conocido también como “sdra64.exe” entre otro.



Y que resulta también esta en otro archivo en ejecución en la memoria. 

El archivo "sdra64.exe" a menudo se asocia con ZeuS, ya que es un nombre de archivo que se ha utilizado en el pasado en relación con esta amenaza. ZeuS a menudo se infiltra en sistemas y se ejecuta con nombres de archivo aparentemente aleatorios o que pueden parecer legítimos para evadir la detección.

ZeuS (también conocido como Zbot) es un troyano bancario altamente peligroso y ampliamente conocido que se ha utilizado para robar información financiera y credenciales de acceso a servicios en línea. Este troyano se ha utilizado en numerosas campañas de ciberataques a lo largo de los años y coincide con la actividad inusual reportada por el empleado en su cuenta bancaria luego de ejecutar el archivo adjunto PDF.