Universidad de Buenos Aires
Facultades de Ciencias Económicas,
Ciencias Exactas y Naturales e Ingeniería

Maestría en Seguridad Informática

Tesis de Maestría

Análisis Forense de Fugas de Información en Ambientes Corporativos

Autor: Marcelo Daniel Bovo

Director de la Tesis: Ing. Hugo Pagola

Fecha de Presentación 26/08/2021

Cohorte 2010

Declaración Jurada de origen de los contenidos

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Tesis vigente y que se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual.

Marcelo Daniel Bovo

FIRMADO

Índice

Agradecimientos

A Victoria y Agustín, que me apoyan incondicionalmente y perdonan cuando mi pasión por la ciberseguridad me quita tiempo para compartir con ellos.

A mis padres, que me inculcaron los valores que hoy me guían de la manera más difícil: con el ejemplo.

Introducción

La información constituye uno de los principales activos de las empresas y como tal se encuentra expuesta a múltiples amenazas que ponen en riesgo su integridad, disponibilidad y confidencialidad.

Tradicionalmente los esfuerzos de las áreas corporativas de seguridad informática o sus equivalentes se centraban en proteger a la información mientras permanecía en los sistemas de la compañía. Esta tarea no resulta para nada trivial y demanda una evolución constante del sistema de control implementado a medida que las amenazas crecen en complejidad y variedad.

La fuga de información en una empresa siempre ha existido como riesgo; listas de precios y clientes, especificaciones técnicas o cualquier otro tipo de documentación interna han sido codiciadas por la competencia y protegidas con mayor o menor éxito. Con la completa digitalización de la información y la aparición de medios masivos de almacenamiento de gran tamaño y conexiones a Internet para todos los empleados el riesgo ha crecido exponencialmente.

Ya no se requiere el acceso físico a expedientes en formato físico o la generación de copias en papel que eran actividades mucho más evidentes y riesgosas. Además, la realización de estas tareas implicaba que quién estaba copiando esta información lo estaba haciendo de una manera consciente y razonada. Hoy en día, con el crecimiento de la información generada por una compañía, un empleado puede incluso tener dificultades para diferenciar qué es información confidencial o propietaria y qué no lo es. Es por eso que es fundamental para toda compañía moderna implementar un plan de prevención de fuga de la información basado fuertemente en un proceso continuo de identificación y clasificación de la información.

Una vez que se ha implementado este proceso de identificación y clasificación de la información y existe un marco normativo que lo respalda puede pensarse en encarar un proceso de prevención de fuga de la información. En otras palabras, es imposible pretender proteger aquello que no se ha identificado y rotulado como información confidencial.

Ahora bien, una vez que este proceso está en marcha y existen controles y alertas para detectar actividades con información confidencial: ¿Cómo analizamos si realmente alguien extrajo documentación sensible fuera de la compañía?

Todo análisis forense pretende obtener la evidencia necesaria que permita generar una cronología de cómo sucedieron los hechos a partir del análisis de las acciones realizadas en los sistemas con los que interactuó la persona que extrajo la información. Lo ideal es poder determinar una serie de acciones realizadas enunciada de la manera menos técnica posible de manera que las personas sin conocimientos forenses puedan entender qué sucedió. Por supuesto estas afirmaciones deberán estar respaldada técnicamente por un análisis forense detallado que se anexa a las conclusiones y que puede ser reproducido por cualquiera de las partes interesadas.

Desde ya hace muchos años, no existe tal cosa como un análisis forense completo. Es decir, no se analizan todos los artefactos forenses e información presentes en el sistema sino sólo aquellos que nos permiten encontrar las respuestas a las preguntas surgidas de la investigación que se está llevando a cabo. Ahora bien, en la mayoría de los casos las preguntas varían de un análisis a otro, con una excepción: la de los análisis de fuga de información. En ellas las preguntas son siempre las mismas:

- ¿Se extrajo información corporativa? ¿Cuál?
- ¿Desde qué repositorios corporativos se copió?
- ¿Qué métodos se utilizaron para su extracción?

Es por eso que resulta posible automatizar la extracción de los artefactos forenses y su pre-procesamiento para este tipo de casos.

El objetivo de este trabajo es identificar los artefactos forenses disponibles en un equipo basado en Windows 10 que nos dan indicios de extracción de información confidencial, su estructura, cómo se analizan y cómo se correlacionan con las actividades realizadas por el usuario de ese equipo.

Para ello se analizarán las posibles formas de extraer información desde un equipo ubicado en una red interna corporativa y, a partir de esa información,

se identificarán que artefactos forenses quedan en el equipo como resultado de esas acciones.

Además, se presentará un método para automatizar la extracción y el procesamiento de estos artefactos ante la necesidad de realizar este tipo de revisiones en forma repetitiva.

Palabras clave: análisis forense, fuga de información, automatización, extracción forense, data leakage.

Fuga de Información en Ambientes Corporativos

Existen múltiples maneras en las que la información puede dejar la empresa de forma no autorizada. Algunas de ellas requieren la intencionalidad del usuario y en otras el usuario es simplemente una víctima. Identificando las vías más comunes analizaremos si es posible determinar su ocurrencia a partir del análisis de un subconjunto limitado de los artefactos forenses disponibles en el equipo del usuario. Para cada artefacto forense identificado, se analizarán en los capítulos siguientes qué información contiene y cómo puede ser decodificada a fin de reconstruir total o parcialmente cómo se fugó determinada pieza de información.[1]

- **Correo electrónico y archivos adjuntos**: La vía más común de intercambio de información en ambientes corporativos ha sido tradicionalmente el correo electrónico. Los usuarios corporativos utilizan en general tanto servidores de correo corporativos como Microsoft Exchange como servicios en la nube tales como Office365. La información puede intercambiarse con destinatarios externos legítimos, con terceros no autorizados o incluso puede ser enviada por el usuario a una casilla externa propia en un servicio como Gmail, Outlook o iCloud.
- **Dispositivos de cómputo portables sin encripción**: Los teléfonos móviles (personales o corporativos) y otros dispositivos personales de cómputo como notebooks a menudo se configuran sin encripción de disco y con solo una password para restringir el acceso. Este método de protección puede eludirse de forma trivial si se tiene acceso físico al dispositivo. Debido a su portabilidad, es frecuente que el usuario los pierda o se los roben. Si un dispositivo de este tipo es comprometido

además de información confidencial, un atacante podría obtener credenciales de acceso válidas para la red corporativa.

- **Servicios de almacenamiento en la nube**: Los empleados a menudo transfieren archivos grandes a través de servicios de almacenamiento o colaboración en la nube. Lo cual expone a la información corporativa a un entorno no controlado por la compañía y bajo unos términos y condiciones que no se conocen. Además, pueden ser utilizados como un medio de extracción de información ya que, al ser un servicio externo, los registros de los archivos transferidos no son tan evidentes.
- **Dispositivos de almacenamiento removibles**: Almacenar información confidencial en un dispositivo de almacenamiento removible constituye no sólo un medio utilizado para extraer información corporativa en forma subrepticia, sino que también exponen a la información cuando son utilizados en forma legítima. Este tipo de dispositivos puede ser fácilmente extraviado o sustraído y al no contar con medios de protección para la información que contienen
- **Controles de acceso inadecuados o vulnerabilidades explotables por un atacante**: Si bien este tipo de vulnerabilidades están siendo explotadas en forma cada vez más frecuente para sustraer información corporativa, su análisis queda fuera del alcance de este trabajo. Existe toda una infraestructura de detección de ataques a servidores e infraestructura de comunicaciones que puede utilizarse para prevenir y detectar ataques de este tipo y cuyo objetivo es prevenir o contener este tipo de incidentes. Una vez que se ha producido, detectado y contenido un ataque de este tipo pueden aplicarse técnicas de análisis forense en estaciones de trabajo, servidores y equipamiento de telecomunicaciones a fin de entender la metodología de intrusión y la extensión del compromiso. Este tipo de análisis, si bien comparte los mismos

principios que el análisis forense que desarrollaremos en este trabajo, utiliza metodologías y artefactos forenses diferentes.

Además de los vectores de exfiltración de datos mencionados, un aspecto fundamental a considerar dentro de las investigaciones de fuga de información es la oportunidad. Es decir, el acceso o recopilación de determinada información puede tener sentido en determinada etapa de la relación laboral del empleado y no tenerlo en otra. Por ejemplo, si se detecta que un empleado organiza, descarga y recopila manuales, planos y procedimientos operativos el día previo a su desvinculación, claramente se trata de un intento de apropiarse de información de la empresa. Si por el contrario esta acción se realiza en plena actividad laboral, puede ser justificable dentro de las tareas que le son asignadas.

Identificación y preservación de la evidencia forense

Antes de realizar un análisis forense, es indispensable identificar y preservar los dispositivos de almacenamiento que contienen la evidencia forense que se pretende analizar.

El primer paso es la identificación. En lo que respecta al usuario, es crucial determinar qué dispositivos la empresa le ha asignado para su utilización. Hoy en día los equipos no se limitan a equipos de cómputo, sino que también incluyen teléfonos celulares, discos rígidos externos, pendrive. En el caso de los equipos de almacenamiento masivo, muchas veces sucede que han sido comprados en forma descentralizada y por lo tanto no existe un registro de su existencia, por lo cual generalmente se descubre su existencia luego de analizar un equipo al que fueron conectados.

Dentro del ámbito corporativo sólo es admisible analizar aquellos equipos con medios de almacenamiento masivo que son propiedad de la empresa y fueron asignados a un determinado usuario para la realización de sus actividades diarias. Aun así, dependiendo de las condiciones en las cuales se recopile la información del dispositivo y de la expectativa de privacidad que tenga el usuario con respecto a la información que almacene en esos dispositivos, la evidencia podrá ser o no considerada válida por un juez.

La realización de estas actividades en presencia de un notario o escribano dependerá principalmente de los usos y costumbres del país en donde se realice el procedimiento y de la utilización que se pretenda hacer de las conclusiones de la investigación. Por ejemplo, en países como Argentina y México es usual utilizar los servicios de un notario para documentar el proceso de toma de imagen forense, mientras que en Brasil, Colombia y Estados Unidos de América en general basta con documentar adecuadamente el proceso seguido. Por otra parte, si las conclusiones de la investigación van a utilizarse dentro del ámbito de la empresa como entrada para mejorar el entorno

de control de la información confidencial, entonces tampoco sería necesario labrar un acta notarial del proceso.[2]

En Argentina, la Procuración General de la Nación emitió el documento “Guía de obtención, preservación y tratamiento de evidencia digital”, la cual si bien está orientada a fuerzas de seguridad contiene principios básicos de aplicación universal para toda investigación forense.[3]

Si no participa un escribano del proceso, al menos es recomendable la confección de un formulario de entrega-recepción. En el mismo, el empleado que tiene el equipo asignado declara que:

- Ese es el equipo que la compañía le ha asignado para el desarrollo de sus tareas
- Que acepta que al equipo se le realice una imagen forense (bit a bit).
- Que el equipo se le devolvió en determinada fecha y hora.

Este documento es una versión simplificada de una cadena de custodia y permite darle un mínimo de formalidad al proceso.

Modelado conductual

El principio de intercambio de Locard, postulado por el científico francés Edmund Locard, dice que “Todo contacto deja una huella”. En el mundo físico esto significa que cuando dos objetos entran en contacto parte del material de uno es transmitido al otro y viceversa. Esta es una de las bases de la ciencia forense. [4]

Es decir, que cada acción que se realice dentro de la computadora a fin de extraer información almacenada en la misma o en otros servidores o aplicaciones de la compañía generará uno o más artefactos forenses que permitirán determinar con mayor o menor grado de certeza que dicha acción se ha realizado. Dependiendo de la operación realizada, el sistema operativo utilizado y las condiciones particulares del servidor o estación de trabajo involucrados será posible obtener más o menos información que nos permita identificar qué acciones se realizaron.

Si bien muchas operaciones que se realizan en un equipo no tienen considerado un registro de actividades, la mayoría de las operaciones realizadas en una aplicación o en el mismo sistema operativo generan artefactos forenses como un efecto colateral. Estos artefactos forenses a menudo no se encuentran documentados por lo que es normal que la comunidad forense o los fabricantes de productos para análisis forenses investiguen y lleguen a conclusiones acerca de su utilidad para comprobar que determinada acción se realizó en un equipo. Debido a que son efectos colaterales de las operaciones que realizan el sistema operativo o las aplicaciones, estos artefactos pueden cambiar de una versión a otra de Sistema Operativo o incluso entre actualizaciones menores, por lo cual se debe considerar con qué versión de Sistema Operativo o aplicación se está trabajando a fin de evitar llegar a conclusiones erróneas.

La manera más lógica de determinar qué artefactos forenses se deben analizar en este tipo de investigaciones consiste en seguir los posibles caminos que pudo haber seguido la información

corporativa desde su recolección o generación hasta su exfiltración y determinar qué artefactos forenses se generan con cada operación.

Al realizar un análisis forense de un equipo de cómputo asignado a un empleado de la compañía en general podremos identificar varias etapas diferenciadas en el proceso de extracción de información:

1. Obtención de la información desde los repositorios de la compañía: En la mayoría de los casos el colaborador no tiene en su equipo la información que desea extraer ya que se encuentra en los repositorios de información corporativos. Los mismos pueden ser CMS (“Content Management System”) como SharePoint, Wordpress, Joomla, Alfresco; Servidores con recursos compartidos vía SMB; ERPs (“Enterprise Resource Planning”) u otro sistema transaccional. Los accesos pueden haber sido otorgados en forma explícita y autorizada o bien la información puede ser accesible por un error en el manejo de los permisos del recurso.
2. Organización de la información: A menudo el empleado genera una estructura de carpetas en su equipo a fin de organizar la información a extraer. Durante este proceso abre los archivos a fin de verificar su contenido e incluso puede mover o copiar información que ya se encontraba en el equipo.
3. Extracción de información: En esta etapa es en donde se pone de manifiesto la complejidad que conlleva la implementación de un sistema de protección contra fuga de información, a menudo llamados “Data Leak Prevention” o DLP en la literatura. Las vías de fuga de información son múltiples y a veces difíciles de detectar. Pueden ir desde la copia en un disco externo USB o un celular, el envío de un correo con adjuntos a una casilla de correo personal hasta técnicas de más bajo volumen y difícil detección como la impresión de un archivo o una foto de la pantalla de un ordenador.

4. Borrado de evidencia: En ocasiones se intenta ocultar que se realizó una extracción de información mediante la utilización de programas de optimización del sistema como CCleaner, el borrado de los archivos copiados del equipo de trabajo del empleado o incluso la reinstalación del sistema operativo. Cada uno de estas operaciones es capaz de borrar evidencia con mayor o menor éxito, pero en todos los casos dejan artefactos forenses recuperables. En estos casos en conveniente recurrir a los registros de los sistemas origen de la información que se supone que se ha filtrado a fin de complementar la información de la que se dispone.

Ya que todo proceso de análisis cuenta con recursos y tiempo limitados, se sugiere comenzar la preservación y análisis por los dispositivos que a priori se supone que pueden contener mayor evidencia de las acciones realizadas. En general se comienza por la preservación de los equipos de cómputo y almacenamiento que la compañía le ha asignado al empleado. Es conveniente determinar, además, si el empleado ha cambiado de equipo recientemente y tratar de preservar el equipo anterior antes de que sea dado de baja o reasignado.

Si el caso lo requiere se podrán realizar imágenes forenses de los servidores y preservar los registros de los sistemas involucrados.

Entendiendo cuáles son las etapas de un proceso de fuga de información, podemos determinar qué artefactos forenses se generan en el sistema y cómo extraerlos, procesarlos y analizarlos. A partir de este modelo podemos generar una lista de los principales artefactos forenses a analizar para determinar si se extrajo información corporativa y cómo ocurrió.

La guía generada contiene una valoración para cada artefacto forense analizado de acuerdo a la posibilidad que tiene de aportar información relevante para la investigación. Es por eso que se sugiere al utilizar esta guía, evaluar primero los artefactos clasificados como de nivel 1, para luego continuar (de ser necesario) con los de nivel 2 y 3.

Artefactos forenses generados en cada etapa

Adquisición

Durante esta etapa el empleado recopila información de los repositorios corporativos a los que tiene acceso. En general en la OT los repositorios de información se encuentran en SharePoint (“on premises” o en la nube), File Servers y aplicaciones web de gestión. También pueden llegar a realizar bajadas de datos desde SAP, tal como el catálogo de clientes/proveedores.

Los principales aplicativos utilizados para acceder a estos datos son los navegadores Web y el explorador de Windows. Ambos dejan rastros forenses no sólo por si mismos sino a través del sistema operativo. En particular para el Internet Explorer/Edge y explorador de Windows la cantidad de artefactos generados es aún mayor debido a la fuerte integración de estas aplicaciones con el sistema operativo.

Acceso a aplicaciones Web (incluyendo SharePoint)

Hoy en día, los navegadores web se han convertido en una de las aplicaciones más utilizadas no sólo para acceder a páginas web sino porque la mayoría de las aplicaciones utiliza interfaces web. La conveniencia y flexibilidad de las interfaces web y el desarrollo de tecnologías web que permiten la generación de aplicaciones de una sola pantalla, han hecho que los clientes de aplicaciones casi hayan desaparecido del escritorio. Es por ello que el análisis de los artefactos forenses que generan los navegadores pueden darnos indicios de la información accedida por un usuario.

- Artefactos forenses de Navegadores: cookies, historial, archivos temporales y descargas.
- URLs ingresadas directamente por el usuario (TypedURLS)
- Logs de creación de archivos en el sistema de archivos de la estación de trabajo (especialmente en directorios críticos tales como: Descargas, Mis Documentos y Escritorio).

- Links
- MRU/Jumplists
- Últimos archivos abiertos mediante la suite Office.
- Archivos salvados o abiertos en Windows.
- Adjuntos de Outlook/Office.
- Archivos recibidos mediante Skype o Microsoft Teams.

Acceso a carpetas Compartidas en Servidores de Archivos

La forma más tradicional de compartir información en un entorno corporativo es a través de un file server con recursos compartidos a través de SMB. Los accesos se definen mediante la combinación de los accesos otorgados a ese recurso más los permisos que el usuario tenga a nivel del sistema de archivos en donde residan efectivamente los archivos.

Si bien es la opción más fácil de implementar, presenta múltiples problemas desde el punto de vista de la trazabilidad del acceso y cambios a los documentos. Por defecto los accesos a los sistemas de archivos en entorno Windows no están habilitados y los administradores rara vez cambian esa configuración ya que se supone que genera un volumen de registros muy alto (lo cual no es así si se realiza en forma correcta). Por lo tanto, es difícil que podamos obtener información relevante de los logs del file server salvo eventos de autenticación que no nos permitirán determinar a qué archivos se accedió.

En el equipo de cómputo, por el contrario, podremos encontrar múltiples artefactos forenses que nos permitirán determinar el acceso del usuario a determinado recurso compartido:

- Historial de Internet Explorer/Edge.
- Shellbags
- Logs del Sistema.
- Registry
- Links

- MRU/Jumplists

Organización

Aunque algunas veces este paso se suele obviar y directamente pasar los archivos desde los repositorios a un dispositivo USB externo, a menudo se ordena la información en una estructura de carpetas o directorios para luego copiarlo ordenado a un disco externo o un proveedor de almacenamiento Cloud. Todos los procesos de creación de archivos quedan registrados en el sistema, con lo cual se puede obtener una prueba más de la existencia de los archivos en el sistema de archivo del equipo asignado.

- Archivos Movidos/Creados
  - Master File Table/Log de Transacciones NTFS (archivos movidos/creados)
  - Aplicaciones utilizadas.
  - Activity Journal.
  - SRUM (System Resource Usage Monitor).

Extracción

En esta etapa, la información a extraer se copia efectivamente al medio de almacenamiento masivo que permitirá extraerlos de la compañía. Los medios pueden ser un pendrive, un disco externo, un celular, una memoria flash, un servidor externo en la nube, impresiones, etc.

1. Discos USB
   a. Artefactos USB: Registry, Archivos lnk, Setupapi.dev.log
   b. Logs del sistema.
2. Trabajos de impresión:
   a. Archivos temporales de impresión.

3. Almacenamiento en la nube[1]

Borrado de evidencia

Al finalizar la extracción de información, en ocasiones se trata de borrar la evidencia de la actividad o al menos hacerla menos obvia.

1. Archivos borrados
    a. Metadata del filesystem NTFS: Usnjrnl/$MFT
    b. Papelera de reciclaje
2. Utilización de herramientas de borrado/limpieza
    a. Shimcache/Prefetch

Con el modelado conductual de quiénes realizan una fuga de información y la determinación de los artefactos forenses a analizar en cada etapa, podemos diseñar un esquema de extracción y procesamiento de los artefactos forenses que nos permitan agilizar nuestro análisis y llegar a conclusiones relevantes en un lapso de tiempo más acotado.

[1] El análisis de artefactos forenses relacionados con el almacenamiento en Cloud está fuera del alcance de este trabajo ya que es un campo que recién está comenzando a tomar fuerza dentro del ámbito corporativo en el que trabajo. Sin embargo, la evidencia puede encontrarse en los logs de los browsers (para los accesos con interfaces web), los registros de ejecución de clientes “standalone” como los de Dropbox y la creación de archivos en las carpetas de sincronización. Estas últimas permiten mediante un proceso activo en el sistema, sincronizar bidireccionalmente toda la información que se coloque en ella con el almacenamiento Cloud y viceversa.

Extracción de los artefactos forenses

La extracción de artefactos forenses puede realizarse de varias maneras: La más común es a partir de una imagen forense del equipo a analizar, aunque actualmente está ganando popularidad la extracción directa de los artefactos forenses a analizar ya que permite acortar los tiempos.

Extracción desde imágenes forenses

Diseño del programa

Para poder realizar la extracción de artefactos forenses a partir de imágenes se desarrolló un utilitario específico a tal fin. Esta herramienta surgió de la necesidad de analizar información desde imágenes forenses que no se encontraban físicamente en el lugar de trabajo de los analistas.

Si bien ha habido una mejora importante en los últimos años, la transmisión de imágenes forenses a través de vínculos WAN sigue siendo prácticamente imposible debido su tamaño. A través del análisis de los artefactos forenses a analizar en un caso de fuga de información llegué a la conclusión de que era posible seleccionar un conjunto de artefactos forenses tales que permitieran un análisis completo del equipo y se pudieran transmitir por un enlace WAN. En el peor de los casos estos artefactos no superan los 4GB.

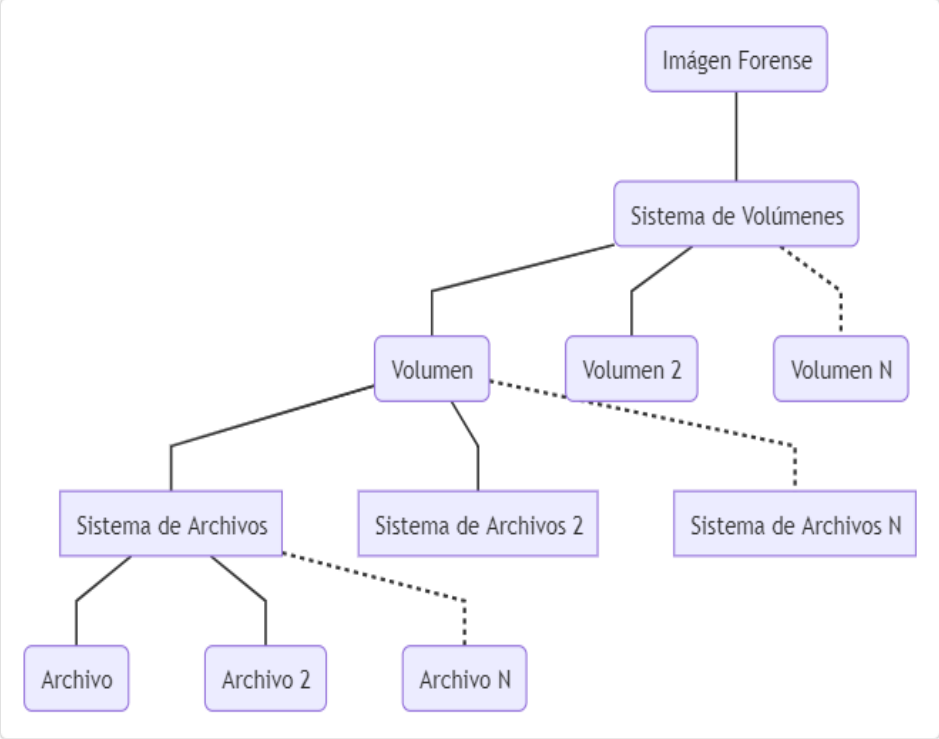
La herramienta a desarrollar debería cumplir con varios requisitos para adaptarse a las particularidades de sus potenciales usuarios:

- Utilizable por usuarios sin ningún conocimiento técnico.
- Utilizable en estaciones de trabajo en las que no se poseen privilegios administrativos.

- Soporte para encripción de disco completo (Bitlocker, Filevault) informando al usuario del ID del drive encriptado para gestionar el código de recovery correspondiente.
- Soporte para varios formatos de imágenes forenses e imágenes particionadas.
- Soporte para múltiples sistemas de archivos.

Luego de analizar las librerías disponibles en Python, se seleccionó dfVFS. dfVFS, o “Digital Forensics Virtual File System” por sus siglas en inglés, provee acceso read-only a sistemas de archivos almacenados en varios tipos de imágenes forenses. El objetivo de dfVFS es proveer una interfaz genérica para acceder a objetos de un sistema de archivos, por lo cual utiliza múltiples back-end que son los que efectivamente implementan el acceso a los diferentes formatos de almacenamiento de datos, volúmenes y sistemas de archivo [5].

Internamente, dfVFS se organiza como puede verse en la siguiente figura:



Cualquiera de los diferentes tipos de objetos modelados puede tener varios tipos y es implementado en el backend por la librería correspondiente.

- **Imagen Forense:** EWF (libewf), QCOW (libqcow), Raw Media (libsmraw), VHD (libvhdi), etc.
- **Tablas de Particiones:** Apple Partition Map (libtsk), GPT (libtsk), MBR (libtsk)
- **Sistemas de Volúmenes:** Apple File System (libfsapfs), Bitlocker Disk Encryption (libbde), FileVault Disk Encryption (libfvde), LVM Logical Volume Management (libvslvm), Volume Shadow Snapshots (VSS) (libvshadow).
- **Sistemas de Archivo**: NTFS (libntfs), ext{2,3,4} (pytsk), FAT (pytsk), HFS/HFS+ (pytsk), UFS (pytsk)

Si bien esto provee gran flexibilidad a la hora de programar trae aparejado un aumento en la complejidad del modelo de objetos en comparación a librerías más básicas como libtsk. Además, debido a que fue desarrollada por Joaquim Metz como base para otros programas como log2timeline la documentación es bastante escasa.

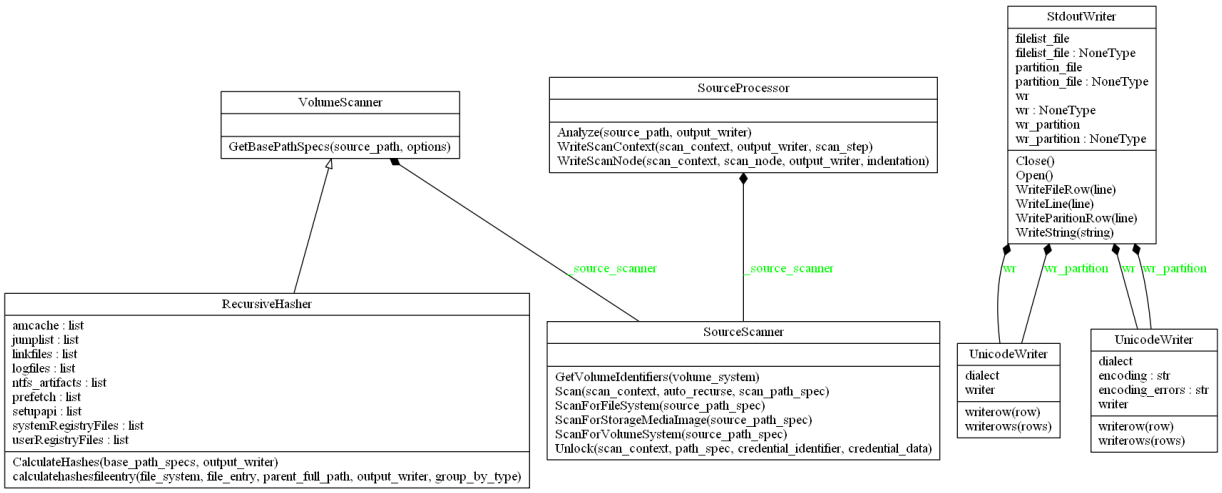
Este programa es una adaptación y mejora de dos ejemplos que vienen con la librería: source_analyzer.py y recursive_hasher.py [6]

Con respecto a los requerimientos iniciales, fueron implementados de la siguiente manera

- Utilizable por usuarios sin ningún conocimiento técnico: Si bien el programa es de línea de comandos y soporta argumentos de entrada para mayor flexibilidad, la configuración por defecto es la que se utiliza normalmente y el usuario sólo tiene que ejecutar el programa arrastrando el archivo de imagen forense al archivo ejecutable de la aplicación. El resultado es un archivo denominado files.zip que se encuentra en el mismo directorio en donde está la imagen forense.

- Utilizable en estaciones de trabajo en las que no se poseen privilegios administrativos: El programa no requiere privilegios administrativos para su uso ya que se compone de un solo archivo ejecutable (mediante py2exe) que no requiere instalación y no monta las imágenes forenses como dispositivos en Windows como hacen Arsenal Image Mounter o FTK Imager.
- Soporte para encripción de disco completo (Bitlocker, Filevault) informando al usuario del ID del drive encriptado para gestionar el código de recovery correspondiente: El progama soporta encripción de disco completo a través de dfVFS y lista los mecanismos de desencripción soportados por la imagen y su ID correspondiente para que el usuario pueda solicitar la clave de desencripción al analista forense.
- Soporte para varios formatos de imágenes forenses e imágenes particionadas y múltiples sistemas de archivos: Al utilizar dfVFS el soporte de formatos y sistemas de archivo del programa es muy amplio, además como se distribuye en un archivo exe *standalone* no es necesario que el usuario instale las librerías que soportan cada uno de estos formatos.
- Extracción de artefactos completamente configurable: A través de un archivo CSV que contiene los artefactos forenses a extraer se puede especificar, no sólo un archivo en particular sino todos los archivos de determinado tipo. Esta flexibilidad permite agregar nuevos artefactos forenses sin necesidad de recompilar el código fuente.

El diagrama de clases del programa es muy simple y puede verse a continuación:



La clase principal es SourceProcessor que se inicializa con las opciones seleccionadas para la extracción

```
 source_analyzer = SourceProcessor(auto_recurse=not
options.no_auto_recurse, process_vss=options.process_vss,
group_by_type=options.group_by_type)
```

Las opciones son:

- Auto recurse: Procesamiento recursivo de todos los directorios de la imagen (default: si)
- Procesar vss: Procesar archivos en virtual snapshots (default: no).
- Agrupar por tipo: Agrupar por tipo (categoría) de archivo o regenerar estructura de directorios (Default: agrupar)

- Al llamar a método principal de la clase source_analyzer.Analyze(options.source, output_writer) se especifican la imagen desde la cual se van a extraer los datos y el objeto de la clase encargada de escribir los archivos .csv que contienen la metadata.

- En caso de haber una partición encriptada se llama a self._PromptUserForEncryptedVolumeCredential(

scan_context, locked_scan_node, source_path, output_writer) para desbloquear la partición.

- Si se especificó el procesamiento recursivo de todos los directorios de la imagen llama a `self.WriteScanContext(scan_context, output_writer)`
- Este a su vez llama a `self.WriteScanNode(scan_context, scan_node, output_writer)` que es el que escribe los datos de la partición en un archivo.
- También tiene adentro el código para obtener el puntero al filesystem y llamar a recursive_hasher = RecursiveProcessor() recursive_hasher.calculatehashesfileentry(file_system, file_entry, u'', output_writer, self._group_by_type)

Por otro lado, en el archivo de configuración llamado “extraction.csv” se pueden configurar los artefactos forenses a extraer. Los campos por fila son los siguientes:

- Path: Determina la ubicación del archivo a extraer. Si alguno de los directorios especificados en el path es ‘*’ busca todas las coincidencias.
- Filemask: Nombre del archivo a extraer o máscara utilizando ‘*’ y ‘?’ como comodines de uno o varios caracteres.
- Group: Grupo de artefactos al que pertenece. Esto se utiliza para cuando se agrupan los artefactos por tipo en vez de por path. Todos los artefactos con el mismo grupo se almacenarán en un directorio con ese nombre.

Un ejemplo puede verse a continuación:

```
Path;Filemask;Group
\Windows\System32\winevt\logs;*.evtx;Logs
\Windows\prefetch;*.pf;Prefetch
\Windows\AppCompat\Programs;RecentFileCache.bcf;ApplicationCompatability
\Windows\AppCompat\Programs;Amcache.hve;ApplicationCompatability
\Windows\AppCompat\Programs;Amcache.hve.LOG*;ApplicationCompatability
\System Volume Information;Syscache.hve;Program Execution
\System Volume Information;Syscache.hve.LOG*;Program Execution
\Users\*\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline;ConsoleHost_history.txt;PowerShell
\;$MFT;FileSystem
\;$LogFile;FileSystem
\$Extend;$UsnJrnl;FileSystem
\Users\*\AppData\Roaming\Microsoft\Windows\Recent;Recursive;LNKFiles
\Users\*\AppData\Roaming\Microsoft\Office\Recent;Recursive;LNKFiles
\Users\*\Desktop;*.LNK;LNKFiles
\Windows\System32\config;SAM.LOG*;Registry
\Windows\System32\config;SECURITY.LOG*;Registry
\Windows\System32\config;SOFTWARE.LOG*;Registry
\Windows\System32\config;SYSTEM.LOG*;Registry
\Windows\System32\config;SAM;Registry
\Windows\System32\config;SECURITY;Registry
\Windows\System32\config;SYSTEM;Registry
\Windows\System32\config;SOFTWARE;Registry
\Windows\System32\config\RegBack;SAM.LOG*;RegistryRegBack
\Windows\System32\config\RegBack;SECURITY.LOG*;RegistryRegBack
\Windows\System32\config\RegBack;SOFTWARE.LOG*;RegistryRegBack
```

Si se ejecuta el programa sin opciones nos imprimirá la guía de utilización que es la siguiente:

```
usage: automator.exe [-h] [--no-auto-recurse] [--group-by-type] [--process-
vss] [--output-file] [image.raw]

extract triage forensic artifacts from a forensic image (supports VSS and
Bitlocker).

positional arguments:
  image.raw             path of the directory or filename of a storage media
image containing the file.

optional arguments:
  -h, --help            show this help message and exit
  --no-auto-recurse, --no_auto_recurse
                        Indicate that the source scanner should not auto-
recurse.
  --group-by-type, --group_by_type
                        Group extracted files by category instead of
preserving directory structure
  --process-vss, --process-vss
                        Process files in Volume Shadow Copies.
  --output-file, --output-file
                        Output zip file name.
```

El código fuente completo de la aplicación puede consultarse en el Apéndice IV.

Resultado

Este programa, con varias evoluciones, ha estado utilizándose regularmente en las diferentes direcciones de auditoría interna de un grupo multinacional desde el año 2017. Ha cumplido perfectamente con los objetivos fijados, permitiendo la obtención de evidencia forense a partir de imágenes forenses ubicadas en diferentes oficinas del grupo alrededor del mundo.

Con la inclusión del soporte de encripción bitlocker fue posible acceder a imágenes de equipos con encripción de disco completo en forma transparente cuando la compañía comenzó a utilizar esta tecnología para proteger sus activos de información.

Por último, la implementación de la configuración de artefactos forenses a extraer desde un archivo ".csv" permitió configurar para cada revisión qué traer en base a los requerimientos y al ancho de banda disponible para la transferencia de los archivos.

Triage desde un equipo operativo

Otra posibilidad extracción de artefactos forenses es a través de una herramienta denominada F-Response [7]. Según lo que declara en su website: "F-Response es una aplicación de conexión y extracción para actividades forenses, de e-discovery y de respuesta a incidentes. F-Response fue diseñada para proveer acceso de solo lectura a equipos físicos remotos (discos, volúmenes, arreglos RAID y memoria) así como a proveedores de almacenamiento Cloud.

Mediante esta herramienta, es posible mapear un volumen de un equipo remoto en el equipo del analista forense y acceder como si fuera un disco local. No sólo el acceso es de sólo lectura, con lo cual se garantiza que no se modifica la información origen, sino que también es posible acceder a información de bajo nivel como los VSS (Virtual Snapshots).

Una vez mapeado el drive remoto, es posible automatizar la extracción remota utilizando Kape. Para eso se desarrolló una configuración de un “Target” de extracción, con los mismos artefactos forenses que la herramienta desarrollada in-house para la extracción desde imágenes forenses. El listado de la misma puede verse en el Apéndice III

Automatización del procesamiento

Para la automatización del procesamiento de los artefactos forenses se decidió utilizar la combinación de Kape [8] en combinación con las siguientes herramientas de análisis forense:

- AmcacheParser
- Amcompatcacheparser
- EvtxEcmd
- JLECmd
- LECmd
- MFTECmd
- PECmd
- RBCmd
- RecentFileCacheParse
- RECmd
- SQLECmd
- SrumCmd
- SUMECmd
- WxTCmd
- BrowsingHistoryView
- USBDetective

Kape provee en forma nativa un módulo denominado EZParser que agrupa todas las herramientas forenses creadas por Eric Zimmerman. A este se le agregaron BrowsingHistoryView de Nirsoft que permite ver el historial de los browsers más conocidos y USB Detective que combina varios artefactos forenses para detectar qué dispositivos USB se conectaron al sistema y la estructura de archivos y carpetas dentro de ellos. El script completo puede verse en el Apéndice III

La salida se compone en una serie de archivos .csv (comma separated value) y de formato Excel organizados en una estructura de directorios según el tipo de información que proveen.

Metodología de Análisis

Información del Sistema Operativo

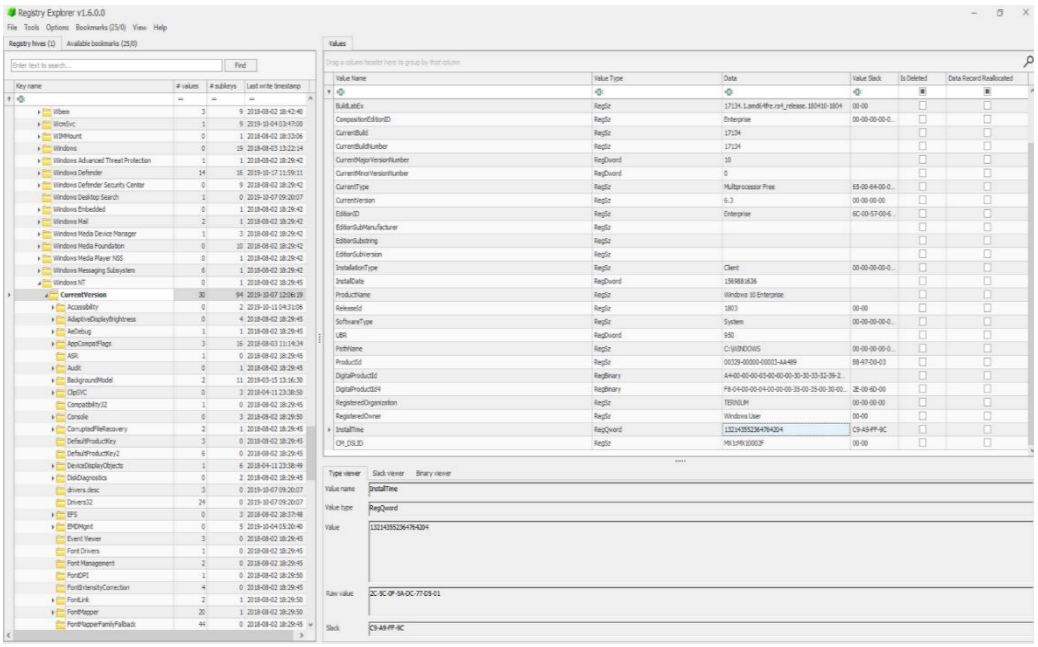
Nivel: 1

El primer paso al comenzar un análisis forense es entender qué sistema operativo estamos analizando. Ya no sólo es importante el tipo de sistema operativo sino la versión que está instalada. Windows 10 principalmente es famoso por cambiar artefactos forenses con cada versión lanzada, que se comportan más como nuevos sistemas operativos que como actualizaciones, debido a la cantidad de cambios que cada una trae.

Además, la fecha de instalación es crucial para determinar si la información forense buscada se encuentra en el equipo. No es poco habitual encontrarse con que al usuario se le reinstaló el sistema operativo o incluso se le cambió el equipo.

La información se encuentra en el registro del sistema, más concretamente en la llave de Software. Una breve introducción acerca del Registro de Windows puede encontrarse en el Apéndice I.

Los artefactos relevantes (Windows OS, service pack e install date) se encuentran en la registry en SOFTWARE\Microsoft\Windows NT\CurrentVersion.



Las propiedades a analizar son:

1. Productname
2. InstallTime (formato Windows Filetime)
3. RegisteredOrganization
4. ReleaseId
5. InstallDate (Unix Seconds UTC)

Se puede analizar manualmente mediante el utilitario RECmd.exe [9], especificando la clave del registro de Windows a extraer

```
.\RECmd.exe --q --f
D:\Kape\Output\C\Windows\System32\config\SOFTWARE --kn
Microsoft\Windows NT\CurrentVersion --vn ProductName
```

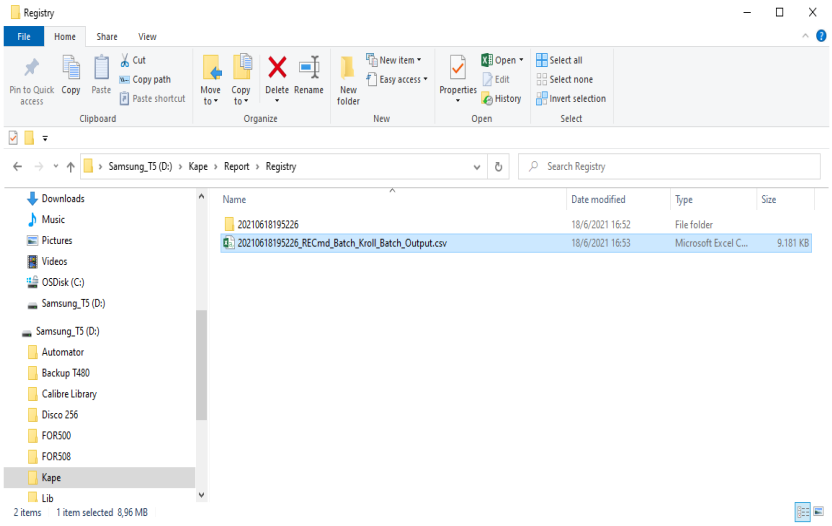
o bien utilizando una de las macros de análisis que provee el programa para analizar toda la Hive de SOFTWARE

```
.\RECmd.exe --q --f
“D:\Kape\Output\C\Windows\System32\config\SOFTWARE” --bn
.\BatchExamples\BasicSystemInfo.reb --csv c:\temp
```

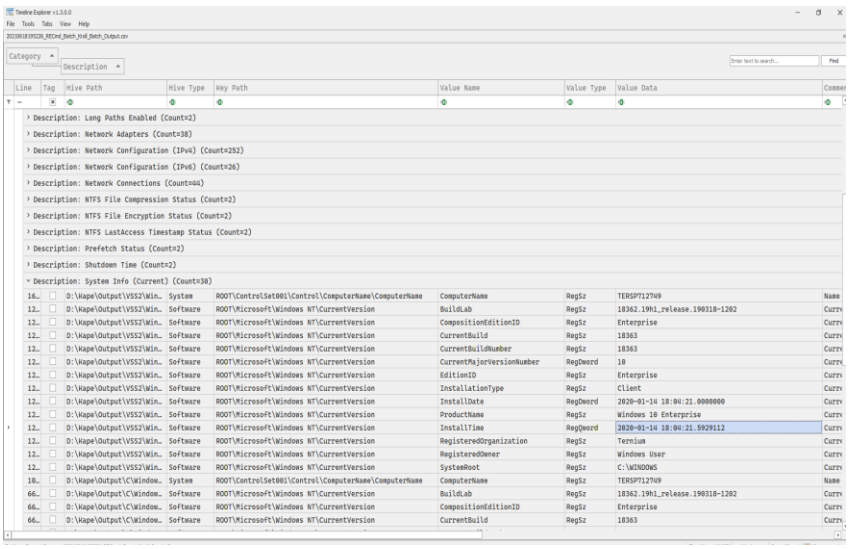
El script de KAPE utilizado, procesa todas las HIVES y llaves relevantes de la registry en forma simultánea y las consolida en un archivo .csv.

Análisis:

En el subdirectorio Registry de la salida de Kape abrir el archivo “RECmd_Batch_Kroll_Batch_Output.csv” en Timeline Explorer.



Analizar los registros con Category “System Info” y Description “System Info”. Los campos de fecha fueron convertidos por la herramienta y ya están en formato ISO (“YYYY-MM-DD HH:MM:SS UTC”).

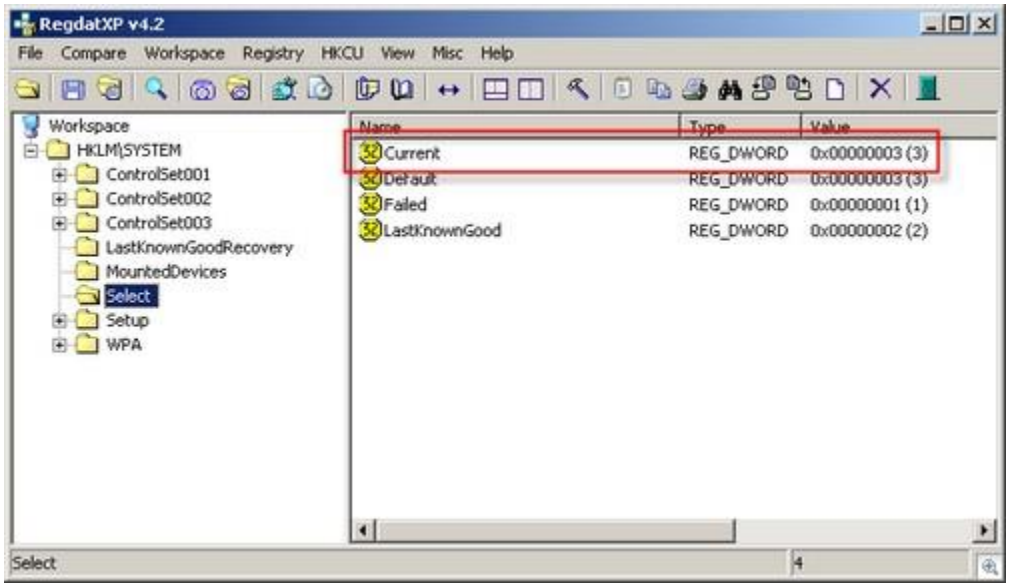


TimeZone

Debido a que Windows guarda la información de varios artefactos forenses en hora local en vez de GMT, es necesario saber en qué zona horaria está configurado el equipo para poder situar los eventos ocurridos adecuadamente. También es necesario tener en cuenta que, para determinada fecha en el pasado, el sistema podría haber estado configurado en otra zona horaria. Esto depende del país en donde se encuentre el equipo y si ese país utiliza cambios de hora para aprovechar más horas de luz solar.

La información de la zona horaria configurada en el equipo, se encuentra almacenada en la rama SYSTEM de la registry.

El primer paso para determinarla es averiguar cuál es al ControlSet activo en el equipo se encuentra en la clave Select de la rama SYSTEM, propiedad Current.



Si, por ejemplo, el valor de Current es 0x0000003, la zona horaria se encuentra en la clave ControlSet003\Control\TimeZoneInformation.

La información de la diferencia de tiempo con Greenwich está almacenada como un entero con signo

```
UTC = LOCALTIME + BIAS
```

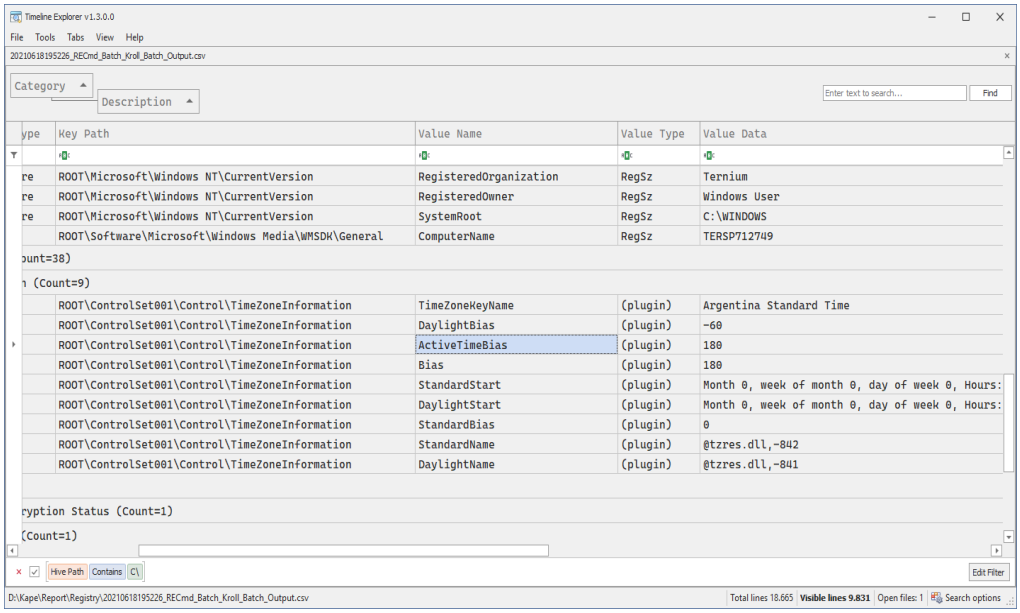
Es decir que si estamos en Argentina (huso horario -3), el ActiveTimeBias será de 3 horas (es decir 180 minutos) y eso es lo que aparecerá en la propiedad correspondiente de la registry.

Un análisis detallado del significado de cada uno de los parámetros se puede consultar en [10].

Análisis:

En el subdirectorio Registry de la salida de Kape abrir el archivo “RECmd_Batch_Kroll_Batch_Output.csv” en Timeline Explorer.

Analizar los registros con Category “System Info” y Description “Time Zone Information”.



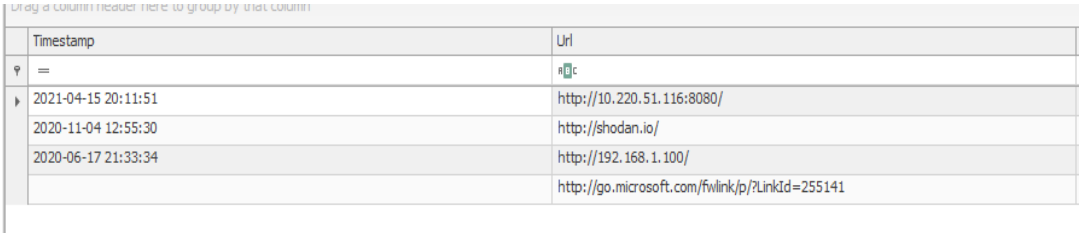
- El valor del registro que contiene la zona horaria configurada es TimeZoneKeyName.
- Para timestamps expresados en hora local, la hora GMT se calcula como: GMT = Timestamp + ActiveTimeBias

TypedURLs

Nivel: 1

Internet Explorer guarda las últimas 50 Urls (25 antes de Internet Explorer 10) que el usuario escribió en la barra de direciones. En la entrada TypedURLsTime (IE10+) está la última vez que esa entrada se utilizó en formato “Windows 64-bit filetime”. Para que se actualice una entrada en esta estructura el usuario la tiene que ingresar mediante el teclado, es decir que no basta hacer click en un link o que una página web redirija al browser, con lo cual se puede probar intencionalidad.

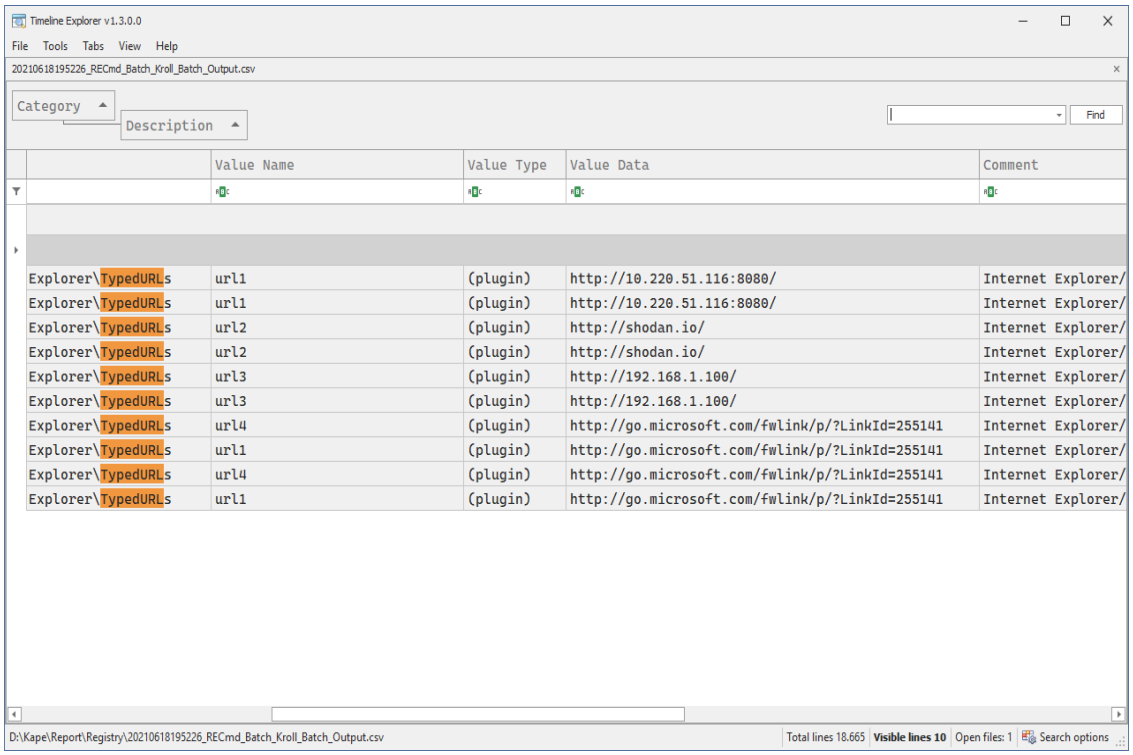
Se encuentran en la siguiente llave de la registry: NTUSER.DAT\Software\Microsoft\Internet Explorer\TypedURLs. Como se trata de NTUSER.DAT habrá una por usuario y esto permite asociar la acción con un usuario en particular del equipo.



Timestamp	Url
=	abc
2021-04-15 20:11:51	http://10.220.51.116:8080/
2020-11-04 12:55:30	http://shodan.io/
2020-06-17 21:33:34	http://192.168.1.100/
	http://go.microsoft.com/fwlink/p/?LinkId=255141

Análisis:

En el subdirectorio Registry de la salida de Kape abrir el archivo “RECmd_Batch_Kroll_Batch_Output.csv” en Timeline Explorer. Analizar los registros con Category “User Activity” y Description “TypedURLs”



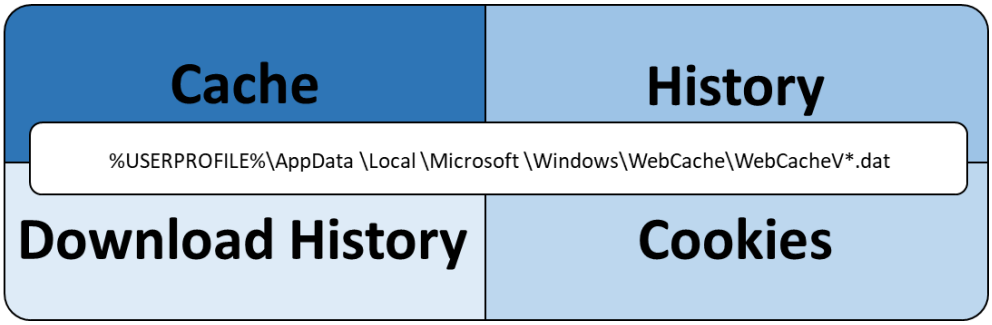
Historial Web

Nivel: 1

Debido a la alta integración entre Internet Explorer y el sistema operativo Windows, el historial de Internet Explorer guarda información no sólo de las páginas web a las que accedió el usuario sino también de los archivos abiertos cuando en discos local y de red cuando no se especifica qué programa utilizar para visualizarlos.

Esto es especialmente útil cuando queremos detectar el acceso a dispositivos externos o drives de red. Como cualquier otro elemento del historial, estos accesos se guardan en los archivos index.dat (IE4—IE9) o en WebCacheV*.dat para IE10+.[11]

En los sistemas Windows 10 en particular la base de datos se encuentra en el archivo %USERPROFILE%\AppData \Local \Microsoft \Windows\WebCache\WebCacheV*.dat. Este es una base de datos con formato ESE database.



Al igual que la mayoría de los artefactos forenses en Windows, esta funcionalidad no fue creada para ser consultada por un analista forense. Su principal propósito es permitir su consulta fin de sugerir al usuario sitios web en función de sus consultas pasadas a medida que el usuario tipea la URL.

Los usuarios pueden borrar su historial a través de la funcionalidad específica de Internet Explorer (u otro browser) o bien borrando los archivos de historial ubicados en el directorio oculto “History”.

```
~\AppData\Local\Microsoft\Windows\WebCache
> ls


    Directory: C:\Users\apamdb\AppData\Local\Microsoft\Windows\WebCache


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         13/5/2021     11:37           8192 V01.chk
-a----         13/5/2021     09:25         524288 V01.log
-a----         12/5/2021     15:12         524288 V0100671.log
-a----         12/5/2021     15:59         524288 V0100672.log
-a----         13/5/2021     09:25         524288 V0100673.log
-a----         31/3/2016     17:34         524288 V01res00001.jrs
-a----         31/3/2016     17:34         524288 V01res00002.jrs
-a----         12/5/2021     13:00         524288 V01tmp.log
-a----          4/5/2021     14:02       82313216 WebCacheV01.dat
-a----          4/5/2021     14:09          16384 WebCacheV01.jfm
```

De la información almacenada en el historial web un investigador puede conocer:

- Qué sitios se han visitados en los últimos n días. El número de días de historial está configurado en la registry en el parámetro DaysToKeep que se almacena en: SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\URL History)
- Qué archivos fueron accedidos en el sistema en los últimos n días (esto es porque Internet Explorer/Edge también registran como historia los archivos locales accedidos en el sistema)
- Cuántas veces se accedió a cada sitio/archivo.
- La cuenta de usuario que se utilizó para acceder al archivo (la información de historial se guarda en el perfil de cada cuenta de usuario en el sistema).
- El momento en el que se accedió por última vez a cada sitio web/archivo.

Análisis:

1. El procesamiento de la información se hace mediante BrowsingHistoryView de Nirsoft [12], que permite no sólo analizar el historial del Internet Explorer sino de todos los más utilizados.

```
.\BrowsingHistoryView.exe /scomma salida_browser_history.csv
/HistorySource 3 HistorySourceFolder C:\Users\
/VisitTimeFilterType 3 /VisitTimeFilterValue 60
```

2. El archivo se encuentra pre procesado en el directorio BrowsingHistory, archivo BrowsingHistory.csv
3. Filtrar por los siguientes ítems:

- file:// (filesystem)
- Url del Sharepoint On Premises
- companyname.sharepoint.com (Sharepoint en Azure)

Grep: Ripgrep https://github.com/BurntSushi/ripgrep#installation

```
rg  --color never -N -i 'URL,Title,Visit Time,Visit
Count,Visited From,Visit Type,Web Browser,User Profile,Browser
Profile,URL Length,Typed
Count|social.ternium.net|file:|ternium.sharepoint.com'
.\salida_browser_history.csv > sharepoint_and_files.csv
```

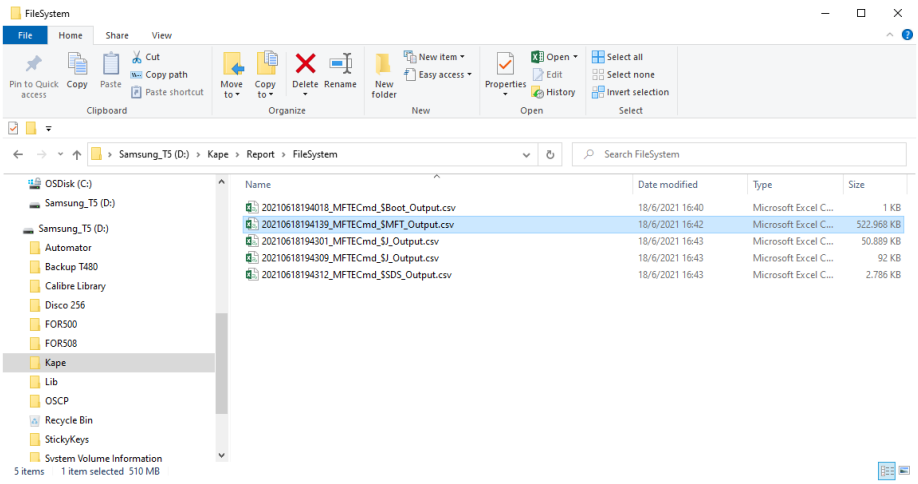
Logs de creación de archivos en el sistema de archivos

Nivel: 2

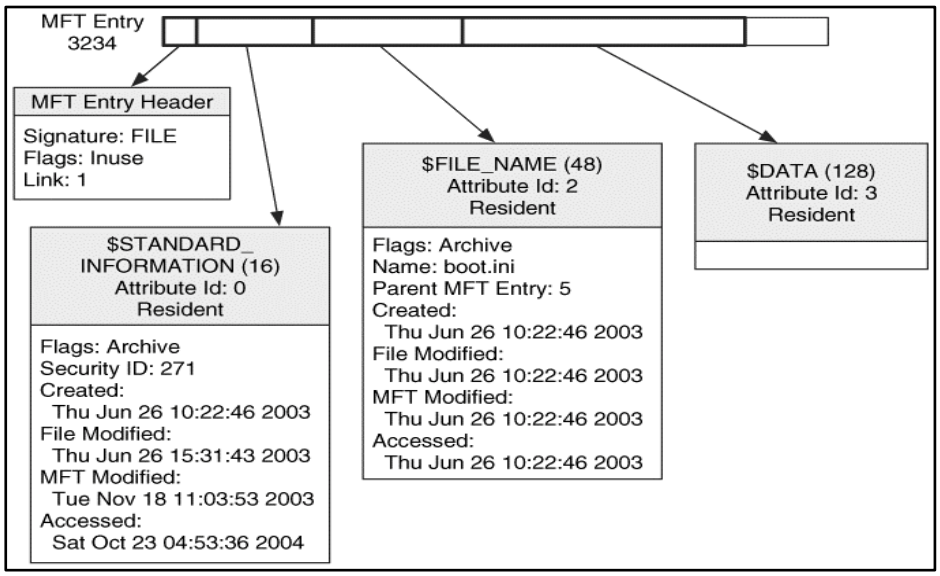
NTFS es lo que se denomina un “journaled filesystem”, es decir que, ante la falla de una operación sobre él, existe un registro que permite volver al último estado válido y por lo tanto prevenir la pérdida de información. Para eso utiliza varios logs en donde registra cada una de las operaciones que se realizan. Esto logs pueden ser utilizados para análisis forense.

Existen tres artefactos del sistema de archivos que pueden utilizarse, en forma conjunta o por separado, para reconstruir las actividades que se han realizado sobre él: $MFT, $Logfile y $USNJrnl.

El metaarchivo $MFT contiene la metadata de todos los archivos almacenados. Entre ellas están las fechas de creación, modificación, acceso y nacimiento (creación en el filesystem).



Cada archivo o directorio en NTFS tiene asociada metadata, la cual se almacena en uno o más atributos. Un archivo típico tiene los atributos $STANDARD_INFORMATION, $FILE_NAME y $DATA.[13]



Existen algunas particularidades del Sistema de archivos NTFS que podemos utilizar para detectar archivos copiados desde otros filesystems. Cuando se copia un archivo desde, por ejemplo, un file server a la máquina local la fecha de modificación del archivo es anterior a la fecha de creación.

La mayoría de los usuarios crean esos archivos en una carpeta de su perfil, por lo cual podemos filtrar por \Users en el path del archivo. Además, TimelineExplorer, nos permite elegir fácilmente el rango de fechas.

Fechas y Atributos

En $STANDARD_INFORMATION se encuentran el conjunto de timestamps principal (aunque no el único) además del dueño del archivo, la seguridad y la información de quota.

El tipo de atributo default es 16 y tiene un tamaño estático de 72 bytes en Windows 2000 y XP. Debido a que Microsoft ordena los atributos dentro de la MFT este es el primer atributo dado que tiene el ID más bajo posible. En este atributo existen 4 fechas/horas diferentes

expresadas en formato Microsoft (valor de 64 bits que representa el número de 0.1 microsegundos transcurridos desde el 1 de enero de 1601. Los valores son:

- File Created
- File Accessed
- File Modified
- MTF last written

Estas 4 fechas se conocen en el campo de análisis forense como “MACB” que corresponden a: Modify, Accessed, Metadata change (modificación de $MFT) y Birth (creación de archivo en el filesystem). Todos los timestamps utilizados en el sistema de archivos NTFS se guardan en UTC (GMT).

Otra copia de los timestamps se guarda en los atributos $File_Name del cual puede haber dos instancias si se almacenan el nombre largo y corto (8.3) del archivo. Esto significa que para un archivo típico existirán 2 o 3 conjuntos de 4 timestamps.

Nota: Debido a que no existe un método directo para modificar los tiemstamps de los atributos $FILE_NAME, en caso de no coincidir con el del $STANDARD_INFORMATION puede ser un indicio de timestomping. Es decir, cuando la información del archivo se altera intencionalmente para ocultar cuándo ciertas actividades se realizaron en el sistema.

Comportamiento de los atributos según la operación

Muchos investigadores forenses han estado investigando por años cómo las diferentes operaciones sobre un archivo modifican los valores de los timestamps en $MFT. Una de las más conocidas es la desarrollada por el SANS Institute para su poster de “Windows Forensic Analysis”[14]

Sin embargo en la misma documentación de los cursos del SANS, se denomina como un “trabajo en progreso” y en efecto difiere a la presentada por Cyberforensicator en su página web [15]. Ante la duda siempre es recomendable hacer pruebas sobre una instalación del sistema operativo que se está analizando (eso incluye versión y buildnumber, ya que en Windows 10 suele haber cambios significativos con cada versión).

Algunas conclusiones interesantes a tener en cuenta para este tipo de análisis son:

- La creación del archivo afecta los 4 timestamps. (MACB)
- Mover un archivo dentro del mismo filesystem sólo afecta la metadata (C) ya que sólo se cambia la

metadata y no se realizan cambios sobre la información del archivo en si.

- Mover un archivo de un volumen a otro afectan el timestamp de último acceso y el de creación (.A.B) cuando la operación se realiza mediante línea de comando, pero sólo el de acceso (A) cuando la misma operación se realiza mediante la interfaz gráfica (cortar y pegar).
- Cuando se copia un archivo se modifican (ACB) pero la fecha de modificación (M) se hereda del original

Es importante remarcar que la modificación a la fecha de acceso se actualiza aun cuando el sistema operativo tenga deshabilitadas las actualizaciones de último acceso, cosa que se realizó durante varias versiones de Windows para mejorar la performance.

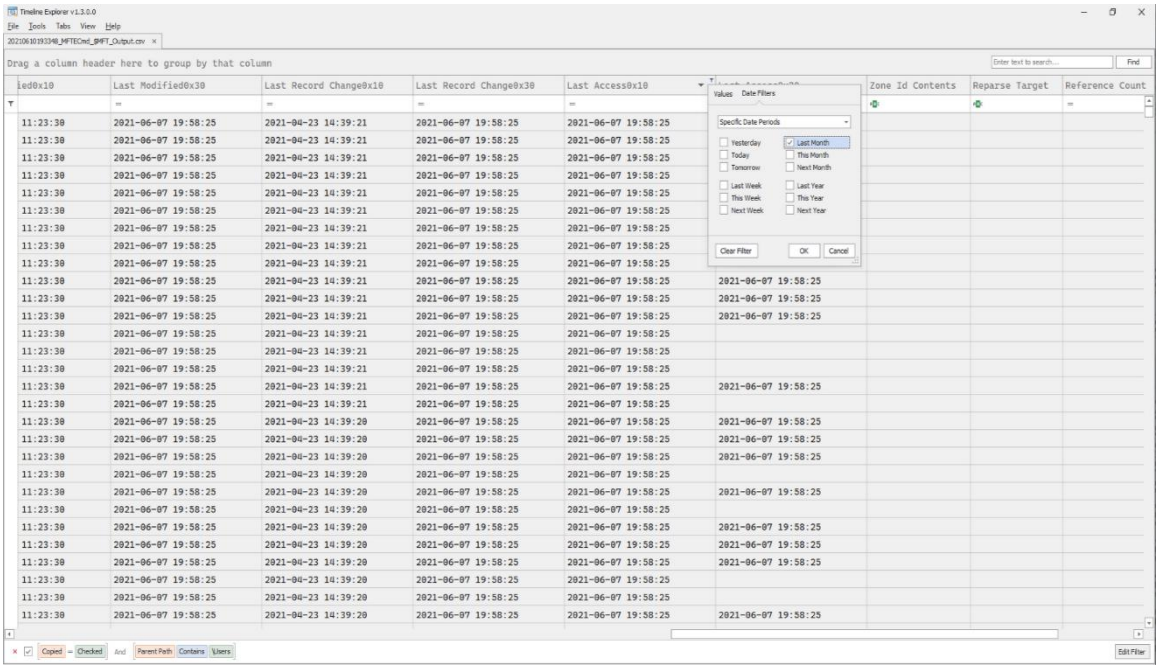
Basados en estas observaciones lo primero que se suele buscar cuando se quieren detectar archivos copiados/movidos es detectar aquellos cuya fecha de modificación es anterior a su fecha de creación. Es claro que debido a la complejidad de estos timestamps y a su carácter de preliminares no deben tomarse como 100% concluyentes.

Análisis:

De la salida de KAPE, analizar el archivo YYYYMMDDHHMMSS_MFTECmd_$MFT_Output.csv ubicado en la carpeta Filesystem utilizando TimelineExplorer

1. Hacer un filtrado por \Users en el path del archivo
2. Seleccionar por “Copied: True” (STANDARD_INFO modified < STANDARD_INFO created time)

3. Ordenar por “**creation time (0x10)”** y filtrar por “Last Month” de esta manera podemos ver los archivos copiados al filesystem en el último mes, por ejemplo:



Shell Items (Shortcut ítems & jumplists)

Nivel: 1

Windows utiliza un tipo de archivo especial que permite apuntar a otro archivo diferente, se los conoce como links o archivos ‘lnk’ (por su extensión). Los shortcuts tienen información relevante para el analista forense incluyendo en qué tipo de medio está el archivo apuntado (drive de red, disco fijo, dispositivo de almacenamiento externo, etc). También incluye el path, el nombre del drive, el número de serie del volumen (no del USB). También hay información de la metadata del archivo apuntado como timestamps, tamaño y atributos.

Desde el punto de vista forense los archivos lnk más relevantes son los denominados “Recent Files”, creados automáticamente por Windows (versiones 10 en adelante) a medida que se acceden/crean archivos mediante el Explorador de Windows. La secuencia de creación es la siguiente:

- Se crea un archivo lnk para el archivo no ejecutable accedido y uno para la carpeta padre.
- En caso de que se trate de una carpeta, se crean links para la carpeta, la carpeta padre y la carpeta superior (abuelo).
- Se guardan hasta un total de 149 archivos.
- Esto sucede incluso si el archivo se encuentra en un dispositivo USB.
- La ubicación del directorio donde se almacenan es:

```
Windows 7+
C:\Users\<user>\AppData\Roaming\Microsoft\Windows\Recent
```

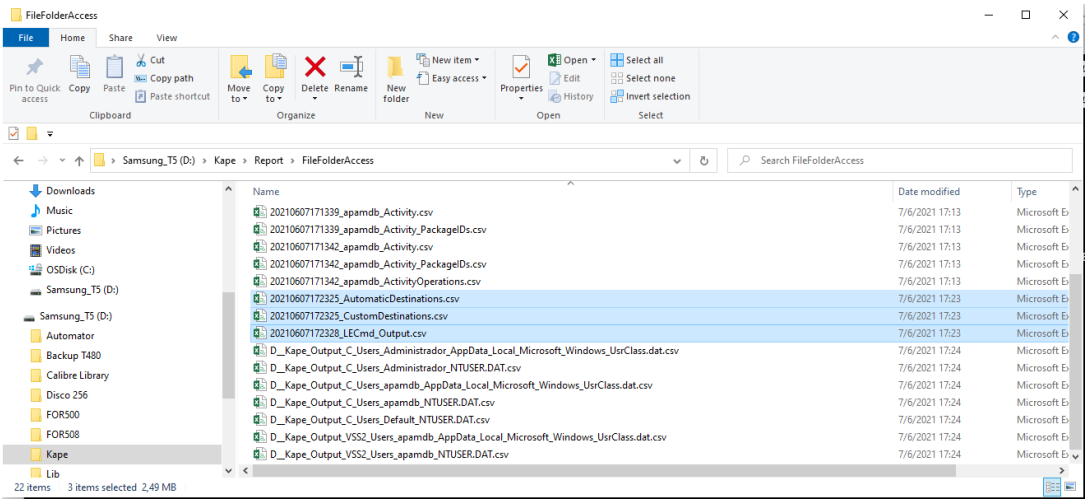
Análisis:

Para determinar los archivos relevantes que fueron accedidos en el sistema, analizar en la carpeta FileFolderAccess los archivos *_AutomaticDestinations.csv, *_CustomDestinations.csv y *_LECmd_Output.csv.

En estos listados buscaremos archivos en drives externos (usb o de red) o accedidos mediante UNC (es decir \\servidor\share\nombrearchivo.ext).

Notas:

- La fecha de creación del archivo ".lnk" es la **primera** vez que se abrió el archivo destino, mientras que la de modificación es la **última**.



Documentos Recientes (Generados por programas)

Nivel: 1

Los documentos abiertos recientemente en sistemas Windows pueden ser analizados fácilmente mediante la llave RecentDocs en la registry de cada perfil de usuario (NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs) . Esta llave y subllaves proveen información valiosa para determinar actividad relacionada con archivos específicos.

La información se encuentra codificada pero la mayoría de los programas de análisis de registry la decodifican sin problemas. Los valores se muestran en orden de MRUlist (es decir, una lista de los más recientemente utilizados ordenados cronológicamente desde la más reciente a las más antigua).

Además, otros programas como Adobe Acrobat Reader y Microsoft Office tienen registro separados para los archivos recientes.

Análisis:

Kape hace un análisis completo de la registry a través de sus batchs en RECmd.exe. El archivo que contiene todas las salidas compiladas es TIMESTAMP_RECmd_Batch_Kroll_Batch_Output.csv en la carpeta del mismo directorio están por separado.

Lo conveniente es utilizar TimelineExplorer y agrupar por “Description”. Categorías a analizar:

- Adobe cRecentFiles y cRecentFolders (RecentDocs para Adobe)
- FirstFolder (muestra la primera carpeta que se ofrece para abrir/salvar por aplicación)
- MicrosoftOfficeMRU y MicrosoftOffice (RecentDocs para Office)
- OpenSavePidlMRU (Archivos abiertos y salvados a través de Windows Shell Dialog)
- TypedPaths y TypedURLs (urls y paths tipeados en explorer/IE: Ver drives de red y externos)
- RecentDocs
- WordWheelQuery (busquedas por nombre de archivo a través de Explorer)

Shellbags

Nivel: 1

Las shellbags son un tipo de artefacto forense asociados con carpetas accedidas a través del explorador de Windows [4]. Windows utiliza la información almacenada en las *shellbags* para almacenar las preferencias del usuario al visualizar esa carpeta a través del Explorador de Windows. Esto permite que si uno hace cambio en la forma en la que se muestra la carpeta (tamaño de la ventana, formato de visualización de archivos, etc), el sistema lo recordará cuando se abra nuevamente la carpeta. Según se ha determinado, para que exista una entrada en la estructura shellbags para una determinada carpeta basta con que la misma sólo haya sido abierta y cerrada una vez, es decir, la simple existencia de una subclave relacionada con una carpeta determinada indica que el usuario analizado visitó la carpeta al menos una vez.[16]

Debido a que Windows modifica la fecha de última modificación para cada clave de la registry también es posible determinar cuándo se visitó por primera y última vez con algunas salvedades.

Si bien el artefacto en principio sólo almacena información de carpetas, bajo algunas circunstancias también almacena información de archivos, como cuando se utiliza la interfaz del explorador de Windows para explorar el interior de archivos comprimidos (especialmente .zip).

Es importante destacar que esta información se almacena **por usuario** por lo cual, si el equipo es utilizado por varias personas a la vez, siempre se podrán asociar determinada actividad con uno de ellos. Otro detalle importante es que la entrada en la Shellbag permanece aun cuando la carpeta haya sido borrada en el disco, o el disco externo haya sido desconectado con lo cual puede obtenerse información de directorios y accesos aun cuando su contenido ya no esté disponible físicamente.

El artefacto se conoce como “shellbag” debido al nombre de una de las claves de la registry involucradas y se almacena en forma separada para cada usuario del equipo. La ubicación de la información varía según la versión del sistema operativo de la siguiente forma:

Windows XP

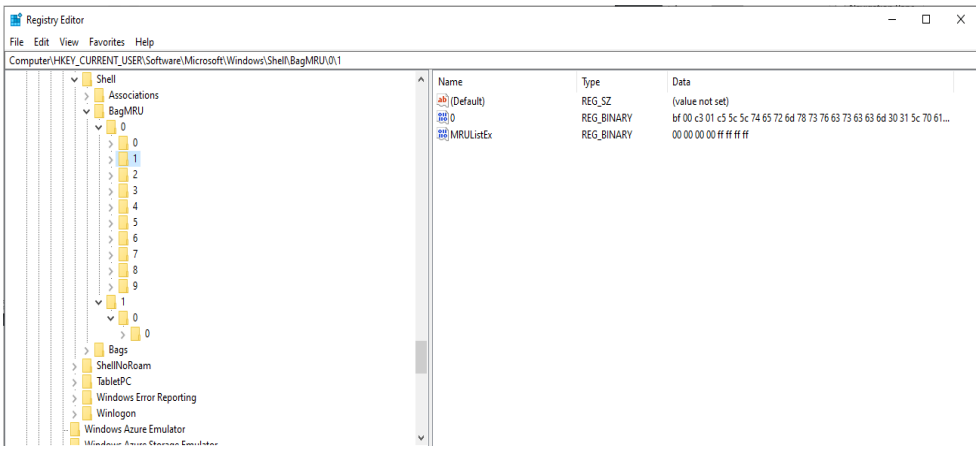
```
Carpetas de Red - NTUSER.DAT\Software\Microsoft\Windows\Shell
Carpetas Locales - NTUSER.DAT\Software\Microsoft\Windows\ShellNoRoam
Carpetas en Dispositivos -
NTUSER.DAT\Software\Microsoft\Windows\StreamMRU
```

Windows 7-10

```
Acceso vía desktop - NTUSER.DAT\Software\Microsoft\Windows\Shell
Acceso vía explorer – UsrClass.dat\Local
Settings\Software\Microsoft\Windows\Shell
```

Estructura

La información relacionada con shellbags se almacena en dos claves de la registry BagsMRU y Bags. La clave BagsMRU almacena los nombres de las carpetas y la estructura de directorios creando una estructura análoga dentro de la registry. A excepción de la clave BagsMRU que representa el escritorio (Desktop), la estructura no tiene posiciones asignadas y se va creando a medida que el usuario explora las carpetas.[17]



Propósito

- Determinar las preferencias de visualización de un usuario en Windows Explorer.
- Determinar si se realizó alguna actividad en una carpeta en particular.
- Determinar si el usuario abrió/cerró/creó/borró o copió una carpeta.
- En algunos casos, determinar los archivos que contenía determinada carpeta

Análisis:

Se analizan utilizando SBCmd o USB Detective. En la carpeta FileFolderAccess hay dos archivos por cada usuario y drive que salen de las hives relevantes de UserClass y NTUSER

- D_Kape_Output_Drive_Users_**Username**_AppData_Local_Microsoft_Windows_UsrClass.dat
- D_Kape_Output_Drive_Users_**Username**_NTUSER.DAT

Se busca evidencia de acceso a shares de red y discos externos mediante el file explorer.

Tener en cuenta que:

- Si se procesaron VSS (Virtual Snapshots) se tendrá un par más por cada VSS.
- En sistemas modernos el grueso de las entradas está en UsrClass.
- Los valores de macb (Modification, Access, Change and Creation/Birth) salen de los valores que tiene el directorio en el filesystem la primera vez que se crea la entrada **no de la fecha de acceso**.
- Los valores a analizar son los de “Folder Last Interacted Time” y “Folder First Interacted Time”.
- Un acceso a un subdirectorio modifica la fecha del MRU del padre (superior, no los intermedios)

Otro método para analizarlos es utilizando los reportes de USB Detective que se encuentran en la carpeta USB Detective de la salida de Kape.

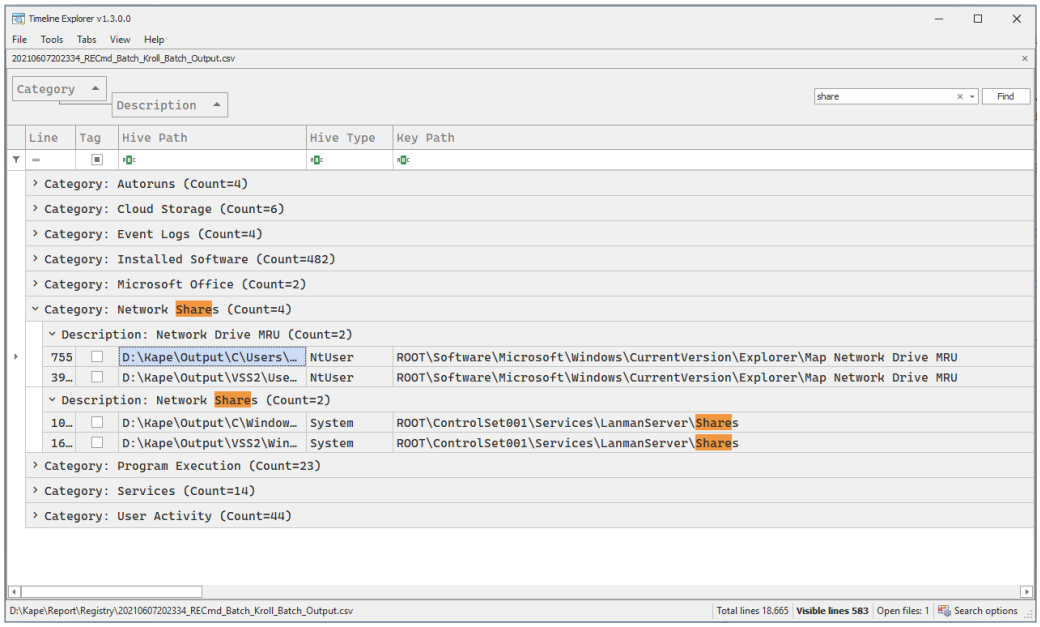
Drives Mapeados

Nivel 2:

Los drives mapeados a una letra, son almacenados en el sistema. La configuración se ubica en la Registry del sistema (SYSTEM) en SYSTEM\CurrentControlSet\Services\lanmanserer\Shares. Hay una entrada por carpeta y el parámetro CSCFlags determina si existe un caché

Análisis:

En el subdirectorio Registry de la salida de Kape abrir el archivo “RECmd_Batch_Kroll_Batch_Output.csv” en Timeline Explorer. Analizar los registros con categorías “Network Shares” y Description “Network Drive MRU”



SRUM (System Resource Usage Monitor)

Nivel: 2

El monitor de utilización de recursos del sistema (SRUM por sus siglas en inglés) es una base de datos que almacena información de performance del sistema, algunas de las cuales pueden verse sumarizadas en la aplicación “Monitor de Recursos”.

Durante los últimos años se ha comenzado a analizar en el marco de investigaciones forenses debido a que puede arrojar información sobre las actividades de un usuario aun cuando se hayan realizado maniobras anti-forenses.

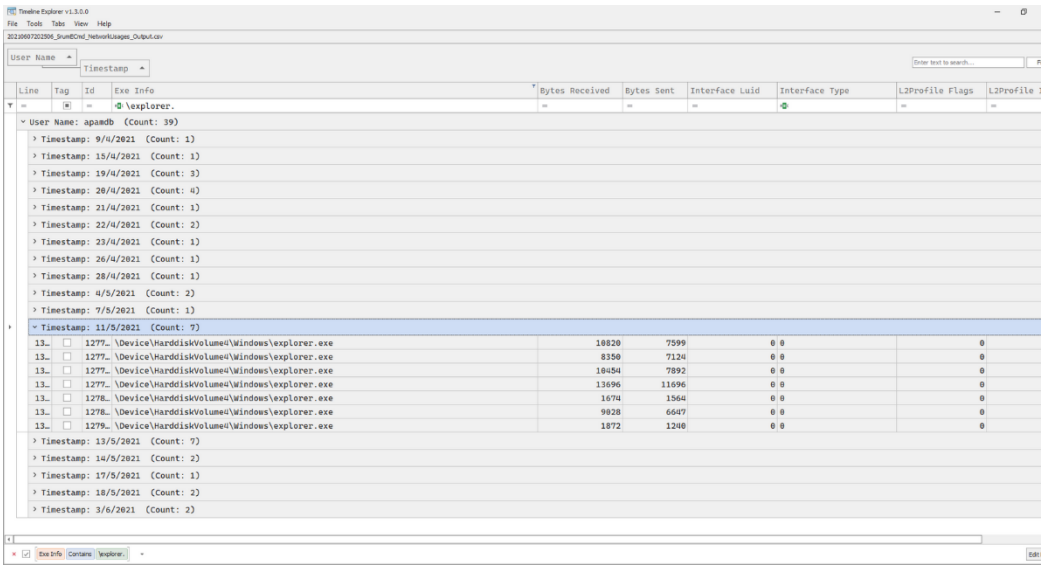
Por ejemplo, es posible detectar con éxito la transferencia de gran cantidad de información desde un equipo remoto al pc analizada,

pudiendo incluso determinar si la actividad la realizó el usuario desde la interfaz gráfica o la transferencia la realizó alguna aplicación.

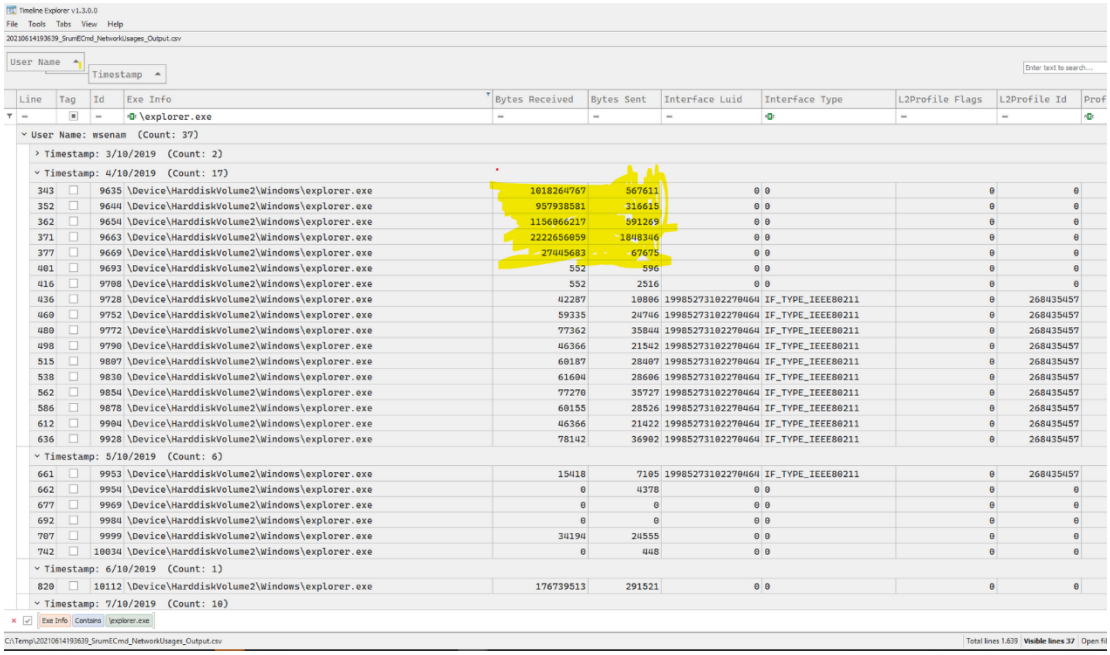
Análisis:

En la carpeta SystemActivity, analizar el archivo: Timestamp_SrumECmd_NetworkUsages_Output.csv filtrando por un ejecutable determinado. En algunos casos, se ve el título de la ventana, que puede darnos pistas de que se accedió a un archivo. En otras, como en este ejemplo, hemos logrado determinar la cantidad de información que se copió desde un share de red a un dispositivo externo a través del Windows Explorer:

Análisis de un equipo en donde no se copiaron datos a dispositivos externos:



El mismo análisis en un equipo en el que se copiaron gran cantidad de archivos:



Timeline Explorer v1.3.0.0

File Tools Tabs View Help

20210614193639_SrumECmd_NetworkUsages_Output.csv

User Name | Timestamp | Enter text to search...

Line	Tag	Id	Exe Info	Bytes Received	Bytes Sent	Interface Luid	Interface Type	L2Profile Flags	L2Profile Id	Prof
			\explorer.exe							
User Name: wsenam (Count: 37)										
Timestamp: 3/10/2019 (Count: 2)										
Timestamp: 4/10/2019 (Count: 17)										
343		9635	\Device\HarddiskVolume2\Windows\explorer.exe	1018264767	567611	0	0	0	0	
352		9644	\Device\HarddiskVolume2\Windows\explorer.exe	957938581	316615	0	0	0	0	
362		9654	\Device\HarddiskVolume2\Windows\explorer.exe	1156066217	591269	0	0	0	0	
371		9663	\Device\HarddiskVolume2\Windows\explorer.exe	2222656059	1048346	0	0	0	0	
377		9669	\Device\HarddiskVolume2\Windows\explorer.exe	27445683	67675	0	0	0	0	
401		9693	\Device\HarddiskVolume2\Windows\explorer.exe	552	596	0	0	0	0	
416		9708	\Device\HarddiskVolume2\Windows\explorer.exe	552	2516	0	0	0	0	
436		9728	\Device\HarddiskVolume2\Windows\explorer.exe	42287	10886	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
460		9752	\Device\HarddiskVolume2\Windows\explorer.exe	59335	24746	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
480		9772	\Device\HarddiskVolume2\Windows\explorer.exe	77362	35844	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
498		9790	\Device\HarddiskVolume2\Windows\explorer.exe	46366	21542	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
515		9807	\Device\HarddiskVolume2\Windows\explorer.exe	60187	28407	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
538		9830	\Device\HarddiskVolume2\Windows\explorer.exe	61604	28606	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
562		9854	\Device\HarddiskVolume2\Windows\explorer.exe	77270	35727	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
586		9878	\Device\HarddiskVolume2\Windows\explorer.exe	60155	28526	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
612		9904	\Device\HarddiskVolume2\Windows\explorer.exe	46366	21422	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
636		9928	\Device\HarddiskVolume2\Windows\explorer.exe	78142	36902	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
Timestamp: 5/10/2019 (Count: 6)										
661		9953	\Device\HarddiskVolume2\Windows\explorer.exe	15418	7105	1998527310227046	4 IF_TYPE_IEEE80211	0	2684354573	
662		9954	\Device\HarddiskVolume2\Windows\explorer.exe	0	4378	0	0	0	0	
677		9969	\Device\HarddiskVolume2\Windows\explorer.exe	0	0	0	0	0	0	
692		9984	\Device\HarddiskVolume2\Windows\explorer.exe	0	0	0	0	0	0	
707		9999	\Device\HarddiskVolume2\Windows\explorer.exe	34194	24555	0	0	0	0	
742		10034	\Device\HarddiskVolume2\Windows\explorer.exe	0	448	0	0	0	0	
Timestamp: 6/10/2019 (Count: 1)										
820		10112	\Device\HarddiskVolume2\Windows\explorer.exe	176739513	291521	0	0	0	0	
Timestamp: 7/10/2019 (Count: 10)										

Exe Info Contains \explorer.exe

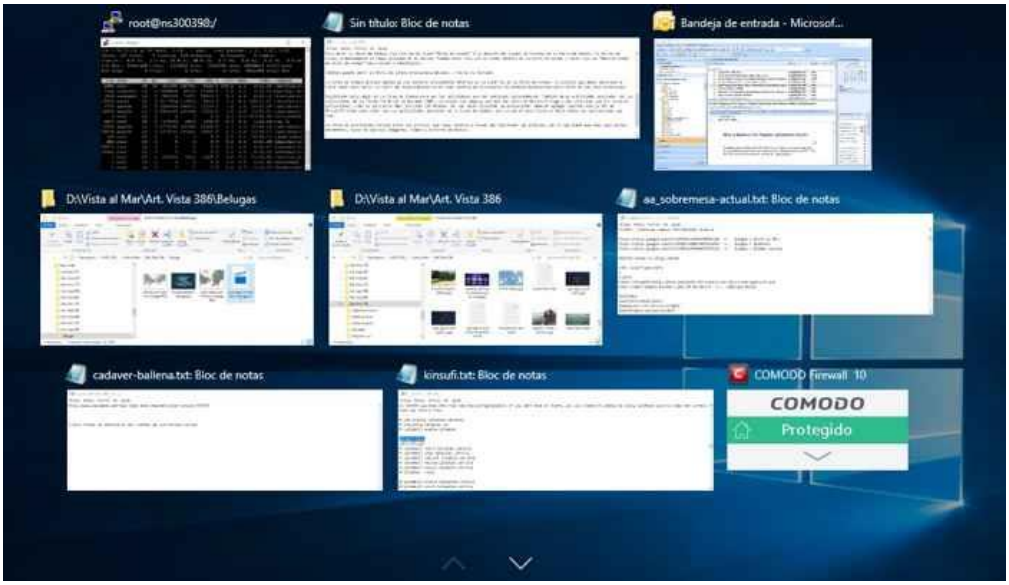
C:\Temp\20210614193639_SrumECmd_NetworkUsages_Output.csv

Total lines 1,639 Visible lines 37 Open fil

Windows 10 Timeline

Nivel: 3

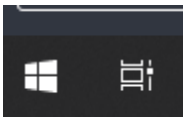
Es una funcionalidad que se agregó en Windows 10, que permite que un usuario vea su actividad reciente en el sistema.



En el timeline se guarda: El historial de Edge, archivos Office2016 y las fotos visualizadas. También se guarda internamente en la base los tiempos en foco y en ejecución de las aplicaciones. La base donde se almacenan está ubicada en:

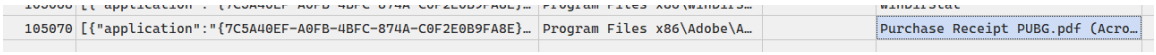
```
%USERPROFILE%\Appdata\Local\ConnectedDevicesPlatform\L.\ActivitiesCache.db
```

Normalmente se accede al historial mediante el ícono a la derecha del botón de Windows



Mediante este timeline es posible acceder a un registro de qué aplicación se ejecutó y por cuánto tiempo. Además, en la columna “Display Text” es posible ver en ocasiones el nombre del archivo que se estaba editando en Word o la página web que se estaba visitando.

Por ejemplo, se puede ver en la imagen el nombre del pdf que se visualizó

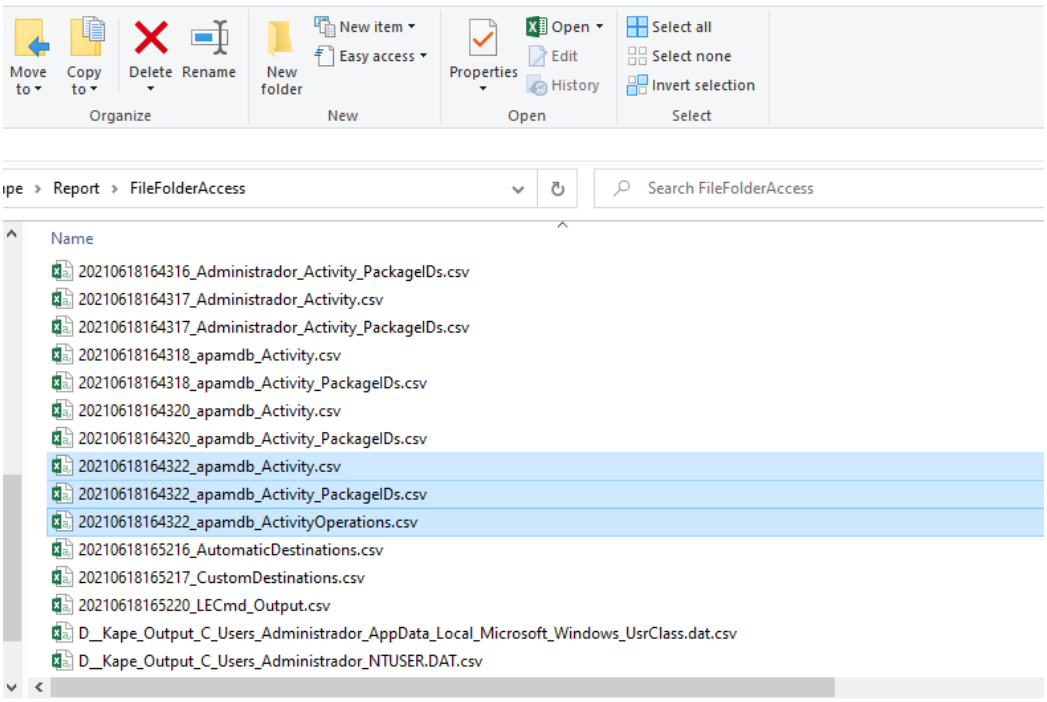


Existe una utilidad denominada WxtCmd que analiza automáticamente la base de datos de actividades[18]

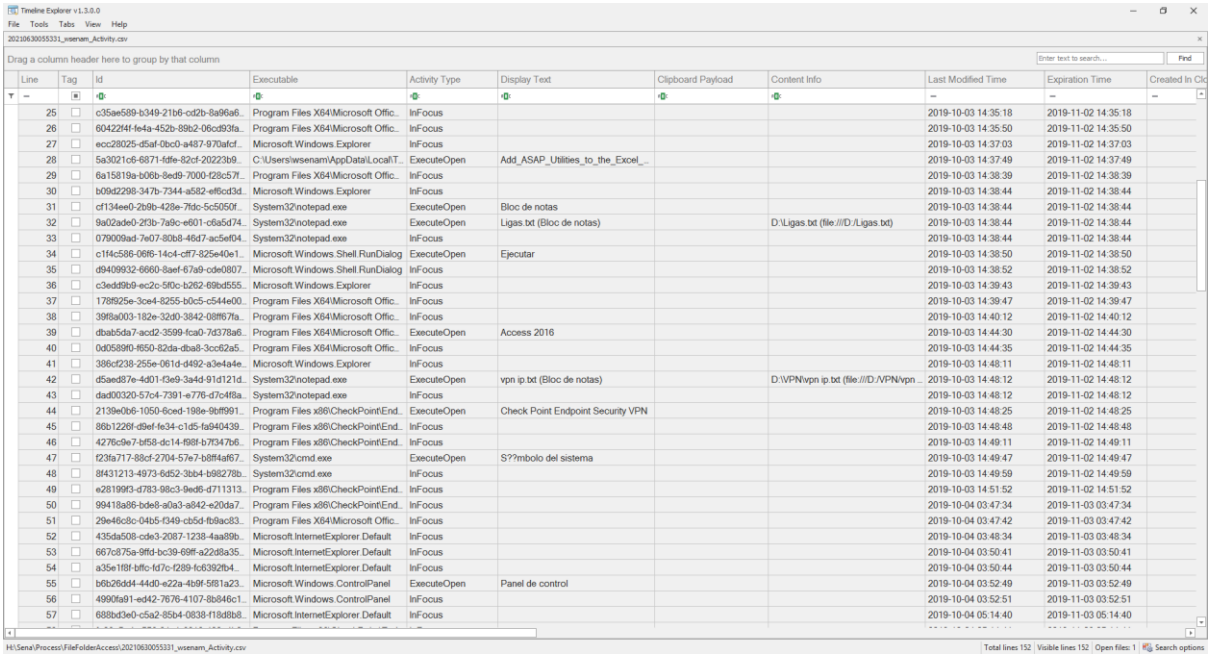
Análisis:

Analizar los archivos generados por WxtCmd en FileFolderAccess para cada perfil/usuario en el sistema (especialmente TIMESTAMP_USUARIO_Activity.csv). Es posible detectar aplicaciones utilizadas, incluidas aquellas destinadas a borrar huellas o aplicaciones Cloud).

–



Ejemplo de análisis en donde se pueden ver dos archivos en un disco externo (Ligas.txt y “vpn ip.txt”) ambos accedidos (acción ExecuteOpen) a través del bloc de notas.



Logs del Sistema

Nivel: 3

Existen una gran cantidad de eventos disponibles en los registros del sistema, aunque dependiendo de la configuración, información puede estar disponible sólo para un rango limitado de tiempo.

Los eventos que más hemos utilizado en revisiones forense son los siguientes:

Acceso a dispositivos externos (USB)

En el log de seguridad se genera el evento 6416 cada vez que se detecta un dispositivo y 4663 cuando se copia un archivo. Si el dispositivo usa Bitlocker se generan los eventos 39/40 en MBAM\Operational que incluyen el dispositivo conectado e ID.[19],[20]

Ubicación del log:

```
C:\Windows\System32\winevt\Logs\Security.evtx
```

Información de referencia de los eventos:

```
https://docs.microsoft.com/en-us/windows/security/threat-
protection/auditing/event-6416
https://docs.microsoft.com/en-us/windows/security/threat-
protection/auditing/event-4663
```

Información provista:

- Fabricante, modelo y número de serie del dispositivo conectado, fecha y hora.
- Path del archivo accedido, información del proceso y usuario que realizaron la acción, fecha y hora.

Conexión de Volumen NTFS

Ubicación del log:

```
C:\Windows\System32\winevt\Logs\Microsoft-Windows-
Ntfs%4Operational.evtx
```

Información de referencia de los eventos:

Microsoft-Windows-Ntfs/Operational, eventos 142 y 145

Información provista:

- ID del volumen conectado.
- Espacio disponible en disco (para poder inferir si se ha copiado información a partir de dos eventos consecutivos).
- Letra asignada por el sistema

```
{
  "EventData": {
    "Data": [
      {
        "@Name": "VolumeGuid",
        "#text": "e5111b3e-5c66-4afd-b04e-e5517263cd02"
      },
      {
        "@Name": "VolumeNameLength",
        "#text": "48"
      },
      {
        "@Name": "VolumeName",
        "#text": "\\\\?\\Volume{e5111b3e-5c66-4afd-b04e-e5517263cd02}"
      },
      {
        "@Name": "LowestFreeSpaceInBytes",
        "#text": "305319936"
      },
      {
        "@Name": "HighestFreeSpaceInBytes",
        "#text": "305319936"
      },
      {
        "@Name": "IsBootVolume",
        "#text": "False"
      }
    ]
  }
}
```


Análisis:

En el directorio EventLogs de la salida de Kape, analizar el archivo **TIMESTAMP**_EvtxECmd_Output.csv, que contiene todos los eventos del sistema.

También es posible obtenerlo directamente desde los archivos de logs extraídos desde el sistema a analizar utilizando únicamente powershell. Por ejemplo, para el evento 4663:

```
Get-WinEvent -Path .\Security.evtx -Oldest | Where-Object { $.Id -eq
‘4663’ -and $.Message -match ‘explorer.exe’} | Format-Table -Wrap -
AutoSize
```

Para el evento 4663 los campos relevantes del log son:

- User Name: Nombre de usuario que accedió al archivo, filtrar los nombres de usuario que terminan en $ ya que corresponden a cuentas de máquina.
- Payload Data 3: Nombre de archivo que fue accedido.
- Executable Info: Programa que accedió al archivo.

Notas: Se ha observado una gran variabilidad en cuanto a los logs generados de acuerdo a la configuración del sistema.

Por ejemplo, en el caso que ilustra la siguiente figura, la mayoría de los eventos están generados por el acceso del antivirus en su escaneo del drive conectado. Si bien esto nos hace perder eventos porque se llena el log (habitualmente hay 3 o 4 días de eventos), también nos puede dar el árbol completo del disco conectado sin necesidad de que el usuario haya entrado a los directorios. También se ven los archivos ejecutables porque los escanea el antivirus, pero no otro tipo de archivos más relevantes en el marco de este tipo de investigaciones como documentos de word, excel, pdf, etc.

El proceso que más accede en este caso es el ccmexec y luego el ntrtscan. Aparentemente se trata de un escaneo de aplicaciones instaladas que hace automáticamente el Configuration Manager.

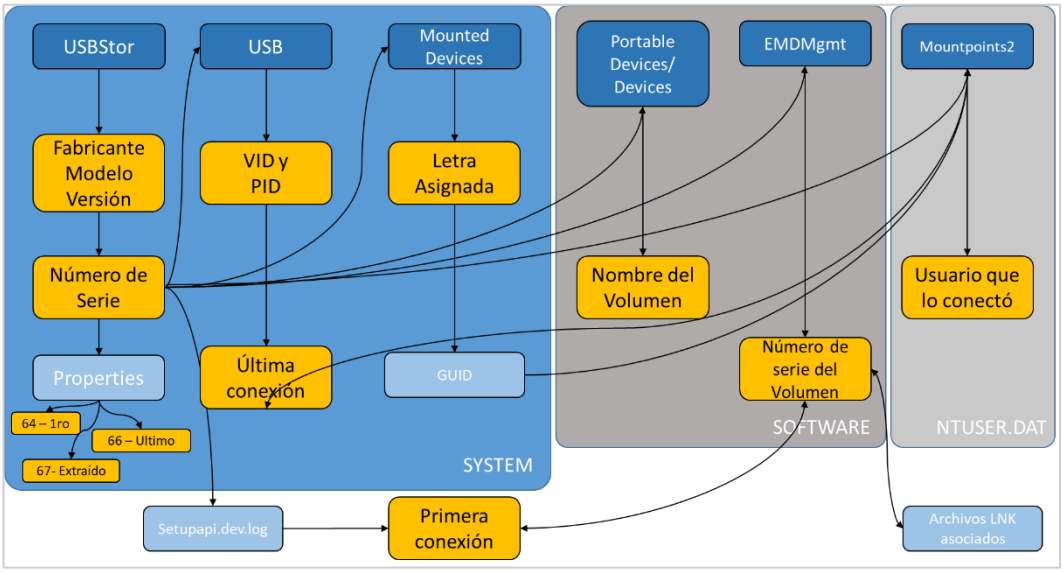
Dispositivos USB

Nivel: 1

Uno de los principales medios de intercambio de información digital es a través de medios de almacenamiento masivo. Los dispositivos actuales son confiables y permiten almacenar una cantidad de datos muy importante. Es más, debido a que la tendencia de las notebooks corporativas es incorporar discos de estado sólido con capacidades de almacenamiento reducida, a menudo se utilizan en forma diaria como almacenamiento auxiliar. [19]

En versiones recientes de Windows se implementó una nueva funcionalidad que elimina las entradas antiguas de las claves USB y USBSTOR del registro del sistema. Existe una tarea programada llamada “Plug and Play Cleanup” que se encarga de la limpieza para aquellos dispositivos USB que no se han detectado luego de un tiempo (aproximadamente 30 días) [11], [21, p. 1]

El análisis forense de dispositivos USB es uno de los más complicados dentro del análisis forense de sistemas Windows. Si bien es posible obtener una gran cantidad de información, es necesario examinar y correlacionar múltiples artefactos diferentes, tal como se muestra en la figura siguiente:



La información que es posible obtener es, entre otra

- Nombre, marca y modelo de dispositivo.
- Tipo de dispositivo.
- Número de serie
- Fecha de primera y última inserción en el sistema
- Letra que se le asignó al dispositivo.
- Qué usuario insertó el dispositivo.
- Directorios y/o archivos que contenía el dispositivo
- Número de serie del sistema de archivos del dispositivo.
- Tamaño y almacenamiento libre.

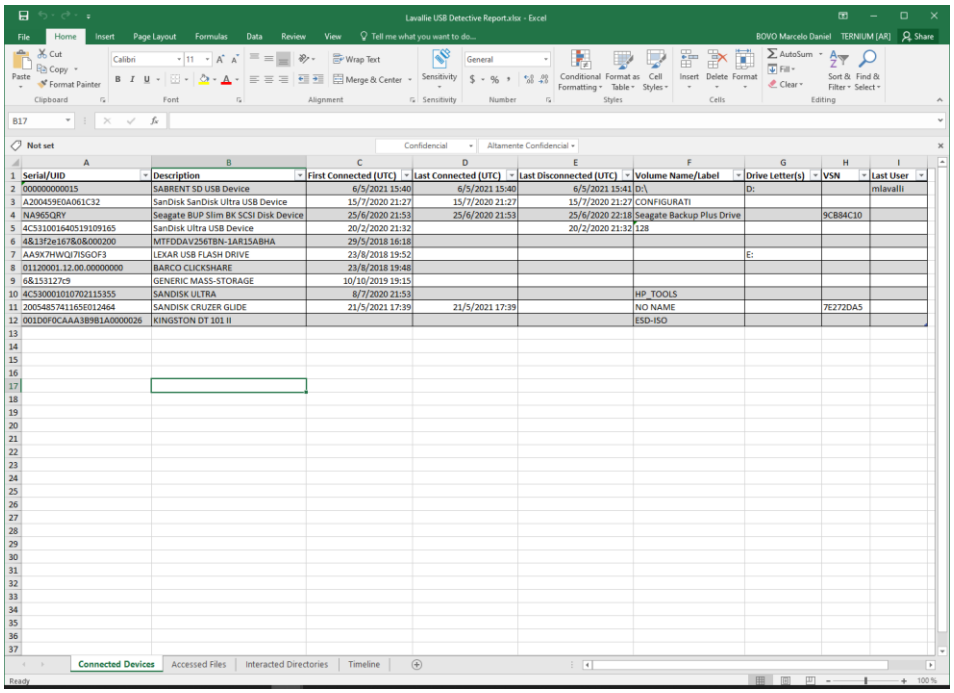
Debido a que esta tarea consume mucho tiempo y es fundamental para análisis de fuga de información, utilizamos USB Detective para automatizar todos estos análisis. Sin embargo, la automatización del análisis no exime al analista forense de conocer al

detalle los artefactos forenses analizados por la herramienta para poder validar las conclusiones.

Análisis con USB Detective:

USB Detective analiza en conjunto varios de los artefactos forenses discutidos anteriormente y genera reportes en formato Excel con los dispositivos USB conectados al sistema, los archivos/directorios accedidos en ellos y una línea de tiempo con la secuencia de accesos. Es la forma más completa y simple de hacer éste análisis, que de otra forma requiere correlacionar múltiples artefactos forenses.

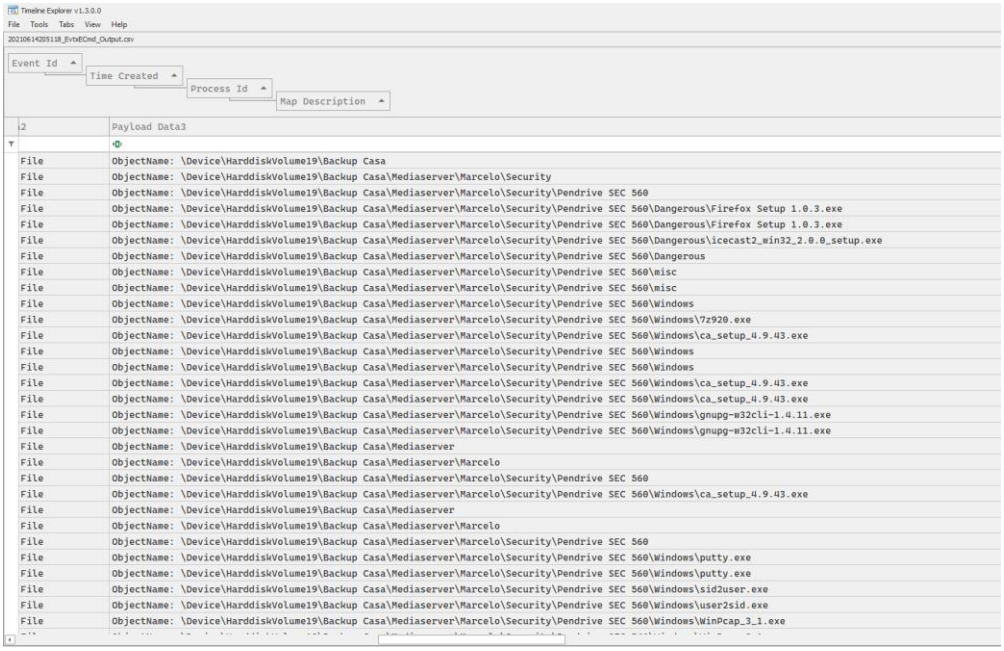
El inconveniente que tiene es que es una herramienta puramente gráfica y no acepta parámetros de línea de comandos. Por ello desarrollamos un Robot programado utilizando la librería FlaUI[22] que simula las acciones del usuario a fin de automatizar la generación de los reportes para la información extraída.



Serial/UID	Description	First Connected (UTC)	Last Connected (UTC)	Last Disconnected (UTC)	Volume Name/Label	Drive Letter(s)	VSN	Last User
000000000015	SABRENT SD USB Device	6/5/2021 15:40	6/5/2021 15:40	6/5/2021 15:41	D:\	D:		mlavalli
A200459E0A061C32	SanDisk SanDisk Ultra USB Device	15/7/2020 21:27	15/7/2020 21:27	15/7/2020 21:27	CONFIGURATI			
NA965QRY	Seagate BUP Slim BK SCSI Disk Device	25/6/2020 21:53	25/6/2020 21:53	25/6/2020 22:18	Seagate Backup Plus Drive		9CB84C10	
4C531001640519109165	SanDisk Ultra USB Device	20/2/2020 21:32		20/2/2020 21:32	128			
4&13f2e167&0&000200	MTFDDAV256TBN-1AR15ABHA	29/5/2018 16:18						
AA9X7HWQI7I5GOF3	LEXAR USB FLASH DRIVE	23/8/2018 19:52				E:		
01120001.12.00.00000000	BARCO CLICKSHARE	23/8/2018 19:48						
6&153127c9	GENERIC MASS-STORAGE	10/10/2019 19:15						
4C530001010702115355	SANDISK ULTRA	8/7/2020 21:53			HP_TOOLS			
200548574116 5E012464	SANDISK CRUZER GLIDE	21/5/2021 17:39	21/5/2021 17:39		NO NAME		7E272DA5	
001D0F0CAAA3B9B1A0000026	KINGSTON DT 101 II				ESD-ISO			

Análisis:

En la carpeta USBDetective de la salida de KAPE se encuentran los reportes generados por la herramienta.



Análisis de Archivos Borrados

Nivel: 2

Existen al menos tres enfoques complementarios para determinar el borrado de archivos en un filesystem NTFS sobre Windows 7 y posteriores:

- Análisis de la papelera de reciclaje
- Análisis del $USNJRNL (más $LOGFILE y $MFT)
- Análisis de Shadow Copies

Papelera de Reciclaje

En Windows 7 y posteriores la información de los archivos borrados a través de la papelera de reciclaje se almacena de la siguiente manera:

- No van a la papelera archivos borrados del sistema o de subcarpetas de “Program Files” (se borran directo).
- La información se encuentra en $Recycle.bin en el directorio raíz de cada volumen (partición). Dentro de ese directorio hay

un directorio por usuario del sistema identificado por su SID. Es decir que es posible determinar qué usuario del sistema borró determinado archivo ya que está en su recycle bin personal.

- Dentro del recycle bin de cada usuario hay dos conjuntos de archivos:
    - Archivos de la forma $Ixxxxx.crt son los archivos que contienen la metadata del archivo borrado: fecha y hora de borrado, nombre original, etc.
    - Archivos de la forma $Rxxxx (con xxxx coincidente con los del archivo de metadata): son los archivos borrador renombrados.

Por otra parte, la fecha de creación del archivo de metadata coincide con la del atributo “Date Recycled”. En el caso del ejemplo, el archivo borrado es en realidad un directorio, por lo cual si vamos a analizar el $Rxxxx correspondiente no tendrá información analizable.

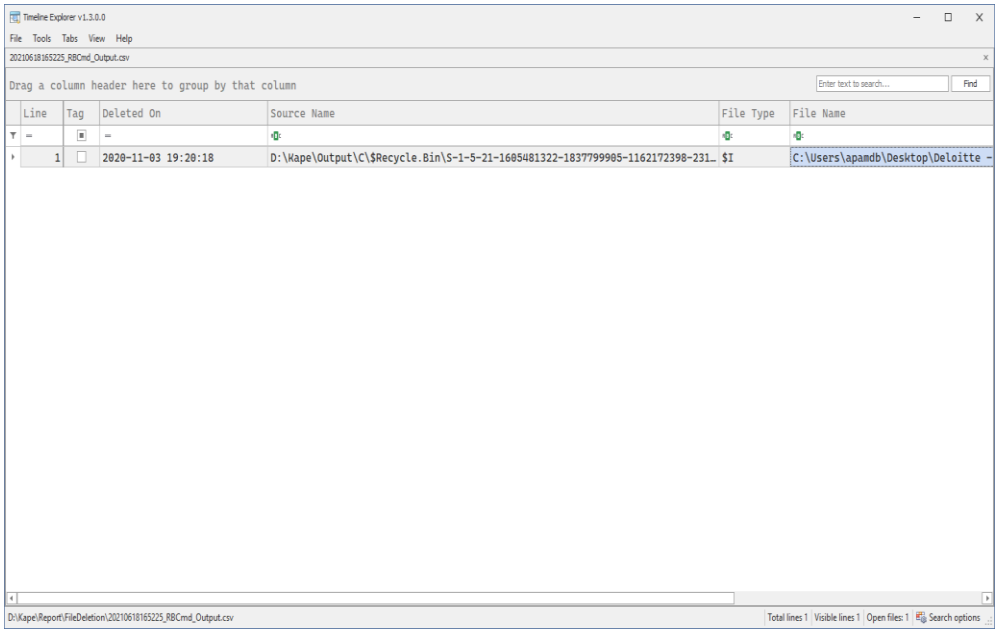
Es posible deducir cuales fueron los archivos borrados que estaban en ese directorio utilizando el hecho de que se mueven a la papelera de reciclaje en forma recursiva, por lo cual ordenando por fecha de creación de archivos $Ixxxx los archivos que se hayan creado inmediatamente después que el perteneciente al directorio padre probablemente serán archivos que estaban en ese directorio. Lo recomendable es ir avanzando hasta que la información que da el FTK de uno de los archivos $Ixxx diga que corresponde a otro directorio.

Si la papelera de reciclaje ya se borró (manualmente o a través del sistema) todos estos archivos estarán borrados (pero generalmente recuperables) en el sistema de archivos.

Una forma de obtener archivos borrados es mediante las “Shadow Copies”. Las “Shadow Copies” o “Volume Snapshot” son una característica del sistema de archivos NTFS que permite tomar una instantánea de los archivos del sistema a ese momento y preservarla.

Análisis:

Analizar el archivo TIMESTAMP_RBCmd_Output.csv de directorio “FileDeletion” de la salida de kape. En el estará decodificada toda la información de los Recycle Bins del sistema que incluye: Nombre original del archivo, ubicación original y fecha de borrado.



El metaarchivo $USNJrnl

La cantidad de registros disponibles depende de la utilización del sistema y puede ir desde un par de días a meses. En los análisis realizados en máquinas corporativas generalmente se pueden obtener algo más de una semana de registros.

El “USN Journal” o “Update Sequence Number Journal” se implementó por primera vez en Windows 2000, pero no se habilitó por defecto hasta Windows Vista[2]. La funcionalidad primaria del log es permitir a los programas determinar qué archivos cambiaron para poder replicarlos, backupearlos o comprimirlos, por ejemplo. [23]

[2] Nota: Se ha observado que está habilitado también en Windows XP, probablemente por la introducción del Service Pack 3

El log de cambios se almacena en el archivo **\$Extend\$UsrJrnl**. Este archivo contiene dos streams de datos (atributo $DATA): **$Max** contiene información básica acerca del log (incluyendo el tamaño máximo del log) y **$J** es el archivo que guarda el registro de las operaciones sobre el sistema de archivos. `$J` contiene un conjunto de registros de longitud variable, cada uno de los cuales contiene la fecha y el cambio realizado sobre el sistema de archivos. Cada uno de estos registros tiene asociado un *Update Sequence Number* (USN), que es un entero de 64 bits. El USN se utiliza para indexar los registros en el log y se guarda en el atributo $STANDARD_INFORMATION del archivo que fue modificado. [24]

En la práctica se ha observado que la cantidad de días cubiertos por este registro varía en gran medida dependiendo de la utilización del filesystem. Puede ir desde un par de meses a un par de días y en general parece ser que los principales responsables de la generación de eventos en el log suelen ser las herramientas antivirus.

Los datos completos de cada entrada del log son:

- Timestamp
- Operación que afectó al archivo
- Atributos del archivo o directorio
- Nombre del archivo o directorio
- Número de referencia del archivo o directorio
- Número de referencia del archivo o directorio padre del modificado
- ID de Seguridad
- “update sequence number” (USN) del registro

TABLE 11-7 Change Journal Change Reasons

Identifier	Reason
USN_REASON_DATA_OVERWRITE	The data in the file or directory was overwritten
USN_REASON_DATA_EXTEND	Data was added to the file or directory
USN_REASON_DATA_TRUNCATION	The data in the file or directory was truncated
USN_REASON_NAMED_DATA_OVERWRITE	The data in a file's data stream was overwritten
USN_REASON_NAMED_DATA_EXTEND	The data in a file's data stream was extended
USN_REASON_NAMED_DATA_TRUNCATION	The data in a file's data stream was truncated
USN_REASON_FILE_CREATE	A new file or directory was created
USN_REASON_FILE_DELETE	A file or directory was deleted
USN_REASON_EA_CHANGE	The extended attributes for a file or directory changed
USN_REASON_SECURITY_CHANGE	The security descriptor for a file or directory was changed
USN_REASON_RENAME_OLD_NAME	A file or directory was renamed; this is the old name
USN_REASON_RENAME_NEW_NAME	A file or directory was renamed; this is the new name
USN_REASON_INDEXABLE_CHANGE	The indexing state for the file or directory was changed (whether or not the Indexing service will process this file or directory)
USN_REASON_BASIC_INFO_CHANGE	The file or directory attributes and/or the time stamps were changed
USN_REASON_HARD_LINK_CHANGE	A hard link was added or removed from the file or directory
USN_REASON_COMPRESSION_CHANGE	The compression state for the file or directory was changed
USN_REASON_ENCRYPTION_CHANGE	The encryption state (EFS) was enabled or disabled for this file or directory
USN_REASON_OBJECT_ID_CHANGE	The object ID for this file or directory was changed
USN_REASON_REPARSE_POINT_CHANGE	The reparse point for a file or directory was changed, or a new reparse point (such as a symbolic link) was added or deleted from a file or directory
USN_REASON_STREAM_CHANGE	A new data stream was added to or removed from a file or renamed
USN_REASON_TRANSACTED_CHANGE	This value is added (ORed) to the change reason to indicate that the change was the result of a recent *commit* of a TxF transaction
USN_REASON_CLOSE	The handle to a file or directory was closed, indicating that this is the final modification made to the file in this series of operations

Análisis del USNJRNL

NTFS es lo que se denomina un “journaled-filesystem”, es decir que, ante la falla de una operación sobre él, existe un registro que permite volver al último estado válido y por lo tanto prevenir la pérdida de información. Para eso utiliza varios logs en donde registra cada una de las operaciones que se realizan. Esto logs pueden ser utilizados para análisis forense.

Existen tres artefactos del sistema de archivos que pueden utilizarse, en forma conjunta o por separado, para reconstruir las actividades que se han realizado sobre él: $MFT, $Logfile y $USNJrnl.

Existen varias utilidades que permiten analizar el USNJournal. La diferencia básica entre ellas es si utilizan sólo la información del $USNJRNL o también analizan el $MFT. El análisis del $MFT es útil para determinar la ubicación exacta del archivo borrado en el filesystem, ya que en el $USNJRNL sólo se almacenan el nombre del

archivo (sin el path) y el puntero hacia el registro correspondiente en la $MFT.

Aunque pueden encontrarse varias utilidades más, solo analizo aquellas que pude probar y cuyos resultados coinciden entre sí.

Al momento de analizar un log, es conveniente recordar lo siguiente:

- El número de referencia del archivo en NTFS es único mientras exista el archivo, pero se reutiliza frecuentemente cuando el archivo se borra. No cambia cuando el archivo es renombrado.
- Cuando el número de referencia (MFTReference) se utiliza para un nuevo archivo, se incrementa el MFTReferenceSeqNo. Esto se puede utilizar para determinar cuándo se trata del mismo archivo.
- Cuando se mueve un directorio a la papelera, cambia el nombre del directorio a un archivo de la forma $Rxxxxx, pero los archivos/subdirectorios no cambian de nombre. Sólo se modifica metadata del archivo mediante la operación USN_SECURITY_CHANGE
- Todos los timestamps en NTFS son UTC, es decir GMT0.

Este es un ejemplo de un MFTReference que se reutiliza varias

1	Offset	FileName	USN	Timestamp	Reason	MFTReference	MFTReferenceSeqNo	MFTParentReference	MFTPa	FileAttributes
219269	0x00000000B1389960	Sen6286.tmp.tag	2973276512	2015-06-30 16:53:12.743:3590	FILE_CREATE	113922	511	105306	1	archive
219270	0x00000000B13899C0	Sen6286.tmp.tag	2973276608	2015-06-30 16:53:12.743:3590	CLOSE+FILE_CREATE	113922	511	105306	1	archive
219271	0x00000000B1389A20	Sen6286.tmp.tag	2973276704	2015-06-30 16:53:12.744:3590	DATA_EXTEND	113922	511	105306	1	archive
219272	0x00000000B1389A80	Sen6286.tmp.tag	2973276800	2015-06-30 16:53:12.744:3590	CLOSE+DATA_EXTEND	113922	511	105306	1	archive
219274	0x00000000B1389B38	Sen6286.tmp.tag	2973276984	2015-06-30 16:53:12.744:3590	CLOSE+FILE_DELETE	113922	511	105306	1	archive
219298	0x00000000B138A3B8	UTC2.tmp.tag	2973279160	2015-06-30 16:53:16.185:2007	FILE_CREATE	113922	512	105032	13	archive
219299	0x00000000B138A410	UTC2.tmp.tag	2973279248	2015-06-30 16:53:16.185:2007	CLOSE+FILE_CREATE	113922	512	105032	13	archive
219300	0x00000000B138A468	UTC2.tmp.tag	2973279336	2015-06-30 16:53:16.185:2007	DATA_EXTEND	113922	512	105032	13	archive
219301	0x00000000B138A4C0	UTC2.tmp.tag	2973279424	2015-06-30 16:53:16.185:2007	CLOSE+DATA_EXTEND	113922	512	105032	13	archive
219303	0x00000000B138A568	UTC2.tmp.tag	2973279592	2015-06-30 16:53:16.186:1995	CLOSE+FILE_DELETE	113922	512	105032	13	archive
219313	0x00000000B138A8D8	ClientAllSetting.ini.tag	2973280472	2015-06-30 16:53:17.295:8663	FILE_CREATE	113922	513	105032	13	archive
219314	0x00000000B138A948	ClientAllSetting.ini.tag	2973280584	2015-06-30 16:53:17.295:8663	CLOSE+FILE_CREATE	113922	513	105032	13	archive
219315	0x00000000B138A9B8	ClientAllSetting.ini.tag	2973280696	2015-06-30 16:53:17.295:8663	DATA_EXTEND	113922	513	105032	13	archive
219316	0x00000000B138AA28	ClientAllSetting.ini.tag	2973280808	2015-06-30 16:53:17.295:8663	CLOSE+DATA_EXTEND	113922	513	105032	13	archive
219318	0x00000000B138AB00	ClientAllSetting.ini.tag	2973281024	2015-06-30 16:53:17.295:8663	CLOSE+FILE_DELETE	113922	513	105032	13	archive
219608	0x00000000B1391180	NewDomain.dat.tag	2973307312	2015-06-30 16:53:21.191:1863	FILE_CREATE	113922	514	105032	13	archive
219609	0x00000000B1391210	NewDomain.dat.tag	2973307408	2015-06-30 16:53:21.191:1863	CLOSE+FILE_CREATE	113922	514	105032	13	archive
219610	0x00000000B1391270	NewDomain.dat.tag	2973307504	2015-06-30 16:53:21.191:1863	DATA_EXTEND	113922	514	105032	13	archive
219611	0x00000000B13912D0	NewDomain.dat.tag	2973307600	2015-06-30 16:53:21.192:1851	CLOSE+DATA_EXTEND	113922	514	105032	13	archive
219613	0x00000000B1391388	NewDomain.dat.tag	2973307784	2015-06-30 16:53:21.192:1851	CLOSE+FILE_DELETE	113922	514	105032	13	archive
219659	0x00000000B1392558	cfgall.ini.tag	2973312344	2015-06-30 16:53:26.618:6655	FILE_CREATE	113922	515	105032	13	archive
219660	0x00000000B13925B0	cfgall.ini.tag	2973312432	2015-06-30 16:53:26.618:6655	CLOSE+FILE_CREATE	113922	515	105032	13	archive
219661	0x00000000B1392608	cfgall.ini.tag	2973312520	2015-06-30 16:53:26.618:6655	DATA_EXTEND	113922	515	105032	13	archive
219662	0x00000000B1392660	cfgall.ini.tag	2973312608	2015-06-30 16:53:26.618:6655	CLOSE+DATA_EXTEND	113922	515	105032	13	archive
219664	0x00000000B1392708	cfgall.ini.tag	2973312776	2015-06-30 16:53:26.619:6643	CLOSE+FILE_DELETE	113922	515	105032	13	archive
219691	0x00000000B13930A0	alerts.ini.tag	2973315232	2015-06-30 16:53:26.974:2383	FILE_CREATE	113922	516	105032	13	archive
219692	0x00000000B13930F8	alerts.ini.tag	2973315320	2015-06-30 16:53:26.974:2383	CLOSE+FILE_CREATE	113922	516	105032	13	archive
219695	0x00000000B13931F0	alerts.ini.tag	2973315568	2015-06-30 16:53:26.974:2383	DATA_EXTEND	113922	516	105032	13	archive
219696	0x00000000B1393248	alerts.ini.tag	2973315656	2015-06-30 16:53:26.974:2383	CLOSE+DATA_EXTEND	113922	516	105032	13	archive

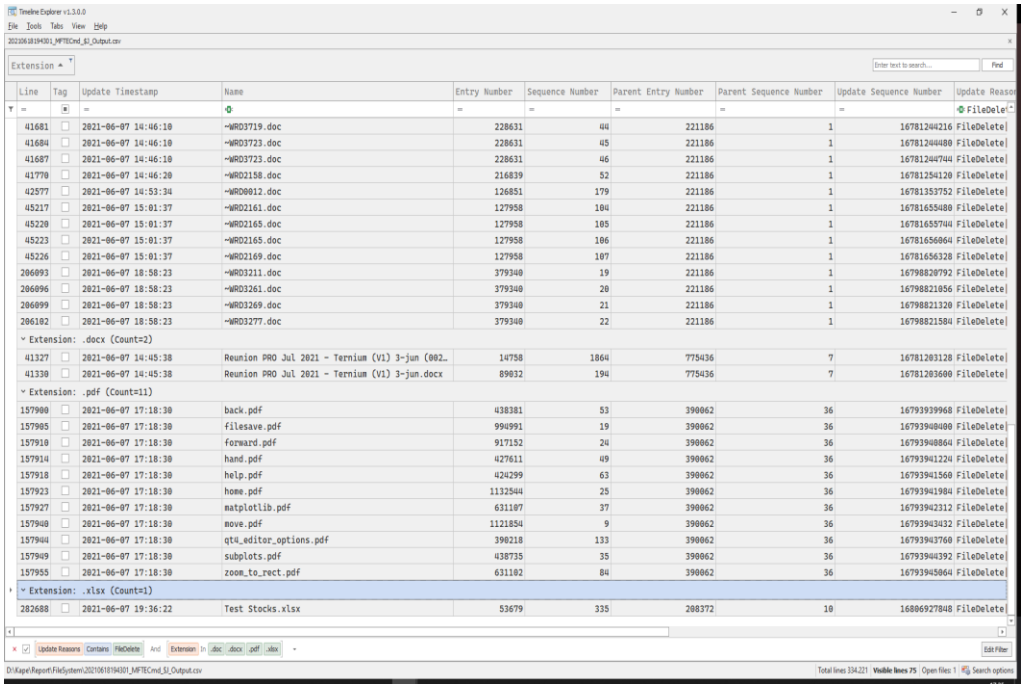
veces:

En la próxima imagen podemos ver el seguimiento de un directorio denominado “Argentina SAP”, que se mueve a la papelera de reciclaje. El movimiento a la papelera hace que cambie de nombre a $RC770DR y se termina borrando cuando se vacía la papelera 3 minutos después. Puede verse que tanto el número de referencia MFT como el número de secuencia permanecen constantes hasta que se elimina.

1	Offset	FileName	USN	Timestamp	Reason	MFTReference	MFTReferenceSeqNo	MFTParentReference	MFTPa	FileAttributes
361056	0x00000000B206F320	Argentina SAP	2986799904	2015-07-02 14:22:14.032:8302	RENAME_OLD_NAME	143874	76	106555	5	directory
361057	0x00000000B206F378	$RC770DR	2986799992	2015-07-02 14:22:14.032:8302	RENAME_NEW_NAME	143874	76	107430	6	directory
361058	0x00000000B206F3C8	$RC770DR	2986800072	2015-07-02 14:22:14.048:4279	CLOSE+RENAME_NEW_NAME	143874	76	107430	6	directory
362585	0x00000000B2096760	$RC770DR	2986960736	2015-07-02 14:22:14.422:7727	SECURITY_CHANGE	143874	76	107430	6	directory
362586	0x00000000B20967B0	$RC770DR	2986960816	2015-07-02 14:22:14.422:7727	CLOSE+SECURITY_CHANGE	143874	76	107430	6	directory
369083	0x00000000B212F2C0	$RC770DR	2987586240	2015-07-02 14:25:39.418:4055	CLOSE+FILE_DELETE	143874	76	107430	6	directory

Análisis:

Examinar el archivo TIMESTAMP_MFTECmd_$J_Output.csv de la carpeta Filesystem de la salida de Kape con TimelineExplorer. Filtrar por UpdateReasons contains “File Delete”, agrupar por “Extension” y filtrar las extensiones por los tipos de archivo relevantes (por ejemplos xls, doc, pdf).



Programas ejecutados (a través de Shimcache/Prefetch)

Nivel: 3

Los artefactos forenses Amcache y Shimcache pueden analizarse para determinar cuándo se ejecutó un programa y la primera y última vez que fue modificado. Adicionalmente proveen información acerca de la ubicación del archivo, su tamaño e incluso su hash (dependiendo de la versión)[25] [26]

La ubicación de Amcache.hve es:

```
\%SystemRoot%\AppCompat\Programs\Amcache.hve
```

Por su parte, el Shimcache (también conocido como AppCompatCache) es parte de la funcionalidad de Windows que permite modificar las condiciones de entorno de programas desarrollados para versiones anteriores del sistema operativo a fin de que puedan correr correctamente.

Este artefacto forense almacena la siguiente información:

- Path completo del archivo
- Tamaño
- Fecha de última modificación (sacada de $Standard_Information)
- Fecha de última modificación del Shimcache
- Flag de ejecución del programa (si fue ejecutado o no).

Esta información se encuentra almacenada en:

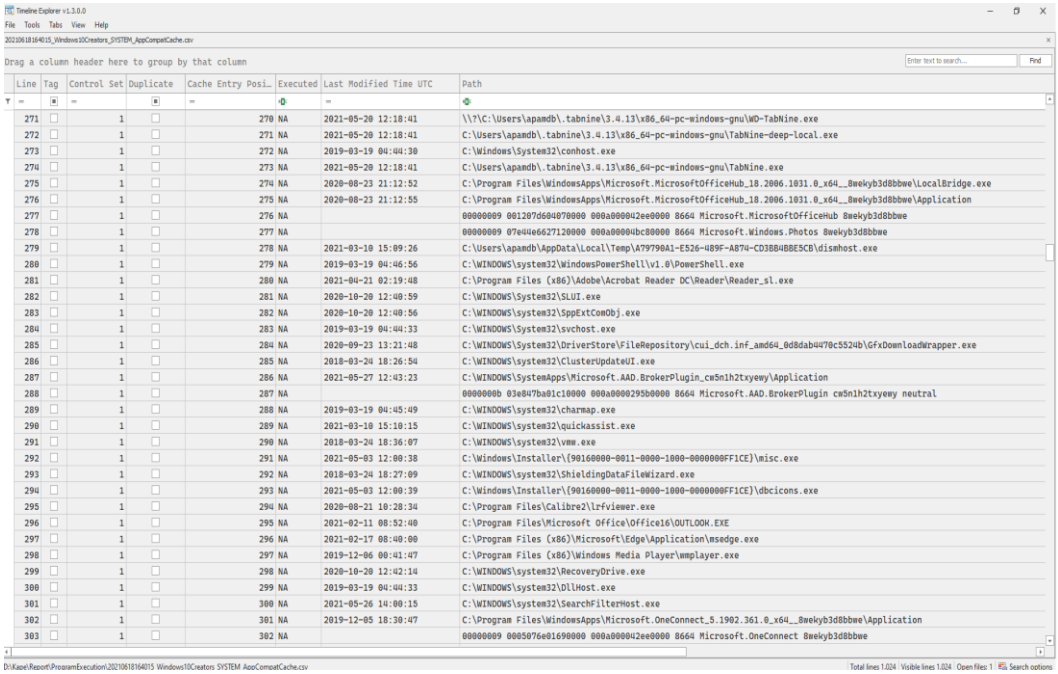
```
HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatCache\
AppCompatCache
```

Windows Server 2003 = 512 entradas

Win7-10, Server 2008/2012/2016 = 1024 entradas

Análisis:

En el directorio ProgramExecution de la salida de Kape analizar los archivos surgidos del análisis de Amcache y AppcompatCache.



Timeline Explorer v1.3.0.0

File Tools Tabs View Help

20210618164015_Windows10Creators_SYSTEM_AppCompatCache.csv

Drag a column header here to group by that column

Enter text to search... Find

Line	Tag	Control Set	Duplicate	Cache Entry Posi…	Executed	Last Modified Time UTC	Path
271		1		270	NA	2021-05-20 12:18:41	\\?\C:\Users\apamdb\.tabnine\3.4.13\x86_64-pc-windows-gnu\WD-TabNine.exe
272		1		271	NA	2021-05-20 12:18:41	C:\Users\apamdb\.tabnine\3.4.13\x86_64-pc-windows-gnu\TabNine-deep-local.exe
273		1		272	NA	2019-03-19 04:44:30	C:\Windows\System32\conhost.exe
274		1		273	NA	2021-05-20 12:18:41	C:\Users\apamdb\.tabnine\3.4.13\x86_64-pc-windows-gnu\TabNine.exe
275		1		274	NA	2020-08-23 21:12:52	C:\Program Files\WindowsApps\Microsoft.MicrosoftOfficeHub_18.2006.1031.0_x64__8wekyb3d8bbwe\LocalBridge.exe
276		1		275	NA	2020-08-23 21:12:55	C:\Program Files\WindowsApps\Microsoft.MicrosoftOfficeHub_18.2006.1031.0_x64__8wekyb3d8bbwe\Application
277		1		276	NA		00000009 001207d604070000 000a0000d2ee0000 8664 Microsoft.MicrosoftOfficeHub 8wekyb3d8bbwe
278		1		277	NA		00000009 07e44e6627120000 000a0000dbc80000 8664 Microsoft.Windows.Photos 8wekyb3d8bbwe
279		1		278	NA	2021-03-10 15:09:26	C:\Users\apamdb\AppData\Local\Temp\A79790A1-E526-4B9F-A874-CD3BB4BBE5CB\dismhost.exe
280		1		279	NA	2019-03-19 04:46:56	C:\WINDOWS\system32\WindowsPowerShell\v1.0\PowerShell.exe
281		1		280	NA	2021-04-21 02:19:48	C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\Reader_sl.exe
282		1		281	NA	2020-10-20 12:40:59	C:\WINDOWS\System32\SLUI.exe
283		1		282	NA	2020-10-20 12:40:56	C:\WINDOWS\system32\SppExtComObj.exe
284		1		283	NA	2019-03-19 04:44:33	C:\WINDOWS\System32\svchost.exe
285		1		284	NA	2020-09-23 13:21:48	C:\WINDOWS\System32\DriverStore\FileRepository\cui_dch.inf_amd64_0d8dab4470c5524b\GfxDownloadWrapper.exe
286		1		285	NA	2018-03-24 18:26:54	C:\WINDOWS\system32\ClusterUpdateUI.exe
287		1		286	NA	2021-05-27 12:43:23	C:\WINDOWS\SystemApps\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\Application
288		1		287	NA		0000000b 03e847ba01c10000 000a0000295b0000 8664 Microsoft.AAD.BrokerPlugin cw5n1h2txyewy neutral
289		1		288	NA	2019-03-19 04:45:49	C:\WINDOWS\system32\charmap.exe
290		1		289	NA	2021-03-10 15:10:15	C:\WINDOWS\system32\quickassist.exe
291		1		290	NA	2018-03-24 18:36:07	C:\WINDOWS\system32\vmw.exe
292		1		291	NA	2021-05-03 12:00:38	C:\Windows\Installer\{90160000-0011-0000-1000-0000000FF1CE}\misc.exe
293		1		292	NA	2018-03-24 18:27:09	C:\WINDOWS\system32\ShieldingDataFileWizard.exe
294		1		293	NA	2021-05-03 12:00:39	C:\Windows\Installer\{90160000-0011-0000-1000-0000000FF1CE}\dbcicons.exe
295		1		294	NA	2020-08-21 10:28:34	C:\Program Files\Calibre2\lrfviewer.exe
296		1		295	NA	2021-02-11 08:52:40	C:\Program Files\Microsoft Office\Office16\OUTLOOK.EXE
297		1		296	NA	2021-02-17 08:40:00	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe
298		1		297	NA	2019-12-06 00:41:47	C:\Program Files (x86)\Windows Media Player\wmplayer.exe
299		1		298	NA	2020-10-20 12:42:14	C:\WINDOWS\system32\RecoveryDrive.exe
300		1		299	NA	2019-03-19 04:44:33	C:\WINDOWS\system32\DllHost.exe
301		1		300	NA	2021-05-26 14:00:15	C:\WINDOWS\system32\SearchFilterHost.exe
302		1		301	NA	2019-12-05 18:30:47	C:\Program Files\WindowsApps\Microsoft.OneConnect_5.1902.361.0_x64__8wekyb3d8bbwe\Application
303		1		302	NA		00000009 0005076e01690000 000a0000d2ee0000 8664 Microsoft.OneConnect 8wekyb3d8bbwe

D:\Kape\Report\ProgramExecution\20210618164015_Windows10Creators_SYSTEM_AppCompatCache.csv

Total lines 1.024 | Visible lines 1.024 | Open files: 1 | Search options

Redacción del Reporte

Consideraciones preliminares

Un informe forense, al menos en el ámbito corporativo, debería contener una descripción del proceso de identificación, extracción, procesamiento y análisis de la imagen forense además de las conclusiones a las que fue posible llegar como resultado de todo el proceso.

El informe debe ser redactado pensando en la audiencia, es decir, si va dirigido a niveles de alta gerencia o dirección es conveniente que se centre en los hechos detectados con sólo los detalles técnicos indispensables que den sustento a la afirmación. Si, en cambio, está dirigido a una audiencia más técnica, podrán agregarse más detalles técnicos del análisis realizado.

Independientemente de la audiencia siempre debe contener la información mínima que permita repetir el análisis para poder contrastar resultados y los datos básicos que permitan afirmar que se ha existido una adecuada cadena de custodia y que la extracción forense no ha contaminado la evidencia.

Por supuesto, sólo se debe afirmar aquello que se está en condiciones de afirmar y, si realmente aporta a las conclusiones, expresar el resto de los hallazgos como evidencia adicional.

Estructura básica del informe

A continuación, se propone una estructura posible de un informe forense basado en las consideraciones preliminares y en las prácticas corporativas habituales. Esto no pretende ser un esquema final sino un punto de partida para que cada quién estructure su reporte de la manera que mejor se adapte a sus propósitos.

- Objetivo: Breve explicación de qué preguntas se pretendieron responder mediante el análisis forense de la información extraída.
- Antecedentes: En esta sección habitualmente se detallan los datos del equipo analizado (marca, modelo, número de serie e inventario), a quién estaba asignado y cómo se extrajo la información forense. Es importante especificar el rango de tiempo durante el cual se realizó el proceso, las herramientas utilizadas y cualquier eventualidad que haya sucedido (por ejemplo, si la máquina estaba en hibernación y al abrirla se inició).
- Conclusión: Este es un resumen en el leguaje más claro y menos técnico posible respondiendo las preguntas que eran el objetivo del trabajo.
- Descripción del proceso: Aquí se detallan todos los procesos que se realizaron sobre el equipo en cuestión y sobre otra información complementaria como, por ejemplo, logs del proxy corporativo, logs de eventos de Active Directory, etc. Es importante que cada conclusión esté soportada por la evidencia que permitió llegar a ella, desde el archivo origen en el filesystem (incluyendo hash de verificación[3]), pasando por la información extraída de ese artefacto, hasta su interpretación en acciones ejecutadas en el sistema. Todos los datos que son demasiado detallados para un informe deben ser preservados para una eventual presentación judicial.
- Anexos: En los anexos se adjunta toda aquella información generada durante los procesos de

[3] El estándar de la industria es calcular tanto el hash MD5 como el SHA1 de cada archivo. Una de las razones de esta práctica es que, si bien existen algoritmos para encontrar colisiones tanto en MD5 como en SHA1, encontrar una colisión simultánea en MD5 y SHA1 es prácticamente imposible. De esta manera podemos asegurar sin lugar a dudas la integridad de los artefactos analizados.

extracción, procesamiento y análisis que es demasiado extensa para formar parte de la descripción del proceso ya que dificultaría su lectura e interpretación. Aquí normalmente pertenecen los logs del proceso de extracción, capturas de pantalla, listado de archivos, etc.

Conclusiones

Cada vez son más frecuentes en organizaciones de distinta índole, el análisis forense de los equipos de los empleados a fin de detectar fugas de información restringida o propietaria.

Los vectores de exfiltración son muy diversos y no siempre es posible detectar que se ha extraído información ya sea porque los dispositivos utilizados están fuera del alcance de la organización o bien porque no existen registros o artefactos forenses que permitan determinar las acciones realizadas en forma fehaciente.

En este trabajo se compilaron las lecciones aprendidas durante varios años de análisis de este tipo y se realizó una correlación entre las diferentes acciones que realiza una persona que pretende extraer información y los artefactos forenses que se generan en su computador asignado a partir de esas acciones.

En primer lugar, se diseñaron dos métodos de extracción automatizada de los artefactos forenses: el primero a partir de una imagen forense del equipo a analizar y mediante un software desarrollado a tal fin; el segundo mediante un proceso de triage remoto utilizando una herramienta comercial denominada F-Response y un software gratuito de extracción llamado Kape.

La herramienta diseñada para extracción forense es capaz de sortear todas las restricciones que aparecen en un trabajo de campo incluyendo: falta de permisos de administración en el equipo utilizado, múltiples formatos de imágenes forenses, equipos con encripción de disco completo y artefactos forenses que varían de caso en caso. El resultado es una herramienta que se ha utilizado ininterrumpidamente desde 2017 en múltiples locaciones y sin problemas relevantes.

La segunda opción de extracción, combina dos herramientas reconocidas en el ambiente forense. F-Response permite mapear la memoria y los dispositivos de almacenamiento de un equipo remoto como si fueran locales y Kape permite automatizar la extracción de

artefactos forenses aún si los mismos se encuentran bloqueados por el sistema operativo o en un Virtual SnapShot. Este método permite la extracción focalizada de aquellos artefactos forenses indispensables para el análisis desde un equipo funcionando, en forma remota y en menor tiempo. Esto permite obtener información desde locaciones en las que no hay posibilidad de tomar una imagen forense a expensas de no tener la información completa si es que se necesita profundizar el análisis.

Una vez realizada la extracción forense, el pre-análisis de la información se realiza mediante otra funcionalidad de Kape que permite automatizar la ejecución de herramientas forenses.

Como se ha dicho en la introducción del trabajo, no existe una revisión forense completa. Es decir, es imposible analizar toda la información forense presente en un equipo y lo que en cambio se hace es analizar los artefactos forenses relevantes para poder responder las preguntas objeto de la investigación. En este sentido la guía de análisis forense desarrollada permite priorizar, a través de una clasificación de prioridad, qué artefactos es necesario analizar en primera instancia. De esta manera, será posible obtener evidencia relevante en forma temprana, orientando la investigación en base a los hallazgos.

La automatización lograda no sólo permite acortar los tiempos de análisis para llegar a conclusiones relevantes, sino que permite estandarizar qué artefactos forenses se analizan y de qué modo. Esta estandarización es fundamental dentro de una organización ya que permite conservar el conocimiento adquirido y garantizar estándares mínimos de calidad en los trabajos realizados.

La guía de análisis no es sólo la consolidación de todos los artefactos forenses que proveen información relevante, sino que también permite consolidar y sistematizar los conocimientos adquiridos durante años de realizar revisiones de este tipo.

Es importante destacar que las conclusiones a las que se llega luego de un análisis de este tipo apuntan a determinar acciones de personas que pueden haber incumplido leyes y/o normas internas de la organización. Es por eso que es fundamental la rigurosidad con la que se encara el análisis y se emiten las conclusiones ya que ellas afectar a una o más personas. Es decir, sólo se debe afirmar aquello de lo que se está 100% seguro.

Apéndice I

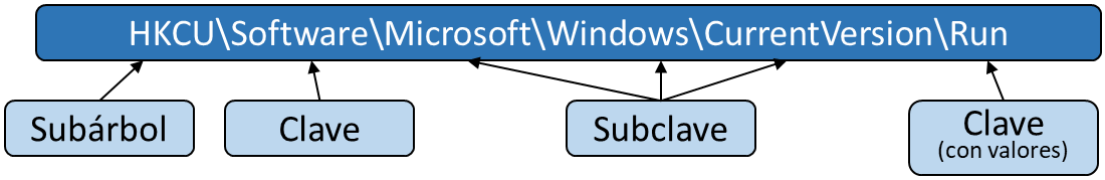
Registro del Sistema

El registro del sistema (Registry según su denominación en inglés), es una colección jerárquica de archivos de base de datos que almacena información de configuración para los sistemas operativos Windows. Almacena configuraciones de hardware, usuarios y privilegios, software instalado en el sistema, preferencias del usuario, etc. Se desarrolló para reemplazar los archivos de configuración .ini, utilizados en sistemas Windows 3.x y anteriores.

Físicamente está compuesta por dos tipos de bases de datos, las que almacenan información general del sistema y afectan a todos los usuarios y las que guardan información de un usuario en particular[4]. El contenido de estas bases de datos es ensamblado por el sistema una estructura jerárquica transparente para el usuario.

Su estructura lógica tiene los siguientes componentes:

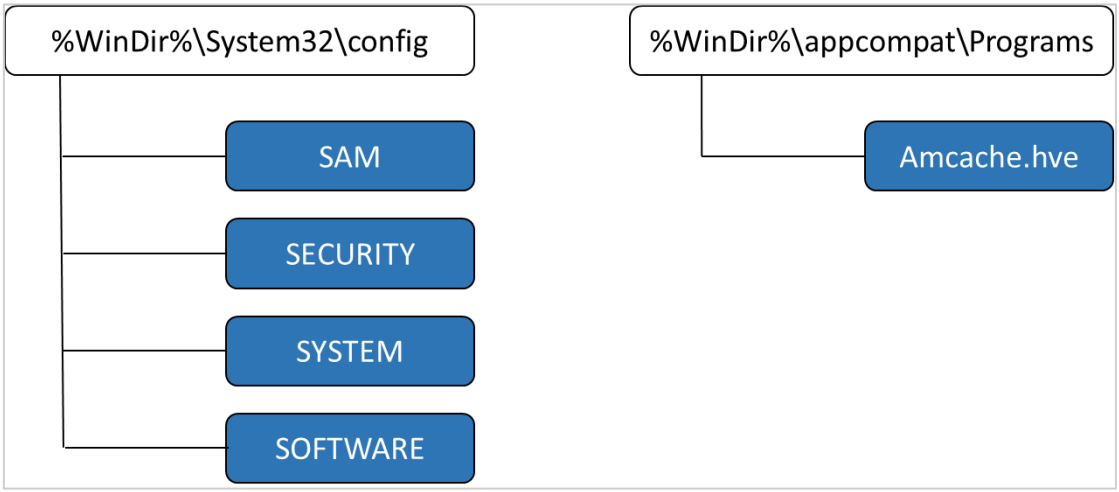
- Claves: Son similares a carpetas (claves) y subcarpetas (sub-claves) y son las que implementan la estructura jerárquica de directorios.
- Valores: Son los datos almacenados dentro de una clave y tienen diferentes tipos de datos según la información que almacenan: entero, cadena, listas o información binaria.



El registro tiene cuatro subárboles (“Hives” en inglés) o secciones lógicas principales:

- HKEY_ CLASSES ROOT: información sobre aplicaciones registradas
- HKEY_CURRENT_USER: preferencias del usuario autenticado actualmente en el sistema
- HKEY_ LOCAL MACHINE: configuraciones específicas del equipo local
- HKEY_ USERS: preferencias de cada uno de los usuarios que tienen un perfil en el sistema. La del usuario actual se monta además en HKEY_CURRENT_USER

Los archivos bases de datos relacionadas con la configuración del sistema se guardan en dos directorios diferentes bajo %WINDIR%\system32, tal como se muestra en la figura a continuación



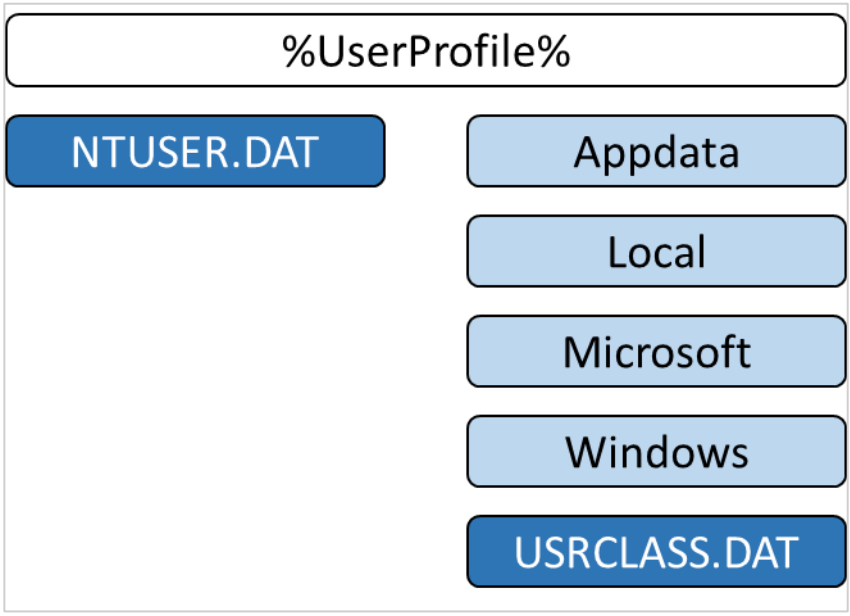
Las bases de datos almacenadas en %WINDIR%\system32\config han estado presentes desde la introducción de la registry, mientras que la base Amcache.hve fue introducida en Windows 8 y está relacionada con los programas que se han corrido recientemente en el sistema.

Además, cada 10 días en sistemas Windows Vista o posteriores se realiza un backup de las bases del sistema en

%WINDIR%\system32\config\backup. Estas pueden ser útil para detectar información que se haya eliminado del registro activo.

El Sistema monta SAM, SECURITY, SOFTWARE y SYSTEM bajo el subárbol HKEY_LOCAL_MACHINE. Cada una de ellas contiene información específica de un aspecto del sistema:

- SAM (HKLM\SAM): Información de usuarios locales y grupos.
- SECURITY (HKLM\Security): Información de seguridad asociada con la SAM.
- SYSTEM (HKLM\System): Información del hardware y servicios del sistema.
- SOFTWARE (HKLM\Software): Configuración del software instalado en el sistema incluyendo Windows.
- AMCACHE.HVE: Utilizada en forma interna por el Sistema para garantizar compatibilidad de aplicaciones.



Las bases de datos NTUSER.DAT y USRCLASS.DAT contienen información relacionada con el usuario, por lo tanto, existe una copia de cada una de ellas por cada perfil de usuario presente en el equipo. Se montan en HKEY_USER o HKEY_CURRENT_USER dependiendo de si el usuario es o no el que inició la sesión activa.

Al igual que sucede con otros formatos de almacenamiento binario, los valores borrados en la registry sólo se marcan como borrados y su información está disponible hasta que el espacio físico que ocupan se sobreescriba con nueva información. La mayoría de las herramientas de análisis forense del registro de Windows son capaces de recuperar información borrada.

Por otro lado, Windows no escribe la información modificada del registro en forma inmediata, sino que primero la guarda en memoria. Luego, realiza lo que se denomina un “hive flush” se guarda temporalmente en archivos de logs denominados con el nombre de la base y la extensión LOG1 y LOG2 (por ejemplo, SAM.LOG1 y SAM.LOG2 para la SAM). Estas transacciones se escriben en su ubicación definitiva una vez por hora o cuando el sistema no está siendo utilizado.

Las implicaciones desde el punto de vista forense son que, si no se fuerza un commit de las transacciones pendientes en los archivos de logs, podemos perder las modificaciones que se hayan realizado en la última hora aproximadamente. Dependiendo del caso también puede ser deseable analizar la registry sin hacer un commit de los logs de transacciones para evitar que algún valor se sobreescriba.[27]

Las herramientas que utilizamos para el análisis automáticamente hacen un commit de los logs de transacciones para mostrarnos la última versión de la registry.

Apéndice II

Análisis de Archivos Borrados

Existen dos enfoques complementarios para determinar el borrado de archivos en un filesystem NTFS sobre Windows 7 y posteriores:

- Análisis del Recycle Bin
- Análisis del $USNJRNL (más $LOGFILE y $MFT)
- “Virtual Snapshot” o “Shadow Copies”

Recycle Bin

En Windows 7 y posteriores la información de los archivos borrados a través de la papelera de reciclaje se almacena de la siguiente manera:

- No van a la papelera archivos borrados del sistema o programa files (se borran directo).
- La información se encuentra en $Recycle.bin en el directorio raíz de cada volumen (partición). Dentro de ese directorio hay un directorio por usuario del sistema identificado por su SID. Es decir que es posible determinar qué usuario del sistema borró determinado archivo ya que está en su recycle bin personal.
- Dentro del recycle bin de cada usuario hay dos conjuntos de archivos:
    - Archivos de la forma $Ixxxxx.crt son los archivos que contienen la metadata del archivo borrado: fecha y hora de borrado, nombre original, etc.
    - Archivos de la forma $Rxxxx (con xxxx coincidente con los del archivo de metadata): son los archivos borrador renombrados.

Por otra parte, la fecha de creación del archivo de metadata coincide con la del atributo Date Recycled. En el caso del ejemplo, el

archivo borrado es en realidad un directorio, por lo cual si vamos a analizar el $Rxxxx correspondiente no tendrá información analizable.

Es posible deducir cuales fueron los archivos borrados que estaban en ese directorio utilizando el hecho de que se mueven a la papelera de reciclaje en forma recursiva, por lo cual ordenando por fecha de creación de archivos $Ixxxx los archivos que se hayan creado inmediatamente después que el perteneciente al directorio padre probablemente serán archivos que estaban en ese directorio. Lo recomendable es ir avanzando hasta que la información que da el FTK de uno de los archivos $Ixxx diga que corresponde a otro directorio.

Si el recycle bin ya se borró (manualmente o a través del sistema) todos estos archivos estarán borrados (pero generalmente recuperables) en el filesystem.

Todo esto se puede corroborar con un análisis del $USNJRNL y/o de la $MFT como se verá a continuación.

Metadata del sistema NTFS

NTFS es lo que se denomina un “journaled-filesystem”, es decir que, ante la falla de una operación sobre él, existe un registro que permite volver al último estado válido y por lo tanto prevenir la pérdida de información. Para eso utiliza varios logs en donde registra cada una de las operaciones que se realizan. Esto logs pueden ser utilizados para análisis forense.

La cantidad de registros disponibles depende de la utilización del sistema y puede ir desde un par de días a meses. En los análisis realizados en máquinas corporativas generalmente se pueden obtener algo más de una semana de registros.

El “USN Journal” o “Update Sequence Number Journal” se implementó por primera vez en Windows 2000, pero no se habilitó por

defecto hasta Windows Vista[4]. La funcionalidad primaria del log es permitir a los programas determinar qué archivos cambiaron para poder replicarlos, backupearlos o comprimirlos, por ejemplo. [23]

El log de cambios se almacena en el archivo \$Extend\$UsrJrnl. Este archivo contiene dos streams de datos (atributo $DATA): $Max contiene información básica acerca del log (incluyendo el tamaño máximo del log) y $J es el archivo que guarda el registro de las operaciones sobre el sistema de archivos. $J contiene un conjunto de registros de longitud variable, cada uno de los cuales contiene la fecha y el cambio realizado sobre el sistema de archivos. Cada uno de estos registros tiene asociado un *Update Sequence Number* (USN), que es un entero de 64 bits. El USN se utiliza para indexar los registros en el log y se guarda en el atributo $STANDARD_INFORMATION del archivo que fue modificado. [24]

Los datos completos de cada entrada del log son:

- Timestamp
- Operación que afectó al archivo
- Atributos del archivo o directorio
- Nombre del archivo o directorio
- Número de referencia del archivo o directorio
- Número de referencia del archivo o directorio padre del modificado
- ID de Seguridad
- “update sequence number” (USN) del registro

Análisis de USNJRNL

Existen varias utilidades que permiten analizar el USNJournal. La diferencia básica entre ellas es si utilizan sólo la información del

[4] Nota: En las imágenes estándar de la OT está habilitado también en Windows XP, probablemente por la introducción del Service Pack 3

$USNJRNL o también analizan el $MFT. El análisis del $MFT es útil para determinar la ubicación exacta del archivo borrado en el filesystem, ya que en el $USNJRNL sólo se almacenan el nombre del archivo (sin el path) y el puntero hacia el registro correspondiente en la $MFT.

Aunque pueden encontrarse varias utilidades más, solo analizo aquellas que pude probar y cuyos resultados coinciden entre sí.

Al momento de analizar un log, es conveniente recordar lo siguiente:

- El número de referencia del archivo en NTFS es único mientras exista el archivo, pero se reutiliza frecuentemente cuando el archivo se borra. No cambia cuando el archivo es renombrado.
- Cuando el número de referencia (MFTReference) se utiliza para un nuevo archivo, se incrementa el MFTReferenceSeqNo. Esto se puede utilizar para determinar cuándo se trata del mismo archivo.
- Cuando se mueve un directorio a la papelera, cambia el nombre del directorio a un archivo de la forma $Rxxxxx, pero los archivos/subdirectorios no cambian de nombre. Sólo se modifica metadata del archivo mediante la operación USN_SECURITY_CHANGE
- Todos los timestamps en NTFS son UTC, es decir GMT0.

Este es un ejemplo de un MFTReference que se reutiliza varias

1	Offset	FileName	USN	Timestamp	Reason	MFTReference	MFTReferenceSeqNo	MFTParentReference	MFTPa	FileAttributes
219269	0x00000000B1389960	Sen6286.tmp.tag	2973276512	2015-06-30 16:53:12.743:3590	FILE_CREATE	113922	511	105306	1	archive
219270	0x00000000B13899C0	Sen6286.tmp.tag	2973276608	2015-06-30 16:53:12.743:3590	CLOSE+FILE_CREATE	113922	511	105306	1	archive
219271	0x00000000B1389A20	Sen6286.tmp.tag	2973276704	2015-06-30 16:53:12.744:3590	DATA_EXTEND	113922	511	105306	1	archive
219272	0x00000000B1389A80	Sen6286.tmp.tag	2973276800	2015-06-30 16:53:12.744:3590	CLOSE+DATA_EXTEND	113922	511	105306	1	archive
219274	0x00000000B1389B38	Sen6286.tmp.tag	2973276984	2015-06-30 16:53:12.744:3590	CLOSE+FILE_DELETE	113922	511	105306	1	archive
219298	0x00000000B138A3B8	UTC2.tmp.tag	2973279160	2015-06-30 16:53:16.185:2007	FILE_CREATE	113922	512	105032	13	archive
219299	0x00000000B138A410	UTC2.tmp.tag	2973279248	2015-06-30 16:53:16.185:2007	CLOSE+FILE_CREATE	113922	512	105032	13	archive
219300	0x00000000B138A468	UTC2.tmp.tag	2973279336	2015-06-30 16:53:16.185:2007	DATA_EXTEND	113922	512	105032	13	archive
219301	0x00000000B138A4C0	UTC2.tmp.tag	2973279424	2015-06-30 16:53:16.185:2007	CLOSE+DATA_EXTEND	113922	512	105032	13	archive
219303	0x00000000B138A568	UTC2.tmp.tag	2973279592	2015-06-30 16:53:16.186:1995	CLOSE+FILE_DELETE	113922	512	105032	13	archive
219313	0x00000000B138A8D8	ClientAllSetting.ini.tag	2973280472	2015-06-30 16:53:17.295:8663	FILE_CREATE	113922	513	105032	13	archive
219314	0x00000000B138A948	ClientAllSetting.ini.tag	2973280584	2015-06-30 16:53:17.295:8663	CLOSE+FILE_CREATE	113922	513	105032	13	archive
219315	0x00000000B138A9B8	ClientAllSetting.ini.tag	2973280696	2015-06-30 16:53:17.295:8663	DATA_EXTEND	113922	513	105032	13	archive
219316	0x00000000B138AA28	ClientAllSetting.ini.tag	2973280808	2015-06-30 16:53:17.295:8663	CLOSE+DATA_EXTEND	113922	513	105032	13	archive
219318	0x00000000B138AB00	ClientAllSetting.ini.tag	2973281024	2015-06-30 16:53:17.295:8663	CLOSE+FILE_DELETE	113922	513	105032	13	archive
219608	0x00000000B13911B0	NewDomain.dat.tag	2973307312	2015-06-30 16:53:21.191:1863	FILE_CREATE	113922	514	105032	13	archive
219609	0x00000000B1391210	NewDomain.dat.tag	2973307408	2015-06-30 16:53:21.191:1863	CLOSE+FILE_CREATE	113922	514	105032	13	archive
219610	0x00000000B1391270	NewDomain.dat.tag	2973307504	2015-06-30 16:53:21.191:1863	DATA_EXTEND	113922	514	105032	13	archive
219611	0x00000000B13912D0	NewDomain.dat.tag	2973307600	2015-06-30 16:53:21.192:1851	CLOSE+DATA_EXTEND	113922	514	105032	13	archive
219613	0x00000000B1391388	NewDomain.dat.tag	2973307784	2015-06-30 16:53:21.192:1851	CLOSE+FILE_DELETE	113922	514	105032	13	archive
219659	0x00000000B1392558	cfgall.ini.tag	2973312344	2015-06-30 16:53:26.618:6655	FILE_CREATE	113922	515	105032	13	archive
219660	0x00000000B13925B0	cfgall.ini.tag	2973312432	2015-06-30 16:53:26.618:6655	CLOSE+FILE_CREATE	113922	515	105032	13	archive
219661	0x00000000B1392608	cfgall.ini.tag	2973312520	2015-06-30 16:53:26.618:6655	DATA_EXTEND	113922	515	105032	13	archive
219662	0x00000000B1392660	cfgall.ini.tag	2973312608	2015-06-30 16:53:26.618:6655	CLOSE+DATA_EXTEND	113922	515	105032	13	archive
219664	0x00000000B1392708	cfgall.ini.tag	2973312776	2015-06-30 16:53:26.619:6643	CLOSE+FILE_DELETE	113922	515	105032	13	archive
219691	0x00000000B13930A0	alerts.ini.tag	2973315232	2015-06-30 16:53:26.974:2383	FILE_CREATE	113922	516	105032	13	archive
219692	0x00000000B13930F8	alerts.ini.tag	2973315320	2015-06-30 16:53:26.974:2383	CLOSE+FILE_CREATE	113922	516	105032	13	archive
219695	0x00000000B13931F0	alerts.ini.tag	2973315568	2015-06-30 16:53:26.974:2383	DATA_EXTEND	113922	516	105032	13	archive
219696	0x00000000B1393248	alerts.ini.tag	2973315656	2015-06-30 16:53:26.974:2383	CLOSE+DATA_EXTEND	113922	516	105032	13	archive

veces:

En este ejemplo podemos ver el seguimiento de un directorio denominado “Argentina SAP”, que se mueve a la papelera de reciclaje. El movimiento a la papelera hace que cambie de nombre a $RC770DR y se termina borrando cuando se vacía la papelera 3 minutos después. Puede verse que tanto el número de referencia MFT como el número de secuencia permanecen constantes hasta que se elimina.

1	Offset	FileName	USN	Timestamp	Reason	MFTReference	MFTReferenceSeqNo	MFTParentReference	MFTPa	FileAttributes
361056	0x00000000B206F320	Argentina SAP	2986799904	2015-07-02 14:22:14.032:8302	RENAME_OLD_NAME	143874	76	106555	5	directory
361057	0x00000000B206F378	$RC770DR	2986799992	2015-07-02 14:22:14.032:8302	RENAME_NEW_NAME	143874	76	107430	6	directory
361058	0x00000000B206F3C8	$RC770DR	2986800072	2015-07-02 14:22:14.048:4279	CLOSE+RENAME_NEW_NAME	143874	76	107430	6	directory
362585	0x00000000B2096760	$RC770DR	2986960736	2015-07-02 14:22:14.422:7727	SECURITY_CHANGE	143874	76	107430	6	directory
362586	0x00000000B20967B0	$RC770DR	2986960816	2015-07-02 14:22:14.422:7727	CLOSE+SECURITY_CHANGE	143874	76	107430	6	directory
369083	0x00000000B212F2C0	$RC770DR	2987586240	2015-07-02 14:25:39.418:4055	CLOSE+FILE_DELETE	143874	76	107430	6	directory

Análisis con X-Ways Forensics

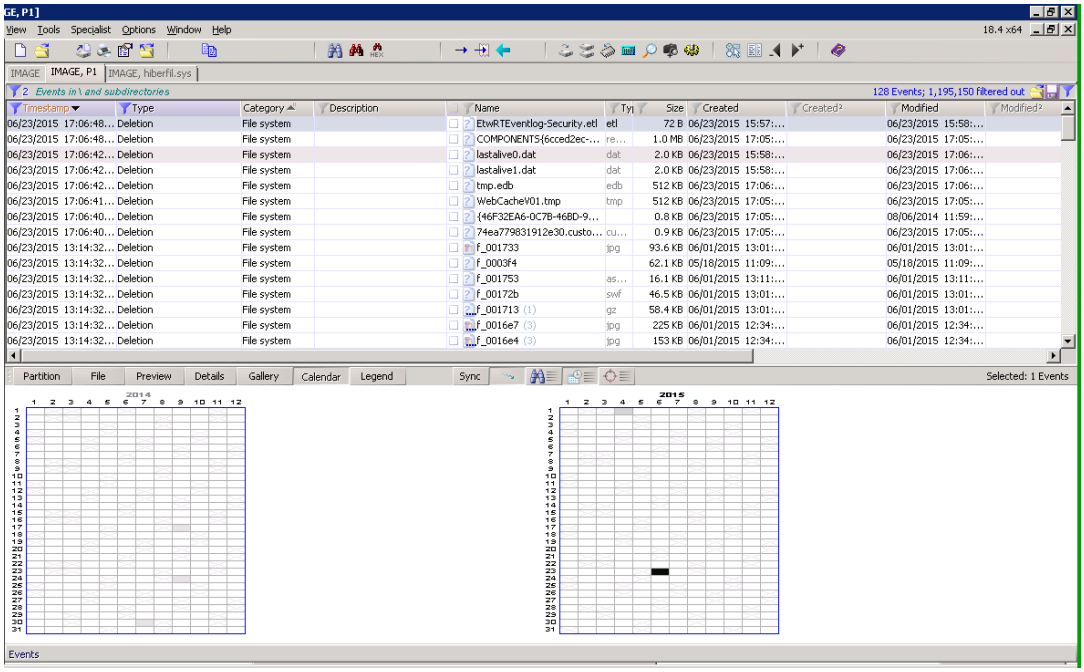
La información de $USNJRNL es interpretada y visualizada en la funcionalidad “Timeline” de X-Ways Forensics, en conjunto con la de otros artefactos forenses del sistema, para visualizar un registro de tiempo de todas las operaciones realizadas.

X-Ways permite la visualización de los diferentes atributos relacionados con “timestamps” de archivos ya sea en la vista de “Directory Browser” o en forma de “timelines” basados en los

diferentes timestamps extraidos del análisis de la imagen a nivel de filesystem.

Cuando se extrae metadata, X-Ways forensics puede compilar una lista de eventos de los timestamps que se encuentran a nivel de sistema de archivos, así como internamente en los archivos y en la memoria RAM. Las fuentes pueden ser historia de los browsers, log de eventos de Windows, registro de Windows, e-mails, etc. Si la lista se ordena cronológicamente trabaja exactamente como un timeline.

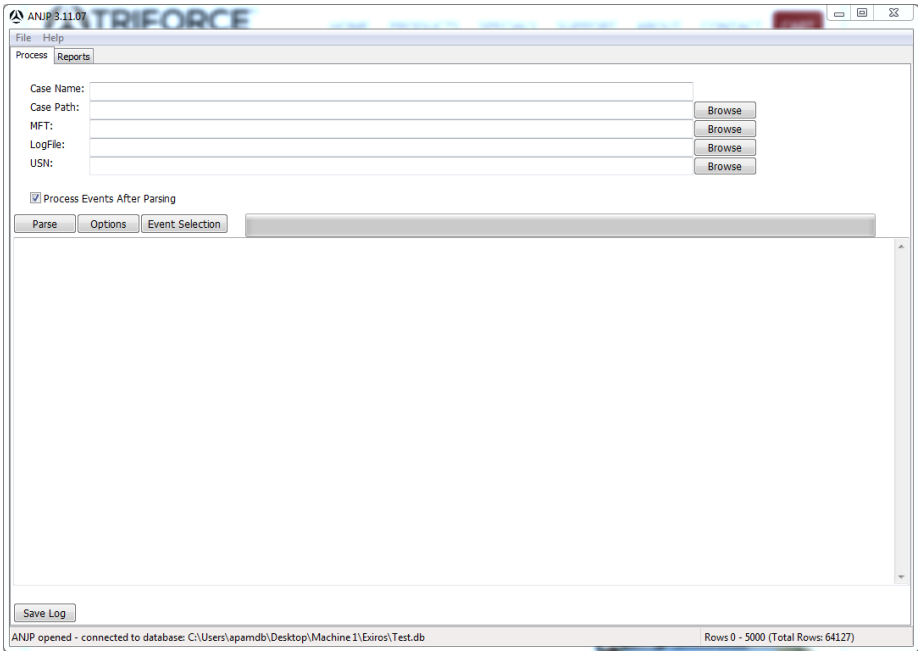
Se puede seleccionar la rama del filesystem de interés utilizando la opción “explorar recursivamente” y se pueden filtrar eventos por cualquier campo (haciendo click en el ícono de “embudo” al lado del en cabezado de cada columna. En el ejemplo se filtró por el tipo de evento “Delete” y por fecha. En el cuadro inferior se muestra un calendario en donde el sombreado de cada día es más o menos oscuro dependiendo de la cantidad de eventos de esa fecha.



Triforce ANJP Free Edition

Este producto, disponible en https://www.gettriforce.com/product/anjp-free/ gratis, va un paso más

allá al permitir correlacionar los eventos del filesystem NTFS guardados en los metaarchivos $MFT, $Logfile y $USNJRNL. Escribe

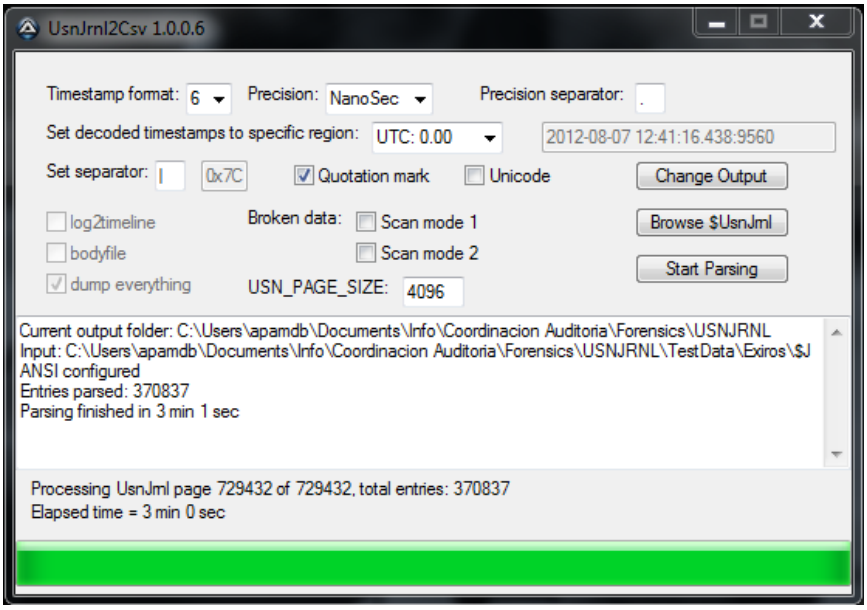


los resultados en una base de datos sqllite y permite consultar la información mediante varios reportes y filtros. Como desventaja tiene que requiere aproximadamente dos horas para correr el análisis.

UsnJrnl2Csv

https://github.com/jschicht/UsnJrnl2Csv

Es una aplicación para Windows en formato ejecutable que, mediante una interfaz gráfica, permite el análisis del archivo $J del $USNJRNL. Teniendo en cuenta que sólo se analiza el metaarchivo $J, no hay información de la ubicación del archivo en el filesystem.



NTFSLog Tracker

https://sites.google.com/site/forensicnote/ntfs-log-tracker

Este analizador, si bien no tiene una interfaz gráfica muy pulida es el mejor en cuanto a prestaciones y velocidad:

- Puede analizar $MFT, $Logfile y $USNJrnl
- Se distribuye en formato ejecutable
- Salida de los datos a una base sqllite con opción de exportar a csv

Apéndice III

Módulo para procesamiento Forense con KAPE

```
Description: Modulo de procesamiento de Offboarding
Category: Misc
Author: Marcelo Bovo (Based on !EZParser)
Version: 1.0
Id: 1d7bd143-aa89-41aa-b616-01bcbf0217fd
ExportFormat: csv
Processors:
-
Executable: AmcacheParser.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: AppCompatCacheParser.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: EvtxECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: JLECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: LECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: MFTECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: PECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: RBCmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: RecentFileCacheParser.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: RECmd_Kroll.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: SBECmd.mkape
CommandLine: ""
ExportFormat: ""
-
```

```
Executable: SQLECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: SrumECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: SumECmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: WxTCmd.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: BrowsingHistoryView.mkape
CommandLine: ""
ExportFormat: ""
-
Executable: USBDetective.mkape
CommandLine: ""
ExportFormat: ""
```

Target para Extracción Forense con KAPE

```
Description: OffBoarding Triage Collection.
Author: Marcelo Bovo (Based on Sans Triage)
Version: 1.1
Id: 7f72507f-e32c-4f9a-989e-7a258cdb6a4b
RecreateDirectories: true
Targets:


    # Event Logs - .\Targets\Windows\EventLogs.tkape
    -
        Name: Event logs XP
        Category: EventLogs
        Path: C:\Windows\System32\config\
        FileMask: '*.evt'
    -
        Name: Event logs Win7+
        Category: EventLogs
        Path: C:\Windows\System32\winevt\logs\
        FileMask: '*.evtx'
    -
        Name: Event logs Win7+
        Category: EventLogs
        Path: C:\Windows.old\Windows\System32\winevt\logs\
        FileMask: '*.evtx'

        # Evidence of Execution -
.\Targets\Compound\EvidenceOfExecution.tkape and
.\Targets\Logs\PowerShellConsole.tkape
    -
        Name: Prefetch
        Category: Prefetch
```

```
    Path: C:\Windows\prefetch\
    FileMask: '*.pf'
-
    Name: Prefetch
    Category: Prefetch
    Path: C:\Windows.old\Windows\prefetch\
    FileMask: '*.pf'
-
    Name: RecentFileCache
    Category: ApplicationCompatability
    Path: C:\Windows\AppCompat\Programs\
    FileMask: RecentFileCache.bcf
-
    Name: RecentFileCache
    Category: ApplicationCompatability
    Path: C:\Windows.old\Windows\AppCompat\Programs\
    FileMask: RecentFileCache.bcf
-
    Name: Amcache
    Category: ApplicationCompatibility
    Path: C:\Windows\AppCompat\Programs\
    FileMask: Amcache.hve
-
    Name: Amcache
    Category: ApplicationCompatibility
    Path: C:\Windows.old\Windows\AppCompat\Programs\
    FileMask: Amcache.hve
-
    Name: Amcache transaction files
    Category: ApplicationCompatibility
    Path: C:\Windows\AppCompat\Programs\
    FileMask: Amcache.hve.LOG*
-
    Name: Amcache transaction files
    Category: ApplicationCompatibility
    Path: C:\Windows.old\Windows\AppCompat\Programs\
    FileMask: Amcache.hve.LOG*
-
    Name: Syscache
    Category: Program Execution
    Path: C:\System Volume Information\
    FileMask: Syscache.hve
-
    Name: Syscache transaction files
    Category: Program Execution
    Path: C:\System Volume Information\
    FileMask: Syscache.hve.LOG*
-
    Name: PowerShell Console Log
    Category: PowerShellConsleLog
    Path:
C:\Users\%user%\AppData\Roaming\Microsoft\Windows\PowerShell\PSRead
line\
    FileMask: ConsoleHost_history.txt

    # File System - .\Targets\Compound\FileSystem.tkape
-
    Name: $MFT
    Category: FileSystem
```

```
        Path: C:\
        FileMask: $MFT
        AlwaysAddToQueue: true
    -
        Name: $LogFile
        Category: FileSystem
        Path: C:\
        FileMask: $LogFile
        AlwaysAddToQueue: true
    -
        Name: $J
        Category: FileSystem
        Path: C:\$Extend\
        FileMask: $UsnJrnl:$J
        AlwaysAddToQueue: true
        SaveAsFileName: $J
    -
        Name: $Max
        Category: FileSystem
        Path: C:\$Extend\
        FileMask: $UsnJrnl:$Max
        AlwaysAddToQueue: true
        SaveAsFileName: $Max
    -
        Name: $SDS
        Category: FileSystem
        Path: C:\
        FileMask: $Secure:$SDS
        AlwaysAddToQueue: true
        SaveAsFileName: $Secure_$SDS
    -
        Name: $Boot
        Category: FileSystem
        Path: C:\
        FileMask: $Boot
        AlwaysAddToQueue: true
    -
        Name: $T
        Category: FileSystem
        Path: C:\$Extend\$RmMetadata\$TxfLog\
        FileMask: $Tops:$T
        AlwaysAddToQueue: true
        SaveAsFileName: $T

        # LNK Files and JumpLists -
.\Targets\Windows\LNKFilesAndJumpLists.tkape
    -
        Name: LNK files from Recent
        Category: LNKFiles
        Path:
C:\Users\%user%\AppData\Roaming\Microsoft\Windows\Recent\
        Recursive: true
        Comment: Also includes automatic and custom jumplist
directories
    -
        Name: LNK files from Microsoft Office Recent
        Category: LNKFiles
        Path:
C:\Users\%user%\AppData\Roaming\Microsoft\Office\Recent\
```

```
        Recursive: true
    -
        Name: LNK files from Recent (XP)
        Category: LNKFiles
        Path: C:\Documents and Settings\%user%\Recent\
        Recursive: true
    -
        Name: Desktop LNK files XP
        Category: LNKFiles
        Path: C:\Documents and Settings\%user%\Desktop\
        FileMask: '*.LNK'
    -
        Name: Desktop LNK files
        Category: LNKFiles
        Path: C:\Users\%user%\Desktop\
        FileMask: '*.LNK'
    -
        Name: Restore point LNK files XP
        Category: LNKFiles
        Path: C:\System Volume Information\_restore*\RP*\
        FileMask: '*.LNK'

        # Recycle Bin and Recycler -
.\Targets\Windows\RecycleBin_DataFiles.tkape and
.\Targets\Windows\RecycleBin_InfoFiles.tkape
    -
        Name: Recycle Bin - Windows Vista+
        Category: FileDeletion
        Path: C:\$Recycle.Bin\
        FileMask: '$R*.*'
        Recursive: true
    -
        Name: RECYCLER - WinXP
        Category: FileDeletion
        Path: C:\RECYCLE*\
        FileMask: 'D*.*'
        Recursive: true
    -
        Name: Recycle Bin - Windows Vista+
        Category: FileDeletion
        Path: C:\$Recycle.Bin\
        FileMask: '$I*.*'
        Recursive: true
    -
        Name: RECYCLER - WinXP
        Category: FileDeletion
        Path: C:\RECYCLE*\
        FileMask: 'INFO2'
        Recursive: true

        # System Registry Files -
.\Targets\Windows\RegistryHivesSystem.tkape
    -
        Name: SAM registry transaction files
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: SAM.LOG*
    -
        Name: SAM registry transaction files
```

```
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SAM.LOG*
-
    Name: SECURITY registry transaction files
    Category: Registry
    Path: C:\Windows\System32\config\
    FileMask: SECURITY.LOG*
-
    Name: SECURITY registry transaction files
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SECURITY.LOG*
-
    Name: SOFTWARE registry transaction files
    Category: Registry
    Path: C:\Windows\System32\config\
    FileMask: SOFTWARE.LOG*
-
    Name: SOFTWARE registry transaction files
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SOFTWARE.LOG*
-
    Name: SYSTEM registry transaction files
    Category: Registry
    Path: C:\Windows\System32\config\
    FileMask: SYSTEM.LOG*
-
    Name: SYSTEM registry transaction files
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SYSTEM.LOG*
-
    Name: SAM registry hive
    Category: Registry
    Path: C:\Windows\System32\config\
    FileMask: SAM
-
    Name: SAM registry hive
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SAM
-
    Name: SECURITY registry hive
    Category: Registry
    Path: C:\Windows\System32\config\
    FileMask: SECURITY
-
    Name: SECURITY registry hive
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SECURITY
-
    Name: SOFTWARE registry hive
    Category: Registry
    Path: C:\Windows\System32\config\
    FileMask: SOFTWARE
-
```

```
    Name: SOFTWARE registry hive
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SOFTWARE
-
    Name: SYSTEM registry hive
    Category: Registry
    Path: C:\Windows\System32\config\
    FileMask: SYSTEM
-
    Name: SYSTEM registry hive
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\
    FileMask: SYSTEM
-
    Name: RegBack registry transaction files
    Category: Registry
    Path: C:\Windows\System32\config\RegBack\
    FileMask: '*.LOG*'
-
    Name: RegBack registry transaction files
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\RegBack\
    FileMask: '*.LOG*'
-
    Name: SAM registry hive (RegBack)
    Category: Registry
    Path: C:\Windows\System32\config\RegBack\
    FileMask: SAM
-
    Name: SAM registry hive (RegBack)
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\RegBack\
    FileMask: SAM
-
    Name: SECURITY registry hive (RegBack)
    Category: Registry
    Path: C:\Windows\System32\config\RegBack\
    FileMask: SECURITY
-
    Name: SECURITY registry hive (RegBack)
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\RegBack\
    FileMask: SECURITY
-
    Name: SOFTWARE registry hive (RegBack)
    Category: Registry
    Path: C:\Windows\System32\config\RegBack\
    FileMask: SOFTWARE
-
    Name: SOFTWARE registry hive (RegBack)
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\RegBack\
    FileMask: SOFTWARE
-
    Name: SYSTEM registry hive (RegBack)
    Category: Registry
    Path: C:\Windows\System32\config\RegBack\
    FileMask: SYSTEM
```

```
-
    Name: SYSTEM registry hive (RegBack)
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\RegBack\
    FileMask: SYSTEM
-
    Name: SYSTEM registry hive (RegBack)
    Category: Registry
    Path: C:\Windows\System32\config\RegBack\
    FileMask: SYSTEM1
-
    Name: SYSTEM registry hive (RegBack)
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\RegBack\
    FileMask: SYSTEM1
-
    Name: System Profile registry hive
    Category: Registry
    Path: C:\Windows\System32\config\systemprofile\
    FileMask: NTUSER.DAT
-
    Name: System Profile registry hive
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\systemprofile\
    FileMask: NTUSER.DAT
-
    Name: System Profile registry transaction files
    Category: Registry
    Path: C:\Windows\System32\config\systemprofile\
    FileMask: NTUSER.DAT.LOG*
-
    Name: System Profile registry transaction files
    Category: Registry
    Path: C:\Windows.old\Windows\System32\config\systemprofile\
    FileMask: NTUSER.DAT.LOG*
-
    Name: Local Service registry hive
    Category: Registry
    Path: C:\Windows\ServiceProfiles\LocalService\
    FileMask: NTUSER.DAT
-
    Name: Local Service registry hive
    Category: Registry
    Path: C:\Windows.old\Windows\ServiceProfiles\LocalService\
    FileMask: NTUSER.DAT
-
    Name: Local Service registry transaction files
    Category: Registry
    Path: C:\Windows\ServiceProfiles\LocalService\
    FileMask: NTUSER.DAT.LOG*
-
    Name: Local Service registry transaction files
    Category: Registry
    Path: C:\Windows.old\Windows\ServiceProfiles\LocalService\
    FileMask: NTUSER.DAT.LOG*
-
    Name: Network Service registry hive
    Category: Registry
    Path: C:\Windows\ServiceProfiles\NetworkService\
```

```
        FileMask: NTUSER.DAT
    -
        Name: Network Service registry hive
        Category: Registry
        Path:
C:\Windows.old\Windows\ServiceProfiles\NetworkService\
        FileMask: NTUSER.DAT
    -
        Name: Network Service registry transaction files
        Category: Registry
        Path: C:\Windows\ServiceProfiles\NetworkService\
        FileMask: NTUSER.DAT.LOG*
    -
        Name: Network Service registry transaction files
        Category: Registry
        Path:
C:\Windows.old\Windows\ServiceProfiles\NetworkService\
        FileMask: NTUSER.DAT.LOG*
    -
        Name: System Restore Points Registry Hives (XP)
        Category: Registry
        Path: C:\System Volume Information\_restore*\RP*\snapshot\
        FileMask: _REGISTRY_*

        # User Registry Files -
.\Targets\Windows\RegistryHivesUser.tkape
    -
        Name: NTUSER.DAT registry hive XP
        Category: Registry
        Path: C:\Documents and Settings\%user%\
        FileMask: NTUSER.DAT
    -
        Name: NTUSER.DAT registry hive
        Category: Registry
        Path: C:\Users\%user%\
        FileMask: NTUSER.DAT
    -
        Name: NTUSER.DAT registry transaction files
        Category: Registry
        Path: C:\Users\%user%\
        FileMask: NTUSER.DAT.LOG*
    -
        Name: NTUSER.DAT DEFAULT registry hive
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: DEFAULT
    -
        Name: NTUSER.DAT DEFAULT registry hive
        Category: Registry
        Path: C:\Windows.old\Windows\System32\config\
        FileMask: DEFAULT
    -
        Name: NTUSER.DAT DEFAULT transaction files
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: DEFAULT.LOG*
    -
        Name: NTUSER.DAT DEFAULT transaction files
        Category: Registry
```

```
        Path: C:\Windows.old\Windows\System32\config\
        FileMask: DEFAULT.LOG*
    -
        Name: UsrClass.dat registry hive
        Category: Registry
        Path: C:\Users\%user%\AppData\Local\Microsoft\Windows\
        FileMask: UsrClass.dat
    -
        Name: UsrClass.dat registry transaction files
        Category: Registry
        Path: C:\Users\%user%\AppData\Local\Microsoft\Windows\
        FileMask: UsrClass.dat.LOG*

        # System Level Artifacts

        # Scheduled Tasks - .\Targets\Windows\ScheduledTasks.tkape
    -
        Name: at .job
        Category: Persistence
        Path: C:\Windows\Tasks\
        FileMask: '*.job'
    -
        Name: at .job
        Category: Persistence
        Path: C:\Windows.old\Windows\Tasks\
        FileMask: '*.job'
    -
        Name: at SchedLgU.txt
        Category: Persistence
        Path: C:\Windows\
        FileMask: SchedLgU.txt
    -
        Name: at SchedLgU.txt
        Category: Persistence
        Path: C:\Windows.old\Windows\
        FileMask: SchedLgU.txt
    -
        Name: XML
        Category: Persistence
        Path: C:\Windows\System32\Tasks\
        Recursive: true
    -
        Name: XML
        Category: Persistence
        Path: C:\Windows.old\Windows\System32\Tasks\
        Recursive: true

        # System Resource Usage Monitor -
.\Targets\Windows\SRUM.tkape
    -
        Name: SRUM
        Category: Execution
        Path: C:\Windows\System32\SRU\
        Recursive: true
    -
        Name: SRUM
        Category: Execution
        Path: C:\Windows.old\Windows\System32\SRU\
        Recursive: true
```

```
        # Thumbcache.db - .\Targets\Windows\ThumbCache.tkape
    -
        Name: Thumbcache DB
        Category: FileKnowledge
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\Explorer\
        FileMask: thumbcache_*.db

        # USB Devices Logs - .\Targets\Windows\USBDevicesLogs.tkape
    -
        Name: Setupapi.log XP
        Category: USBDevices
        Path: C:\Windows\
        FileMask: setupapi.log
    -
        Name: Setupapi.log Win7+
        Category: USBDevices
        Path: C:\Windows\inf\
        FileMask: setupapi.dev.log
    -
        Name: Setupapi.log Win7+
        Category: USBDevices
        Path: C:\Windows.old\Windows\inf\
        FileMask: setupapi.dev.log
    -
        Name: WindowsIndexSearch
        Category: FileKnowledge
        Path:
C:\programdata\microsoft\search\data\applications\windows\
        FileMask: Windows.edb
    -
        Name: WBEM
        Category: WBEM
        Path: C:\Windows\System32\wbem\Repository\
        Recursive: true
    -
        Name: WBEM
        Category: WBEM
        Path: C:\Windows.old\Windows\System32\wbem\Repository\
        Recursive: true

        # User Communication

        # Skype - .\Targets\Apps\Skype.tkape
    -
        Name: main.db (App <v12)
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Packages\Microsoft.SkypeApp_*\LocalSt
ate\*\
        FileMask: main.db
    -
        Name: skype.db (App +v12)
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Packages\Microsoft.SkypeApp_*\LocalSt
ate\*\
        FileMask: skype.db
```

```
    -
        Name: main.db XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Skype\*\
        FileMask: main.db
    -
        Name: main.db Win7+
        Category: Communications
        Path: C:\Users\%user%\AppData\Roaming\Skype\*\
        FileMask: main.db
    -
        Name: s4l-[username].db (App +v8)
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Packages\Microsoft.SkypeApp_*\LocalSt
ate\
        FileMask: s4l-*.db
    -
        Name: leveldb (Skype for Desktop +v8)
        Category: Communications
        Path: C:\Users\%user%\AppData\Roaming\Microsoft\Skype for
Desktop\IndexedDB\*.leveldb\
        FileMask: '*.log'

        # Web Browser Artifacts -
.\Targets\Compound\WebBrowsers.tkape
    -
        Name: Chrome bookmarks XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Bookmarks*
    -
        Name: Chrome Cookies XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Cookies*
    -
        Name: Chrome Current Session XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Current Session
    -
        Name: Chrome Current Tabs XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Current Tabs
    -
        Name: Chrome Favicons XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Favicons*
    -
        Name: Chrome History XP
```

```
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: History*
    -
        Name: Chrome Last Session XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Last Session
    -
        Name: Chrome Last Tabs XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Last Tabs
    -
        Name: Chrome Preferences XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Preferences
    -
        Name: Chrome Shortcuts XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Shortcuts*
    -
        Name: Chrome Top Sites XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Top Sites*
    -
        Name: Chrome Visited Links XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Visited Links
    -
        Name: Chrome Web Data XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\
        FileMask: Web Data*
    -
        Name: Chrome bookmarks
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Bookmarks*
    -
        Name: Chrome Cookies
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Cookies*
    -
```

```
        Name: Chrome Current Session
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Current Session
    -
        Name: Chrome Current Tabs
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Current Tabs
    -
        Name: Chrome Favicons
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Favicons*
    -
        Name: Chrome History
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: History*
    -
        Name: Chrome Last Session
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Last Session
    -
        Name: Chrome Last Tabs
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Last Tabs
    -
        Name: Chrome Preferences
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Preferences
    -
        Name: Chrome Shortcuts
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Shortcuts*
    -
        Name: Chrome Top Sites
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Top Sites*
    -
        Name: Chrome Visited Links
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Visited Links
```

```
    -
        Name: Chrome Web Data
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\
        FileMask: Web Data*
    -
        Name: Chrome Extension Files
        Category: Communication
        Path: C:\Users\%user%\AppData\Local\Google\Chrome\User
Data\*\Extensions\
        Recursive: true
    -
        Name: Chrome Extension Files XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Application Data\Google\Chrome\User Data\*\Extensions\
        Recursive: true
    -
        Name: Edge folder
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wek
yb3d8bbwe\
        Recursive: true
    -
        Name: WebcacheV01.dat
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\WebCache\
    -
        Name: Firefox Places
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: places.sqlite*
    -
        Name: Firefox Downloads
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: downloads.sqlite*
    -
        Name: Firefox Form history
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: formhistory.sqlite*
    -
        Name: Firefox Cookies
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: cookies.sqlite*
    -
        Name: Firefox Signons
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
```

```
        FileMask: signons.sqlite*
    -
        Name: Firefox Webappstore
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: webappstore.sqlite*
    -
        Name: Firefox Favicons
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: favicons.sqlite*
    -
        Name: Firefox Addons
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: addons.sqlite*
    -
        Name: Firefox Search
        Category: Communications
        Path:
C:\Users\%user%\AppData\Roaming\Mozilla\Firefox\Profiles\*\
        FileMask: search.sqlite*
    -
        Name: Firefox Places (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: places.sqlite*
    -
        Name: Firefox Downloads (XP)
        Category: Communications (XP)
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: downloads.sqlite*
    -
        Name: Firefox Form history (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: formhistory.sqlite*
    -
        Name: Firefox Cookies (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: cookies.sqlite*
    -
        Name: Forefox Signons (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: signons.sqlite*
    -
        Name: Firefox Webappstore (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
```

```
Data\Mozilla\Firefox\Profiles\*\
        FileMask: webappstore.sqlite*
    -
        Name: Firefox Favicons (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: favicons.sqlite*
    -
        Name: Firefox Addons (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: addons.sqlite*
    -
        Name: Firefox Search  (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Mozilla\Firefox\Profiles\*\
        FileMask: search.sqlite*
    -
        Name: Index.dat History
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\History\History.IE5\
        FileMask: index.dat
    -
        Name: Index.dat History subdirectory
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\History\History.IE5\*\
        FileMask: index.dat
    -
        Name: Index.dat temp internet files
        Category: Communications
        Path: C:\Documents and Settings\%user%\Local
Settings\Temporary Internet Files\Content.IE5\
        FileMask: index.dat
    -
        Name: Index.dat cookies (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Cookies\
        FileMask: index.dat
    -
        Name: Index.dat UserData (XP)
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Microsoft\Internet Explorer\UserData\
        FileMask: index.dat
    -
        Name: Index.dat Office XP
        Category: Communications
        Path: C:\Documents and Settings\%user%\Application
Data\Microsoft\Office\Recent\
        FileMask: index.dat
    -
        Name: Index.dat Office
        Category: Communications
        Path:
```

```
C:\Users\%user%\AppData\Roaming\Microsoft\Office\Recent\
        FileMask: index.dat
    -
        Name: Local Internet Explorer folder
        Category: Communications
        Path: C:\Users\%user%\AppData\Local\Microsoft\Internet
Explorer\
        Recursive: true
    -
        Name: Roaming Internet Explorer folder
        Category: Communications
        Path: C:\Users\%user%\AppData\Roaming\Microsoft\Internet
Explorer\
        Recursive: true
    -
        Name: IE 9/10 History
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\History\
        Recursive: true
    -
        Name: IE 9/10 Cache
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\Temporary Internet
Files\
        Recursive: true
    -
        Name: IE 9/10 Cookies
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\Cookies\
        Recursive: true
    -
        Name: IE 9/10 Download History
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\IEDownloadHistory\
        Recursive: true
    -
        Name: IE 11 Metadata
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\WebCache\
    -
        Name: IE 11 Cache
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\INetCache\
        Recursive: true
    -
        Name: IE 11 Cookies
        Category: Communications
        Path:
C:\Users\%user%\AppData\Local\Microsoft\Windows\INetCookies\
        Recursive: true

        # Windows Timeline -
.\Targets\Windows\WindowsTimeline.tkape
```

```
    -
        Name: ActivitiesCache.db
        Category: FileFolderAccess
        Path:
C:\Users\%user%\AppData\Local\ConnectedDevicesPlatform\*\
        FileMask: ActivitiesCache.db
    -
        Name: ActivitiesCache.db-shm
        Category: FileFolderAccess
        Path:
C:\Users\%user%\AppData\Local\ConnectedDevicesPlatform\*\
        FileMask: ActivitiesCache.db-shm
    -
        Name: ActivitiesCache.db-wal
        Category: FileFolderAccess
        Path:
C:\Users\%user%\AppData\Local\ConnectedDevicesPlatform\*\
        FileMask: ActivitiesCache.db-wal

        # Compound targets

        Name: Discord Cache and LevelDB Files
        Category: Apps
        Path: Discord.tkape
```

Apéndice IV

automator.py

```
from forensic_extractor import SourceProcessor
from info_writer import InfoWriter
from file_helper import FileHelper
import sys
import argparse
import os

def process_options():

    argument_parser = argparse.ArgumentParser(description=(
        'extract triage forensic artifacts from a forensic image
(supports VSS and Bitlocker). '))

    argument_parser.add_argument(
        'source', nargs='?', action='store', metavar='image.raw',
        default=None, help=('path of the directory or filename of
 a storage '
                            'media image containing the file.'))

    argument_parser.add_argument(
        u'--no-auto-recurse', u'--
no_auto_recurse', dest=u'no_auto_recurse',
        action=u'store_true', default=False, help=(
            u'Indicate that the source scanner should not auto-
recurse.'))

    argument_parser.add_argument(
        u'--group-by-type', u'--
group_by_type', dest=u'group_by_type',
        action=u'store_true', default=False, help=(
            u'Group extracted files by category instead of preser
ving directory structure'))

    argument_parser.add_argument(
        u'--process-vss', u'--process-vss', dest=u'process_vss',
        action=u'store_true', default=False, help=(
            u'Process files in Volume Shadow Copies.'))

    argument_parser.add_argument(
        u'--output-file', u'--output-file', dest=u'output_file',
        action=u'store_true', default='files.zip', help=(
            u'Output zip file name.'))

    options = argument_parser.parse_args()

    if options.source:
        return options

    print(u'Source image value is missing.')
```

```
    print(u'')
    argument_parser.print_help()
    print(u'')
    sys.exit(0)


def main():
    """The main program function.

    Returns:
      A boolean containing True if successful or False if not.
    """

    # Process command line options
    options = process_options()

    # Create FileHelper object
    fh = FileHelper()
    FileHelper.temp_dir = fh.create_temp_dir()
    FileHelper.base_dir = os.getcwd()

    output_writer = InfoWriter()

    if not output_writer.open():
        print(u'Unable to open output writer.')
        print(u'')
        return False

    return_value = True

    # Initialize analyzer object
    source_analyzer = SourceProcessor(auto_recurse=not options.no
_auto_recurse, process_vss=options.process_vss,
                                      group_by_type=options.group
_by_type)
    try:
        source_analyzer.Analyze(options.source, output_writer)
        print(u'Completed.')
    except KeyboardInterrupt:
        return_value = False
        print(u'Aborted by user.')

    output_writer.close()

    # Create zip with extracted files and delete temporary files
    fh.zip_delete_dir(options.output_file)

    return return_value


if __name__ == '__main__':
    main()
```

file_helper.py

```
import tempfile
import os
import zipfile
import shutil
import sys

class FileHelper:
    """Helper class for all filesystem related operations."""
    base_dir = None
    temp_dir = None

    def __init__(self):
        pass

    def create_temp_dir(self)-> str:
        """ Creates a temporary directory for storing extracted
 files and image metadata
        """
        if self.temp_dir is None:
            print("Creating temp dir to store files")
            self.temp_dir = tempfile.mkdtemp()

        return self.temp_dir

    def create_subdir(self,dirname: str):
        """Creates subdirectory of base_dir

        Args:
            dirname (str): Name of the directory to create.

        """
        if not self.temp_dir:
            self.create_temp_dir()

        subdir_str = ''.join([self.temp_dir, "/", dirname])

        # Create metadata subdirectory
        try:
            os.makedirs(subdir_str)
            return subdir_str
        except OSError as err:
            print(f"Couldn\'t create {subdir_str}: {err}")
            sys.exit(1)


    def zipdir(self, path:str, filename: str):
        """Moves the files on the given path to a zip file loca
ted on os current directory.

        Args:
            path (str): Path where the files to be compressed a
re located.
            filename (str): Name of the zip file.
```

```
        """
        tfile = zipfile.ZipFile(filename, 'w')
        print("zipping file")

        for root, _, files in os.walk(path):
            for f in files:
                tfile.write(os.path.join(root, f), os.path.relp
ath(root, path) + "/" + f)
        tfile.close()


    def delete_tempdir(self):
        """Delete the temporary directory and all its contents.
        """
        shutil.rmtree(self.temp_dir)

    def zip_delete_dir(self, filename:str='extracted_files.zip'
):
        """Zips all the files in the given path and deletes the
 temporary directory.

        Args:
            path (str): Path where the files to be compressed a
re located.
            filename (str): Name of the zip file.
        """
        self.zipdir(self.temp_dir, filename)
        self.delete_tempdir()
```

hash_helper.py

```
import hashlib

class HashHelper:

    def __init__(self):

         # Prepare objects to calculate hashes
        self.md5hash = hashlib.md5()
        self.sha1hash = hashlib.sha1()


    def process_chunk(self, data):
        # Append data to hashes object
        self.md5hash.update(data)
        self.sha1hash.update(data)

    def get_hash(self)->tuple:

        # Calculate hashes
        return [ self.md5hash.hexdigest(), self.sha1hash.hexdig
est()]
```

info_writer.py

```
from file_helper import FileHelper
import unicodecsv as csv

class InfoWriter:

    def __init__(self):
        """Stdout output writer."""
        self._filelist_file = None
        self._partition_file = None
        self._wr = None
        self._wr_partition = None

        fh = FileHelper()
        self.metadata_dir = fh.create_subdir('metadata')


    def open(self)->bool:
        """Opens the output writer object.
           Creates metadata directory and output files. Write h
eader row on both
    Returns:
      bool: True if open was successful or False if not.
    """

        # Open csv file for filesystem listing, set encoding to
 UTF-8 to handle accents and binary output in order
        # to avoid extra blank lines.
        filelist = ''.join([self.metadata_dir, "/filelist.csv"]
)
        try:
            self._filelist_file = open(filelist, 'wb')
        except IOError:
            return False

        self._wr = csv.writer(self._filelist_file, quoting=csv.
QUOTE_ALL)
        self.write_row_archives_file(
            ['Partition', 'Inode', 'Filename', 'Full Path', 'Mo
dification Time', 'Access Time', 'Change Time',
             'Creation Time', 'Size', 'MD5 Hash', 'SHA1 Hash'])

        # Create file for partition list
        self._partition_file = open(self.metadata_dir + "/parti
tions.csv", 'wb')

        try:
            self._wr_partition = csv.writer(self._partition_fil
e, quoting=csv.QUOTE_ALL)
        except IOError:
            return False

        self.write_row_partitions_file(['Number', 'Type', 'Star
t Offset', 'Start Offset Bytes', 'Length'])
```

```
        return True

    def close(self):
        """Closes the output writer object."""
        self._partition_file.close()
        self._filelist_file.close()

    def write_row_archives_file(self, line: str):
        """Writes a line of text metadata file.

        Args:
            line (str): Row of information to be written
        """
        self._wr.writerow(line)

    def write_row_partitions_file(self, line: str):
        """Writes a line of text to partition file.

        Args:
            line (str): Row of information to be written
        """
        self._wr_partition.writerow(line)
```

forensisc_extractor.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
"""Script to extract triage forensic artifacts from a forensic
image (supports VSS and Bitlocker).
   Marcelo D. Bovo
   https://dfvfs.readthedocs.io/_/downloads/en/latest/pdf/
   https://github.com/forensicmatt/PancakeViewer
"""

# TODO: Alternate data streams handling.

from hash_helper import HashHelper

import datetime
import errno
import fnmatch
import getpass
import locale
import logging
import os
import sys

import pybde
import pytsk3
from dfvfs.credentials import manager as credentials_manager
from dfvfs.helpers import source_scanner, volume_scanner
from dfvfs.lib import definitions
from dfvfs.resolver import resolver
from file_helper import FileHelper
```

```
# from FileExtractor import FileExtractor

class RecursiveProcessor(volume_scanner.VolumeScanner):
    """Class that recursively calculates message digest hashes
of files."""

    # Class constant that defines the default read buffer size.
    _READ_BUFFER_SIZE = 32768

    # Dictionaries para almacenar la información a extraer
    extraction_data = {}
    extraction_data_category = {}

    def __init__(self, current_partition:str):
        super().__init__()
        import csv
        with open('./extraction.csv') as csv_file:
            csv_reader = csv.reader(csv_file, delimiter=';')
            for line_count, row in enumerate(csv_reader):
                if line_count > 0:
                    if row[0] in self.extraction_data.keys():
                        self.extraction_data[row[0]].append(row
[1])
                    else:
                        self.extraction_data[row[0]] = [row[1]]
                        self.extraction_data_category[row[0]] =
 [row[2]]
        self.current_partition = current_partition


    def CalculateHashes(self, base_path_specs, output_writer, g
roup_by_type):
        """Recursive calculates hashes starting with the base p
ath specification.

        Args:
          base_path_specs: a list of source path specification
(instances
                           of dfvfs.PathSpec).
          output_writer: the output writer (instance of InfoWri
ter).
          group_by_type: group by file type or by absolute path
        """
        for base_path_spec in base_path_specs:
            file_system = resolver.Resolver.OpenFileSystem(base
_path_spec)
            file_entry = resolver.Resolver.OpenFileEntry(base_p
ath_spec)
            if file_entry is None:
                logging.warning(
                    u'Unable to open base path specification:\n
{0:s}'.format(
                        base_path_spec.comparable))
                continue

            self.ProcessFilesystem(
```

```
                file_system, file_entry, u'', output_writer, gr
oup_by_type)

    def ProcessFilesystem(
            self, file_system, file_entry, parent_full_path, ou
tput_writer, group_by_type):
        """Recursively scans filesystems detecting files to ext
ract acording to config file specs.

        Args:
          file_system: the file system (instance of dfvfs.FileS
ystem).
          file_entry: the file entry (instance of dfvfs.FileEnt
ry).
          parent_full_path: the full path of the parent file en
try.
          output_writer: the output writer (instance of InfoWri
ter).
          group_by_type: group by file type or by absolute path
        """
        # Since every file system implementation can have their
 own path
        # segment separator we are using JoinPath to be platfor
m and file system
        # type independent.
        full_path = file_system.JoinPath([parent_full_path, fil
e_entry.name])

        # Extracts files if matches one of the specified in the
 config file.
        if file_entry.IsFile():
             for key in self.extraction_data.keys():
                if fnmatch.fnmatch(parent_full_path, key):
                    # Estamos en un directorio con archivos rel
evantes
                    for valores in self.extraction_data[key]:
                        if valores == 'Recursive':
                            valores = '*'
                        if fnmatch.fnmatch(file_entry.name, val
ores) == True:
                            # Es un archivo que quiero extraer
                            dump_dir = self.extraction_data_cat
egory[key][0]
                            self._process_file(file_entry, pare
nt_full_path, output_writer, dump_dir, group_by_type)
                            break

        for sub_file_entry in file_entry.sub_file_entries:
            self.ProcessFilesystem(
                file_system, sub_file_entry, full_path, output_
writer, group_by_type)

    @staticmethod
    def get_macb(filestats) -> tuple[str, str, str, str]:

        format_mask = '%Y-%m-%d %H:%M:%S'
```

```
        macb_tuples = []

        for attribute in ['crtime', 'atime', 'mtime', 'ctime']:
            if hasattr(filestats, attribute):
                macb_tuples.append(datetime.datetime.fromtimest
amp(getattr(filestats, attribute))
                    .strftime(format_mask))
            else:
                macb_tuples.append('N/A')

        return macb_tuples


    def _process_file(self, entryObject, parentPath, output_wri
ter, dump_dir=None, group_by_type=False):
        """Extract selected files from filesystem, get metadata
 and hash contents.

        Args:
            entryObject: dfvfs object to be extracted.
            parentPath (str): current filesystem parent path.
            output_writer ([type]): Metadata writer object.
            dump_dir (str, optional): Dir to extract the file t
o. Defaults to None.
            group_by_type (bool, optional): Group the files by
category or by original path. Defaults to False.
        """
        md5digest = "N/A"
        sha1digest = "N/A"

        _READ_BUFFER_SIZE = 32768

        # Get file object to process
        file_object = entryObject.GetFileObject()
        filestats = entryObject.GetStat()

        if entryObject.number_of_data_streams > 1:
            for data_stream in entryObject.data_streams:
                print(data_stream.name)

        # En este momento lee todo el sparse file
        if entryObject.name.lower() == "$usnjrnl":
            file_object = entryObject.GetFileObject('$J')
        # else:
        #     data_stream_name = None

        if not file_object:
            return None

        try:
            #TODO: use path function to do this
            if group_by_type:
                exported_file_path = FileHelper.temp_dir + '\\'
 + dump_dir + '\\' + self.current_partition + "_" + str(
                    filestats.ino) + "_" + entryObject.name
            else:
```

```
                # Exports file from image to disk
                exported_file_path = FileHelper.temp_dir  + '\\
' + self.current_partition + parentPath + '\\' + entryObject.na
me
            # Create Directory is not exists
            if not os.path.exists(os.path.dirname(exported_file
_path)):
                try:
                    os.makedirs(os.path.dirname(exported_file_p
ath))
                except OSError as exc:  # Guard against race co
ndition
                    if exc.errno != errno.EEXIST:
                        raise

            try:
                exported_file = open(exported_file_path, 'wb')
            except OSError:
                print("Error reading ", exported_file_path)
                return None

            # Extract file and calculate hashes
            hasher = HashHelper()

            data = file_object.read(self._READ_BUFFER_SIZE)
            while data:
                # Append data to hashes object
                hasher.process_chunk(data)

                # Write file chunk
                exported_file.write(data)

                data = file_object.read(self._READ_BUFFER_SIZE)

            exported_file.close()
            # Calculate hashes
            [md5digest, sha1digest] = hasher.get_hash()

        except IOError as exception:
            logging.warning((
                'Unable to read from path specification:\n{0:s}
'
                'with error: {1!s}').format(
                file_object.path_spec.comparable, exception))
            return None

        # Output file metadata to csv file
        out_tuple = self.get_macb(filestats)
        [birth_time, access_time, modification_time, change_tim
e] = out_tuple

        output_writer.write_row_archives_file(
            [self.current_partition, str(filestats.ino), entryO
bject.name,
             parentPath + entryObject.name,
             birth_time,
```

```
            access_time,
            modification_time,
            change_time,
            filestats.size, md5digest, sha1digest])

        if file_object is not None:
           file_object = None

class SourceProcessor(object):
    """Class that recursively analyze forensic image contents."
""

    # Class constant that defines the default read buffer size.
    _READ_BUFFER_SIZE = 32768

    def __init__(self, auto_recurse:bool=True, process_vss:bool
=False, group_by_type:bool=True):
        """Initializes the source analyzer object.
    Args:
      auto_recurse: optional boolean value to indicate if the s
can should
                    automatically recurse as far as possible. T
he default
                    is True.
      process_vss: optional boolean value to indicate if conten
ts from
                    virtual snapshots (VSS) should be extracted
. The default
                    is False.
      group_by_type: optional boolean value to indicate if extr
acted files should
                    be grouped by forensic artifact type or by
directory structure. The default
                    is True.
    """
        super(SourceProcessor, self).__init__()
        self._auto_recurse = auto_recurse
        self._process_vss = process_vss
        self._group_by_type = group_by_type
        self._encode_errors = u'strict'
        self._preferred_encoding = locale.getpreferredencoding(
)
        self._source_scanner = source_scanner.SourceScanner()
        self.current_partition = None

    def _encode_string(self, string):
        """Encodes a string in the preferred encoding.
    Returns:
      A byte string containing the encoded string.
    """
        try:
            # Note that encode() will first convert string into
 a Unicode string
            # if necessary.
            encoded_string = string.encode(
```

```
                self._preferred_encoding, errors=self._encode_e
rrors)
        except UnicodeEncodeError:
            if self._encode_errors == u'strict':
                logging.error(
                    u'Unable to properly write output due to en
coding error. '
                    u'Switching to error tolerant encoding whic
h can result in '
                    u'non Basic Latin (C0) characters to be rep
laced with "?" or '
                    u'"\\ufffd".')
                self._encode_errors = u'replace'
            encoded_string = string.encode(
                self._preferred_encoding, errors=self._encode_e
rrors)
        return encoded_string

    def _UnlockEncryptedVolume(self, scan_context, locked_scan_
node, source_path, output_writer):
        """Prompts the user to provide a credential for an encr
ypted volume.

    Args:
      scan_context: the source scanner context (instance of
                    SourceScannerContext).
      locked_scan_node: the locked scan node (instance of Sourc
eScanNode).
      output_writer: the output writer (instance of InfoWriter)
.
    """
        credentials = credentials_manager.CredentialsManager.Ge
tCredentials(
            locked_scan_node.path_spec)

        # For BDE: Print volume description and identifiers
        if locked_scan_node.type_indicator == definitions.TYPE_
INDICATOR_BDE:
            print(u'Found a BitLocker encrypted volume.')
            ids = self._get_BDE_protectors(locked_scan_node)
        else:
            print(u'Found an encrypted volume ({locked_scan_nod
e.type_indicator}).')


        credential_data, credential_identifier = self._get_unlo
ck_password(credentials, ids)
        if credential_identifier != 'skip':
            try:
                result = self._source_scanner.Unlock(
                    scan_context, locked_scan_node.path_spec, c
redential_identifier,
                    credential_data)

                if not result:
                    print(u'Unable to unlock volume.')
```

```
                    print(u'')

            except OSError as err:
                print(u'Unable to unlock volume:', err)
                print(u'')


    def _get_unlock_password(self, credentials, ids):

        credential_data = None
        credential_identifier = None

        # Print Credentials
        credentials_list = list(credentials.CREDENTIALS)
        credentials_list.sort()
        credentials_list.append(u'skip')

        #  check which credentials are available.
        print(u'Supported credentials:\n')

        for index, name in enumerate(credentials_list):
            if name in ids.keys():
                print(u'  {0:d}. {1:s} ({2:s})'.format(index, n
ame, ids[name]))
            else:
                print(u'  {0:d}. {1:s}'.format(index, name))
        print(u'')

        print('Select a credential to unlock the volume:', end=
'')
        print(u'', end='')
        input_line = sys.stdin.readline()

        while credential_identifier is None:
            if input_line in credentials_list:
                credential_identifier = input_line
            else:
                try:
                    credential_identifier = int(input_line, 10)
                    credential_identifier = credentials_list[cr
edential_identifier]
                except (IndexError, ValueError):
                    print(u'Unsupported credential: {0:s}'.form
at(input_line))

        if credential_identifier != u'skip':
            getpass_string = u'Enter credential data: '
            credential_data = getpass.getpass(getpass_string)
            print(u'')

        return credential_data, credential_identifier


    def _get_BDE_protectors(self, locked_scan_node):

        bde_volume = pybde.volume()
```

```
        file_object = resolver.Resolver.OpenFileObject(
            locked_scan_node.path_spec.parent, resolver_context
=self._source_scanner._resolver_context)
        bde_volume.open_file_object(file_object)

        print("Volume Description: " + bde_volume.description)

        # De la documentación
            # https://github.com/libyal/libbde/blob/master/docu
mentation/BitLocker%20Drive%20Encryption%20(BDE)%20format.ascii
doc#key_protection_types
            # 0x0000 (0) VMK protected with clear key (Basicall
y this is an unprotected VMK)
            # 0x0100 (256) VMK protected with TPM
            # 0x0200 (512) VMK protected with startup key
            # 0x0500 (1280) VMK protected with TPM and PIN
            # 0x0800 (2048) VMK protected with recovery passwor
d
            # 0x2000 (8192) VMK protected with password

        protectors_id = {8192: 'password', 2048:  'recovery_pas
sword', 512: 'startup_key'}

        return {
            protectors_id[protector.type]: protector.identifier
.split('-')[0]
            for protector in bde_volume.key_protectors
        }

    def Analyze(self, source_path, output_writer):
        """Analyzes the source image looking for partitions and
 filesystems.

    Args:
      source_path: the source path.
      output_writer: the output writer (instance of InfoWriter)
.

    Raises:
      RuntimeError: if the source path does not exists, or if t
he source path
                    is not a file or directory, or if the forma
t of or within
                    the source file is not supported.
    """
        if not os.path.exists(source_path):
            raise RuntimeError(u'No such source: {0:s}.'.format
(source_path))

        scan_context = source_scanner.SourceScannerContext()
        scan_path_spec = None

        scan_context.OpenSourcePath(source_path)

        while True:
            self._source_scanner.Scan(
```

```
                scan_context, auto_recurse=self._auto_recurse,
                scan_path_spec=scan_path_spec)

            # If no new items are found after las iteration, th
en all partitions are already identified.
            if not scan_context.updated:
                break

            # If not recursive scan is selected, then process o
nly top-level.
            if not self._auto_recurse:
                self.ProcessScanNode(scan_context, None,  outpu
t_writer)

            # The source is a directory or file.
            if scan_context.source_type in [
                definitions.SOURCE_TYPE_DIRECTORY, definitions.
SOURCE_TYPE_FILE]:
                break

            # The source scanner found a locked volume, e.g. an
 encrypted volume,
            # and we need a credential to unlock the volume.
            for locked_scan_node in scan_context.locked_scan_no
des:
                self._UnlockEncryptedVolume(
                    scan_context, locked_scan_node, source_path
, output_writer)

            if not self._auto_recurse:
                scan_node = scan_context.GetUnscannedScanNode()
                if not scan_node:
                    return
                scan_path_spec = scan_node.path_spec

        if self._auto_recurse:
            # If it's a recursive analysis, then get the root s
can node and process it.
            print('Source type\t\t: {0:s}'.format(scan_context.
source_type))
            print('')
            scan_node = scan_context.GetRootScanNode()
            self.ProcessScanNode(scan_context, scan_node, outpu
t_writer)
            print('')


    def ProcessScanNode(self, scan_context, scan_node, output_w
riter, indentation=''):
        """Writes the source scanner node to stdout.
    Args:
      scan_context (SourceScannerContext): the source scanner c
ontext.
      scan_node (SourceScanNode): the scan node.
      indentation (Optional[str]): indentation.
       outputwriter (InfoWriter):
```

```
    """
        if not scan_node:
            return

        values = []

        for attribute in ['part_index', 'store_index', 'start_o
ffset', 'location']:
            part_index = getattr(scan_node.path_spec, attribute
, None)
            if  part_index is not None:
                values.append(str(part_index))

        values = ', '.join(values)

        flags = ''
        if scan_node in scan_context.locked_scan_nodes:
            flags = ' [LOCKED]'

        # Add row to partitions table
        output_writer.write_row_partitions_file([indentation, s
can_node.path_spec.type_indicator, values, flags])

        # If the ScanNode is a filesystem, then process the fil
es in it.
        if scan_node.IsFileSystem():
            file_system = resolver.Resolver.OpenFileSystem(scan
_node.path_spec)
            if file_system:
                fstype = getattr(scan_node.path_spec.parent, 't
ype_indicator')
                # Skip VSS if VSS processing is disabled.
                if fstype == definitions.TYPE_INDICATOR_VSHADOW
 and not self._process_vss:
                    print("skipping VSS")
                else:
                    if fstype == definitions.TYPE_INDICATOR_BDE
:
                        self.current_partition = scan_node.path
_spec.parent.parent.location.replace("/", "")
                    else:
                        self.current_partition = scan_node.path
_spec.parent.location.replace("/", "")

                    print("Analyzing partition:", self.current_
partition)

                    file_entry = resolver.Resolver.OpenFileEntr
y(scan_node.path_spec)

                    recursive_hasher = RecursiveProcessor(self.
current_partition)
                    recursive_hasher.ProcessFilesystem(
                        file_system, file_entry, u'', output_wr
iter, self._group_by_type)
```

```
        # Recursive Subnode processing.
        for sub_scan_node in scan_node.sub_nodes:
            self.ProcessScanNode(scan_context, sub_scan_node, o
utput_writer, indentation=indentation+'\t')
```

Bibliografía

[1] Mmorais, «Vectors of Data Loss and Exfiltration», ~#, 26-sep-2015. [Online]. Disponible en: http://tcpflag.blogspot.com.ar/2015/09/vectors-of-data-loss-and-exfiltration.html. [Accedido: 07-nov-2017]

[2] B. Nikkel, *Practical Forensic Imaging: Securing Digital Evidence with Linux Tools*. No Starch Press, 2016.

[3] «Resolución PGN N ° 756 /16. “Guía de obtención, preservación y tratamiento de evidencia digital”». Procuración General de La Nación, 31-mar-2016.

[4] H. Carvey, *Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry*. Syngress, 2016.

[5] «Home · log2timeline/dfvfs Wiki». [Online]. Disponible en: https://github.com/log2timeline/dfvfs/wiki. [Accedido: 19-may-2017]

[6] *log2timeline/dfvfs*. log2timeline, 2020 [Online]. Disponible en: https://github.com/log2timeline/dfvfs. [Accedido: 26-may-2020]

[7] «F-Response Enterprise». [Online]. Disponible en: https://www.f-response.com/software/ee. [Accedido: 10-jul-2021]

[8] «Kroll Artifact Parser and Extractor - KAPE». [Online]. Disponible en: https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape. [Accedido: 10-jul-2021]

[9] E. Zimmerman, *EricZimmerman/RECmd*. 2021 [Online]. Disponible en: https://github.com/EricZimmerman/RECmd. [Accedido: 09-jul-2021]

[10] «Manual Identification of Suspect Computer Time Zone». [Online]. Disponible en: http://www.digital-detective.net/manual-identification-of-suspect-computer-time-zone-2/. [Accedido: 03-jun-2015]

[11] Lee, Rob, *FOR500: Windows Forensic Analysis*, 2018.ª ed., vol. 5, 6 vols. SANS, Institute, 2018.

[12] «BrowsingHistoryView - View browsing history of your Web browsers». [Online]. Disponible en: https://www.nirsoft.net/utils/browsing_history_view.html. [Accedido: 09-jul-2021]

[13] B. Carrier, *File System Forensic Analysis*. Addison-Wesley Professional, 2005.

[14] «Windows Forensic Analysis | SANS Poster». [Online]. Disponible en: https://www.sans.org/posters/windows-forensic-analysis/. [Accedido: 19-jul-2021]

[15] «Windows 10 Time Rules – Cyber Forensicator». [Online]. Disponible en: http://cyberforensicator.com/2018/03/25/windows-10-time-rules/. [Accedido: 03-abr-2018]

[16] Y. Zhu, P. Gladyshev, y J. James, «Using shellbag information to reconstruct user activities», *Digit. Investig.*, vol. 6, pp. S69-S77, sep. 2009, doi: 10.1016/j.diin.2009.06.009.

[17] V. Lo, «Windows ShellBag Forensics in Depth», p. 33.

[18] «Introducing WxTCmd!» [Online]. Disponible en: https://binaryforay.blogspot.com/2018/05/introducing-wxtcmd.html. [Accedido: 29-jun-2021]

[19] H. Carvey, «HowTo: USB Thumb Drives», *Windows Incident Response*, 04-feb-2012. [Online]. Disponible en: http://windowsir.blogspot.com.ar/2012/02/howto-usb-thumb-drives.html. [Accedido: 10-abr-2018]

[20] Y. Name, «USB storage forensics in Win10 #1 - Events», *Forensics|Exchange*. [Online]. Disponible en: /posts/19_08_03_usb_storage_forensics_1/. [Accedido: 30-jun-2021]

[21] «Windows 8.1 clearing plug-and-play state if device not used for 30 days». [Online]. Disponible en: https://social.msdn.microsoft.com/Forums/sqlserver/en-US/e1cb0ce0-bfe1-4366-85d3-1b385b6db7d7/windows-81-clearing-plugandplay-state-if-device-not-used-for-30-days?forum=wdk. [Accedido: 10-jul-2021]

[22] *FlaUI/FlaUI*. FlaUI, 2021 [Online]. Disponible en: https://github.com/FlaUI/FlaUI. [Accedido: 11-jul-2021]

[23] D. Cowen, «Hacking Exposed Computer Forensics Blog: Daily Blog #51: Understanding the artifacts USNJrnl». [Online]. Disponible en: http://www.hecfblog.com/2013/08/daily-blog-51-understanding-artifacts.html. [Accedido: 19-may-2015]

[24] M. Russinovich, D. Solomon, y A. Ionescu, *Windows Internals, Part 2*, 6 edition. Redmond, Wash: Microsoft Press, 2012.

[25] A. Fortuna, «Amcache and Shimcache in forensic analysis», *So Long, and Thanks for All the Fish*, 16-oct-2017. [Online]. Disponible en: https://www.andreafortuna.org/cybersecurity/amcache-and-shimcache-in-forensic-analysis/. [Accedido: 04-may-2018]

[26] «Secrets of the Application Compatilibity Database (SDB) – Part 1 – Alex Ionescu's Blog». [Online]. Disponible en: https://www.alex-ionescu.com/?p=39. [Accedido: 30-jun-2021]

[27] msuhanov, *msuhanov/regf-samples*. 2021 [Online]. Disponible en: https://github.com/msuhanov/regf-samples/blob/cfa0fae0700d68f4678d078d478464e24611b226/8.1-unreconciled/Flush%20strategies%20in%20the%20Windows%20registry.md. [Accedido: 11-jul-2021]