CIRA: Un framework de file Carving como solución a una necesidad detectada en la generación de un PURI - Proceso Unificado de Recuperación de Información.

Ana Haydée Di Iorio[1], Martín Castellote[2],
Ariel Podestá[3], Fernando Greco[4],
Bruno Constanzo[5], Julián Waimann[6]

[1] Ingeniero Informático, Docente e Investigadora en Universidad FASTA, diana@ufasta.edu.ar.

[2] Ingeniero Informático. Docente e Investigador de la Facultad de Ingeniería de la Universidad FASTA castellotemartin@yahoo.com.ar

[3] Ingeniero Informático. Docente e Investigador de la Facultad de Ingeniería de la Universidad FASTA, arielpodesta@gmail.com

[4] Ingeniero Informático, Docente e Investigador en Universidad FASTA, fmartingreco@gmail.com

[5] Técnico Informático. Auxilliar de Investigación Alumno, Facultad de Ingeniería de la Universidad FASTA. Bru.constanzo@gmail.com

[6]Analista Informático. Auxilliar de Investigación Alumno, Facultad de Ingeniería de la Universidad FASTA. julianw@ufasta.edu.ar

CAPITULO: Informática y Telecomunicaciones Forenses

1. Introducción

En los últimos diez años, la sociedad ha experimentado un proceso gradual de digitalización, lo que trajo aparejado una dependencia prácticamente total de los sistemas informáticos para manipular información. A su vez, tareas cada vez más críticas son realizadas por software, desde intervenciones médicas hasta complejas operaciones militares.

Los cambios en las tecnologías, plataformas, medios de almacenamiento, legislaciones y aplicaciones de software, hace cada vez más necesario el uso de procesos, métodos, estándares y buenas prácticas que permitan garantizar la recuperación de información contenida, y sobre todo, que permitan asegurar que se realizaron todas las tareas posibles con los mecanismos adecuados.

En el Grupo de Investigación en Sistemas Operativos e Informática Forense de la Facultad de Ingeniería de la Universidad FASTA se detectó la necesidad de los Informáticos Forenses de contar con un proceso de recuperación de información que sirva de guía en las tareas a realizar, que conste de una metodología, que haya sido probado, evaluado, y que sea reproducible en instancias de juicio. Como resultado de la elaboración del proceso PURI – Proceso Unificado de Recuperación de la Información - se detectaron diversos aspectos carentes de técnicas y herramientas, entre los que se encuentra el proceso de File Carving.

Se presenta en este trabajo el proceso PURI definido en sentido amplio y la arquitectura de Framework de File Carvers - CIRA - con el objeto de que ambas soluciones propuestas puedan ser conocidas, evaluadas, ampliadas y utilizadas, tanto por los profesionales forenses como por la comunidad científica.

2. Propuesta de un Proceso Unificado de Recuperación de la Información

Desde el año 2001 diferentes autores y organizaciones han estado trabajando en guías de buenas prácticas en informática forense que si bien constituyen un excelente aporte procedimental, entendemos no constituyen un proceso unificado. Del analisis realizado sobre las guías existentes, se encuentra que una gran cantidad solo abarca solo una parte del proceso, otras son muy generales y varias focalizan únicamente en los temas delictivos. Por otro lado desde el punto de vista práctico, estas guías no abordan las técnicas existentes para realizar ciertas tareas, las herramientas comerciales y gratuitas disponibles en el mercado, así como tampoco las diferentes alternativas de acuerdo a la plataforma de software del equipo a periciar, o del equipo base del forense.

El Proceso Unificado de Recuperación de la Información generado por este equipo de investigación se compone de un conjunto de fases, etapas, tareas, técnicas y herramientas.
En la tabla siguiente se detalla el proceso resumido, con las tareas que permanecerán en el tiempo independientemente de la tecnología que se utilice para realizarlas. A partir de la definición de este proceso se detectaron necesidades en la etapa de extracción física de la información, tema que se decide con el proyecto CIRA.

PROCESO PURI RESUMIDO

FASE ADQUISICIÓN:

Adquisición de medio de almacenamiento persistente

- Bloqueo del medio de almacenamiento (impedir escrituras en el mismo)
- Captura y resguardo de la imagen (copia exacta del contenido del medio de almacenamiento)
- Opcional: compresión y división de la imagen
- Validación de original y copia (verificación de que es copia fiel del original)

Adquisición de medio de almacenamiento volátil

- Captura y resguardo de la imagen (copia de la información volátil que el equipo encendido manipula)

Adquisición de tarjetas SIM

- Lectura de los registros almacenados
- Opcional: Clonación en otra SIM

Adquisición de Tarjetas de Memoria Extraíble Persistente del Dispositivo

- Bloqueo de la tarjeta (impedir escrituras en la misma)
- Captura y resguardo de la imagen
- Opcional: compresión y división de la imagen
- Validación de original y copia

Adquisición de Memoria ROM interna del Dispositivo

- Captura de la imagen (copia exacta de la memoria ROM)

Adquisición de Memoria Volátil (RAM) del Dispositivo móvil

- Captura y resguardo de la imagen (copia de la información volátil que el dispositivo encendido manipula)

FASE PREPARACIÓN:

Restauración de la Imagen

- Ensamblado de las divisiones de la imagen
- Descompresión de la imagen
- Opcional: Validación de original y copia

Preparación de la Extracción

- Preparación para Extracción Lógica (configuración de lectores de sistemas de archivos)
- Preparación Extracción física (configuración de acceso al contenido en crudo del archivo de imagen)

Identificación

- Identificación de cantidad y tipos de Sistemas Operativos Presentes
- Identificación de cantidad de discos, particiones y tipos de sistemas de archivo presentes
- Opcional: Identificar y quebrar medios de encriptación de información (ejemplo: Bitlocker)
- Opcional: Identificar y quebrar medios de ocultación de información (ejemplo: HPA - Host Protected Area)
- Identificación de Máquinas virtuales presentes

Preparación de ambiente de trabajo

- Preparación de ambiente de examinación: de acuerdo a características de lo adquirido

FASE ANÁLISIS:

Extracción Lógica

- Recuperación de archivos eliminados
- Extracción de información a examinar por tipo de archivo (determinación de formatos de archivo ignorando la extensión)
- Extracción de metadatos de archivo presentes en el sistema de archivos
- Extracción de metadatos propios del archivo (Lectura de los atributos del archivo contenidos en la propia información del archivo)
- Extracción de archivos protegidos con contraseña
- Extracción de archivos comprimidos
- Extracción de archivos encriptados
- Búsqueda de determinado tipo de archivo oculto (determinar en todo el contenido aquellos tipos de archivos que no se corresponden con su extensión)
- Búsqueda de información en el área de paginado de la memoria virtual
- Búsqueda de Información de Configuración presente en el equipo
- Búsqueda de Información en Procesos en Memoria

Extracción Física

- Búsqueda de palabras en disco
- Extracción de archivos en espacio desalojado no fragmentado. Carving Primera Generación.
- Extracción de archivos en espacios desalojados, que puedan estar fragmentados. Carving Segunda generación.

Análisis de Relaciones

- Identificación de relaciones entre conjunto de archivos
- Verificación de aplicaciones instaladas

FASE PRESENTACIÓN:

Armado del Informe detallando metodología utilizada y asegurando trazabilidad y reproducción

Preparación de la información a presentar / entregar

3. El FrameWork CIRA

4. CIRA es una solución de File Carving, compuesta de un Framework y de una herramienta

implementada con esta arquitectura.

El File Carving es el proceso de extracción de archivos u objetos del disco en ausencia de metadatos del sistema de archivo, es decir, accediendo directamente al contenido de los bloques [1]. El proceso de file carving se basa en recuperar información que ha sido eliminada o es inaccesible debido a daños del dispositivo o del sistema de archivos. Su uso es vital en la Informática Forense, ya sea para recuperar archivos eliminados que puedan ser utilizados como prueba, como para recuperar información comercial o personal valiosa [2].

Existen varias técnicas de File Carving, algunas implementadas en herramientas, y otras ún no. Estas técnicas varían desde las más básicas, basadas en la lectura del header y footer de un archivo, hasta otras mucho más complejas como Bifragment Gap (Garfinkel), Smart Carving (Pal, Memon et al) o Semantic Carving (Garfinkel). Incluso algunas tienen varios enfoques, como por ejemplo Header/Footer carving que puede aplicarse en una sola o en múltiples pasadas.

El proceso de File Carving ha ido evolucionando en los últimos años, sin embargo no cuenta áun con una definición flexible, adaptable e integradora, que permita describir y utilizar las técnicas que mejor se adapten a cada estructura de archivos.

Por otro lado, los File Carvers actuales (herramientas que implementan file carving) presentan varias limitaciones. Las herramientas más populares suelen presentar resultados incompletos, una tasa muy alta de falsos positivos y recuperar archivos dañados o no válidos. También ocurre que aquellas herramientas con muy buena performance recuperan grandes cantidades de archivos y muchos de ellos inválidos, lo que dificulta el acceso a los resultados de interés.

A partir de las necesidades detectadas los alumnos Bruno Constanzo y Julián Waimann deciden complementar el trabajo realizado en PURI y abordar el desarrollo de una solución de file carving “CIRA” como proyecto final de graduación de la carrera de Ingeniería Informática [3].

El framework desarrollado – CIRA - se estructura alrededor de un proceso de tres etapas definidas: preprocesamiento, carving y postprocesamiento. Además del proceso en etapas, el framework está estructurado de manera tal que el algoritmo de carving propiamente dicho es ajeno a los detalles de acceso a la imagen de disco y a la extracción de los archivos. Ésto permitiría, por ejemplo, que se extienda el framework para operar a través de una red con una imagen de disco residente en otra computadora que actúe de servidor de archivos, o, modificar el extractor de archivos para que en lugar de crear archivos físicos en la computadora genere los metadatos para la creación de un file

system virtual[1], entre otras cosas.

Como parte del desarrollo se implementaron dos soluciones de preprocesamiento, cuatro algoritmos de *file carving*, dos soluciones de postprocesamiento y un logger de extracción, junto con otros objetos asociados que fueron necesarios para mantener un equilibro entre el nivel de abstracción deseado en cada parte y el rendimiento del producto. Es destacable que, si bien se mantuvo un alto grado de abstracción que permite la fácil implementación de algoritmos de carving y componentes de pre y postprocesamiento, el rendimiento no tuvo un impacto significativo, y en condiciones similares es posible acercarse al rendimiento de "Scalpel", reconocido por su foco en la alta performance y bajo consumo de recursos [6].

Los preprocesadores implementados permiten excluir y extraer bloques del análisis. Uno de los preprocesadores permite que se excluyan bloques arbitrarios, definidos como una cadena de texto y generar un archivo con los rangos que se desean excluir. El otro preprocesador realiza un análisis estadístico de los bloques disponibles y decide, en base a la media aritmética y la entropía, si los bloques deben excluirse del análisis. Esta técnica facilita, por ejemplo, la selección de bloques que contienen datos binarios, excluyendo los datos ASCII usualmente asociados con archivos de texto. Este preprocesador se encuentra en una fase de experimentación y ajuste, que está planeado realizar como parte del trabajo futuro.

Con respecto a los algoritmos de *carving*, se implementaron tres variantes de *header/footer carving* y se realizó una implementación de *carving* basado en la estructura interna de archivos.

Los algoritmos de *header footer carving* implementados fueron denominados *Single Format Carve*, *Multiple Format Carve* y *Maximum Length Carve*. Todos son variantes de *header footer carving*, es decir que generan los archivos desde la ocurrencia de un encabezado de archivo hasta la ocurrencia de una cadena, el *footer*, que delimita el fin de un archivo. En el orden que fueron presentados, puede considerarse como la evolución de la variante más simple de la técnica de *header footer carving* hacia su versión más compleja.

Con respecto al algoritmo de *carving* basado en la estructura interna de los archivos, durante el trabajo con los validadores de archivo se descubrió que al comenzar el análisis de validez de archivo de un determinado en posiciones arbitrarias de la imagen de disco, era posible encontrar y extraer archivos JPG que resultaban problemáticos para el algoritmo de *Single Format Carve*, sobre el que se estaba trabajando en ese momento. Esta experiencia se tomó como base para la

1 Esta técnica fue estudiada por autores como Richard, Roussev, Marziale y otros [4][5]

implementación de un carver que combina una parte del funcionamiento de *Multiple Format Carve* y utiliza el Framework de Validación para llevar a cabo la extracción de archivos válidos luego de analizar su estructura. Pese a que su funcionamiento presenta ventajas con respecto a las técnicas de *header/footer*, aún pueden implementarse mejoras, tanto los validadores como el *carver* de estructura interna.

Finalmente, para la etapa de postprocesamiento se desarrollaron dos postprocesadores, uno que realiza la verificación de los archivos extraídos por medio del framework de validación, y otro que calcula los hashes MD5 y SHA-1 de los archivos extraídos, que suelen utilizarse para verificar su integridad respecto a otras versiones del mismo archivo disponible.

En paralelo y promoviendo la integración del proyecto CIRA, desde el equipo de investigación de PURI se trabajó en la creación de un framework de validación de archivos, tomando como base y continuando el trabajo de Garfinkel[7][8], pero iniciando un nuevo desarrollo en lenguaje Python.

Los nichos carentes detectados por PURI coinciden con las áreas que se sugiere como temas de investigación en Forensics Wiki [3].

4. Reflexiones y Conclusiones

El proceso PURI presentado se validó para las plataformas: Linux Ubuntu, Android, Windows y Mac OSX. A partir de esta validación se detectaron áreas carentes de técnicas y/o de herramientas que las implementen. Una de estas áreas es el File Carving, y CIRA es una propuesta de solución a estas necesidades.

Partiendo de un proyecto abarcador, teórico y general como el desarrollo de un PURI, se logró el desarrollo de una solución de *file carving* específica y práctica que está a la altura de herramientas ya consolidadas, tanto en performance, como en funcionalidad y calidad de los resultados.

Además, CIRA provee una arquitectura extensible y con posibilidades de desarrollo futuro.

Para la continuación del proyecto se prevee implementar otros algoritmos de *carving*, continuar la optimización de los módulos actuales y construir nuevos módulos de pre y posprocesamiento.

Referencias

[1] MEROLA A.: Data Carving Concepts, SANS Institute (2008)

[2] CONSTANZO, Bruno; WAIMANN, Julián El estado actual de las Técnicas de File Carving y la necesidad de Nuevas Tecnologías que implementen Carving Inteligente. Journal CADI (2012)

[3] DI IORIO, Ana et al La recuperación de la información y la informática forense: Una propuesta de proceso unificado, Journal CADI (2012)

[4] "In-Place File Carving", Golden G. Richard III, Vassil Roussev, and Lodovico Marziale, IFIP 2007

[5] http://ocfa.sourceforge.net/libcarvpath/ accedido el 12 de Julio de 2013

[6] GOLDEN G. RICHARD III, VASSIL ROUSSEV, "Scalpel: A Frugal, High Performance File Carver", DFRWS 2005

[7] GARFINKEL, SIMSON "Carving contiguous and fragmented files with fast object validation", DFRWS 2007.

[8] GARFINKEL, SIMSON, S2 carver source code, 2006 http://sandbox.dfrws.org/2006/garfinkel/

[9] http://www.forensicswiki.org/wiki/Research_Topics accedido el 12 de Julio de 2013