El framework cira, un aporte a las técnicas de file carving

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

Resumen

File Carving es el proceso de extraer archivos de un medio de almacenamiento en ausencia de metadatos del sistema de archivo, es decir, directamente a partir del análisis del contenido de los bloques de disco. Pese a la existencia de varias técnicas y algoritmos de file carving, no hay definido aún un proceso que sea aplicable a todos. En este trabajo se presenta una solución de file carving – CIRA –, su implementación, su Framework, su arquitectura y algunos resultados obtenidos.

Abstract

File carving is the process of extracting files from a disk in the absence of file system meta-data, through the analysis of the remaining files content in the disk. Despite having a wide array of algorithms and techniques, there is not a defined process that is applicable to all of them. In this paper, we present a file carving solution, - CIRA - the implementation, the architecture and some results.

Keywords: File carving, Digital forensics, File recovery, CIRA, PURI

[1] Ingeniera en Informática, Docente e Investigadora en la Facultad de Ingeniería de la Universidad FASTA. **E mail:**diana@ufasta.edu.ar

[3] Ingeniero en Informática, Docente e Investigador de la Facultad de Ingeniería de la Universidad FASTA. **E mail:** castellotemartin@yahoo.com.ar

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

[4] Ingeniero en Informática, Docente e Investigador de la Facultad de Ingeniería de Universidad FASTA. **E mail:**fmartingreco@gmail.com

[5] Técnico en Informática, Auxilliar de Investigación Alumno, Facultad de Ingeniería de la Universidad FASTA. **E mail:** Bru.constanzo@gmail.com

[6] Analista en Informática, Auxilliar de Investigación Alumno, Facultad de Ingeniería de la Universidad FASTA. **E mail:** julianw@ufasta.edu.ar

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.

Surge entonces la idea de los alumnos Bruno Constanzo y Julián Waimann de complementar el trabajo realizado en PURI y abordar el desarrollo de una solución de file carving como proyecto final de graduación de la carrera de Ingeniería Informática [1].

Se presenta en este trabajo una arquitectura de Framework de File Carvers – CIRA (Carving Inteligente para la recuperación de Archivos) – con el objeto de que pueda ser conocida, ampliada y utilizada por la comunidad científica.

2. File Carving

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 [2]. El proceso de file carving se basa en recuperar información que ha sido eliminada o es inaccesible debido a daños en el 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 en caso de algún sistema de archivos o disco dañado [3].

Existen varias técnicas de File Carving, algunas implementadas en herramientas, y otras aún 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) [5], Smart Carving (Pal, Memon et al) [7] o Semantic Carving (Garfinkel) [5]. Incluso algunas tienen varios enfoques, como por ejemplo Header/Footer1 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 aún 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.

2.1. Historia del File Carving

En el año 1999 el Laboratorio de Informática Forense del Departamento de Defensa de Estados Unidos (Defense Computer Forensics Lab), presenta el programa CarvThis que consistía en una herramienta que permitía recuperar de un dispositivo de almacenamiento aquellos archivos que estuvieran desvinculados de los metadatos que los representaban en el filesystem. Luego, se sucedieron una serie de proyectos fomentados por éste trabajo inicial que resultaron en el desarrollo de la aplicación "Foremost" por la Oficina de Investigaciones Especiales de la Fuerza Aérea del mismo país, (US Air Force Office of Special Investigations). Foremost es una herramienta open source para realizar file carving que en sus inicios implementaba únicamente header/footer carving, pero en el año 2005 extendió su funcionalidad para trabajar con la estructura interna de los archivos [4].

En el año 2005, Goldman y Roussev reimplementaron Foremost con una nueva base de código, creando Scalpel, un carver enfocado en la velocidad de procesamiento y el bajo consumo de recursos. Scalpel logró ubicarse como una herramienta de referencia en el ámbito forense. Ya en el año 2011, con la versión 2.0 de Scalpel, se agregó la capacidad de ejecutarse en multiprocesadores y procesadores gráficos de propósito general (GPGPUs), incrementando su performance. [5] Scalpel actualmente logra recuperar archivos de FAT, NTFS, ext2 / 3, HFS+ y desde particiones sin formato.

En paralelo a estos desarrollos se fueron dividiendo las investigaciones del File Carving en tres ramas.

Por un lado, comenzando en el año 2003, Memon, Shanmugasundaram y Pal comenzaron a presentar trabajos describiendo algoritmos para recuperar archivos fragmentados considerando al carving como un problema de grafos [6]. Su trabajo resultó en algoritmos que fueron implementados en el llamado Smart Carving™. Las características del Smart Carving hacen que sea una técnica altamente eficaz, y capaz de recuperar archivos fragmentados en diversas condiciones, además, separa el proceso de carving en etapas lo que brinda flexibilidad y permite aplicar diversas optimizaciones para la performance. [7]

Por otro lado, a partir del año 2007 con la incursión de Garfinkel en esta temática se comienza a trabajar en el desarrollo de un nuevo algoritmo basado en las técnicas de header/footer carving pero más complejo y capaz de recuperar archivos fragmentados bajo condiciones específicas pero estadísticamente relevantes.

Por último, Cohen en 2007 presenta un análisis profundo del problema de File Carving y un prototipo de un software desarrollado bajo una arquitectura de características muy avanzadas, donde propone la inclusión del preprocesamiento de imágenes, la discriminación de bloques,

la recuperación de archivos fragmentados y la validación de archivos.

Pese a todos estos avances, en la actualidad hay pocas herramientas que implementen los nuevos algoritmos de carving. Muchas herramientas dicen implementar carving avanzado, y en realidad implementan variantes del header/footer carving, la técnica más básica. Por otro lado, pocas herramientas que si los implementan lograron superar la etapa teórica para convertirse en programas maduros y usables en un entorno forense. Además, aún está presente el problema de los falsos positivos que presentan algunos algoritmos. [8]

2.2. Herramientas actuales de File Carving

A continuación se procede a describir los File Carvers más utilizados actualmente.

PhotoRec es una herramienta que implementa la técnica de Header/Footer y Header – File Sctructured Based Carving2. Se basa en el análisis de la imagen y en un enfoque inteligente de búsqueda de encabezados de archivos, logrando una mejor performance. Esta herramienta, además, agrega un paso de validación para evitar la extracción de archivos dañados o inválidos.

Scalpel es una de las herramientas basadas en Header/Footer más avanzadas y más utilizadas en la actualidad. Cuenta con una amplia gama de características, y se destaca por su alta velocidad. Sin embargo, tiende a extraer una gran cantidad de archivos dañados o inválidos.

A partir del año 2002 se comenzó a trabajar en las técnicas que luego se convirtieron en Smart Carving. Estas técnicas intentan resolver el problema de la fragmentación de archivos mediante una orientación a grafos. Se han publicado documentos posteriores muy prometedores, sin embargo, es recién a partir de 2009 cuando las principales implementaciones tienen lugar. Ha habido algunas investigaciones y trabajos académicos que implementaron Smart Carving [4], pero encontraron diversos problemas relacionados con las funciones de peso de las aristas del grafo. El único producto que implementa correctamente esta técnica es comercial y de alto costo. La herramienta Revit, basada en Smart Carving, se presentó en dos oportunidades en los desafíos DFRWS (Digital Forensic Research WorkShop) de los años 2006, 2007 pero a pesar lograr resultados interesantes su desarrollo se encuentra aún en fase experimental.

La técnica de Semantic Carving propone identificar el idioma utilizado en un bloque, para poder relacionarlo con bloques que se encuentran en el mismo idioma. De esta manera, un Semantic Carver debería relacionar los bloques en forma coherente para reconstruir un texto. S. Garfinkel ha presentado en el congreso DFRWS 2006 un prototipo de carver semántico llamado S2, escrito en C++, que permite incorporar distintos tipos de carvers y validadores para tipos de archivo específicos. Esta técnica es especialmente útil para formatos de archivos que almacenan texto.

La técnica de Bifragment Carving Gap fue presentada por Garfinkel en 2007 como una manera de buscar una solución al problema de la fragmentación de archivos. A partir de un análisis de amplio espectro genera un índice de "Relevancia de la bi fragmentación", y posteriormente desarrolla un algoritmo para la reconstrucción de archivos bajo ese escenario. No se han encontrado herramientas que implementen esta técnica.

El Carving con validación, también propuesto por Garkinkel en el año 2007, se basa en el uso de visores o validadores que comprueben la estructura de los archivos y verifiquen que se respete el formato al que pertenece. Puede utilizarse tanto durante la etapa de reconstrucción de archivos como en una etapa posterior para filtrar falsos positivos. Se entiende que la validación de archivos debe proporcionarse en forma independiente al algoritmo a utilizar, e incluso, independiente del software, y que debe dividir los resultados en carpetas en lugar de la prevención de la extracción de archivos no válidos. De esta forma, datos válidos pueden ser analizados rápidamente, y los datos no válidos pueden ser tratados posteriormente.

La técnica de Repackaging Carving se utiliza cuando hay archivos parcialmente recuperados. Trabaja a partir de agregar bloques al archivo recuperado hasta obtener un archivo válido. Esto genera que el resultante no sea idéntico al archivo original, pero permite recuperar partes del mismo que de otra manera sería ilegible [5].

In-place File Carving propone que en lugar de extraer los archivos de la imagen de disco analizada, se creen metadatos nuevos que referencian a los bloques presentes en la imagen. Utilizando un filesystem virtual se accede a los archivos como si se hubieran extraído las copias de la imagen. Ésta técnica mejora la performance, ya que la extracción no se realiza o se realiza en diferido,

reduciendo así el espacio de almacenamiento que acarrea el uso de un file carver [8].

El preprocesamiento de imagen más común que los file carvers realizan es el análisis de los metadatos del sistema de archivos para determinar qué bloques de la imagen están asignados y cuales no, a fin de obtener un subconjunto de bloques que serán procesados. Es importante recordar que el proceso de file carving se basa en recuperar información que ha sido eliminada o es inaccesible debido a daños en el sistema de archivos, en cuyo caso ese subconjunto a analizar sería el dispositivo completo.

El objetivo del preprocesamiento es intentar reducir la cantidad de bloques a analizar tanto como sea posible, a fin de lograr que algoritmos complejos como Smart Carving o Bifragment Carving tengan una mayor performance.

Una de las características que aún no ha sido implementada es la posibilidad de marcar los bloques ya recuperados en un proceso de carving. De esta manera, en un proceso de carving de tipo iterativo la cantidad de bloques a analizar podría reducirse en el tiempo. Otros tipos de preprocesamientos que serían deseables son el aislamiento de los bloques de tipo ASCII y UNICODE de los bloques de tipo binarios, con el fin de lograr disminuir la cantidad de bloques a analizar para la extracción de un tipo específico de formato de archivo.

2.3. Métricas para el File Carving

Existen distintos tipos de indicadores que se pueden utilizar para evaluar la performance de un file carver, su velocidad de procesamiento y la calidad de sus resultados. Los más utilizados son:

1. Cantidad de archivos recuperados
2. Cantidad de archivos válidos recuperados
3. Cantidad de archivos parcialmente recuperados
4. Cantidad de archivos no recuperados
5. Cantidad de falsos positivos.
6. Cantidad de falsos negativos.
7. Tasa de Precisión de carving: Coeficiente que surge de dividir la cantidad de archivos válidos sobre el total de archivos recuperados.
8. Tasa de Carving recall: Coeficiente que surge de dividir la cantidad de archivos recuperados sobre el total de archivos presentes en el dispositivo.
9. Tasa de Overcarving: Coeficiente que surge de dividir el tamaño de los archivos recuperados sobre el tamaño del dispositivo analizado. Idealmente debería ser un número <= 1.

   Un rendimiento óptimo en la evaluación de una herramienta es obtener una tasa de carving recall igual a 1, una tasa de precisión cercana a 1 y un overcarving lo más pequeño posible. Igualmente, el indicador clave para la medición de los algoritmos de carving es el recall, dado que asegura recuperar la totalidad de los archivos presentes en el dispositivo. Los indicadores de precisión y overcarving pueden ser ajustados con la aplicación de técnicas complementarias al carving, como In-place File Carving.

3. El producto CIRA

Si bien hay herramientas de propósito general que realizan file carving, ocasionalmente se escriben nuevas herramientas o scripts especializados para extraer archivos específicos con características particulares. Tal práctica genera una multiplicidad de scripts y programas demasiado especializados, poco probados y rara vez reutilizados, para realizar una tarea forense que debería ser reproducible, auditable y validable.

Surge entonces la necesidad de desarrollar un Framework de file carving – CIRA - que signifique un aporte a los productos disponibles en la actualidad, constituyendo un marco flexible y extensible, capaz de aplicar diferentes algoritmos y de añadir módulos de preprocesamiento, posprocesamiento y validadores.

El Framework de CIRA se estructura alrededor de un proceso definido en tres etapas: preprocesamiento, carving y posprocesamiento y está pensado de manera tal que el algoritmo de carving propiamente dicho es ajeno a los detalles de acceso a la imagen de disco "Lectura" y extracción de los archivos "Escritura". Esto permite, por ejemplo, que se extienda el Framework para operar, a través de una red, con una imagen de disco residente en otra computadora como podría ser un 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 filesystem virtual – una

técnica estudiada por autores como Richard, Roussev, Marziale y otros [8] [9].

Como parte del desarrollo se implementaron dos soluciones de preprocesamiento, cuatro algoritmos de file carving, dos soluciones de posprocesamiento 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 la performance final. 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 post procesamiento, la performance no tuvo un impacto significativo, y en condiciones similares es posible acercarnos a los valores de la herramienta con mejor performance hoy día que es Scalpel [6].

Los preprocesadores implementados en CIRA permiten excluir bloques del análisis y extracción. Uno de los preprocesadores permite que se excluyan bloques arbitrariamente, definidos como una cadena de texto. Esta característica se implementa a partir de la generación de un archivo de configuración con los rangos de bloques 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 permite, 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 ajustes, que están planeados como parte del trabajo futuro para mejorar sus capacidades de clasificación y selección.

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 archivos se descubrió que al comenzar el análisis de validez de un archivo 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. Esta experiencia se tomó como base para la 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.

Finalmente, para la etapa de posprocesamiento se desarrollaron dos posprocesadores, uno que realiza la verificación de los archivos extraídos por medio de un módulo de validación, y otro que calcula los hashes MD5 y SHA-1 de los archivos extraídos. Esta última característica suele utilizarse para verificar la integridad de los archivos extraídos o para compararlos con otras versiones disponibles y excluirlos, como por ejemplo, en el caso de haber recuperado archivos del sistema operativo.

4. Conclusiones y Trabajos Futuros

Partiendo de un proyecto de Investigación "PURI" cuyo objetivo fue la generación de un Proceso Unificado de Recuperación de la Información, se llegó al desarrollo de una solución de file carving CIRA que está a la altura de herramientas ya establecidas, tanto en sus prestaciones, como en el rendimiento obtenido y calidad de los archivos recuperados.

CIRA provee una arquitectura extensible y con grandes posibilidades de desarrollo.

A futuro se prevé implementar otros algoritmos de carving que le brinden mayor potencialidad; optimizar el código crítico para mejorar la performance; agregar nuevos módulos de pre y post procesamiento; adaptar CIRA a entornos distribuidos y por último incorporar técnicas de inteligencia computacional al análisis de imágenes o textos recuperados que permitan discriminar de un conjunto aquellos archivos que contienen un dato en particular.

Queda mucho por hacer y mucho por mejorar. Los autores estamos presentando este trabajo, fruto de la conjunción de un proyecto de investigación y un proyecto final de la carrera de Ingeniería Informática de la Universidad FASTA, a disposición de la comunidad para que pueda utilizarse en el ámbito forense o académico y CIRA pueda ser

conocido, utilizado, y pueda seguir creciendo

Header/Footer Carving es una técnica de recuperación de archivos que trabaja a partir de la búsqueda de un encabezado y un fin de archivo. De esta manera, se busca entre los bloques del dispositivo un header que indique el comienzo de un archivo de determinado tipo y a partir de este se añaden en secuencia los bytes contiguos hasta encontrar el footer que indica que el archivo llegó a su fin.

2 File Structure Based Carving es un algoritmo que se basa en analizar las estructuras internas del tipo de archivo, además de los datos que se encuentran en la cabecera. Es especialmente útil en formatos que guardan sus datos en bloques, como AVI, MP3, JPG, entre otros.

Referencias

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

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

[3] 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)

[4] POISELY, R., TJOA, S., and TAVOLATO, P.: Advanced File Carving Approaches for Multimedia Files, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA) 01/201

[5] GARFINKEL, S.: Carving contiguous and fragmented files with fast object validation. Digital Forensics Research Workshop -DFRWS (2007)

[6] RICHARD, G., ROUSSEV, V.: SCALPEL: A Frugal, High performance File Carver, DFRWS (2005)

[7] PAL, A., MEMON, N.: The Evolution of File Carving. IEEE Signal Processing Magazine, (2009) 59 –71

[8] RICHARD, G., ROUSSEC, V., MARZIALE, L.: In-place File Carving. International Federation for Information Processing - IFIP (2007)

[9] http://ocfa.sourceforge.net/libcarvpath/ accedido el 10 de Julio de 2013