RECONSTRUCCIÓN DE VOLÚMENES RAID

INTRODUCCIÓN

El Grupo de Investigación en Sistemas Operativos e Informática Forense de la Universidad FASTA lleva desde el año 2011 trabajando para definir y mejorar un Proceso Unificado de Recuperación de Información (PURI)[1]. Éste trabajo inicial se extendió para adaptarlo y aplicarlo tanto a Smart-phones como a Sistemas Distribuidos.

Durante el desarrollo de este proceso se detectaron aspectos de informática forense con fuertes carencias de herramientas y técnicas, que fueron llamados “nichos carentes”[2]. Múltiples proyectos finales de alumnos de la universidad FASTA se enfocaron en estos aspectos a fin de brindar soluciones a estas necesidades[3][4].

En el estudio de los Sistemas Distribuidos y entornos de “Cloud Computing” se analizaron cuestiones específicas de entornos de computación distribuida, ya sean servidores, máquinas virtuales o sistemas en la nube. Un aspecto de especial interés, tanto por el desafío técnico como por la necesidad de los peritos informáticos, fue la reconstrucción de arreglos de discos.

El uso de arreglos RAID, en sistemas de mediano a gran tamaño, presenta varios desafíos a los informáticos forenses, entre los que se encuentran el no contar con la capacidad de almacenamiento para realizar una adquisición completa del volumen RAID, que la adquisición del arreglo RAID no haya seguido un procedimiento adecuado para obtener la información completa, o que exista daño físico en el sistema. Si no se siguen los procedimientos correctos el resultado es una pila de discos (o imágenes de discos) potencialmente sin valor para la investigación judicial, ya que sería complicado acceder a la información en forma coherente.

En este trabajo se van a exponer conceptos básicos de RAID y sistemas de archivos, plantear una situación de problema, un entorno de pruebas y una técnica propuesta para realizar la reconstrucción de un arreglo RAID.

MARCO TEÓRICO

Arreglos RAID

RAID (*Redundant Array of Independent Disks*) es una tecnología que permite combinar múltiples dispositivos de almacenamiento y los presenta al *host* como un solo volumen virtual[5][6]. RAID establece una sinergia entre los dispositivos que conforman el arreglo, brindando las siguientes ventajas:

- **Rendimiento:** el funcionamiento en conjunto de los múltiples dispositivos abre la posibilidad de realizar operaciones de lectura y escritura en forma paralela, que no serían posibles si se tratara de un único dispositivo.
    - Velocidad: se logra una mayor tasa de transferencia al distribuir las lecturas y escrituras en múltiples dispositivos.

    - Operaciones E/S por segundo: cuando se puede paralelizar el acceso a las tramas en distintos discos se puede responder mayor cantidad de operaciones pequeñas sobre los discos.
- **Tolerancia a fallos:** RAID, en algunos de sus modos de operación, permite tener redundancia en los datos. En estos casos la falla de un disco no compromete la información, pero el rendimiento se ve degradado hasta reemplazar el dispositivo y restaurar el arreglo.
- **Capacidad:** como consecuencia de combinar los dispositivos, se obtiene un dispositivo virtual de igual o mayor tamaño que cada uno de los dispositivos individuales.
- **Economía:** éstas características se obtienen de combinar discos reales con un costo relativamente bajo. Si se buscara un dispositivo único real con las mismas características que un arreglo de discos, en caso de existir, probablemente sería mucho más costoso.

Dependiendo la configuración de RAID que se utilice, se refuerzan en mayor o menor medida estos aspectos. Como se verá más adelante, hay configuraciones que sacrifican la capacidad de almacenamiento por redundancia, o al revés, sacrifican la redundancia por mayor capacidad de almacenamiento. También hay configuraciones que establecen un balance entre capacidad y redundancia, a un costo de rendimiento.

Los metadatos de configuración de los discos que componen un arreglo RAID se almacenan en una estructura llamada *superbloque RAID*, o *estructura DDF* según la nomenclatura del SNIA[7]. Esta estructura guarda la información relevante para determinar a qué arreglo y unidad virtual pertenece cada disco, y los parámetros de configuración, como tipo de paridad, tamaño de banda, caché, entre otros.

La especificación de RAID define siete niveles de RAID. A continuación se detallarán los cuatro niveles más utilizados:

- **RAID 0:** los discos del arreglo RAID se fraccionan en tramas (*chunks*) de acuerdo a bandas, también llamadas *stripes*, lo que permite utilizar los discos en paralelo para realizar operaciones de lectura/escritura en cada disco. Este nivel de RAID brinda excelente rendimiento, pero no cuenta con redundancia de datos.
- **RAID 1:** también llamado “arreglo espejo”, define un disco de datos y uno o más discos espejo. Los discos espejo son copias idénticas del disco de datos y pueden reemplazarlo automáticamente si ocurre una falla. Este nivel de RAID permite la lectura en paralelo de las partes espejadas, pero la ventaja principal de este modo es que realiza un “respaldo automático” del disco de datos que permite proteger la información de la rotura física de todos menos uno de los discos que componen el arreglo.
- **RAID 5:** los discos del arreglo RAID se fraccionan en bandas, lo que permite realizar lectura y escritura en paralelo en los múltiples discos. Para cada banda, el segmento que corresponde a cada disco se denomina *chunk*, y se calcula una función XOR de los N-1 *chunks* de datos de la banda. El *chunk* de paridad se distribuye entre discos para distintas bandas del arreglo, y así se evita que uno solo de los discos cargue con la tarea de escritura asociada con el almacenamiento de paridad. Esto permite un mejor rendimiento de todo el arreglo, ya que el uso de los discos tiende a ser uniforme. Los

chunks de paridad permiten al arreglo soportar la falla de uno de los discos y continuar funcionando en forma degradada, a la espera del cambio del disco, y reconstruir el disco perdido cuando haya un reemplazo disponible. RAID 5 es un balance entre capacidad, rendimiento y redundancia de datos.

- **RAID 6:** puede considerarse como una variante de RAID 5 en donde se calculan dos funciones de paridad distintas, lo que permite recuperar el arreglo con la falla de dos discos. Nuevamente, es un balance entre capacidad, rendimiento y redundancia, más resistente a fallas que RAID 5. En particular, este nivel de RAID permite recuperarse del caso donde se produce la rotura de un segundo disco durante la reconstrucción.

En los niveles 5 y 6 es necesario además especificar cómo se van a distribuir los *chunks* de paridad entre las distintas bandas. Usualmente se aplican tres algoritmos estándar[8]: *Left Asymmetric*, *Left Symmetric* y *Right Asymmetric*, La diferencia entre las tres radica en el modo en el que se distribuye la paridad y el orden de los *chunks* de datos en cada banda, tal como se muestra en la Ilustración 1.

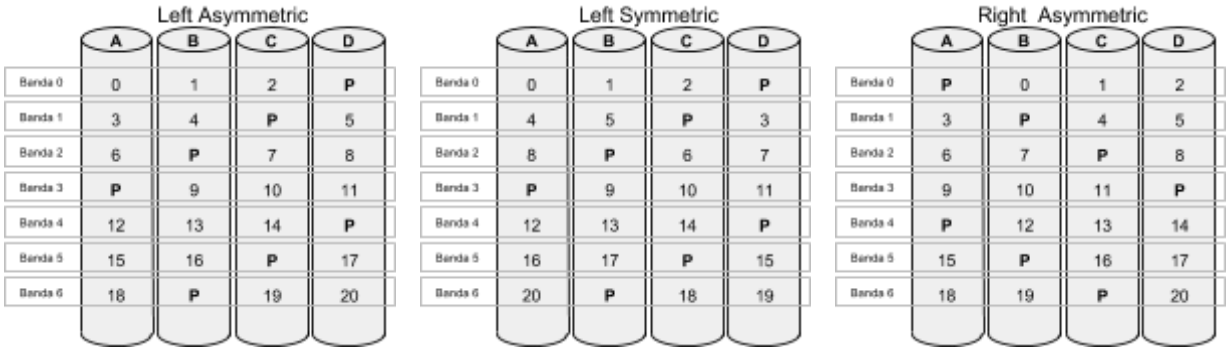

Ilustración 1 - Distribución de *chunks* de datos y *chunks* de paridad (P) de acuerdo a los distintos algoritmos, para un arreglo compuesto por cuatro discos.

RAID puede implementarse tanto por software como por hardware. Las implementaciones de RAID por software permiten flexibilidad y brindan las ventajas de los distintos niveles, usualmente a un costo de rendimiento ya que las tareas de administración de los discos y cálculo de paridades deben llevarlas a cabo el sistema operativo, introduciendo al procesador como parte del camino crítico del sistema de almacenamiento.

En el caso de un RAID implementado por hardware se pierde algún grado de flexibilidad, pero se gana en rendimiento: el sistema operativo delega en la controladora RAID las tareas de administración y cálculo, y se expone al *host* la controladora como uno o varios discos virtuales. Además, las controladoras RAID pueden implementar cachés de escritura y circuitos específicos para el cálculo de las funciones de paridad, que permiten acercar el arreglo al desempeño máximo teórico.

Particiones, MBR y GPT

Los discos, tanto reales como virtuales, deben particionarse para formatear la partición con un sistema de archivos. Las particiones son exactamente eso, partes de un disco completo a las que se le asigna un sistema de archivos configurado con parámetros determinados. En la década de 1980 IBM definió el *Master Boot Record* (*MBR*) como un sector del disco en el cual guardar código del cargador de arranque (*bootloader*) e información de las particiones del disco, que permite definir 4 particiones, almacenar unos 400 bytes de código de arranque y manejar particiones de hasta 2 TiB de tamaño. Con el paso de los años se fueron haciendo extensiones y modificaciones a MBR, pero era claro que el estándar necesitaba un reemplazo.

GPT (*GUID Partition Table*), definido por Intel, reemplaza a MBR, permitiendo definir cientos de particiones, y al utilizar 64 bits para almacenar el tamaño, su tamaño puede ser de hasta 8 ZiB. Además, GPT está diseñado para ser compatible con MBR y facilitar la transición. Actualmente casi todos los sistemas informáticos modernos utilizan GPT para definir su tabla de particiones.

El concepto de particiones es importante para la reconstrucción de un RAID porque el enfoque de éste trabajo se concentra en particiones tipo NTFS. Si hay una partición NTFS presente en el arreglo RAID, es posible aplicar la técnica y deducir el orden de los discos para todo el arreglo.

Conceptos NTFS

New Technology File System (NTFS), es un sistema de archivos desarrollado por Microsoft para sus sistemas operativos que presenta características especiales[9] que facilitan la técnica de reconstrucción de arreglos RAID planteada en este trabajo.

En NFTS todo el volumen se encuentra compuesto por bloques. El tamaño de dichos bloques es establecido al momento de la creación del sistema de archivos y siempre corresponderá a un múltiplo de dos por el tamaño del sector de disco. A su vez cada bloque puede contener información válida de un solo archivo.

Una particularidad realmente distintiva de NTFS es que en él “todo es un archivo”. Toda información válida dentro del volumen se encuentra contenida en un archivo, incluso la misma tabla de archivos y el sector de arranque. Así, todos los metadatos y estructuras de administración del sistema de archivos estan comprendidos en archivos del sistema. Probablemente el archivo de sistema más relevante es la *Master File Table* (tabla de archivos), identificada con el nombre $MFT. Esta tabla, por el hecho de encontrarse implementada como un archivo, presenta las mismas características que cualquiera de ellos. Por ejemplo, puede ubicarse en cualquier sector del volumen, crecer dinámicamente o incluso hallarse fragmentada.

La MFT cuenta con un registro por cada archivo o directorio contenido en el volumen al que pertenece. Esto no excluye a los archivos de sistema. De esta manera también existe un registro referente a la misma tabla, denominado $MFT, y un registro al archivo que contiene el sector de arranque del volumen denominado $Boot, entre otros. La MFT se implementa como una secuencia de registros de tamaño fijo (1024 bytes) que representan los metadatos de cada archivo o directorio en el sistema, lo que brinda la oportunidad de utilizarla como recurso en la reconstrucción de un arreglo de discos.

Cada registro se compone de un encabezado y una estructura dinámica de atributos no necesariamente presentes en todos los registros. A su vez, cada atributo también se materializa mediante un encabezado y contenido dinámicos. Esto haría pensar que el tamaño del registro también debería ser dinámico pero no lo es. Cuando los datos de los atributos no caben en los 1024 bytes asignados por registro, se almacenan direcciones a bloques de contenido en lugar del contenido en sí. En este caso el contenido se denomina “no residente” por el hecho de que el dato no reside en la misma tabla $MFT.

Para el método de reconstrucción de discos que se verá más adelante, lo que interesa de la MFT son los datos del encabezado de cada registro. Entre otros datos cada uno de ellos cuenta con un identificador: el número de registro MFT. Este identificador se encuentra implementado como un número entero de 4 bytes que crece secuencialmente con cada registro MFT.

EL PROBLEMA

Se plantea una situación hipotética para establecer las condiciones del entorno de pruebas y las condiciones en las que se tiene que trabajar para intentar reconstruir el arreglo:

> *Una empresa informática sufrió una falla en uno de sus servidores, y el arreglo RAID 5 de N discos se corrompió. Por malas políticas del departamento TIC, no cuentan con información sobre la configuración del arreglo de discos. Para empeorar las cosas, un intento fallido de recuperación resultó en la re-escritura de los superbloques RAID, por lo tanto tampoco se pueden utilizar para recuperar la configuración original. Es decir, se cuenta con N discos de los cuales se desconoce el orden, tamaño de chunk y algoritmo de distribución de la paridad, de los cuales es urgente recuperar información crítica para el negocio.*

Se plantea como caso un inconveniente no judicial para mostrar que ésta técnica es aplicable a cualquier situación de recuperación de la información. Su aplicabilidad en entornos judiciales/forenses es igualmente válida, siempre y cuando se documente el proceso para garantizar la reproducibilidad y replicabilidad del procedimiento.

ENTORNO DE PRUEBAS

Para simular el problema propuesto, se trabajó en un sistema Debian Linux con la utilidad *mdadm*, que permite realizar arreglos RAID por software. El proceso de creación de un caso es prueba es el siguiente:

1. Se generan *N* archivos vacíos, de nombre aleatorio, que van a funcionar como discos virtuales. Cada archivo se monta como un disco a través de la interfaz de *loopback* de Linux.

2. Se crea el arreglo RAID 5 con *mdadm*, utilizando los *N* dispositivos *loopback*. El tamaño de *chunk* y algoritmo de distribución de paridad son datos aleatorios que no se almacenan.

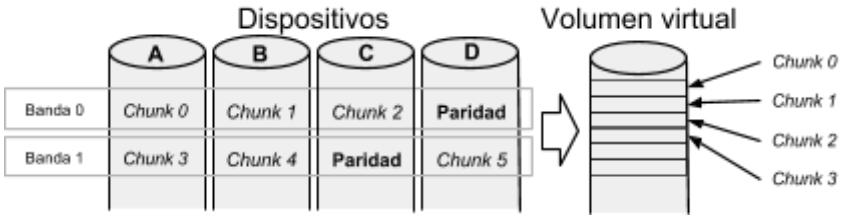

Ilustración 2 - RAID 5 de 4 discos (A a D) con paridad *Left Asymmetric*.

3. Se monta el arreglo RAID recién creado, y se crea en él una partición NTFS con un desplazamiento aleatorio del comienzo del disco (alineado a sector). Esto sirve para simular que la partición se encuentra en el medio del disco virtual RAID, como si hubiera otras particiones presentes en el disco.

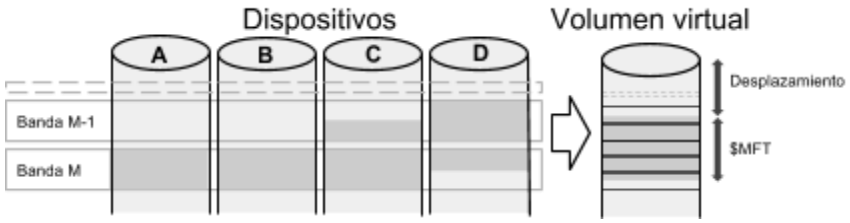

Ilustración 3 - Ejemplo de distribución de la MFT en los dispositivos físicos y el virtual.

En la ilustración 3 se ejemplifica cómo se vería si la MFT ocupase 6 *chunks* comenzando en la banda M-1 del dispositivo C y finalizando en el *chunk* de la banda M del dispositivo D. También se ejemplifica cómo se vería la misma MFT en el dispositivo virtual ocupando 5 *chunks* si existiera 1 *chunk* de paridad en los dispositivos físicos. Cabe destacar que para que el estudio sea posible la MFT debe ocupar una cantidad considerablemente mayor de *chunks*.

4. La partición NTFS se monta y se crean en ella 3 archivos de 1 MiB con contenido aleatorio. De éstos archivos se calcula el digesto (*hash digest*) MD5 para verificar luego que la reconstrucción ha sido exitosa.

5. Dentro de la unidad NTFS se crean 17.000 archivos pequeños con contenido “Hola mundo”. Sin estos archivos se corre el riesgo de la MFT no sea lo suficientemente grande y no se pueda deducir la distribución de paridad.

6. Se desmonta el volumen RAID y se desconecta de la interfaz *loopback*. Luego, se detiene el arreglo RAID con *mdadm*.

7. Se eliminan los superbloques de cada disco con *mdadm*.

8. Finalmente se desconectan los archivos de la interfaz *loopback*.

Se creó un *script* de *bash* para generar automáticamente nuevos casos de prueba con distinta cantidad de discos y configuraciones de forma fácil, para validar la técnica propuesta con múltiples pruebas.

TÉCNICA PROPUESTA

Hasta aquí se tienen *N* archivos que representan los *N* dispositivos, denominados en las ilustraciones como A, B, C y D. Si bien en los ejemplos se muestran en orden por claridad, este orden no es conocido a priori. A continuación se describe el proceso de reensamblado:

Paso 1: Se buscan en todos los discos del arreglo las cabeceras NTFS y los registros FILE. En un sistema de archivos NTFS siempre se encuentran dos cabeceras, una al principio de la partición y otra al final. Debido a la redundancia de RAID 5 es posible que se encuentren copias de una o ambas cabeceras en el *chunk* de paridad de la banda correspondiente. Encontrar las cabeceras NTFS ayuda a establecer los posibles puntos de comienzo del sistema de archivos dentro del arreglo RAID.

Paso 2: De los registros FILE interesa el número de registro MFT, que ayuda a determinar el orden de los discos. Si un mismo conjunto de registros está presente en dos discos, ésto indica que uno de esos discos, para esa banda, contiene un *chunk* de paridad y los demás están vacíos.

Paso 3: Hay dos cuestiones para analizar con respecto al número de registro MFT:

En primer lugar, se debe analizar la longitud de los conjuntos de registros con numeración contigua por cada dispositivo. Si en el dispositivo A se encuentran los registros 3001, 3002, 3003, …, 4000, 7001, 7002, 7003, ... Puede apreciarse un salto en la secuencia del 4000 al 7001 y se puede deducir que el *chunk* tiene la capacidad de almacenar 1000 registros FILE de la MFT y así se determina su tamaño. La ilustración 4 muestra este concepto.

En segundo lugar, se deben ubicar los registros FILE de la MFT iniciales, y seguir la secuencia cuando salta de un disco a otro. De éste análisis se empieza a determinar el orden de los discos, aunque no alcanza para determinar cuál es el disco inicial.

El seguimiento de la secuencia de registros FILE de la MFT también permite detectar los *chunks* de paridad. En la ilustración 4 puede verse un arreglo RAID 5 con distribución *Left Asymmetric* de 4 dispositivos en orden, con los registros FILE de la MFT y la secuencia indicada con flechas. Puede apreciarse cómo la presencia de un *chunk* de paridad altera el orden común de salto de la secuencia de registros entre dispositivos, y cómo también indica la secuencia que siguen los dispositivos en el arreglo.

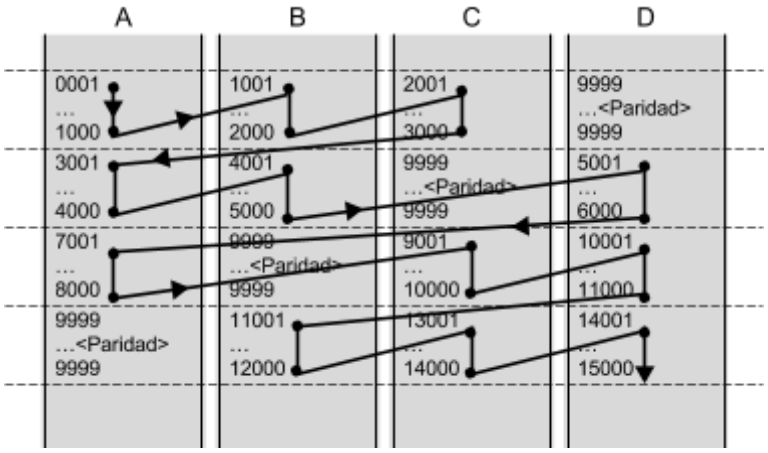

Ilustración 4 - Distribución de los registros de la MFT en un arreglo RAID 5 *Left Asymmetric*. Se asume que entran 1000 registros FILE de MFT por *chunk* del arreglo y se muestra el orden.

Para el caso de la distribución *Left Symmetric*, el salto de un disco a otro siempre es el mismo, pero el *chunk* de paridad introduce un salto a la siguiente banda del arreglo.

Nota: La técnica busca determinar el orden de los discos, o si eso no es posible, reducir la cantidad de combinaciones que deben intentarse para determinar el orden real. En el caso de la distribución *Left Symmetric* no se puede determinar el disco inicial, sin embargo se logra reducir las combinaciones posibles de *N*! a *N*. En el caso de las distribuciones *Left Asymmetric* y *Right Asymmetric*, si el tamaño de *chunk* es demasiado grande, puede no haber suficiente información para determinar el primer dispositivo del arreglo.

Paso 4: Las combinaciones posibles para reconstruir el arreglo se verifican utilizando *mdadm* en el modo en que ignora la información de los superbloques y monta el arreglo RAID con la configuración suministrada manualmente, siguiendo las deducciones previas.

Paso 5: Con el arreglo reconstruido, se monta la unidad en un directorio. Si se puede acceder a la misma, casi con seguridad se ha tenido éxito en la reconstrucción. Para verificar estrictamente el éxito de la técnica, se comparan los digestos MD5 de los archivos grandes con los digestos MD5 calculados previo al desarmado del arreglo RAID y la eliminación de los superbloques.

Se realizaron múltiples pruebas con el entorno de pruebas y la técnica propuesta, y fue posible la validación por medio del montaje de la unidad y la verificación de los digestos MD5 de los archivos grandes.

CONCLUSIONES Y TRABAJO FUTURO

La técnica propuesta permite reconstruir un arreglo RAID 5 contando con *N* imágenes, una por cada dispositivo del arreglo, en condiciones en las que no se conoce el orden de las mismas, ni el tamaño de *chunk* ni el algoritmo de distribución de paridad. Si bien la técnica se apoya en las estructuras del sistema de archivos NTFS, es posible extender esta técnica para aplicar a otros sistemas de archivos.

Esta técnica debería utilizarse como último recurso, ya que si se procediera con las tareas de recolección y adquisición de los discos de forma ordenada y metódica no sería necesaria. Ésta técnica y las tareas asociadas para la correcta adquisición se incorporaron como una extensión al proceso PURI.

Como trabajo futuro, se está evaluando implementar esta técnica en una herramienta informática para simplificar la tarea de reconstrucción. También es posible extender la técnica para poder trabajar en base a archivos conocidos, y de ésta forma lograr independencia del sistema de archivos de las particiones que se desee reconstruir.

Si el arreglo es RAID 6, o si utiliza un algoritmo de distribución de paridad no estándar, la técnica tal como se expuso presenta algunas dificultades. Con un estudio detallado de RAID 6 y de los algoritmos de distribución de paridad propietarios, la técnica puede adaptarse para funcionar en estas situaciones.

REFERENCIAS

- [1] DI IORIO, Ana H., SANSEVERO, Rita E., CASTELLOTE, Martín A., PODESTÁ, Ariel, GRECO, Fernando, CONSTANZO, Bruno, WAIMANN, Julian. (2012) "La recuperación de la información y la informática forense: Una propuesta de proceso unificado", Congreso Argentino de Ingeniería CADI 2012.
- [2] DI IORIO, Ana H., SANSEVERO, Rita E., CASTELLOTE, Martín A., PODESTÁ, Ariel, GRECO, Fernando, CONSTANZO, Bruno, WAIMANN, Julian. (2013) "Determinación de aspectos carentes en un Proceso Unificado de Recuperación de Información digital", Jornadas Argentinas de Informática Forense JAIF 2013.
- [3] CONSTANZO, Bruno, WAIMANN, Julián. “El Estado Actual de las Técnicas de File Carving y la Necesidad de Nuevas Tecnologías que Implementen Carving Inteligente”. (2012). 1er. Congreso Argentino de Ingeniería.
- [4] DI IORIO, Ana H., CASTELLOTE, Martín A., PODESTÁ, Ariel, GRECO, Fernando, CONSTANZO, Bruno, WAIMANN, Julian. “El framework CIRA, un aporte a las técnicas de file carving”. (2013). Revista Argentina de Ingeniería.
- [5] TANENBAUM, Andrew S. “Sistemas Operativos Modernos”, Capítulo 4, Prentice Hall Hispanoamericana, 1993.
- [6] TANENBAUM, Andrew S. “Structured Computer Organization”, páginas 89 a 93, 5ta edición, Pearson Prentice Hall, 2006.
- [7] SNIA. (2008) “Common RAID Disk Data Format Specification”.
- [8] FAY-WOLFE. (2008). “RAID Rebuilding 101”. CSC-486 Network Forensics, University of Rhode Island. Disponible en http://media.uri.edu/cs/csc486_wmv/RaidRebuilding_TOC.pdf

[9] RUSSON, Richard, FLEVEL, Yuval, “NTFS Documentation” antes disponible en http://linux-ntfs.sourceforge.net/ntfs/index.html, copia disponible en http://dubeyko.com/development/FileSystems/NTFS/ntfsdoc.pdf

Autores
Hugo Curti. Ingeniero en Informática, Docente e Investigador en Universidad FASTA y Universidad Nacional del Centro, hcurti@gmail.com
Ariel Podestá. Ingeniero en Informática, Docente e Investigador en Universidad FASTA, arielpodesta@gmail.com
Bruno Constanzo. Ingeniero en Informática, Investigador en Universidad FASTA, bconstanzo@ufasta.edu.ar
Juan Ignacio Iturriaga. Ingeniero en Informática, Docente e Investigador en Universidad FASTA, Juan@ufasta.edu.ar
Martín Castellote. Ingeniero en Informática, Docente e Investigador en Universidad FASTA, tinchocapoeira@ufasta.edu.ar

Resumen
La proliferación de entornos distribuidos en la informática ha generado un cambio de paradigma que influye prácticamente en todas las actividades asociadas, incluso la informática forense.
El Grupo de Investigación de Sistemas Operativos e Informática Forense de la Facultad de Ingeniería de la Universidad FASTA ha desarrollado un Proceso Unificado de Recuperación de Información (PURI) que sirve de guía, tanto a informáticos forenses como operadores judiciales, en los pasos a seguir para recuperar la información almacenada digitalmente en un equipo de computación.
Se presenta en este trabajo una propuesta de los pasos a seguir por los informáticos forenses en el caso particular de encontrarse con un arreglo de discos RAID, del que se desconoce su estructura, al cual se le debe realizar una pericia.
Palabras clave: recuperación de información - arreglos RAID - informática forense

Abstract
The proliferation of distributed environments in information technology has generated a paradigm shift that affects practically in every associated activities, even digital forensics. The Operating Systems and Digital Forensics Research Group of Universidad FASTA has developed a Unified Process for Information Recovery (PURI) which serves as guide, both for digital forensics experts and judicial employees, in the steps to follow to recover the information that is digitally stored in a computer.
This work proposes the steps that a digital forensics expert must follow in the special case of finding a RAID array, from which they do not know its structure, that must be examined.
Keywords: information recovery - RAID arrays - digital forensics