
*Tema 3: Análisis forense de Windows.
Sistema de ficheros.*

IES Zaidín-Vergeles

02 de febrero de 2021

Tabla de contenidos

1. Introducción	1
2. MBR (Master Boot Record)	3
2.1. ¿Qué es el MBR?	3
2.2. Estructura y tareas del MBR	4
2.3. El MBR está dañado ¿qué hago?	5
2.4. Alternativas a MBR	6
3. GPT	9
3.1. ¿En qué se caracterizan las particiones GPT?	9
3.2. Esquema de la GPT	10
3.3. Estructura de la cabecera GPT	11
3.4. Estructura de la entrada de partición	12
3.5. ¿En qué casos se utiliza la tabla de particiones GUID?	13
4. Análisis forense de particiones de discos MBR y GPT	15
4.1. Forense de GPT	16
4.2. Extracción de particiones GPT	18
4.3. Artefactos GPT	18
5. Sistema de ficheros FAT, File Allocation Table	21
5.1. Convención de nomenclatura de FAT	22
6. Sistema de ficheros NTFS, New Technology File System	25
6.1. Clústeres y sectores en una partición NTFS	26
6.2. MFT (Master File Table)	26
6.2.1. 6.2.1 Metadatos almacenados de la MFT	27
6.3. Resumen de la estructura	29
7. Borrado de ficheros en Windows	31
7.1. Slack Space en detalle	32
7.1.1. 7.1.1 File Slack	32
7.1.2. 7.1.2 RAM Slack	33
7.2. Análisis de fichero y carving	33
7.3. Orphan Files	36

CAPÍTULO 1

Introducción

El objetivo que se persigue ante un análisis pericial es poder responder completamente a las preguntas que se plantean. Una vez que podemos determinar esta cuestión, el detallar los sucesos a través de una secuencia temporal de eventos será la base para definir el cuándo. Si estamos trabajando en incident response, especificar el punto de entrada o inicio del suceso es fundamental. Poder detectar la vulnerabilidad explotada, poder llegar al quién y al porqué del problema serían puntos clave si se piensa en un proceso judicial.

El procedimiento de análisis dependerá del caso y tipo de incidente, pero en general se trabaja con las imágenes de los sistemas de ficheros. Siempre se va a realizar un timeline o análisis de secuencia temporal para establecer el marco en el tiempo. Se realizarán búsquedas ciegas por contenidos específicos para acotar las pruebas, siempre se puede recurrir a la recuperación de binarios y documentos (borrados o corruptos) y dependiendo del tipo de caso se realizará un análisis de código (pueden existir virus, troyanos, rootkits, etc.).

Por ejemplo, son muy habituales los casos de competencia desleal y fuga de información por parte de empleados de una empresa. Si nos encargan hacer una pericial que pruebe este tipo de comportamientos haremos clonado de los discos duros de los equipos implicados. Dentro de un primer análisis siempre es muy útil buscar datos en el slack space, archivos ocultos, procesos no usuales, sockets abiertos, cuentas de usuario extrañas y es crítico determinar el nivel de seguridad del sistema, posibles agujeros, etc... En más de una ocasión empleados «molestan» dejan sorpresas en sus equipos o en el servidor de la empresa que quedan latentes. Si por el contrario estamos ante un incident response, los pasos a seguir en una investigación

requieren el ir recabando datos de más volatilidad a menos volatilidad. Primero, se empieza recabando información de conexiones de red y luego se desconecta la red. El siguiente paso es adquirir los procesos en ejecución y la memoria del Sistema. Una vez hecho esto desconectamos el equipo de la corriente y se hacen las imágenes de los discos duros (clonado), se continúa verificando el incidente estudiando logs, IDS, logs de SSOO, aplicaciones, se hacen correlaciones.

Otra cuestión importante es que dentro de una investigación no podemos confiar en el

sistema que se está analizando ya que el atacante lo puede haber comprometido. Las herramientas usadas para examinar un sistema en vivo deben ser herramientas o comandos del SSOO, se deben usar los mínimos recursos del propio sistema y se debe alterar lo mínimo posible al sistema comprometido. Entre los datos no volátiles tenemos logs, ficheros, emails y toda la información relevante del sistema

CAPÍTULO 2

MBR (Master Boot Record)

El MBR es una reliquia de los primeros días de la tecnología de la PC. Se introdujo por primera vez en 1983 con el IBM PC DOS 2.0 y desde entonces ha sido un elemento esencial, sobre todo en los ordenadores con Windows. Aquí te mostramos cómo este pequeño elemento, al principio de soportes de datos con formato, es capaz de arrancar sistemas operativos altamente complejos. Además, te explicamos qué hacer si un MBR deja de cumplir su función de arrancar el PC.

2.1 ¿Qué es el MBR?

El MBR o master boot record es el primer sector físico de un portador de datos (por ejemplo, un disco duro, una memoria USB) que se utiliza para arrancar (iniciar) los ordenadores. Para esto, el ordenador debe disponer de un BIOS y un sistema operativo x86.

Nota: x86 se refiere a una arquitectura de procesador específica introducida por Intel en 1976. Entre otras cosas, procesa registros de instrucciones especiales, desarrollados por ejemplo por los fabricantes de chips Intel y AMD, para controlar el sistema operativo. Al principio, predominaban los procesadores x86 con una arquitectura de 32 bits (capacidad de procesamiento: 32 bits), los sistemas actuales funcionan con procesadores x86 de 64 bits más potentes (capacidad de procesamiento: 64 bits). El estándar de 64 bits x86 también se conoce como x64.

El MBR siempre tiene la misma dirección estándar en los soportes de datos. Cilindro 0, cabeza 0, sector 1. Normalmente tiene un tamaño de 512 bytes, que corresponde al tamaño de un sector en un medio de almacenamiento.

El MBR se puede encontrar en casi todos los **medios de almacenamiento externo** (por ejemplo, las memorias USB) que son compatibles con la tecnología de PC (arquitectura x64/x86) y pueden funcionar con Windows. En los soportes de datos que no están diseñados para arrancar un PC, el MBR no está integrado operativamente, sino que solo sirve

como **fuente de información legible**. Por ejemplo, los reproductores de archivos de audio leen allí la información sobre la ubicación y el tamaño de las particiones que contienen los archivos MP3 que se vayan a reproducir.

2.2 Estructura y tareas del MBR

El master boot sector siempre consta de al menos cuatro componentes:

- Programa de inicio (bootloader)
- Soporte de datos, firma de disco (a partir de Windows 2000)
- Tabla de particiones maestra
- MBR o firma de arranque (Magic number)

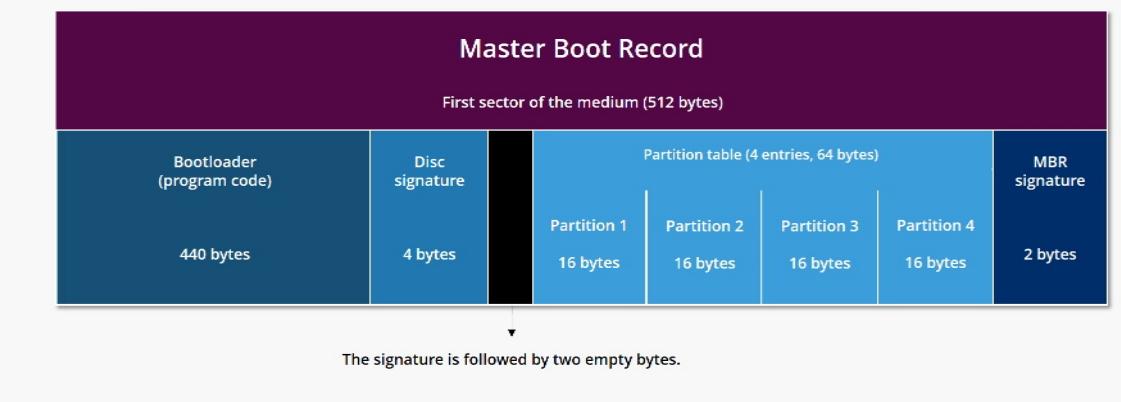
El **programa de inicio** se encuentra en los primeros 446 bytes del MBR. El software, de pequeño tamaño, se activa cuando se arranca el PC e inicia el proceso de arranque a nivel operativo. Esto pone en funcionamiento una rutina más extensa con pasos de procesamiento estandarizados, que culmina con el sistema operativo (por ejemplo, Windows) listo para su uso. Para poder controlar este proceso, se dirige la memoria principal instalada en el ordenador. Se activa si el ordenador puede utilizar energía eléctrica para las operaciones de procesamiento necesarias cuando lo encienden.

Los ordenadores Windows (Windows 2000 y posteriores) identifican los soportes de datos con una tabla de partición a través de la **firma del disco o del soporte de datos**.

La **tabla de particiones** documenta las divisiones de los soportes de datos en secciones de almacenamiento separadas. Con este fin, utiliza cuatro entradas de 16 bytes cada una, que indican la ubicación y el tamaño de cada partición. Esto indica dónde comienza y termina una partición C: o D:. La tabla también contiene información sobre el tipo de soporte de datos, por ejemplo, “FAT32”, “LINUX Native” o “Soporte de datos dinámico”. El orden de las particiones en la tabla de particiones no siempre corresponde al orden físico en el disco duro.

El **MBR o firma de arranque** contiene las cadenas 55 y AA en dos bytes. Gracias a su codificación característica, que siempre se encuentra al final del sector MBR, un registro de arranque principal está claramente identificado como tal. Si esta información no está presente, el sector de arranque principal no se identifica y el proceso de arranque se cancela con un mensaje de error.

Structure of a Master Boot Record (MBR)



Las actividades del MBR requieren un activador encendido por el BIOS (Basic Input/Output System) al encender el PC. La **BIOS** es un software especial, también conocido como **firmware** (firm: firme en este contexto). Se encuentra en la placa base de un PC con arquitectura x86, donde está incrustado en un chip especial (por ejemplo, un chip EPROM, una memoria flash). La BIOS permanece como un componente fijo, incluso cuando el ordenador está apagado.

La propia BIOS no necesita saber exactamente cómo está particionado un disco. Solo asegura que el **bootloader** del MBR se cargue en la memoria y se ejecute. Si se ha leído el sector de arranque principal y su bootloader está activo en la memoria de trabajo, la partición activa (o sea, de arranque) de un disco duro dividido se determina primero por medio de la **tabla de particiones**.

Cuando este se encuentra, se produce una reacción en cadena según el principio chain loading. El sector de arranque identificado de la partición direccionada se integra operativamente y el **bootloader de la partición** toma el control en la memoria principal. Después, se ejecutan procesos y rutinas más extensos que se encargan de la puesta en marcha real del sistema operativo. Como el propio bootloader de la partición realiza tareas más complejas, suele ser más grande que el programa de arranque del MBR.

Si el ordenador tiene instalado **más de un sistema operativo**, el proceso de arranque se detiene antes de terminar hasta que el usuario hace una elección (por ejemplo, entre Windows 7 y Windows 10). Estos **bootmanager** especiales suelen estar temporizados; si no hay ninguna entrada externa, al cabo de un periodo de tiempo determinado el sistema operativo preferido se inicia automáticamente.

2.3 El MBR está dañado ¿qué hago?

Si un PC x86 no arranca, a menudo se debe a un error en el sector de arranque principal. Para esto, basta con que la BIOS no pueda leer los dos bytes de la firma MBR. En esos casos, existen diversas **estrategias de solución de problemas**, que dependen principalmente del sistema operativo instalado. Los usuarios de Windows de hoy en día tienen dos métodos principales a su disposición:

- reparación automática del sistema con un medio de emergencia (CD, DVD, memoria USB),

- reparación manual a través de la línea de comandos.

Con el **método automático**, primero se debe cambiar el medio de arranque principal en la BIOS. De lo contrario, la rutina de arranque se interrumpirá, ya que buscará en vano un MBR intacto en la partición activa del disco duro integrado. El ordenador arranca después de la modificación de la BIOS, por ejemplo, a partir de un DVD de Windows 7, se puede seleccionar la opción “Opciones de reparación del ordenador” después de algunos pasos intermedios. Despues de otros pasos intermedios, finalmente se llega a la reparación del sistema, que restablece automáticamente el MBR.

Con el **método manual**, hay que reparar el registro de arranque principal con la herramienta de línea de comandos del símbolo del sistema de comandos de Windows (cmd.exe) e iniciar allí las rutinas de reparación con comandos estandarizados (bootrec/fixmbr, bootrec/fixboot). La manera de llegar a la línea de comandos depende en gran medida del sistema operativo instalado. Además, el procedimiento depende de si Windows se puede iniciar o si, en cambio, es imposible acceder a este.

Sin embargo, aparte de los errores en el sector de arranque principal, puede haber otras causas. Podría incluso tratarse de un **virus de arranque** que se haya almacenado en el sector de arranque y que se active al encender el ordenador. Si es posible, solo usuarios experimentados deben realizar reparaciones en el MBR, ya que existe el riesgo de pérdida de datos. Por esto, hay que acordarse de hacer una copia de seguridad del sistema y de los datos importantes del usuario. Si la recuperación de la partición MBR no tiene el éxito deseado o si se produce un error, el hardware puede quedar dañado de forma irreparable.

2.4 Alternativas a MBR

El registro de arranque principal tiene algunas desventajas, como la falta de mecanismos para la resolución de problemas (por ejemplo, si el MBR se corrompe), las restricciones en el tamaño del disco duro o la manipulación de las particiones. De este modo, solo se admiten las particiones MBR con una capacidad de almacenamiento de hasta 2 TB. En su forma original, el esquema de particiones clásico solo puede manejar **4 particiones**.

En la práctica, las limitaciones se pueden compensar parcialmente mediante algunos trucos y alternativas. De esta manera se puede modificar el sector de arranque y crear una **partición extendida**, que a su vez se divide en **particiones lógicas** más pequeñas. Sin embargo, esos “trucos solo son soluciones de emergencia que no son realmente adecuadas al ritmo del desarrollo actual de la tecnología informática.

Los sistemas actuales están dejando de lado el formato MBR para adoptar gradualmente el sistema de partición del disco duro conocido como **formato GPT**. GPT demuestra su superioridad en el ámbito de la seguridad de datos, puede manejar discos duros más grandes y varias particiones. Los soportes de datos con formato GPT trabajan estrechamente integrados con una interfaz de firmware, que desde hace tiempo está sucediendo a la BIOS. Por ejemplo, a partir de Windows Vista Service Pack 1 (versión x86 de 64 bits), es posible utilizar el **Unified Extensible Firmware (UEFI)**. La nueva tecnología de arranque es una especie de “sistema operativo en miniatura que soporta procesadores de 64 bits de fábrica de forma nativa.

Como ya existen alternativas más flexibles y eficientes, el MBR ha ido perdiendo importancia con el tiempo. Hoy en día, los sectores de arranque principal se utilizan principal-

mente para arrancar **ordenadores con hardware antiguo**. Además, la tecnología MBR se sigue utilizando ampliamente por **motivos de compatibilidad**.

CAPÍTULO 3

GPT

La **tabla de particiones GUID** es un estándar para **configurar las tablas de particiones** de los medios de almacenamiento, especialmente los discos duros. La GPT forma parte de la UEFI, siglas de Unified Extensible Firmware Interface (interfaz de firmware extensible unificada), una especificación que define la interfaz entre el firmware y los sistemas operativos durante el proceso de arranque, desarrollada y lanzada en 2000 como sucesora de BIOS. Las particiones GPT también pueden **utilizarse independientemente de la UEFI**, con algunas limitaciones. En este caso, el requisito previo es que el sistema operativo y el disco duro a particionar sean compatibles con el estándar. Gracias a las mejoras que ofrece, el estilo GPT ha sustituido en gran medida al registro de arranque principal o Master Boot Record (MBR), el tipo de arranque tradicional.

Nota: Las siglas GUID se refieren al Globally Unique Identifier (en castellano, identificador único global). Se trata de un número exclusivo de 128 bits (16 bytes) que se asigna a cada archivo o documento y que, en el caso de los discos duros, también permite identificar a los medios de almacenamiento y tipos de particiones de forma inequívoca.

3.1 ¿En qué se caracterizan las particiones GPT?

En la actualidad, una GPT partition es recomendable por varias razones. La más importante es que la tabla de particiones GUID utiliza entradas de 64 bits para el direccionamiento, por lo que el **tamaño máximo de cada partición es de 18 exabytes**, lo que corresponde a unos 18 mil millones de gigabytes. Esta característica es imprescindible para la nueva generación de soportes de datos, que ya suelen ofrecer varios terabytes de **espacio de almacenamiento**, incluidos los de uso doméstico. El registro de arranque principal clásico solo admite un máximo de dos terabytes por partición, lo que no basta para muchos discos duros modernos. El resto de **ventajas y características de la partición GPT** pueden resumirse de la siguiente manera:

- **Particiones primarias ilimitadas:** aunque, en teoría, la tabla de particiones admite un número ilimitado de particiones primarias para estructurar el espacio de almacenamiento, en la práctica, los sistemas operativos establecen un límite. Por ejemplo, el valor de Windows asciende a 128, lo que resulta más que suficiente.
- **Protección mediante sumas de verificación CRC32:** las sumas de verificación garantizan la integridad de la cabecera GPT, ya que permiten detectar los sectores defectuosos que la dañan, entre otras cosas.
- **Identificación clara de particiones y medios de almacenamiento:** como ya hemos mencionado, con la tecnología GUID, todas las particiones y medios de almacenamiento obtienen un número de identificación único.
- **Copia de seguridad de la cabecera:** la cabecera de la tabla de particiones GUID no solo queda respaldada por la suma de verificación que hemos descrito, sino también por una copia de seguridad idéntica, lo que aumenta la seguridad de los metadatos de la partición y minimiza el riesgo de perderlos en caso de un fallo del hardware.
- **Compatibilidad con sistemas anteriores:** el llamado Protective Master Boot Record (MBR protector) del sector 0, el primer bloque de datos de un disco duro GPT, asegura que casi todos los sistemas operativos, servicios y herramientas diseñados para la partición MBR funcionen también con la GPT.

3.2 Esquema de la GPT

La tabla de particiones GUID proporciona un esquema claro de la división de la memoria del disco duro. En general, su estructura consta de los siguientes cuatro sectores:

- **Registro de arranque principal protector:** en primer lugar, está el mencionado MBR protector, que garantiza la compatibilidad con otros estilos de partición anteriores.
- **Tabla de partición GUID primaria:** cabecera GPT y entradas de partición.
- **Particiones:** a la cabecera y las entradas de partición les siguen las correspondientes unidades en las que se divide el espacio de almacenamiento, es decir, las diferentes particiones.
- **Tabla de partición GUID secundaria:** copia de seguridad de la cabecera GPT y las entradas de partición, reflejando el mismo orden.

En el siguiente esquema, se muestran todos los elementos de esta estructura. Los **bloques LBA** (del inglés Logical Block Addressing, o direccionamiento de bloque lógico) establecidos en cada caso corresponden a un sector del disco duro y, por lo tanto, a 512 bytes.

LBA 0	Protective Master Boot Record			
LBA 1	Primary GPT Header			
LBA 2	entry 1	entry 2	entry 3	entry 4
LBA 3 bis 33	entry 5 - 128			
LBA 34	partition 1 partition 2 remaining partitions			
LBA -34	entry 1	entry 2	entry 3	entry 4
LBA -33 bis -2	entry 5 - 128			
LBA -1	Secondary GPT Header			

Las entradas de partición van precedidas por la cabecera GPT y seguidas de la cabecera secundaria.

3.3 Estructura de la cabecera GPT

En la cabecera (en inglés, header) de la tabla de particiones GUID, se describen los **bloques utilizables** del disco duro y el **tamaño de cada entrada de partición**, entre otras cosas. Por lo tanto, es imprescindible para la funcionalidad de las particiones GPT. Como puedes ver en el esquema, la cabecera GPT siempre se guarda en el **segundo sector** del medio de almacenamiento (LBA 1), directamente después del MBR protector. En el disco también se almacena una **copia de seguridad** de la cabecera, concretamente, en el último sector (LBA -1). Las ubicaciones exactas de ambas versiones, protegidas por una **suma de verificación**, también se almacenan en la cabecera.

Nota: Las sumas de verificación CRC32 generadas automáticamente para la cabecera y las entradas de partición son verificadas por el firmware, el gestor de arranque o el sistema operativo.

Si la analizamos con más detalle, la cabecera GPT, cuyo tamaño es de 92 o 512 bytes (incluido el espacio reservado a cero), contiene los siguientes datos:

GUID Partition Table Header / GPT-Header		
Startbyte	Bytes	Contents
0	8	signature („EFI PART“)
8	4	revision number (information about current GPT version)
12	4	header size in Bytes (default: 92)
16	4	CRC32 checksum (header)
20	4	reserved (must be „0“)
24	8	location of the header (LBA 1)
32	8	location of the backup header (LBA -1)
40	8	first usable logical block (LBA) for partitions
48	8	last usable logical block (LBA) for partitions
56	16	unique disk GUID
72	8	starting LBA of partition entries
80	4	number of partition entries (partitions)
84	4	size of a single entry (default: 128 bytes)
88	4	CRC32 checksum of partition entries
92	420+	reserved, must be zeroes (420 bytes for a default sector size of 512 bytes, otherwise more)

El sector principal de la cabecera de la tabla de particiones GUID tiene un tamaño de 92 bytes. El resto del bloque debe estar ocupado por una secuencia de bytes en cero, para que no pueda almacenar otros datos.

3.4 Estructura de la entrada de partición

Después de la cabecera principal, vienen las entradas que describen todas las particiones GPT. Cada **entrada consta de 128 bytes**, de modo que siempre pueden almacenarse **cuatro entradas por bloque lógico** (es decir, por LBA). Para ello, en el esquema estándar de la tabla de particiones GUID, se proponen los bloques 2 a 33, lo que corresponde a **128 particiones**, un modelo que se implementa en los sistemas operativos Windows, por ejemplo. El número de sectores libres para las entradas de partición también puede aumentarse según sea necesario, por lo que, en teoría, el número de particiones posibles es ilimitado, como se explica en el apartado de las características del GPT. La única limitación es el espacio de almacenamiento disponible.

Al margen del número de entradas o particiones, su **estructura**, de acuerdo con la especificación de la GPT o la UEFI, es relativamente sencilla, como puedes ver en el siguiente esquema:

GPT partition entry format		
Startbyte	Bytes	Contents
0	16	partition type GUID (unique ID describing partition type)
16	16	partition GUID (unique ID of the partition)
32	8	first logical block (LBA) of the partition
40	8	last logical block (LBA) of the partition
48	8	attribute flags (e. g. „system partition“, „read-only“ or „hidden“)
56	72	Partition name (36 UTF-16LE code units)
128		

La función de las entradas de partición GPT es definir las particiones de un disco duro GPT.

3.5 ¿En qué casos se utiliza la tabla de particiones GUID?

Desde 2005, el uso de la tabla de particiones GUID como método de partición para discos duros HDD y SSD ha ido en aumento. El principal motivo es que la UEFI no ha dejado de ganarle terreno al BIOS: por ejemplo, el **hardware** y los **sistemas operativos modernos** utilizan el nuevo estándar de interfaz cada vez más y, por lo tanto, también la partición GPT. Además de las ediciones actuales de Windows, como **Windows 10, 8 o 7**, varias de las versiones más recientes de **macOS** y **Linux** figuran entre los sistemas operativos compatibles con GPT.

Las tablas de particiones GUID se utilizan en prácticamente **todos los sistemas informáticos modernos** que incorporan medios de almacenamiento en el rango de los gigabytes o terabytes. Por supuesto, este estilo de partición también suele aplicarse a los **discos duros externos**, especialmente porque muchos de ellos ofrecen más de dos terabytes de memoria en la actualidad y, por lo tanto, no son compatibles con la partición MBR. Otro **ámbito de aplicación típico** de la GPT son las memorias USB de arranque. En este caso, su uso siempre dependerá del firmware y el sistema operativo para el que hayan sido diseñadas y de si debe utilizarse UEFI (es decir, GPT) o BIOS.

Obviamente, la partición GPT también es una buena opción para los **dispositivos USB** que no se utilizan como arranque, sino como un **mero soporte de datos**, sobre todo porque ofrecen una **mayor protección frente a la pérdida de información** en caso de defecto del hardware, un riesgo que suele afectar más a los medios de almacenamiento portátiles (incluidos los discos duros externos).

CAPÍTULO 4

Análisis forense de particiones de discos MBR y GPT

Cuando se arranca un sistema operativo se ejecuta la BIOS (Basic Input/Output System) en primer lugar ya que inicializa los componentes de hardware. El siguiente paso es ejecutar el sistema operativo y por último las aplicaciones. Actualmente se utiliza la tecnología UEFI (Unified Extensible Firmware Interface) como nuevo estándar reemplazando a la BIOS.

Es decir, la BIOS (habitualmente ejecución en modo 16 bits) es un firmware almacenado en una memoria ROM que inicializa el sistema y selecciona el dispositivo de arranque. Con el sistema UEFI llegaron mejoras a nivel de seguridad como la capacidad de ejecutar código en 32 o 64 bits en modo protegido en la CPU. Además, reduce el tiempo de inicio y reanudación y utiliza el modo Secure Boot (solo admite un boot loader firmado).

MBR (Master Boot Record) es el formato de particiones que se venía utilizando en la mayor parte de sistemas operativos, pero con la aparición de UEFI es GPT (GUID Partition Table) el sistema que se impone porque da unas prestaciones mucho mayores que MBR.

GPT en sistema Windows empezó a utilizarse con la versión server 2003 SP1 y ha ido reemplazando a MBR totalmente. Dentro de las limitaciones originales de MBR están que los discos no pueden superar los 2 TB o que solo admite 4 particiones primarias con lo que se debía crear una extendida si se deseaba superar este límite. GPT sin embargo no tiene más limitaciones que las impuestas por el sistema operativo (en Windows admite hasta 128 particiones). En este sistema, a cada partición se le asigna un identificador único (GUID).

Por otro lado, la fiabilidad de GPT no la tiene MBR porque GPT crea múltiples copias redundantes de la información sobre las particiones y eso hace que, si se produce algún error, la tabla de particiones

se pueda recuperar de forma automática desde cualquier copia. En contrapartida, MBR solo guarda la información de la tabla de particiones en el primer sector del disco, si se produce algún problema localizado en él se pierde la información. Es importante destacar que, a nivel de sistema operativo, Windows solo puede arrancar desde discos con tabla de particiones GPT en sus versiones de 64 bits y desde la versión Windows Vista, Cuando

se utiliza un Windows de 32 bits, se puede leer y escribir en discos con GPT, pero no se puede arrancar con ellos. Las versiones actuales de MacOS son compatibles con GPT a pesar de tener su propia APT (Apple Partition Table). Por supuesto, Linux es compatible también.

Existen varias formas de saber si un disco utiliza GPT o MBR: la herramienta de Administrador de Discos de Windows, herramientas de particionado o diskpart (comando que también existe en la consola MS-DOS) son algunas posibilidades

```
> diskpart  
> list disk
```

```
Administrator: Command Prompt - diskpart  
Microsoft Windows [Version 10.0.10586]  
(c) 2015 Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>diskpart  
  
Microsoft DiskPart version 10.0.10586  
  
Copyright (C) 1999-2013 Microsoft Corporation.  
On computer: HP-ENVY  
  
DISKPART> list disk  
  
Disk ###  Status     Size      Free   Dyn  Gpt  
-----  -----  -----  -----  ---  
Disk 0    Online    465 GB    0 B    *  
Disk 1    Online    1863 GB   0 B    *  
Disk 2    Online    3861 MB   0 B
```

Como se puede observar en la imagen anterior se ve una columna GPT. Los discos que utilicen es tabla de particiones tienen un asterisco en ella.

4.1 Forense de GPT

La herramienta «diskpart» provee más comandos para el manejo de un disco GPT bajo Windows. Por ejemplo, «detail disk» informa sobre el GUID del disco, «detail partition» sobre el tipo de partición.

El comando «mmls» del Sleuthkit proporciona una vista de los sectores de inicio y fin de c partición, por ejemplo, mostraría algo del tipo:

```
# sudo mmls /dev/sda
```

```

root@dd-ubuntu1404d-x64:/home/dd
root@dd-ubuntu1404d-x64:/home/dd# fdisk -lu /dev/sda
WARNING: GPT (GUID Partition Table) detected on '/dev/sda'! The util fdisk doesn't support GPT. Use GNU Parted.

Disk /dev/sda: 17.2 GB, 17179869184 bytes
256 heads, 63 sectors/track, 2080 cylinders, total 33554432 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Device Boot      Start        End     Blocks  Id System
/dev/sda1          1   33554431  16777215+ ee GPT
root@dd-ubuntu1404d-x64:/home/dd# mmls /dev/sda
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors
          Slot  Start       End     Length  Description
 00: Meta  0000000000  0000000000  0000000001  Safety Table
 01: ----  0000000000  0000000033  0000000034  Unallocated
 02: Meta  0000000001  0000000001  0000000001  GPT Header
 03: Meta  0000000002  0000000033  0000000032  Partition Table
 04: 00    0000000034  0000001057  0000001024  gptboot0
 05: 01    0000001058  0004195361  0004194304  swap0
 06: 02    0004195362  0033554398  0029359037  zfs0
 07: ----  0033554399  0033554431  0000000033  Unallocated
root@dd-ubuntu1404d-x64:/home/dd#

```

Cuando las herramientas no soportan discos GPT o no aportan información suficiente, se puede hacer un análisis manual de la cabecera GPT y de las entradas de las particiones. Si abrimos un visor hexadecimal el sector 1 (no el 0 que sería en caso de MBR), tenemos la cabecera GPT del tipo

```
# sudo dd if=/dev/sdc count=1 bs=512 skip=1 2> /dev/null | xxd
```

El backup de cabeceras GPT se puede ver direccionando el visor hexadecimal al último sector del disco. El backup de la tabla de particiones se puede ver trabajando hacia atrás desde el último sector del disco.

```

root@server-backup: ~
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: gpt
Identificador del disco: 76ADA50A-49AC-4780-9E34-7E4534BBC065

/dev/sdal      2048      999423     997376   487M Arranque de BIOS
/dev/sda2     999424    16623615   15624192    7,5G Linux swap
/dev/sda3    16623616  7813730303  7797106688  3,6T Sistema de ficheros de Linux
root@server-backup:~# dd if=/dev/sda count=1 bs=512 skip=1 2> /dev/null | xxd
00000000: 4546 4920 5041 5254 0000 0100 5c00 0000  EFI PART....\...
00000010: d009 11cf 0000 0000 0100 0000 0000 0000  .....
00000020: ff10 bcd1 0100 0000 2200 0000 0000 0000  .....
00000030: de10 bcd1 0100 0000 0aa5 ad76 ac49 8047  .....v.I.G
00000040: 9e34 7e45 34bb c065 0200 0000 0000 0000  .4~E4..e....
00000050: 8000 0000 8000 0000 dbfd aec8 0000 0000  .....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  .....


```

4.2 Extracción de particiones GPT

El sistema de ficheros que reside en un disco GPT es totalmente independiente del esquema de particiones. Se pueden extraer GUIDs individuales y analizarlos igual que cualquier otro tipo de partición.

Por ejemplo, se puede clonar una única partición en lugar de un disco:

```
# dc3dd if=/dev/sdc2 of=/particion2.dd
```

Se puede utilizar una herramienta de Sleuthkit para hacer una extracción raw, pero si es un disco GPT se usará «-t gpt» para indicarlo:

```
# mmcatt -t gpt /dev/sdb 3 > /particion.dd
```

En este ejemplo el número 3 no hace referencia a la tercera partición, es el número con que la herramienta mms lo identifica.

A los discos que contienen un HPA (Host Protected Area) o DCO (Device Configuration Overlay) se les puede crear una imagen con un clonado estándar, pero hay que tener en cuenta un dato importante: si se eliminara el HPA o DCO se extiende el área accesible para el usuario y esto cambia la localización del último sector. El sistema GPT asume que el último sector del disco contiene un backup (copia secundaria) de la cabecera de particiones y entradas de particiones por lo que un disco GPT con HPA o DCO eliminados no tendrá el backup esperado al final del disco. Esto haría que algunas herramientas forenses aportaran datos erróneos.

4.3 Artefactos GPT

El esquema GPT proporciona varios identificadores únicos globales, los GUIDs, que identifican de forma unívoca al disco y a cada partición individual. Los EFI GUIDs siguen el formato definido en el RFC 4122. Si el sistema operativo tiene logs, historial u otros artefactos que reflejan los GUIDs de las particiones usadas, este dato proporciona evidencias vinculando un disco duro a un sistema.

Los GUIDs de la versión Y pueden contener información codificada en el mismo string, existen ejemplos donde contienen un timestamp o incluso una MAC address. Según el estándar EFI, el GUID en su primera versión contiene el timestamp que refleja la creación del header GPT y de cada entrada de partición, aunque muchos fabricantes usan UUIDs generados aleatoriamente que contienen evidencias relevantes por protección de privacidad.

Una herramienta útil para decodificar el formato de los GUID y de los UUID es la «uuid tool» Linux. Se muestra un ejemplo de un GUID versión I que contiene un timestamp y una Mac address

```
# sudo uuid -d 1499ccie-2987-de-12 149pple-2937-11de-a42e-
→00id7a479397
```

Otra información que se debe buscar en determinadas periciales es la que se puede ocultar. En un sistema tan flexible como GPT es fácil crear áreas de datos que sean ocultas.

Por ejemplo, espacios entre particiones definidas, poner datos en espacio no particionado hasta el final del disco (siempre sin borrar el Backup GPT), etc. En estos casos, dependiendo de la implementación del estándar GPT por el fabricante, se pueden modificar las cabeceras GPT para crear zonas en el disco que permiten ocultar datos. Por ejemplo, se puede mover el inicio o fin de una zona particionada, se puede incrementar el número o tamaño de una entrada de partición también. Existen áreas designadas como reservadas por las especificaciones GPT que también se pueden utilizar para ocultar información

CAPÍTULO 5

Sistema de ficheros FAT, File Allocation Table

Un sistema de archivos es la estructura subyacente que un equipo usa para organizar los datos de un disco duro. Si está instalando un disco duro nuevo, tiene que realizar las particiones y formatearlo empleando un sistema de archivos para poder comenzar a almacenar datos o programas.

Antes de explicar conceptos de disco y formato se debe aclarar que un cluster es un concepto lógico que agrupa sectores que son el concepto físico. Los sectores se agrupan formando bloques más grandes y cada uno de ellos puede ser usado únicamente por un fichero. El tamaño del cluster viene definido por el tamaño del disco e influye en el rendimiento de la utilización del disco. La estructura es la siguiente, por orden en los sectores: Boot Record, sectores reservados. Copia FAT 1, Copia FAT 2, root directory (en FAT 12 y 16), zona de datos. Las tablas FAT y el directorio raíz deben almacenarse en una ubicación fija para que los archivos de arranque del sistema se puedan localizar correctamente.

En Windows, las tres opciones del sistema de archivos que tiene para elegir son NTFS, FAT32 Y las anteriores FAT (FAT12 y FAT 16). La tarea principal de la File Allocation Table es mantener información actualizada de los clusters (agrupación lógica de los sectores). Existen cuatro entradas posibles en FAT: allocated (contiene la dirección del clúster asociado con datos), unallocated, fin fichero y sectores dañados.

Para proporcionar redundancia (por si los datos se corrompen), se guardan dos copias de la FAT en el sistema de ficheros. La segunda FAT es un duplicado de la primera. El sistema de FAT mirroring se puede deshabilitar en un disco FAT32 lo que provoca que cualquiera de las dos FATs se convierta en primaria, por lo que es posible encontrar la primera copia de la FAT vacía; esto puede desorientar en una pericial

El Directorio raíz, a veces denominado Carpeta raíz, contiene una entrada para cada archivo y directorio almacenados en el sistema de archivos. Esta información incluye el nombre del archivo, el número de cluster inicial y el tamaño del archivo. Esta información se cambia cada vez que se crea un archivo o se modifica posteriormente. El directorio raíz tiene un tamaño fijo de 512 entradas en un disco duro. Con FAT32 se puede almacenar en cualquier lugar dentro de la partición, aunque en versiones anteriores siempre se encuentra inmediatamente después de la región FAT.

El Boot Record, las FATs y el Root Directory se denominan «Área de Sistema». El espacio restante en la unidad lógica se llama «Área de Datos», que es donde se almacenan los archivos. Cabe señalar que cuando el sistema operativo elimina un archivo, los datos almacenados en el Área de datos permanecen intactos hasta que se sobrescribe.

Importante destacar que es en el Boot Record donde está definido el tamaño del clúster. Tamaños de clúster grandes implican pérdida de espacio porque los ficheros no utilizan todo el espacio del mismo. El tamaño del cluster puede variar desde un único sector (512 bytes) a 128 sectores (65536 bytes). Dentro de un cluster los sectores son continuos. Por ello, si tenemos un fichero de 2 KB en un cluster de 32 KB, se pierden 30 KB de espacio (slack space) del que hablaremos más adelante en este mismo capítulo

FAT12 es el tipo más antiguo de este sistema de ficheros, que usa entradas de 12 bits en la tabla. FAT 16 utiliza entradas de 16 bits y posteriormente apareció FAT 32. Esta versión se usa en sistemas Windows 95, Windows 98, Windows Millennium Edition, Windows 2000 y Windows XP. FAT32 no tiene la misma seguridad que NTFS como se explicará, cualquier usuario que tenga acceso al equipo puede leer y escribir en dicha partición. FAT32 también tiene limitaciones de tamaño, ya que no puede almacenar un archivo mayor que 4GB. La razón principal de explicar FAT32 es que es muy probable encontrar equipos en peritajes que tengan instalado Windows XP o anterior aún a día de hoy y probablemente aún se encontrarán durante unos cuantos años más. FAT es con diferencia el sistema de archivos más simple de aquellos compatibles con Windows NT.

Un disco con formato FAT se asigna en clústeres, cuyo tamaño viene determinado por el tamaño del volumen. Cuando se crea un archivo, se crea una entrada en el directorio y se establece el primer número de clúster que contiene datos. Esta entrada de la tabla FAT indica que este es el último clúster del archivo o bien señala al cluster siguiente. La actualización de la tabla FAT es muy importante y requiere mucho tiempo. Si la tabla FAT no se actualiza con regularidad, podría producirse una pérdida de datos. Requiere mucho tiempo porque las cabezas lectoras de disco deben cambiar de posición y ponerse a cero en la pista lógica de la unidad cada vez que se actualiza la tabla FAT. No hay ninguna organización en cuanto a la estructura de directorios de FAT, y se asigna a los archivos la primera ubicación libre de la unidad. Además, FAT solo es compatible con los atributos del sistema, ocultos y de solo lectura.

5.1 Convención de nomenclatura de FAT

FAT utiliza la convención de nomenclatura tradicional, ocho caracteres para el nombre y tres para la extensión (8.3); además, todos los nombres de archivo deben crearse con el conjunto de caracteres ASCII. El nombre debe empezar con una letra o un número y puede contener cualquier carácter excepto los siguientes:

. " / \ [] : ; | = ,

Si se utiliza cualquiera de estos caracteres, pueden producirse resultados inesperados. El nombre no puede contener espacios en blanco. Los nombres siguientes están reservados:

CON, AUX, COM1, COM2, COM3, COMA, LPT1, LPT2, LPT3, PRN, NUL

Todos los caracteres se convertirán en mayúsculas.

Esto se debe tener en cuenta en el momento de realizar búsquedas, ya que dependiendo del software que usemos o el SSOO puede ser sensible a las mayúsculas/minúsculas.

CAPÍTULO 6

Sistema de ficheros NTFS, New Technology File System

NTFS es el sistema de archivos utilizado en las nuevas versiones de Windows. Tiene muchos beneficios respecto al sistema de archivos FAT32, entre los que se incluye, por ejemplo, la capacidad de recuperarse a partir de algunos errores relacionados con el disco automáticamente, lo que FAT32 no puede hacer.

También incluye una compatibilidad mejorada para discos duros más grandes y, sobre todo, una mejor seguridad porque puede utilizar permisos y cifrado para restringir el acceso a archivos específicos para usuarios.

Se utiliza desde Windows NT 3.1, 2000, XP, Microsoft Windows Vista y los más actuales Windows 7, 8 y 10. Se basa en una estructura llamada «tabla maestra de archivos» o MFT, la cual puede contener información detallada de los archivos. Desde el punto de vista del rendimiento, el acceso a los archivos en una partición NTFS es más rápido que en una partición de tipo FAT, esto es porque usa un árbol binario de alto rendimiento para localizar a los archivos.

NTFS ofrece muchas mejoras y ventajas como la capacidad de recuperarse a partir de algunos errores relacionados con el disco y esto lo hace automáticamente, frente a FAT32 que no lo puede hacer. Tiene una compatibilidad mejorada para discos duros más grandes además de ofrecer mejor seguridad porque puede utilizar permisos y cifrado para restringir el acceso a archivos específicos para usuarios aprobados. Además, aparecen dos características fundamentales:

1. **Journaling:** El concepto de journaling se refiere, por ejemplo, a que si se arranca el sistema sin haberlo cerrado correctamente no es necesario hacer un chequeo ya que la recuperación sucede de forma automática a partir de su último estado. NTFS es un sistema seguro ante fallos que puede autocorregirse en casi todas las situaciones,

Por tanto, NTFS journaling proporciona capacidad de recuperación del sistema de archivos, ya que consiste en la grabación de todas las operaciones necesarias para cualquier transacción que altera las estructuras de datos del sistema de archivos importantes. Esto se hace antes de que estas operaciones se produzcan en el disco. El journaling asegura

que, si el sistema se bloquea, las transacciones parcialmente terminadas se pueden rehacer o deshacer cuando el sistema vuelva a estar disponible

2. **Compresión:** Los archivos en un volumen NTFS tienen un atributo denominado «compressed», que permite que cualquier archivo se guarde de forma comprimida con el propósito de ahorrar espacio, esa compresión es transparente para las aplicaciones. La compresión se lleva a cabo por bloques de 16 clústeres y se usa un cluster virtual

Con respecto a la seguridad tiene un descriptor que asegura que ningún proceso puede acceder a un archivo a menos que disponga de los permisos otorgados por el administrador del sistema o por el propietario del archivo. Por lo tanto, antes de que un proceso pueda abrir un handler a cualquier tipo de objeto, el sistema comprueba que tienen la autorización adecuada,

El esquema general de un sistema NTFS estaría formado por:

- **Boot Partition Récord:** En los primeros 8Kb está la información sobre el volumen (tipo de partición, tamaño, etc.), junto con el bloque del código básico para iniciar al sistema operativo.
- **MFT:** La tabla maestra contiene el dónde y el cómo están almacenados los archivos junto con todos los atributos asociados a estos.
- **Archivos del Sistema:** Contiene la información sobre los datos y operaciones que se realizan sobre el sistema de archivos: espacio libre, log de transaccionalidad, etc.

Área de archivos: Donde realmente se almacenan los datos del usuario.

6.1 Clústeres y sectores en una partición NTFS

Los clústeres en un volumen NTFS se numeran secuencialmente desde el principio de la partición. NTFS almacena todos los objetos del sistema de archivos en la tabla maestra de archivos (MFT), similar en estructura a una base de datos.

En los volúmenes NTFS, la información comienza en el sector cero que contiene el bloque de los parámetros de la BIOS (almacena información acerca de la disposición del volumen y las estructuras del sistema de archivos). Continúa la MFT que contiene toda la información de los archivos de la partición NTFS.

6.2 MFT (Master File Table)

Es el archivo que contiene la información de los sistemas con formato NTFS, es decir, la información de todos los ficheros y carpetas que se encuentran en la partición, que continuamente se actualiza mientras el sistema busca, lee y escribe la información. Por tanto, cuando un volumen se formatea en NTFS, se crean un fichero MFT (similar a una tabla) y otros archivos de metadatos. En NTFS, los ficheros de metadatos son los archivos usados para implementar la estructura del sistema de ficheros, NTFS reserva los primeros 16 registros de la MFT para referenciar estos metadata files.

La MFT es una sucesión lineal de registros de tamaño fijo (1 KB). Cada registro de MFT describe un archivo o directorio y contiene los atributos del archivo: su nombre, marcas de tiempo, y la lista de direcciones de disco donde están sus bloques de datos. Si un archivo es muy grande, puede ser necesario usar dos o más registros MFT para poder contener la lista de todos sus bloques. Si esto sucede, el primer registro MFT (registro base), apunta a los demás registros MFT.

Los dieciséis primeros archivos referenciados son los llamados Metafiles, éstos son la única parte del disco que tiene una posición fija y son responsables del funcionamiento del sistema. Son los únicos archivos inmóviles en el sistema NTFS, el resto de archivos ante un problema físico del disco podrían desplazarse o fragmentarse. En la siguiente imagen se puede ver a modo esquemático un resumen:

Esquema representando Sistema de Ficheros NTFS

Cada registro de la MFT consiste en una secuencia de pares, (encabezado atributo, valor). Cada atributo se inicia con un encabezado que indica de qué atributo se trata y qué longitud tiene, porque algunos valores de atributos son de longitud variable, como el nombre archivo y los datos. Si el valor del atributo es lo bastante corto como para que quepa en el registro MFT, se coloca ahí. Si es demasiado largo, se coloca en otro lugar del disco y en el registro MFT se pone un apuntador a él.

6.2.1 6.2.1 Metadatos almacenados de la MFT

Como se desprende de lo explicado, en el sistema de archivo NTFS cada componente es tratado como un archivo, hasta la propia información del sistema. Los metadata files están en el directorio raíz y su nombre empieza con el símbolo \$. Por tanto, cada registro describe un archivo normal que tiene atributos y bloques de datos, igual que cualquier otro archivo

Windows necesita una forma de encontrar el primer bloque del archivo MFT, para encontrar el resto de información del sistema de archivos. Esto se hace examinando el bloque de arranque, donde la dirección del primer bloque del archivo MFT se crea en el momento de dar formato a la partición

El registro 1 es una copia de la primera parte del archivo MFT (\$MFTMirr).

El registro 2 es el archivo de registro o log (\$LogFile). Este es un archivo que contiene todas las acciones llevadas a cabo en la partición. Cuando se efectúan cambios estructurales al sistema de archivos, como añadir un nuevo directorio o eliminar uno que ya existe, la acción se asienta aquí antes de efectuarse para aumentar la posibilidad de una recuperación correcta en caso de que se presente una falla durante la operación. Los cambios a los atributos de archivos también se registran aquí. Los únicos cambios que no se asientan aquí son los efectuados a datos de usuario

El registro 3 contiene información acerca del volumen, como su tamaño, etiqueta y versión

El archivo \$AttrDef es donde se definen los atributos. La información acerca de este archivo está en el registro 4 de la MFT. Luego viene el directorio raíz, que también es un archivo y puede crecer hasta una longitud arbitraria. Se describe en el registro 5 de la MFT.

Se lleva el control del espacio libre en el volumen con un bitmap. Éste también es un archivo y sus atributos y direcciones en disco están en el registro 6 de la MFT. El siguiente registro de la MFT apunta al archivo del cargador de auto arranque. El registro 8 sirve para enlazar todos los bloques defectuosos y así asegurarse de que nunca formen parte de un archivo. El registro 9 contiene información de seguridad. El registro 10 se usa para la conversión de mayúsculas y minúsculas.

Por último, el registro 11 es un directorio que contiene archivos diversos para cosas como cuotas de disco, identificadores de objetos, puntos de re análisis, etc, Los últimos 4 registros de la MFT se han reservado para usos futuros.

Cada entrada de la MFT consiste en un encabezado seguido de una secuencia de pares (atributo, valor). El header del registro contiene un magic number que es un número consecutivo que se actualiza cada vez que el registro se reutiliza para un nuevo archivo, un contador de referencias al archivo, el número actual de bytes en el registro utilizado, el identificador (índice, número de secuencia) del registro base y otros datos

Tabla 1: Fichero de metadatos MTF raíz

Sistema de archivos	Nombre del archivo	Registro MFT	Finalidad del fichero
Tabla maestra de archivos	\$MFT	0	Contiene un registro para cada archivo y carpeta. Su información que se guarda de un archivo o carpeta e demasiado grande para un solo registro, se utilizan más registros
Mirror de la tabla maestra de archivos	\$MFTMirr	1	Garantiza el acceso a la MFT en caso de fallo de un sector Es una imagen duplicada de los primeros cuatro registros de la MFT localizados en la mitad del disco.
Log	\$LogFile	2	Contiene información utilizada por NTFS para hacer recuperación del sistema. El tamaño del archivo depende del tamaño del volumen.
Volumen	\$Volumen	3	Contiene información sobre el volumen, por ejemplo, la etiqueta de volumen y la versión
Definiciones de atributos	\$AttrDef	4	Define los tipos de atributos soportados en el volumen indica si se pueden indexar y recuperar si fuera necesario
Root	\$.	5	La carpeta raíz.
Cluster Bitmap	\$Bitmap	6	Representa el volumen, mostrando clústeres libres y sin usar.
Sector de arranque	\$Boot	7	Incluye el BPB usado para montar el volumen y el código del gestor de arranque adicional que se utiliza cuando el volumen es de arranque,
Bad Cluster	\$BadClus	8	Contiene sectores defectuosos, dañados... de un volumen
Archivo de seguridad	\$Secure	9	Contiene los descriptores de seguridad únicos para todos los archivos de un volumen.

continué en la próxima página

Tabla 1 – proviene de la página anterior

Sistema de archivos	Nombre del archivo	Registro MFT	Finalidad del fichero
Tabla Uppercase	\$Upcase	10	Tabla de concordancia de mayúsculas y minúsculas en los nombres de archivo en el volumen actual. Muy necesario porque en NTFS los nombres se almacenan en Unicode.
Extensión de archivo NTFS	\$Extend	11	Se utiliza para distintas finalidades opcionales, por ejemplo: cuotas o identificadores de objetos.
		12-15	Reservado para uso futuro.

Destacar, como se muestra en la imagen de la figura 5, que la segunda copia de los primeros registros, por su importancia, se almacena exactamente en la mitad del disco. La parte restante de la MFT puede almacenar como cualquier otro archivo en cualquier lugar del disco.

Siempre se puede establecer su posición utilizando como base el primer elemento de la MFT.

6.3 Resumen de la estructura

En NTFS se pueden tener longitudes de clústeres desde 512 bytes hasta 64 KBytes. El cluster de 4

KBytes se considera estándar siempre que la partición sea mayor de 2 GBytes. Los discos NTFS se dividen, simbólicamente, en dos partes:

1. El primer 12 % del disco se asigna al área de la MFT, es el espacio donde crece este archivo. No se pueden almacenar datos en esta área,
2. El resto es el área de datos. No obstante, cualquier informe del SO acerca del espacio libre en el disco incluye el área MFT. El mecanismo del área MFT es el siguiente: cuando no haya espacio para almacenar más archivos se toma espacio del área MFT y se reduce su longitud una vez que vuelve a existir espacio el área MFT vuelve a crecer

CAPÍTULO 7

Borrado de ficheros en Windows

Al eliminar un fichero en Windows y si es en un disco mecánico, ya que los SSD tienen el garbage collector, solo se eliminan las referencias al fichero, pero no el contenido del mismo. En la MFT el espacio que ocupaba el fichero se marca como libre y esto hace que cuando se sigan guardando datos, este espacio podrá ser utilizado

Cuando un fichero ocupa menos que el tamaño de un cluster (unidad más pequeña de almacenamiento de un disco, la cual está formada por varios sectores), el espacio no ocupado se denomina slack space y contiene información de ficheros que han sido previamente eliminados. Se muestra a continuación un ejemplo gráficamente:



El Fichero 1 ocupa 2 sectores.



Se eliminan las referencias al fichero y se declara el espacio como disponible.



Se guarda el Fichero 2 en el espacio disponible.

No se debe menospreciar el poder del slack space para arrojar luz en los análisis forenses. Existen herramientas que permiten ocultar información en este espacio.

Tampoco se debe confundir lo que es file slack con el unallocated space. Cuando hablamos

del unallocated se está haciendo referencia al espacio libre en el disco que no está asignado a ninguna partición, volumen o unidad local.

En un análisis forense, después de completar la revisión de la estructura de archivos, nos concentraremos en analizar el espacio no asignado y el slack de archivos. El uso de una herramienta de software para facilitar el proceso es la manera más fácil de lograr esta parte del análisis.

Incluso con la ayuda de herramientas de software, este proceso puede ser muy largo. Además, los resultados de la extracción de archivos borrados pueden ser voluminosos.

Otro punto importante, todos los archivos identificados deben ser revisados. No podemos revisar Simplemente hasta que encontramos algo del material que estamos buscando, o material que ayude a nuestro caso, y detener la búsqueda. Eso sería una evaluación injusta e incompleta de la potencial evidencia.

Por lo tanto, para acelerar el proceso de revisión de archivos extraídos de espacio no asignado, se puede utilizar una utilidad de software llamada «dtSearch», por ejemplo. Con todos los archivos extraídos en una ubicación, se crean términos de búsqueda en dtSearch o con la PowerShell o cualquier Shell si nos encontramos en un sistema Unix, el programa busca a través de los archivos para encontrar lo que encaja con esos términos de búsqueda solicitados.

Al igual que en la revisión de la estructura de archivos lógicos, cuando se encuentran pruebas potenciales, se debe registrar su dirección en el disco duro. Sin embargo, debido a que el unallocated space y el slack space del archivo están fuera del esquema de direccionamiento lógico en esta revisión, se registrar la dirección física de cualquier evidencia, incluyendo esencialmente su cluster y dirección de sector (por ejemplo, cluster 11155, sector 357517)

7.1 Slack Space en detalle

Para entender bien este concepto y lo reflejado en las figura anterior se puede hacer un ejercicio práctico, crear un nuevo archivo de texto. Se edita usando el Bloc de notas y se puede escribir una simple palabra en él, se guarda y cierra. Al hacer clic derecho en el archivo y para comprobar sus propiedades se puede ver que los dos atributos «Tamaño» y «Tamaño disco son distintos. El tamaño siempre será menor o igual al tamaño en disco, la diferencia es el slack space que puede ser de tres tipos: File Slack, RAM Slack y Drive Slack.

7.1.1 File Slack

Los tamaños de archivo varían por lo que, para almacenarlos, el sistema de archivos utiliza contenedores de tamaño fijo o bloques llamados Cluster. Los clusters no son más que grupos de sectores que se utilizan para asignar el espacio de almacenamiento en disco en los sistemas operativos. Por lo tanto, a cualquier archivo nuevo se le asigna una serie de clústeres tales que:

```
tamaño de archivo <= núm. de clústeres * tamaño de un solo  
→cluster
```

Obviamente, los tamaños de archivo rara vez coinciden perfectamente con el tamaño de uno o varios clústeres. Como resultado de esto, queda un espacio entre el final del contenido del archivo y el final del último clúster asignado a él. Este espacio se llama File Slack. File Slack se crea en el momento en que se guarda un archivo en el disco. El File Slack se puede dividir en RAM Slack y Drive Slack. Definamos estos dos términos en detalle.

7.1.2 RAM Slack

Los sistemas basados en Microsoft Windows normalmente escriben en bloques de 512 bytes llamados Sectores. Esto significa que cuando el sistema operativo desea escribir en el sistema de archivos, escribiría en espacios de 512 bytes con un mínimo de 512 bytes. Por lo tanto, si no hay suficientes datos para llenar el último sector en el último grupo, el sistema operativo escribe datos aleatorios de memoria (RAM) a la zona sin llenar en el último sector. Es posible que esa área de la memoria pudiese contener algo sensible, como la contraseña para un disco cifrado o partición, que se habría montado en algún momento en el pasado. Esta área que se llena con los datos aleatorios de la memoria RAM se llama RAM Slack.

El RAM Slack sólo se aplica al último sector de un archivo, pero los sectores restantes que forman parte del último clúster asignado al archivo tampoco se rellenan con ningún archivo de datos este caso no se escribe nada a los sectores restantes del clúster. En definitiva, lo que fue almacenado en esa zona del disco permanece allí y podría contener restos de archivos previamente eliminados o incluso los datos que existían antes del último formateo. También se puede analizar el File Slack para identificar los usos previos del equipo en cuestión y puede contener fragmentos de mensajes de correo electrónico, documentos, etcétera.

7.2 Análisis de fichero y carving

Existen muchas herramientas para recuperar ficheros borrados, por ejemplo: Hfind, Recuvafree Exiftool, FocaFree... pero no todas hacen carving. Este proceso de carving es utilizado principalmente en la informática forense para extraer información a partir de una cantidad de datos en bruto sin necesidad de conocer el sistema de ficheros con el que se han creado los archivos.

En general, todos los tipos de ficheros tienen características comunes. Por ejemplo, atendiendo a su estructura, algunos tipos de ficheros JPG/JFIF empiezan por FF D8 FF E1 como se puede ver en el siguiente ejemplo:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	yoyá	Exif	II*
0	FF	D8	FF	E1	00	18	45	78	69	66	00	00	49	49	2A	00	ÿþýá	Exif	II*
16	08	00	00	00	00	00	00	00	00	00	00	00	FF	EC	00	11	ÿí		
32	44	75	63	6B	79	00	01	00	04	00	00	00	4D	00	00	FF	Ducky	M	ÿ
48	E1	03	81	68	74	74	70	3A	2F	2F	6E	73	2E	61	64	6F	á	http://ns.ado	
64	62	65	2E	63	6F	6D	2F	78	61	70	2F	31	2E	30	2F	00	be.ccm/xap/1.0/		
80	3C	3F	78	70	61	63	6B	65	74	20	62	65	67	69	6E	3D	<?xpacket begin=		
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Y terminan por FF D9; como se puede ver en el final de la imagen, referida al mismo fichero JPG:

A hex dump of file content. The bytes are grouped into pairs. A red box highlights the sequence 03 FF D9, which is the magic number for JPEG files. To the right of the dump, there are four columns of text: P, E, P, E; Š(, Š(, ¢; and Š(, Š(, ¢.

14	50	01	45	14	50	01	45	14	50	01	45	14	50	30	A2	P	E	P	E
8A	28	00	A2	Š(¢	Š(¢												
8A	28	00	A2	8A	28	00	A2	8A	28	03	FF	D9			Š(¢	Š(¢	

Es importante tener en cuenta que dentro de la misma familia de codecs de compresión de imagen, aun siendo ficheros con extensión jpg, varía algún dato, por ejemplo, la cabecera de un fichero JPG de formato «Still Picture Interchange File Format» (SPIFF) es FF D8 FF E8

Otro ejemplo, la cabecera de un fichero JPEG de formato «SAMSUNG D500» es FF D8 FF E3 o el de un JPEG «CANNON EOS» es FF D8 FF E2. Sabiendo esto, en bloques de datos, se pueden localizar aquellos que correspondan a ficheros «jpg» en base al comienzo y fin de su estructura.

Esta técnica es realmente útil en los casos en los que los dispositivos de almacenamiento se hayan corrompido o dañado

Todos los tipos de ficheros tienen una estructura similar. Utilizan una constante conocida como Magic Number, la cual permite identificar el correspondiente tipo de fichero. En la siguiente tabla se indican una serie de magic numbers representativos con el tipo de fichero asociado a modo de muestra:

Description	Extension	Magic Number
Adobe Illustrator	.ai	25 50 44 46 [%PDF]
Bitmap graphic	.bmp	42 4D [BM]
Class File	.class	CA FE BA BE
JPEG graphic file	.jpg	FFD8
JPEG 2000 graphic file	.jp2	0000000C6A5020200D0A [....jP..]
GIF graphic file	.gif	47 49 46 38 [GIF89]
TIF graphic file	.tif	49 49 [II]
PNG graphic file	.png	89 50 4E 47 .PNG
WAV audio file	.png	52 49 46 46 RIFF
ELF Linux EXE	.png	7F 45 4C 46 .ELF
Photoshop Graphics	.psd	38 42 50 53 [8BPS]
Windows Meta File	.wmf	D7 CD C6 9A
MIDI file	.mid	4D 54 68 64 [MThd]
Icon file	.ico	00 00 01 00
MP3 file with ID3 identity tag	.mp3	49 44 33 [ID3]
AVI video file	.avi	52 49 46 46 [RIFF]
Flash Shockwave	.swf	46 57 53 [FWS]
Flash Video	.flv	46 4C 56 [FLV]
Mpeg 4 video file	.mp4	00 00 00 18 66 74 79 70 6D 70 34 32 [....ftypmp42]
MOV video file	.mov	6D 6F 6F 76 [....moov]
Windows Video file	.wmv	30 26 B2 75 8E 66 CF
Windows Audio file	.wma	30 26 B2 75 8E 66 CF
PKZip	.zip	50 4B 03 04 [PK]
GZip	.gz	1F 8B 08
Tar file	.tar	75 73 74 61 72
Microsoft Installer	.msi	D0 CF 11 E0 A1 B1 1A E1
Object Code File	.obj	4C 01
Dynamic Library	.dll	4D 5A [MZ]
CAB Installer file	.cab	4D 53 43 46 [MSCF]
Executable file	.exe	4D 5A [MZ]
RAR file	.rar	52 61 72 21 1A 07 00 [Rar!...]
SYS file	.sys	4D 5A [MZ]
Help file	.hlp	3F 5F 03 00 [?_..]
VMWare Disk file	.vmdk	4B 44 4D 56 [KDMV]
Outlook Post Office file	.pst	21 42 44 4E 42 [!BDNB]
PDF Document	.pdf	25 50 44 46 [%PDF]

Existen diferentes técnicas de «file carving»:

- **Basadas en la cabecera de un fichero y en el final** o, si se desconoce éste, en el tamaño máximo de archivo (dato disponible en la cabecera).
- **Basadas en la estructura de un fichero:** cabecera, pie, cadenas significativas, tamaño, etc
- **Basadas en el contenido del fichero:** entropía, reconocimiento del lenguaje, atrí-

butos estáticos, etc. Ejemplo: HTML, XML, etcétera. También podemos encontrar muchas herramientas, tanto open source como propietarias que permiten realizar el proceso de «file carving» con mayor o menor efectividad y utilidad. A continuación, se describen algunas de ellas:

- **Foremost:** (<http://foremost.sourceforge.net>) es una herramienta de código abierto desarrollada por la Oficina de Investigaciones Especiales de las Fuerzas Aéreas estadounidenses. Permite trabajar tanto con imágenes de dispositivos (dd, encase, etc.) como directamente sobre el dispositivo. Está orientada a la recuperación de información en entornos Linux. Presenta la limitación de que únicamente es capaz de procesar ficheros de hasta 2 GB.
- **Scalpel:** (<https://github.com/sleuthkit/scalpel>) es una herramienta de código abierto, basada en Foremost, aunque mucho más eficiente, incluida en The Sleuth Kit. Está orientada a la recuperación de información tanto en entornos Linux como OSX o Windows.
- **Forensic Toolkit (FTK):** (<http://accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk?/solutions/digital-forensics/ftk>) es una suite muy completa para realizar análisis forenses. Una de las muchas funcionalidades que incorpora es la de carvings avanzado que le permite especificar criterios de búsqueda como tamaño de archivo, tipo de datos y tamaño de píxel, para reducir la cantidad de datos irrelevantes extraídos.
- **X-Way Forensics (WinHex):** (<http://www.x-ways.net/forensics/>) al igual que FTK es una suite completa para realizar análisis forenses. Sus prestaciones de “carving” no son demasiado configurables, pero sí potentes.

7.3 Orphan Files

Un orphan file es el que ya no tiene información de la carpeta en la que estaba, es decir, si se elimina una carpeta y se sobre escribe la entrada de esa carpeta en la SMFT, los archivos que contenía siguen existiendo, aunque borrados, son ficheros huérfanos.

Cuando borramos una carpeta y sus ficheros, si sus entradas quedan intactas en la SMFT, tanto el padre (la carpeta), como los hijos (los ficheros que contenía), se pueden recuperar de forma sencilla

CAPÍTULO 8

Conclusiones

El sistema de ficheros a estudiar en una pericial determina las herramientas a utilizar y la forma de proceder. En alguna ocasión he tenido periciales de discos duros antiguos, cuyo sistema operativo era un Windows 95 y debía sacar todos los metadatos posibles de sus documentos de Microsoft Word. Bien, trabajar con FAT16 ya es poco habitual pero además tratar con Microsoft Office 9 es curioso. En este caso pocos datos pude extraer más que nombre del autor, fechas de creación y modificación o tiempo de edición, pero suficientes para lo que se pedía demostrar. Por tanto, no subestimemos el estudio de una FAT, aún existen más ordenadores de los que nos imaginamos ca ese sistema y nunca se sabe cuándo nos puede tocar peritar uno.

Por otra parte, el unallocated space también puede proporcionar muchos datos. En algunas periciales he podido recurrir a este espacio y ha proporcionado textos muy esclarecedores de planes e intenciones en casos de competencia desleal. Es por esto que muchas veces, aunque se haga carving estudiar el espacio unallocated si no se ha encontrado nada relevante por otros medios puede ser muy útil.