📝 Guia Rápido - Informática Forense

🔍 1. Conceitos Fundamentais

- **Informática Forense** → Processo de recolha, preservação, análise e apresentação de evidências digitais.
- **Evidência Digital** → Qualquer dado armazenado ou transmitido digitalmente que pode ser usado como prova.
- **Cadeia de Custódia** → Processo que garante a integridade das evidências digitais.

🔍 Tipos de Análise Forense

✅ **Análise Live** → Sistema ligado, análise de RAM, processos e ligações ativas.
✅ **Análise A Frio** → Dispositivo desligado, análise de discos e ficheiros apagados.

🔍 Processo de Análise Forense Digital

1️⃣ **Identificação** → Determinar as fontes de evidência.
2️⃣ **Preservação** → Criar cópias forenses sem alterar dados.
3️⃣ **Análise** → Examinar ficheiros, logs, registos e atividades.
4️⃣ **Relatório** → Documentar descobertas e apresentar prova.

🛠️ Ferramentas Utilizadas

- **Autopsy** → Investigação de discos e ficheiros apagados.
- **Wireshark** → Análise de tráfego de rede.
- **Volatility** → Análise de memória RAM.
- **FTK Imager** → Aquisição e visualização de imagens forenses.
- **Kali Linux** → Conjunto de ferramentas forenses e de segurança.

🚫 Técnicas Anti-Forenses

- **Estenografia** → Esconder ficheiros dentro de outros ficheiros.
- **Encriptação** → Tornar dados ilegíveis sem chave de acesso.
- **Manipulação de timestamps** → Alteração das datas de criação e modificação de ficheiros.
- **Eliminação segura de dados** → Ferramentas que sobrescrevem ficheiros apagados.

🔑 Funções de Hash e Integridade de Dados

- **MD5 e SHA-1** → Já não são considerados seguros devido a colisões.
- **SHA-256 e SHA-512** → Algoritmos recomendados para garantir integridade.

📂 Organização dos Discos

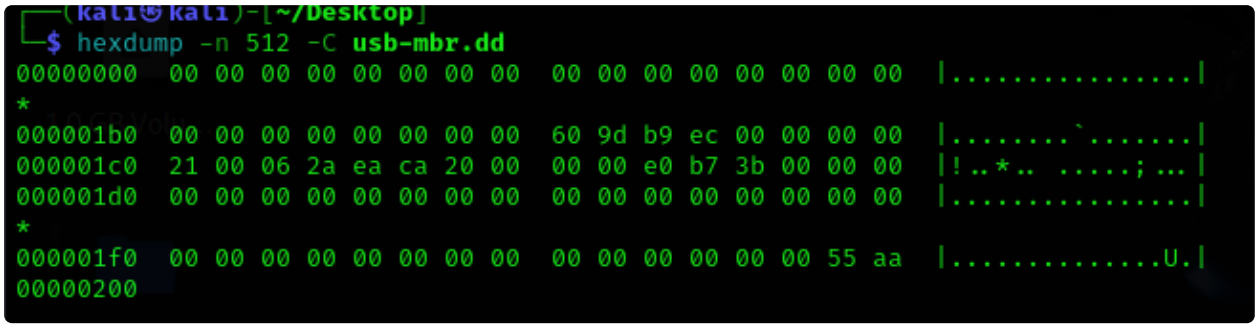
- **Master Boot Record (MBR)** → Ocupa os primeiros 512 bytes do disco e contém a tabela de partições.
- **GUID Partition Table (GPT)** → Método mais moderno que suporta discos maiores e mais partições.
- **Partições** → Divisões lógicas do disco onde diferentes sistemas de ficheiros são aplicados.

📌 Master Boot Record (MBR)

🔹 *Estrutura Genérica*

Endereço		Descrição		Tamanho (bytes)
Hex	Dec			
+0x0000	+0	Código de arranque (*boot*)		446
+0x01BE	+446	Partição #1	Tabela de partições primárias	16
+0x01CE	+462	Partição #2		16
+0x01DE	+478	Partição #3		16
+0x01EE	+494	Partição #4		16
+0x01FE	+510	55	Assinatura	2
+0x01FF	+511	AA		
Tamanho total : 446 + 4×16 + 2				**512**

🔍 Análise de um Disco



📌 O que os dados mostram?

1️⃣ O **MBR tem 512 bytes**, seguindo a estrutura clássica:

- **0x0000 - 0x01BD (446 bytes)** → Código de arranque (**bootloader**).
- **0x01BE - 0x01FD (64 bytes)** → **Tabela de Partições Primárias** (4 entradas de 16 bytes).
- **0x01FE - 0x01FF (2 bytes)** → Assinatura `0x55AA` (**indica que o MBR é válido**).

2️⃣ **Analisando o hexdump:**

- O final do ficheiro (offset `0x01FE`) contém `55 AA`, confirmando um **MBR válido**.
- As entradas na **Tabela de Partições** (offset `0x01BE` até `0x01FD`) contêm **informações sobre partições**.
- Algumas partições podem estar vazias ou não inicializadas (**cheias de** `00`).

📌 **Como podes continuar a análise?**

1️⃣ **Decodificar as Entradas da Tabela de Partições**

Cada partição tem **16 bytes**, contendo:

- **Status** (*ativo* `0x80` ou *inativo* `0x00`).
- **CHS (Cylinder-Head-Sector)** – Informação do disco.
- **Tipo de partição** (ex: **NTFS**, **FAT32**, **EXT4**).
- **Endereço LBA inicial** e **tamanho da partição**.

📌 **Estrutura do Campo de Partição**

Posição relativa	Descrição	Tamanho (bytes)
0	Indicador de *boot* (80h)	1
1	Inicio de partição (CHS)	3
4	Tipo de partição	1
5	Fim de partição (CHS)	3
8	Setor inicial (LBA)	4
12	Tamanho da partição (em setores)	4
	Tamanho total:	**16**

Tipos de Numeração de Setores no Disco

📌 **CHS** (*Cylinder/Head/Sector*) → Método mais antigo.
📌 **LBA** (*Logical Block Addressing*) → Método mais recente.

📌 **Exemplo de Entrada de Partição**

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	**80**	01
1C0	01	00	**0B**	1F	3F	33	3F	00	00	00	41	99	01	00	00	00
1D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA

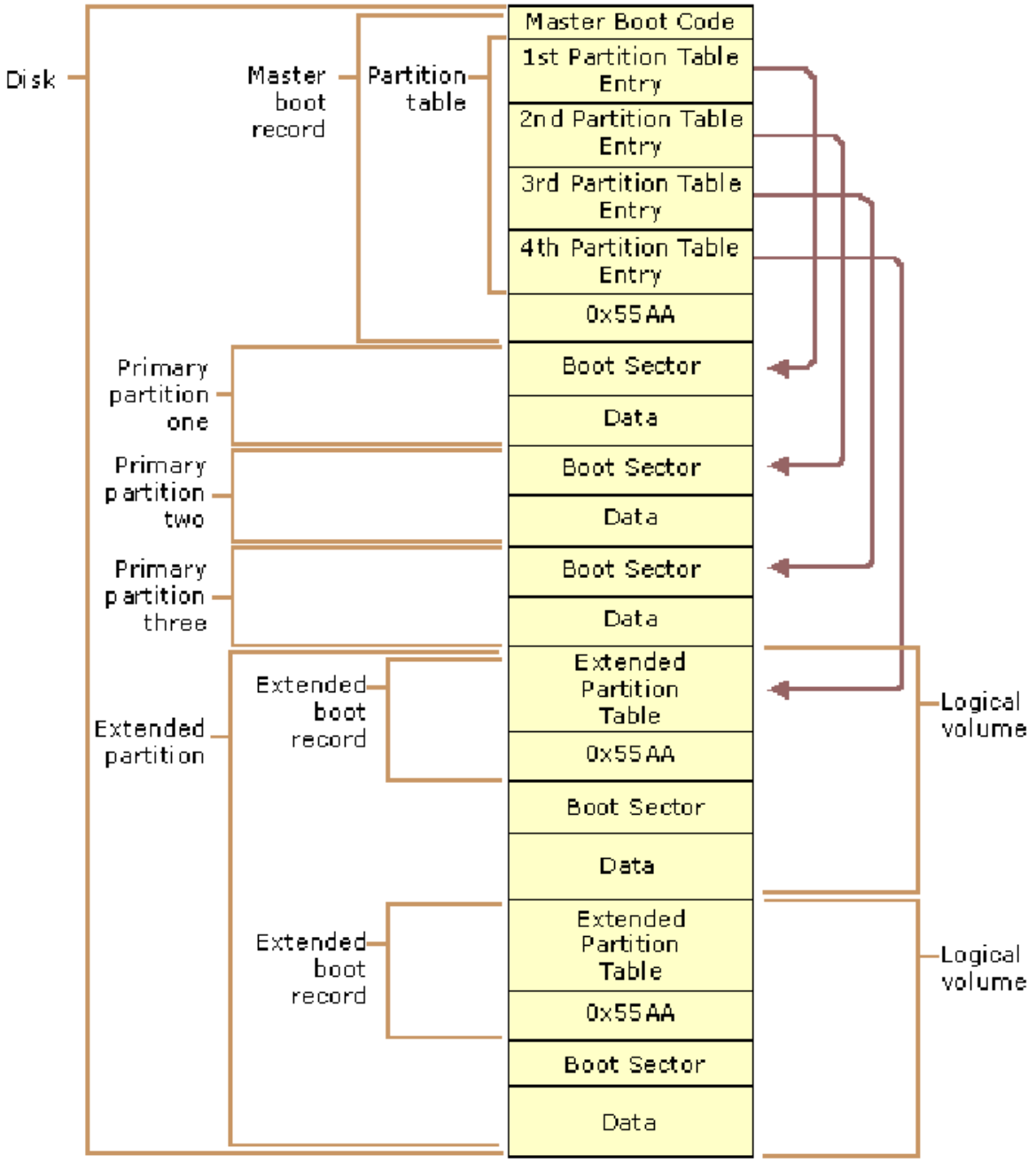
Detalhes do Exemplo

- **0x1BE** → Partição está ativa (**valor** `80h`).
- **0x1BF** → Setor inicial da partição **CHS(0,1,1)**.
- **0x1C2** → Tipo de partição (**0B** → **FAT32**).

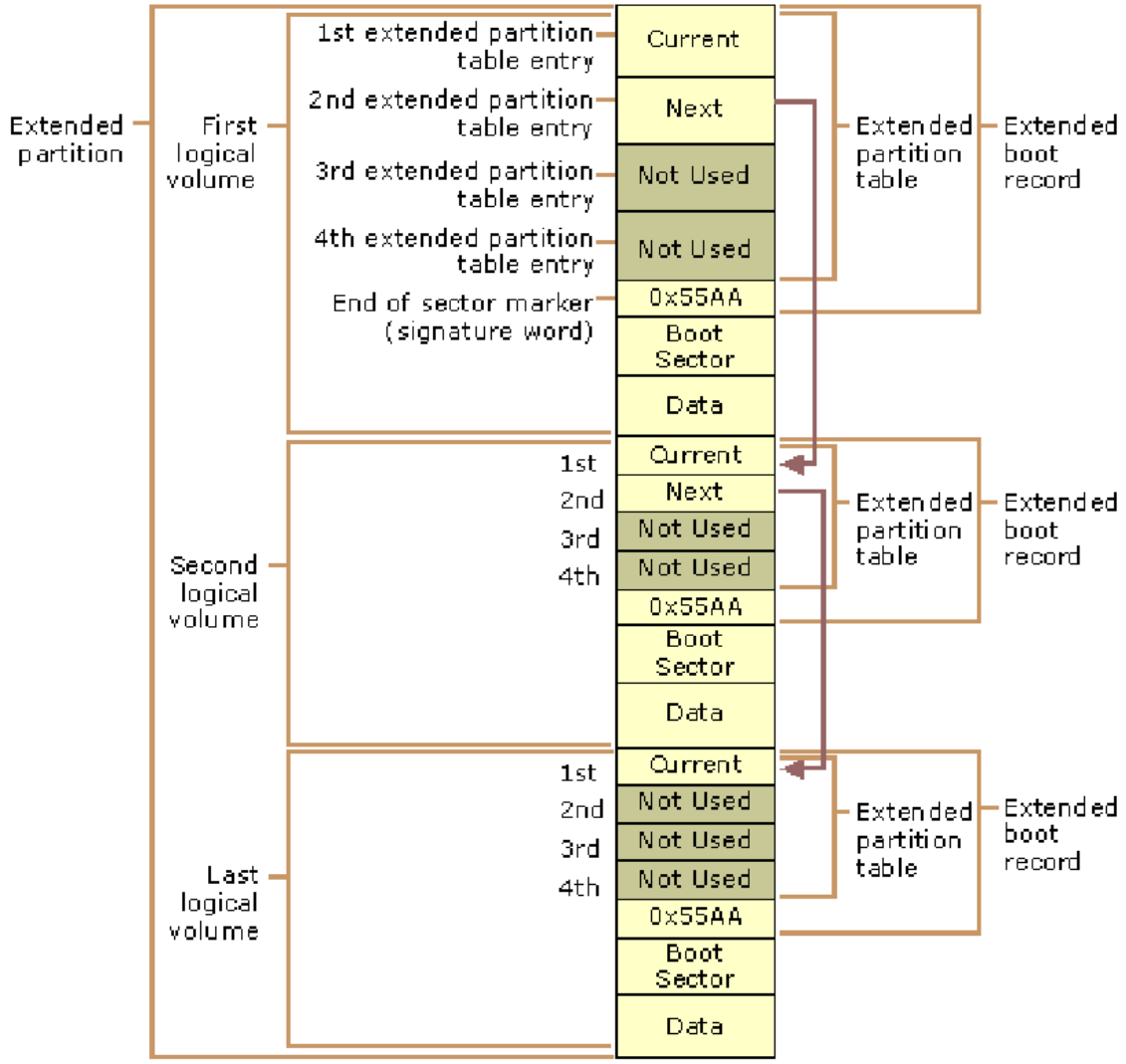
📌 Estrutura Genérica do Disco

Posição relativa ao início do EBR		Descrição	Tamanho (bytes)
Hex	Dec		
000 – 1BD	000 – 445	Tipicamente vazia (zeros)	**446**
1BE – 1CD	446 – 461	**Partição #1** – Descreve a partição atual	**16**
1CE – 1DD	462 – 477	**Partição #2** – Descreve a próxima partição	**16**
1DE – 1ED	478 – 493	**Partição #3** – Não utilizada (zeros)	**16**
1EE – 1FD	494 – 509	**Partição #4** – Não utilizada (zeros)	**16**
1FE - 1FF	510 - 511	Assinatura	**2**
		Tamanho total: 446 +4×16 +2 =	**512**

📌 Layout de um Disco com MBR



📌 Layout de um EBR (Extended Boot Record)



📌 GUID Partition Table (GPT)

🔹 **GPT (GUID Partition Table)** é o esquema de partições da norma **UEFI (Unified Extensible Firmware Interface)**.

🔹 **Principais Características:**
✅ Usa **endereçamento por LBA** (blocos de **512 bytes**).
✅ **Suporta partições maiores** que o MBR.
✅ Pode ser usada em sistemas com **BIOS**, desde que suportado pelo SO (ex: **Linux**).
✅ **LBA-0 (512 bytes iniciais)** pode ser:

- **Ignorado** (não usado pela GPT).
- **Protective MBR** (para compatibilidade com sistemas antigos).

🔍 **Vantagens da GPT sobre MBR:**

- ✔ Suporta **mais de 4 partições primárias**.
- ✔ Sem limite de **2 TB** por partição.
- ✔ Melhor tolerância a falhas (cópia redundante da tabela de partições).

```
aap@pc:~$ hexdump -n 1024 -C disk-image.dd
00000000 33 c0 8e d0 bc 00 7c 8e c0 8e d8 be 00 7c bf 00 |3.....|......|..|
00000010 06 b9 00 02 fc f3 a4 50 68 1c 06 cb fb b9 04 00 |.......Ph.......|
00000020 bd be 07 80 7e 00 00 7c 0b 0f 85 0e 01 83 c5 10 |....~..|........|
00000030 e2 f1 cd 18 88 56 00 55 c6 46 11 05 c6 46 10 00 |.....V.U.F...F..|
00000040 b4 41 bb aa 55 cd 13 5d 72 0f 81 fb 55 aa 75 09 |.A..U..]r...U.u.|
00000050 f7 c1 01 00 74 03 fe 46 10 66 60 80 7e 10 00 74 |....t..F.f`.~..t|
00000060 26 66 68 00 00 00 00 66 ff 76 08 68 00 00 68 00 |&fh....f.v.h..h.|
00000070 7c 68 01 00 68 10 00 b4 42 8a 56 00 8b f4 cd 13 ||h..h...B.V.....|
00000080 9f 83 c4 10 9e eb 14 b8 01 02 bb 00 7c 8a 56 00 |............|.V.|
                    (LINHAS REMOVIDAS POR FALTA DE ESPACO)
00000150 10 eb f2 f4 eb fd 2b c9 e4 64 eb 00 24 02 e0 f8 |......+..d..$...|
00000160 24 02 c3 49 6e 76 61 6c 69 64 20 70 61 72 74 69 |$..Invalid parti|
00000170 74 69 6f 6e 20 74 61 62 6c 65 00 45 72 72 6f 72 |tion table.Error|
00000180 20 6c 6f 61 64 69 6e 67 20 6f 70 65 72 61 74 69 | loading operati|
00000190 6e 67 20 73 79 73 74 65 6d 00 4d 69 73 73 69 6e |ng system.Missin|
000001a0 67 20 6f 70 65 72 61 74 69 6e 67 20 73 79 73 74 |g operating syst|
000001b0 65 6d 00 00 00 63 7b 9a -- 1. PA RT -- -> 00 fe |em...c{.}n......|
000001c0 ff ff ee fe ff ff 01 00 00 00 a3 70 3d 3a <- -- |...........p=:..|
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
*
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa |..............U.|
00000200 45 46 49 20 50 41 52 54 00 00 01 00 5c 00 00 00 |EFI PART....\...|
00000210 7f 95 63 62 00 00 00 00 01 00 00 00 00 00 00 00 |..cb............|
00000220 a3 70 3d 3a 00 00 00 00 22 00 00 00 00 00 00 00 |.p=:............|
00000230 82 70 3d 3a 00 00 00 00 a8 4d 00 00 93 38 00 00 |.p=:.....M...8..|
00000240 3d 11 00 00 f3 62 00 00 02 00 00 00 00 00 00 00 |=....b..........|
00000250 80 00 00 00 80 00 00 00 81 d5 7a 4e 00 00 00 00 |..........zN....|
00000260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
*
00000400
```

📌 **Estrutura Genérica**

Posição relativa	Descrição	Tamanho (bytes)
LBA0	Não utilizado (MBR)	512
LBA1	Cabeçalho GPT principal	512
LBA2	Entradas das partições 1 a 4 (128 bytes cada)	512
LBA3	Entradas das partições 5 a 128 (128 bytes cada)	512
. . .		...
LBA33		512
. . .	Dados / Partições	...
LBA-33	Cópia das Entradas das partições 5 a 128 (128 bytes cada)	512
. . .		...
LBA-3		512
LBA-2	Cópia das entradas das partições 1 a 4 (128 bytes cada)	512
LBA-1	Cópia do cabeçalho GPT	512

LBA-1 refere-se ao último LBA do disco.

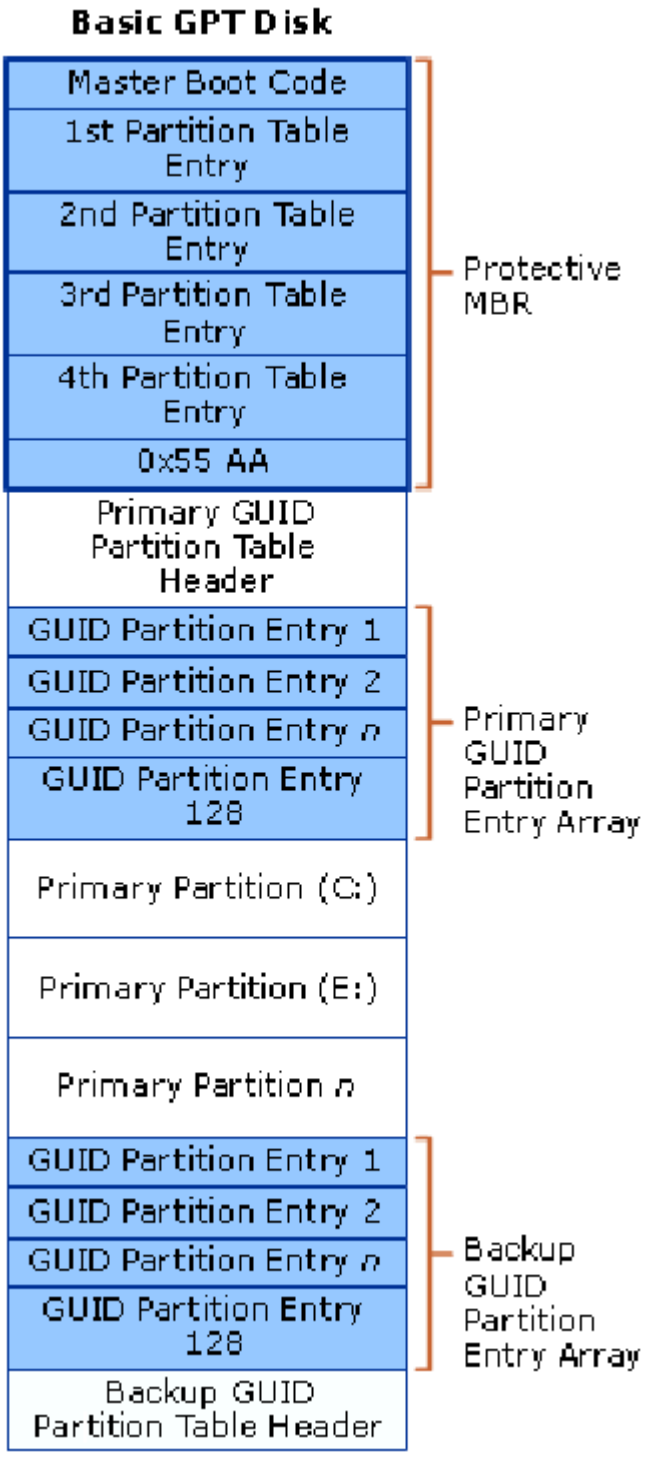
📌 **Estrutura do cabeçalho GPT**

Posição relativa	Tamanho	Contents
0 (0x00)	8 bytes	Assinatura (45h 46h 49h 20h 50h 41h 52h 54h)
8 (0x08)	4 bytes	Revisão (00h 00h 01h 00h para GPT ver 1.0)
12 (0x0C)	4 bytes	Tamanho (5Ch 00h 00h 00h = 92 bytes)
16 (0x10)	4 bytes	CRC32 do cabeçalho
20 (0x14)	4 bytes	Reservado (zero)
24 (0x18)	8 bytes	LBA do cabeçalho
32 (0x20)	8 bytes	LBA da cópia do cabeçalho
40 (0x28)	8 bytes	Primeiro LBA útil
48 (0x30)	8 bytes	Último LBA útil
56 (0x38)	16 bytes	GUID do disco
72 (0x48)	8 bytes	LBA da lista de partições
80 (0x50)	4 bytes	Número de partições na lista
84 (0x54)	4 bytes	Tamanho da cada entrada de partição (128 bytes)
88 (0x58)	4 bytes	CRC32 da lista de partições
92 (0x5C)	*	Reservado, zeros até ao fim

📌 **Estrutura de entrada de partição em GPT**

Posição relativa	Descrição	Tamanho (bytes)
0	GUID do tipo de partição	16
16	GUID da partição (único)	16
32	LBA inicial	8
40	LBA final	8
48	Atributos	8
56	Nome da partição (UTF-16LE)	72
	Tamanho total:	**128**

📌 Layout de um disco com GPT

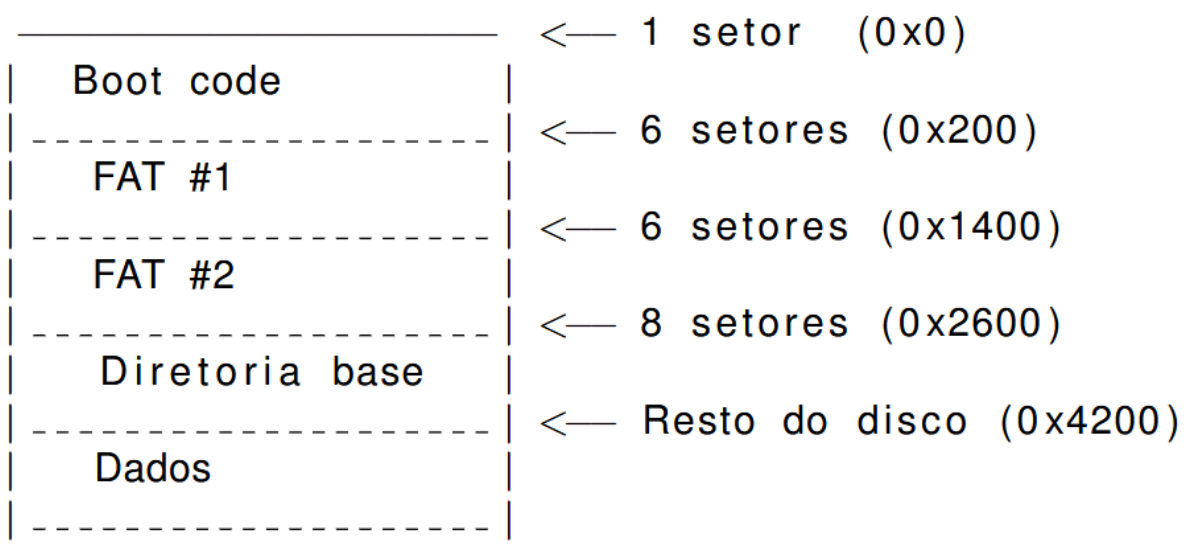


📌 Sistemas de Ficheiros

💻 FAT (File Allocation Table)

- Sistema de ficheiros **simples e popular**.
- Utilizado em **DOS, Windows, USBs, cartões de memória e smartphones**.

📌 Layout de uma Partição FAT

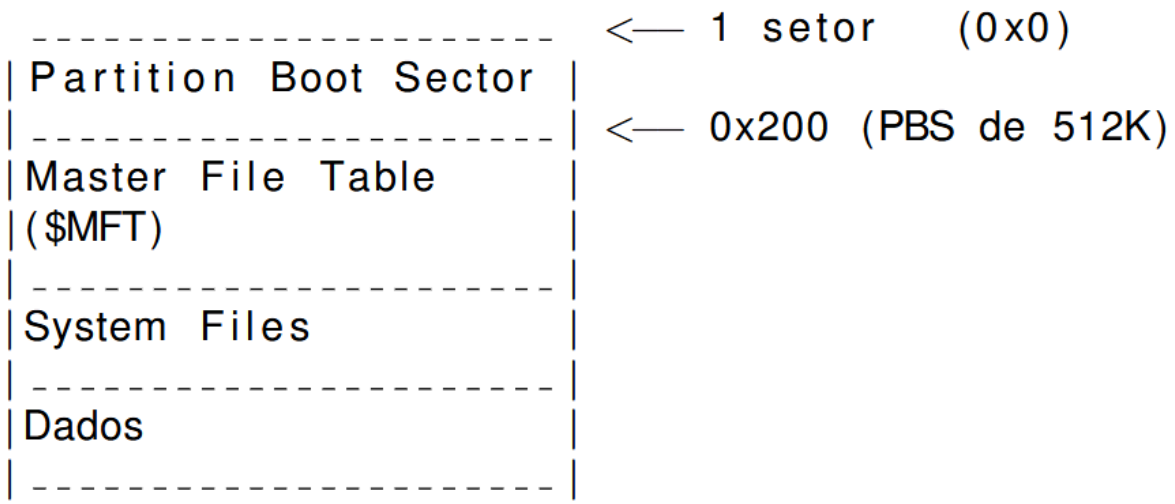


✅ **Boot Code** → Normalmente vazio.
✅ **Root Directory** → Diretoria base do sistema.
✅ **FAT #2** → Cópia de segurança da FAT #1.

🖥️ NTFS (New Technology File System)

- Sistema de ficheiros usado no **Windows NT, 2000, XP, 7+**.
- **Proprietário da Microsoft**.
- Suporta **ficheiros superiores a 4GB**.
- Todos os registos são **ficheiros**, incluindo a própria **$MFT**.

📌 Layout de uma Partição NTFS

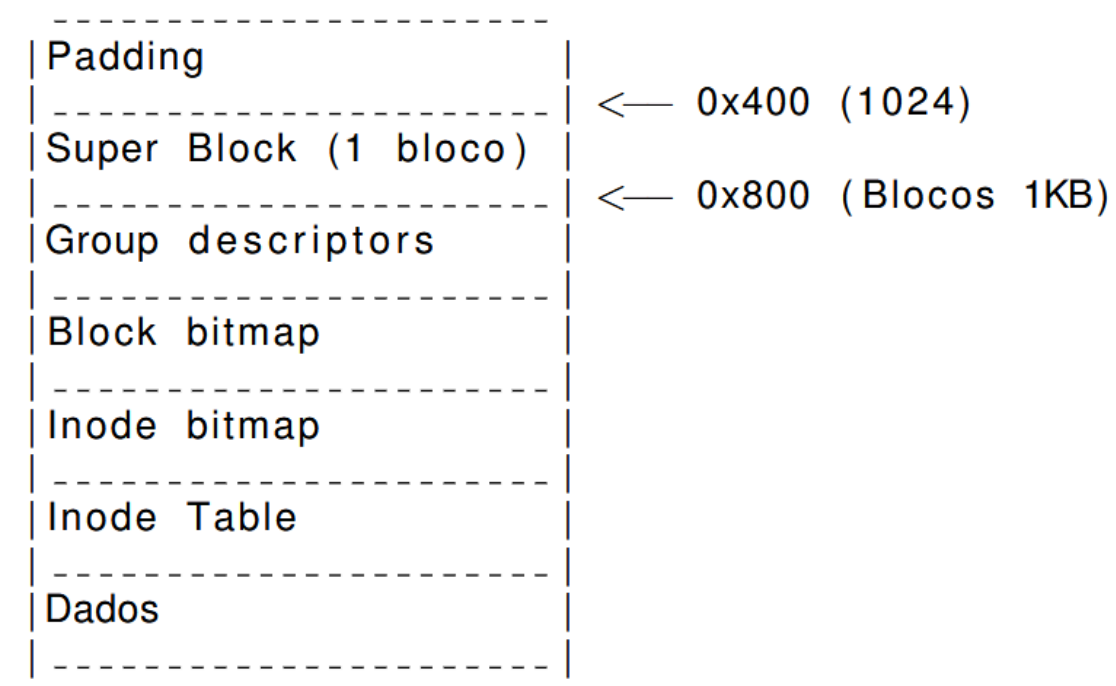


✅ **Partition Boot Sector** → Pode ocupar **1 a 16 setores**.
✅ Usa vários ficheiros de sistema (**$MFT, $Bitmap, $LogFile**).
✅ Mantém assinatura **0x55AA** no offset **0x1FE**.

🖥️ **EXT (Extended File System - Linux)**

🔹 **Usado em sistemas Linux**.
🔹 Organizado em **blocos de 1KB a 64KB**.

📌 **Layout de uma Partição EXT**



✅ **Organizado em blocos**.
✅ **Bloco inicial de 1024 bytes é ignorado**.

🔢 **Representação de Dados**

- **Números** → Decimal, binário, hexadecimal e octal.
- **Ordem de bytes**:
  - **Big-endian** → O byte mais significativo vem primeiro.
  - **Little-endian** → O byte menos significativo vem primeiro.
- **Codificação de caracteres**:
  - **ASCII** → 8 bits, utilizado historicamente.
  - **Unicode (UTF-8, UTF-16, UTF-32)** → Suporte a múltiplas línguas.

📌 **Autopsy - Módulos Importantes**

🔹 **Recent Activity**

📌 **Extrai informações dos últimos 7 dias**, incluindo:
✅ **Utilização da Internet** (pesquisas recentes, histórico).
✅ **Programas instalados** e **arquivos executados recentemente**.
✅ **Dispositivos USB conectados**.
✅ **Processamento do Registry Hive**.
📌 Resultados armazenados em **Extracted Content** → **Results**.

🛠 **Exemplo:**

🔍 **Investigação de um suspeito de fraude**:

- O Autopsy pode revelar que **um USB foi conectado recentemente**, contendo ficheiros Excel suspeitos.
- Pode mostrar **pesquisas no Google** como *"como apagar evidências de um computador"*.

🔹 **Hash Lookup**

📌 **Compara hashes MD5** dos ficheiros encontrados com bases de dados conhecidas.
✅ **Known Hash Sets** → Lista de ficheiros inofensivos que podem ser ignorados.
✅ **Notable Hash Sets** → Lista de ficheiros suspeitos ou criminosos.

🛠 **Exemplo:**

🔍 **Investigação de um ataque de malware**:

- O perito usa **Hash Lookup** para comparar ficheiros suspeitos com a base de dados **VirusShare**.
- O Autopsy detecta um ficheiro **"trojan.exe"**, cujo hash corresponde a um **ransomware conhecido**.

📌 **Bases de dados usadas:**

- **NSRL (National Software Reference Library)** → nsrl.nist.gov
- **VirusShare** → virusshare.com

🔹 **File Type Identification**

📌 **Identifica o tipo real dos ficheiros**, independente da extensão.
✅ Usa **Apache Tika** (tika.apache.org).
✅ Ajuda a detectar ficheiros **renomeados para esconder o verdadeiro conteúdo**.

🛠 **Exemplo:**

🕵️ **Esconder pornografia infantil**:

- O suspeito renomeia ficheiros **".jpg"** para **".docx"** para evitar detecção.
- O **File Type Identification** identifica que o ficheiro **não é um DOCX, mas sim uma imagem**.

🔹 **Embedded File Extraction**

📌 **Extrai ficheiros comprimidos** dentro de ZIP, RAR, DOCX, PPTX, XLSX.
✅ Permite analisar **ficheiros escondidos dentro de documentos**.
📌 Resultados aparecem em **Archives** → **File Types**.

🛠 **Exemplo:**

🕵️ **Corrupção financeira**:

- Um perito forense encontra um **Excel suspeito** num email.
- O **Embedded File Extraction** revela que há um **ZIP escondido dentro do Excel**, contendo **documentos de lavagem de dinheiro**.

🔹 **EXIF Parser**

📌 **Extrai metadados de imagens (EXIF)**.
✅ **Geolocalização (GPS)**.
✅ **Data e hora da foto**.
✅ **Modelo da câmara, configurações de captura**.
📌 Resultados em **EXIF Metadata → Extracted Content**.

🛠 **Exemplo:**

🔍 **Caso de stalking ou abuso**:

- O agressor envia **uma foto anónima** à vítima.
- O **EXIF Parser** revela **as coordenadas GPS** da imagem, indicando **onde foi tirada**.

🔹 **Keyword Search**

📌 **Pesquisa palavras-chave dentro dos ficheiros**.
✅ Usa **Apache Solr** para indexação rápida.
✅ Suporta **Texto, Office, PDF, E-mails**.
✅ Em ficheiros não suportados, usa **String Extraction**.

🛠 **Exemplo:**

🔍 **Investigação de terrorismo**:

- O perito adiciona palavras-chave como **"bomba", "alvo", "explosivo"**.
- O **Keyword Search** encontra um **documento Word com planos detalhados de um ataque**.

📌 **Listas predefinidas incluem:**
✔ **Endereços web (URLs)**.
✔ **Endereços IP**.
✔ **Números de telefone**.
✔ **E-mails**.

🔹 **Email Parser**

📌 **Extrai emails de ficheiros MBOX e PST**.
✅ Recupera mensagens e anexos.

🛠 **Exemplo:**

🔍 **Caso de phishing**:

- O suspeito nega ter enviado **emails fraudulentos**.
- O **Email Parser** analisa um ficheiro **PST do Outlook** e encontra **emails de phishing enviados para várias vítimas**.

🔹 **Extension Mismatch Detector**

📌 **Detecta ficheiros cuja extensão não corresponde ao conteúdo real**.
✅ Útil para **descobrir ficheiros camuflados**.

🛠 **Exemplo:**

🔍 **Esconder ficheiros ilegais**:

- Um ficheiro **"documento.pdf"** na verdade é um **executável malicioso**.
- O **Extension Mismatch Detector** alerta que **a extensão não corresponde ao formato interno**.

🔹 **E01 Verifier**

📌 **Verifica checksums de ficheiros E01**.
✅ Detecta **imagens forenses corrompidas**.

🛠 **Exemplo:**

🔍 **Caso de integridade de evidência**:

- O perito quer garantir que **uma imagem de disco E01** não foi alterada.
- O **E01 Verifier** calcula os hashes e confirma que **a evidência é autêntica**.

🔹 **Interesting Files Identifier**

📌 **Gera alertas quando encontra ficheiros suspeitos**.
✅ Critérios:

- **Nome, Caminho, Extensão**.
- **Tamanho e Tipo MIME**.

🛠 **Exemplo:**

🔍 **Caso de pedofilia**:

- O perito configura o **Autopsy** para alertar sobre **nomes de ficheiros suspeitos** como *"kids.zip"*.
- O **Interesting Files Identifier** encontra **pastas escondidas com nomes suspeitos**.

🔹 **PhotoRec Carver**

📌 **Recupera ficheiros eliminados do espaço não alocado**.
✅ Suporta **vários tipos de ficheiros**.
✅ Permite adicionar **assinaturas personalizadas**.
📌 Necessário ativar **"Process Unallocated Space"**.

🛠 **Exemplo:**

🔍 **Recuperação de provas apagadas**:

- Um criminoso apaga **imagens incriminatórias** antes da apreensão do disco.
- O **PhotoRec Carver** recupera **fotos e vídeos apagados**.

🔹 **Virtual Machine Extractor**

📌 **Identifica e analisa discos de máquinas virtuais**.
✅ Suporta **VMWare (VMDK)** e **Microsoft VHD**.

🛠 Exemplo:

🕵️ **Investigação de hacking**:

- O suspeito usa **máquinas virtuais para ocultar atividades**.
- O **Virtual Machine Extractor** monta e analisa **discos virtuais**, revelando **logs de acessos remotos**.

📌 Resumo Rápido

Módulo	Função	Exemplo
Recent Activity	Histórico de uso	USBs conectados, pesquisas feitas
Hash Lookup	Verificação de hashes	Detecta malware conhecido
File Type ID	Verifica tipo real do ficheiro	Detecta imagens escondidas em DOCX
Embedded File Extraction	Extrai ficheiros ZIP/RAR	ZIP escondido dentro de um Excel
EXIF Parser	Extrai metadados de imagens	Coordenadas GPS de uma foto
Keyword Search	Pesquisa por palavras-chave	Palavras como "bomba" em emails
Email Parser	Extrai emails de PST/MBOX	Encontra phishing num PST
Extension Mismatch	Detecta ficheiros disfarçados	Executável renomeado para PDF
PhotoRec Carver	Recupera ficheiros apagados	Recupera fotos eliminadas

---

🕵️ Análise a Sistemas Ligados (Live)

A investigação forense digital em sistemas ligados (live) foca-se na extração de informação volátil, como **dados em RAM, processos em execução e ligações de rede**. Essa abordagem é útil para obter informações que podem desaparecer após o desligamento do sistema.

🏛️ Especificidades e Limitações

- **Minimizar alterações no sistema analisado**, pois qualquer ação pode modificar evidências.
- **Dependência do sistema analisado**, pois algumas ferramentas podem ser bloqueadas ou afetadas por malware.
- **Requer acesso ao sistema**, tornando a análise intrusiva.

📌 Estratégias de Investigação Live

1️⃣ **Execução de utilitários ou scripts** → Comandos para recolher informações sem capturar a RAM.
2️⃣ **Recolha de uma imagem da memória RAM (RAM dump)** → Permite análise posterior da memória volátil.

Ambas implicam alteração do estado do equipamento em análise (intrusivas).

📂 Informação Passível de Recolha

- **Lista de processos em execução** (`pslist` no Linux, `tasklist` no Windows).
- **Lista de ficheiros abertos** (`lsof` no Linux, `handle` no Windows).
- **Ligações de rede ativas** (`netstat` no Linux/Windows).
- **Dump de memória RAM** (`WinPmem` para Windows, `LiME` para Linux).

🖥️ Dados em RAM

- Aplicações não protegem conteúdo armazenado na RAM.
- Palavras-passe e informação sensível podem ser recuperadas.
- Dados podem persistir mesmo após o encerramento do programa ou reinício.

📌 Análise de Cópias de Memória (RAM) - Volatility

🔹 **Volatility** é uma ferramenta **open-source** para extrair informação de dumps de memória RAM.
📌 Permite recuperar **processos, ligações de rede, passwords, clipboard e mais**.

🔹 **Comandos Essenciais do Volatility**

1️⃣ **pslist - Listar Processos**

✅ Mostra **todos os processos em execução** no momento do dump.

```
volatility -f memory.dmp --profile=Win7SP1x64 pslist
```

🛠 **Exemplo:** 🔍 **Análise de malware** → Descobrir **processos desconhecidos ou suspeitos** em execução.

2️⃣ **hivelist - Identificar Localizações do Registry**

✅ Mostra **endereços de memória** onde as chaves do **Windows Registry** estão armazenadas.

```
volatility -f memory.dmp --profile=Win7SP1x64 hivelist
```

🛠 **Exemplo:** 🔍 **Identificar chaves do Registry** modificadas por **malware**.

3️⃣ **hashdump - Extrair Hashes de Passwords**

✅ Extrai **hashes de passwords do Windows** a partir da RAM.

```
volatility -f memory.dmp --profile=Win7SP1x64 hashdump
```

🛠 **Exemplo:** 🔍 **Recuperação de credenciais** → Permite obter **hashes de passwords** armazenadas na RAM.

4️⃣ **pstree - Processos em Forma de Árvore**

✅ Mostra processos em **formato hierárquico**, útil para ver **processos pai e filhos**.

```
volatility -f memory.dmp --profile=Win7SP1x64 pstree
```

🛠 **Exemplo:** 🔍 **Detectar processos ocultos** → Alguns malwares iniciam processos **como filhos de processos legítimos**.

5️⃣ psscan - Identificar Processos Escondidos

✅ Pesquisa por processos **ativos ou terminados**.

```
volatility -f memory.dmp --profile=Win7SP1x64 psscan
```

🛠 **Exemplo:** 🔍 **Malware Injection** → Descobrir processos que **foram terminados, mas ainda residem na RAM**.

6️⃣ psxview - Múltiplas Listagens de Processos

✅ Compara **diferentes métodos** de listagem para encontrar **processos escondidos**.

```
volatility -f memory.dmp --profile=Win7SP1x64 psxview
```

🛠 **Exemplo:** 🔍 **Processos Rootkit** → Alguns rootkits **não aparecem** em `pslist`, mas aparecem em `psxview`.

7️⃣ connections - Ligações de Rede Ativas

✅ Lista **ligações de rede ativas** no momento do dump.

```
volatility -f memory.dmp --profile=Win7SP1x64 connections
```

🛠 **Exemplo:** 🔍 **Investigação de ataque** → Identificar **ligações remotas suspeitas**.

8️⃣ connscan - Ligações de Rede Identificadas por Análise de Memória

✅ Pesquisa **ligações de rede recentes**, mesmo as já fechadas.

```
volatility -f memory.dmp --profile=Win7SP1x64 connscan
```

🛠 **Exemplo:** 🔍 **Detecção de Malware** → Encontrar conexões que **não aparecem em logs do sistema**.

9️⃣ netscan - Ligações de Rede (Windows 10)

✅ Alternativa ao `connections`, **específico para Windows 10**.

```
volatility -f memory.dmp --profile=Win10x64 netscan
```

🛠 **Exemplo:** 🕵️ **Detectar comunicação de malware** → Verificar ligações remotas de **backdoors**.

🔟 notepad - Extrair Texto Aberto no Notepad

✅ Recupera **texto visível** de sessões ativas do Notepad.

```
volatility -f memory.dmp --profile=Win7SP1x64 notepad
```

🛠 **Exemplo:** 🕵️ **Investigação de espionagem** → Alguém pode ter digitado **informações confidenciais** num Notepad aberto.

1️⃣1️⃣ clipboard - Ver Conteúdo do Clipboard

✅ Mostra **o que estava copiado na área de transferência**.

```
volatility -f memory.dmp --profile=Win7SP1x64 clipboard
```

🛠 **Exemplo:** 🕵️ **Recuperação de senhas** → Muitas vezes, utilizadores **copiam senhas para o clipboard** sem perceber.

📌 Resumo Rápido

Comando	Função	Exemplo de Uso
pslist	Lista processos ativos	Ver processos suspeitos em execução
hivelist	Identifica localização do Registry	Ver chaves modificadas por malware
hashdump	Extrai hashes de passwords	Recuperação de credenciais Windows
pstree	Exibe processos em árvore	Encontrar processos filhos suspeitos
psscan	Identifica processos escondidos	Rootkits e malwares ocultos
psxview	Compara listagens de processos	Detectar processos invisíveis
connections	Lista ligações de rede ativas	Ver comunicação de malware
connscan	Ligações de rede recentes	Ver conexões fechadas
netscan	Ligações de rede (Win10)	Identificar backdoors remotos
notepad	Extrai texto do Notepad	Recuperação de notas digitadas
clipboard	Ver conteúdo do Clipboard	Encontrar senhas copiadas

---

📱 Investigação Digital a Telemóveis

A investigação digital forense em dispositivos móveis foca-se na **extração e análise de dados** armazenados em telemóveis, cartões SIM e cartões de memória. Esses dispositivos contêm informações valiosas como **registos de chamadas, mensagens, emails, dados de localização e redes sociais**.

🏛️ Estrutura da Investigação

- **Cartões SIM** → Contém identificadores, registos de chamadas, SMS e contatos.
- **Telemóveis** → Armazenam IMEI, ficheiros, registos de aplicações, redes Wi-Fi e GPS.
- **Cartões de Memória** → Podem conter ficheiros multimédia e backups de dados.
- **Cloud e Operadoras** → Algumas informações podem ser obtidas mediante autorização legal.

- **UICC (Universal Integrated Circuit Card)** → Componente electronica que pode conter múltiplos cartões SIM.
- **SIM**
  - **Autenticação** → Usa algoritmos próprios (Algoritmo A3).
  - **Registo** → Faturação, tendo por base os IDs do SIM e do equipamento.
- **Identificadores**:
  - **ICCID (Integrated Circuit Card Identifier)** → Número único do cartão SIM(19 a 20 dígitos).
  - **IMSI (International Mobile Subscriber Identity)** → Identifica o cliente e a operadora.
  - **MSISDN (Mobile Station International Subscriber Directory Number)** → Número de telefone associado.

📌 **Prova Potencial em Cartões SIM (UICC/SIM)**

🔹 **O que pode ser usado como prova forense?**

✅ **Identificadores únicos**

- **ICCID (Integrated Circuit Card Identifier)** → Identifica o cartão SIM.
- **IMSI (International Mobile Subscriber Identity)** → Identifica o assinante na rede.

✅ **Dados armazenados no SIM**

- **Agenda e contactos** → Números de telefone salvos.
- **Histórico de chamadas** → Chamadas efetuadas, recebidas e perdidas.

✅ **Mensagens SMS**

- Incluindo **mensagens apagadas**, se ainda não forem sobrescritas.

🛠 **Exemplo de Investigação**

🔍 **Caso de tráfico de drogas:**

- A polícia apreende um telemóvel e extrai o **IMSI** do SIM para confirmar que o dispositivo pertence ao suspeito.

- O **histórico de chamadas** mostra ligações frequentes para um fornecedor conhecido.
- O **SMS recuperado** contém mensagens relacionadas à venda de substâncias ilegais.

📱 Telemóveis e IMEI

- **O que é o IMEI?** → International Mobile Equipment Identity, um identificador único de cada telemóvel.
- **Estrutura do IMEI**:
  - **TAC (Type Allocation Code)** → Primeiros 8 dígitos que indicam marca e modelo.
  - **Número de Série** → 6 dígitos identificando o dispositivo.
  - **Dígito de verificação** → Último dígito para validação.
- **Exercício**: Verificar IMEI com `*#06#` e validar TAC online.

📌 Métodos de Aquisição de Dados

- **Aquisição Lógica** → Extrai apenas os dados acessíveis pelo sistema operativo.
  - Aplicações forenses (ex.: XRY, UFED) pedem permissões ao dispositivo.
  - Não permite recuperar dados apagados.
- **Aquisição Física** → Copia bit a bit os dados do dispositivo.
  - Permite recuperação de ficheiros eliminados.
  - Requer exploração de vulnerabilidades ou permissões root.

🔒 Isolamento da Rede

- **Motivo:** Evitar alterações ou eliminação remota de dados.
- **Técnicas:**
  - Ativar **modo avião**.
  - Remover **cartão SIM e desligar Wi-Fi**.
  - Usar **caixas ou sacos de Faraday** para bloqueio de sinais.

🛠️ Ferramentas Utilizadas

- **XRY** → Extrai dados de telemóveis bloqueados.
- **Autopsy** → Análise forense de sistemas Android.
- **FTK Imager** → Aquisição de cartões de memória.
- **SQLiteBrowser** → Análise de bases de dados de aplicações.
- **strings & file** → Identificação de tipos de ficheiros e extração de texto.

📌 Conclusões - Análise Forense de Dispositivos Móveis

🔹 A recordar...

✅ **IMEI** identifica um **dispositivo específico**.
✅ O **cartão SIM** contém:

- **ICCID** (*não requer PIN para leitura*).
- **IMSI** (*identifica o utilizador na rede*).
  ✅ O **equipamento elimina registos de chamadas** quando um novo **SIM** é inserido.
  ✅ **Isolar sempre os dispositivos** da rede para evitar eliminação remota de dados.
  ✅ Os resultados da análise **dependem do método de extração** e do equipamento.
  ✅ Dados podem estar armazenados em:
- **Memória do equipamento**.
- **Cartão SIM**.
- **Cartões de memória externos**.
  ✅ Também podem ser encontrados **fora do equipamento**:
- **Cloud, backups, operadores de telecomunicações**.

🔹 Informação disponível nos operadores

✅ **Detalhes do subscritor** (nome, morada, método de pagamento).
✅ **Registos de chamadas efetuadas/recebidas**.
✅ **Registos de SMS e MMS enviados/recebidos**.
✅ **Voicemail armazenado nos servidores da operadora**.
✅ **Identificadores do dispositivo**:

- **ICCID** (Cartão SIM).
- **IMSI** (Identidade da Rede Móvel).
- **IMEI** (Identificação do Telemóvel).
- **Número de telefone (MSISDN)**.
  ✅ **Código PUK** → Permite redefinir um **novo PIN** caso o SIM seja bloqueado.

🔍 Exemplo de Investigação
📌 Caso de sequestro

- O suspeito troca de cartão SIM para evitar rastreamento.
- A polícia solicita **registos de chamadas e SMS** à operadora.
- Com o **IMEI**, conseguem rastrear o dispositivo **mesmo com outro SIM**.

📱 Análise Forense de Aplicações Mobile

A análise forense de aplicações móveis foca-se na extração de informações sobre permissões, comunicações, bases de dados e código-fonte. Aplicações podem armazenar e transmitir dados sensíveis, tornando-as um alvo essencial para investigações digitais.

🛡️ Permissões em Aplicações

- **Controle de permissões:** Sistemas operativos móveis (Android e iOS) impõem restrições ao acesso de funcionalidades.
- **Permissões normais vs. perigosas:**
  - **Normais:** Baixo risco, ativadas por padrão (ex.: alterar fuso horário).
  - **Perigosas:** Requerem aprovação explícita (ex.: GPS, acesso a contactos).
- **Grupos de permissões importantes:**
  - **Localização** → Acesso a GPS e redes Wi-Fi.
  - **Armazenamento** → Leitura/escrita em dispositivos.
  - **SMS/Chamadas** → Monitorização de mensagens e registos de chamadas.
- **Riscos de segurança:**
  - Aplicações podem enganar utilizadores para obter permissões desnecessárias.
  - Algumas escondem comportamento malicioso por trás de permissões legítimas.

🛡️ Permissões perigosas

Grupo	Permissões
CALENDAR	READ_CALENDAR, WRITE_CALENDAR
CAMERA	CAMERA
CONTACTS	READ_CONTACTS, WRITE_CONTACTS, GET_ACCOUNTS
LOCATION	ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION
MICROPHONE	RECORD_AUDIO
PHONE	READ_PHONE_STATE, CALL_PHONE, READ_CALL_LOG WRITE_CALL_LOG, ADD_VOICEMAIL, USE_SIP PROCESS_OUTGOING_CALLS
SENSORS	BODY_SENSORS
SMS	SEND_SMS, RECEIVE_SMS, READ_SMS, RECEIVE_WAP_PUSH, RECEIVE_MMS
STORAGE	READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE

📡 Análise de Comunicações

- **HTTP vs HTTPS** → Aplicações devem usar HTTPS para proteger dados.
- **Captura de tráfego**:
  - Pode revelar comunicações inseguras ou dados expostos.
  - Ferramentas como `mitmproxy` permitem analisar conexões cifradas.
- **Autenticação e APIs:**
  - Algumas aplicações utilizam OAuth para autenticação.
  - APIs sem autenticação podem permitir vazamento de dados.

🗄️ Bases de Dados Locais

- **Armazenamento interno**:
  - Muitas aplicações usam **SQLite** para armazenar dados.
  - Algumas bases de dados não são cifradas, permitindo fácil extração.
- **Análise de bases de dados:**
  - Ferramentas como **SQLiteBrowser** permitem visualizar conteúdos armazenados.
  - Logs de atividades e mensagens podem ser recuperados.

🛠️ Análise de Código

- **Ficheiros APK**:
  - Aplicações Android são distribuídas como `.apk` (arquivos ZIP).

    - Podem ser descompiladas para análise com ferramentas específicas.
- **Ferramentas de análise:**
    - **APKTool** → Descompila APKs para análise do código.
    - **JADX-GUI** → Permite leitura do código-fonte Dalvik/Java.
    - **Bytecode Viewer** → Análise de binários compilados.
- **Extração de credenciais e links ocultos:**
    - Muitas aplicações armazenam tokens ou URLs sensíveis em seu código.
    - APKs podem conter credenciais hardcoded.

🌐 Análise Forense a Redes de Computadores

A análise forense de redes investiga **tráfego de rede, logs e dispositivos conectados** para identificar ataques, fugas de informação e atividades maliciosas. Permite a captura e interpretação de pacotes de dados.

📌 Complexidade na Análise Forense de Redes

🔹 Por que a análise forense de redes é complexa?

📌 A investigação forense em redes envolve **múltiplos fatores** que dificultam a recolha e interpretação de dados.

Principais desafios:

✅ **Número de equipamentos a analisar**

- **Servidores, PCs, dispositivos móveis, IoT, roteadores, switches**.
  ✅ **Heterogeneidade do ambiente**
- Diferentes **sistemas operativos (Windows, Linux, macOS)**.
- Vários **serviços e aplicações (HTTP, FTP, SSH, DNS, VPNs, etc.)**.
  ✅ **Grande volume de dados**
- **Milhões de pacotes** de rede podem ser gerados em minutos.
- Necessário **filtrar e identificar tráfego relevante**.
  ✅ **Diferentes formas de obtenção da informação**
- **Captura passiva** → Sniffers de rede (Wireshark, tcpdump).
- **Captura ativa** → Ataques man-in-the-middle, análise de logs de firewall e IDS.
- **Recolha de artefactos** → Logs de servidores, registos de autenticação.

🔍 Exemplo de Investigação

📌 Caso de ataque de ransomware numa empresa

- O analista deve inspecionar **centenas de dispositivos** para encontrar a origem do ataque.
- Logs de **diferentes sistemas operativos** precisam ser correlacionados.
- O tráfego capturado no **Wireshark** contém milhões de pacotes a serem analisados.
- A análise precisa ser precisa para evitar **falsos positivos** e isolar **o atacante real**.

📌 Características Intrínsecas da Análise Forense de Redes

1️⃣ Segmentação

📌 Os dados são transmitidos em **pacotes IP**, podendo ser fragmentados e enviados por rotas diferentes.
✅ **Desafio forense** → A reconstrução completa da comunicação pode ser difícil se houver **perda ou manipulação de pacotes**.

🛠 Exemplo:

🔍 Interceptação de tráfego HTTP

- Um atacante pode capturar pacotes **não cifrados** usando **Wireshark** e reconstruir páginas web visitadas.

2️⃣ Encriptação

📌 A maior parte das comunicações **atualmente é cifrada** para garantir a privacidade (**HTTPS, VPNs, TLS**).
✅ **Desafio forense** → A análise forense muitas vezes **não pode ver o conteúdo** do tráfego sem acesso às chaves de encriptação.

🛠 Exemplo:

🔍 Análise de tráfego HTTPS

- Um analista quer investigar um site suspeito visitado pelo utilizador.

- O tráfego capturado via **Wireshark** está encriptado, impossibilitando a leitura direta.
- **Solução**: Usar **mitmproxy** para capturar tráfego desencriptado dentro do browser.

3️⃣ Temporalidade

📌 O tráfego de rede **só pode ser capturado em tempo real**, não pode ser recuperado após a transmissão.
✅ **Desafio forense** → Se a captura não foi realizada no momento certo, a evidência pode ser perdida.

🛠 Exemplo:

🔍 Ataque DDoS a um servidor

- Um ataque ocorreu, mas **não havia uma captura de pacotes ativa** na hora do incidente.
- **A única evidência disponível são logs de firewall e IDS**, que podem não conter todo o tráfego.
- **Solução**: Implementar captura contínua com **tcpdump** ou **Security Information and Event Management (SIEM)**.

4️⃣ Localização

📌 O tráfego pode vir de **qualquer parte do mundo** e os atacantes podem **ocultar ou falsificar sua localização**.
✅ **Desafio forense** → Um atacante pode usar **VPNs, proxies ou Tor** para dificultar a identificação.

🛠 Exemplo:

🔍 Investigação de um hacker

- O analista tenta rastrear um ataque de **phishing** vindo de um IP **localizado nos EUA**.
- Após análise detalhada, descobre-se que o atacante **está na Rússia**, usando uma VPN para mascarar a origem.

- **Solução**: Correlacionar logs de múltiplos servidores e analisar **técnicas de fingerprinting de rede**.

📌 Fases da Investigação Forense de Redes

1️⃣ **Identificação da Rede** → Reconhecimento de dispositivos, endereços IP e topologia.
2️⃣ **Captura de Tráfego** → Uso de ferramentas como `tcpdump` e `Wireshark` para registar pacotes.
3️⃣ **Análise Protocolar** → Verificação de protocolos suspeitos (HTTP, SMTP, POP3, etc.).
4️⃣ **Filtragem e Extração** → Aplicação de filtros BPF para encontrar pacotes relevantes.
5️⃣ **Correlação de Eventos** → Comparação com logs e outros sistemas de monitorização.

🏛️ Técnicas de Captura de Tráfego

- **Port Mirroring** → Reencaminha pacotes de uma porta de switch para um dispositivo de monitorização.
- **Network Taps** → Dispositivos físicos que duplicam tráfego de rede.
- **Packet Sniffing** → Captura passiva de pacotes em redes Wi-Fi e com fio.

🛠️ Ferramentas de Captura e Análise

- **Wireshark** → Captura e análise gráfica de pacotes.
- **Tcpdump** → Captura de pacotes via linha de comandos.
- **Nmap** → Descoberta de dispositivos e serviços ativos.
- **ngrep** → Pesquisa de padrões em pacotes de rede.
- **tshark** → Versão CLI do Wireshark para filtragem e análise rápida.

📌 Portas de Monitorização em Cisco (Port Mirroring - SPAN)

As **portas de monitorização** permitem **espelhar o tráfego de uma ou mais interfaces** para análise forense de rede.

🔹 Como configurar a porta Fa0/10 como mirror da porta Fa0/1 (ambas as direções)

```
monitor session 1 source interface Fa0/1 both
monitor session 1 destination interface Fa0/10
```

✅ **Configuração SPAN** → Espelha o tráfego da interface **Fa0/1** para a interface **Fa0/10**.
✅ Captura **tráfego de entrada e saída** da porta analisada.

🔹 Como consultar os port mirroring existentes

```
show monitor

Session 1
---------
Type : Local Session
Source Ports:
RX Only: None
TX Only: None
Both: Fa0/1
Source VLANs:
RX Only: None
TX Only: None
Both: None
Source RSPAN VLAN: None
Destination Ports: Fa0/10
...
```

✅ Exibe as **sessões de monitorização ativas** no switch.
✅ Permite verificar se uma interface já está sendo usada para **captura de tráfego**.

🔹 Como enviar o tráfego de uma porta (Fa0/2) para duas portas (Fa0/11 e Fa0/12)

```
monitor session 2 source interface Fa0/2 both
monitor session 2 destination interface Fa0/11
monitor session 2 destination interface Fa0/12
```

✅ Permite que **duas portas recebam o mesmo tráfego** capturado.

✅ Útil para enviar tráfego a **múltiplos sistemas de monitorização (ex: Wireshark + IDS)**.

🔹 Como replicar pela porta Gi1/2 todo o tráfego enviado para a VLAN 10

```
monitor session 1 source vlan 10 rx
monitor session 1 destination interface Gi1/2
```

✅ Espelha **todo o tráfego da VLAN 10** para a interface **Gi1/2**.
✅ Importante para **monitorização de VLANs específicas em ambientes empresariais**.

🎯 **Berkeley Packet Filters (BPF)**

Aplicações de recolha de tráfego de rede capturam todos os pacotes, exceto se for especificada uma expressão BPF.

BPF permite filtrar pacotes durante a captura. Exemplos:

- Capturar tráfego HTTP:

  ```
  tcpdump -i eth0 port 80
  ```

- Capturar apenas pacotes de um IP:

  ```
  tcpdump -i eth0 src host 192.168.1.10
  ```

- Capturar pacotes DNS:

  ```
  tcpdump -i eth0 port 53
  ```

Capturar todos os pacotes com porta 80 e guardar em ficheiro.

```
[aap@eb-aap ~] $ sudo tcpdump -w http.pcap port http
tcpdump: listening on enp0s25, link-type EN10MB,
capture size 262144 bytes

^C
49 packets captured
49 packets received by filter
0 packets dropped by kernel
```