Digital Forensics

Federico Conti

2024/25

Contents

console.dd

The purpose of this analysis is to investigate a provided disk image (`console.dd`), which appears to be an unpartitioned FAT filesystem containing only a single JPEG file. There is suspicion that the volume was recently reformatted to conceal prior data. The goal is to reconstruct the original partition scheme and recover as much of the original content as possible.

Initial Setup

```
diff console.dd.sha256 <(sha256sum console.dd)

file console.dd
# Output:
 console.dd: DOS/MBR boot sector, code offset 0x3c+2, OEM-ID "mkfs.fat", sectors/cluster 4, root entrie

fdisk -l console.dd
# Output:
    Disk console.dd: 4 MiB, 4194304 bytes, 8192 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
    Disklabel type: dos
    Disk identifier: 0x00000000

sudo mount -oro console.dd /mnt/console1

strings -e S xbox.jpg | hexdump
# Output:
    00000000  ff d8 ff e0 0a 4a 46 49  46 0a 32 22 33 2a 37 25  |.....JFIF.2"3*7%|

# SHA xbox.jpg
    88f83bef7713f5e38aa59da9b71ec53081fe373593758879a4ce06a658ccead0  xbox.jpg
```

- The hash verification confirms that the image file `console.dd` is intact and unaltered.
- The image is detected as a FAT12 filesystem.
- No active partitions were listed, reinforcing that the current filesystem is unpartitioned.
- Upon mounting, only a single JPEG file (`xbox.jpg`) was present.
- The JPEG magic number (`FFD8 FFE0`) confirms the file type.

At this stage, the disk image appears as an unpartitioned FAT12 filesystem with a single valid JPEG file. However, further analysis is required to detect remnants of any previous partitioning.

Analysis Process

```
mmstat console.dd
# Output:
    gpt

mmls console.dd
# Output:
    GUID Partition Table (EFI)
    Offset Sector: 0
    Units are in 512-byte sectors
```

```
      Slot      Start        End          Length       Description
000:  -------   0000000000   0000002047   0000002048   Unallocated
001:  002       0000002048   0000008158   0000006111   Linux filesystem
002:  Meta      0000008159   0000008190   0000000032   Partition Table
003:  -------   0000008159   0000008191   0000000033   Unallocated
004:  Meta      0000008191   0000008191   0000000001   GPT Header
```

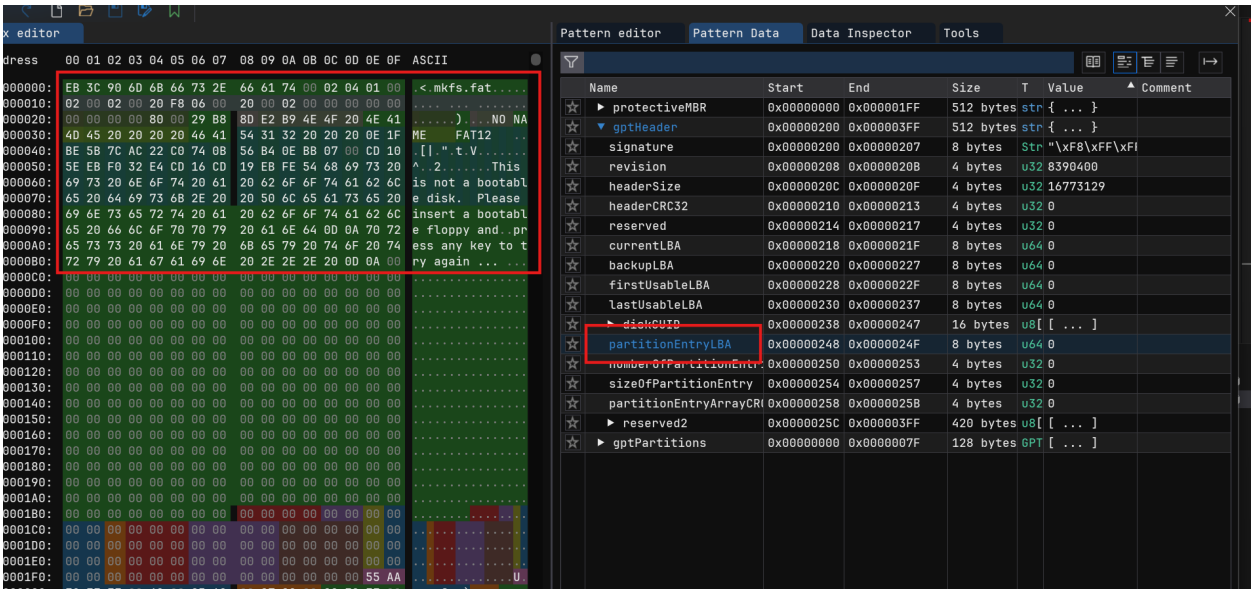
```
img_stat console.dd
# Output:
    IMAGE FILE INFORMATION
    --------------------------------------------
    Image Type: raw

    Size in bytes: 4194304
    Sector size:    512
```

Surprisingly, `mmstat` detected traces of a GUID Partition Table (GPT), which is inconsistent with a simple FAT12 format. This indicates remnants of a previous partition scheme.

Findings

- A previous GPT structure is partially detectable.
- There's evidence of a Linux filesystem starting at sector 2048.
- The presence of unallocated spaces and partition table metadata confirms that the disk was likely reformatted over an existing partitioned structure.

Using ImHex



Analysing the disk image with ImHex revealed further information on the GPT structure:

- It appears that the disk was quickly reformatted to Logical Block Addressing (LBA) 0 with a FAT12 filesystem. This action overwritten the primary GPT header, rendering it partially corrupted and unable to identify the original partition entries.
- Despite this, remnants of the GPT structure are still detectable, suggesting that the disk previously contained a

more complex partition scheme.

Given:

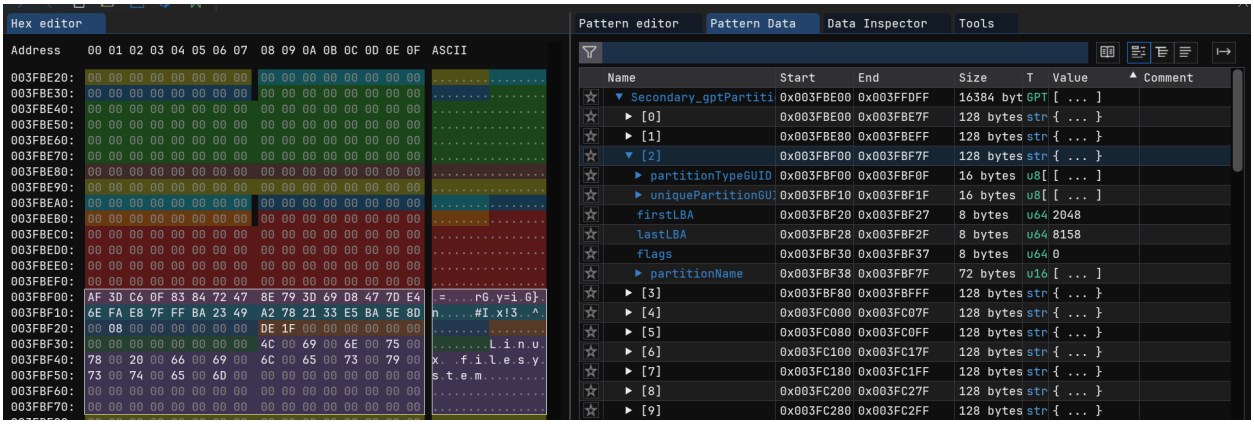
1. The disk image size is 4,194,304 bytes.
2. The sector size is 512 bytes.

We confirm there are exactly 8192 sectors. According to the GPT standard, the backup GPT header resides at the last sector (LBA 8191).

```
#include <std/mem.pat>

struct PartitionEntry {
    u8  bootIndicator;
    u8  startCHS[3];
    u8  partitionType;
    u8  endCHS[3];
    u32 relativeSectors;
    u32 totalSectors;
};

struct GPTHeader {
    char signature[8];          // "EFI PART"
    u32  revision;              // typically 0x00010000
    u32  headerSize;            // usually 92 (0x5C)
    u32  headerCRC32;           // CRC32 of the header
    u32  reserved;              // must be zero
    u64  currentLBA;            // LBA of this header
    u64  backupLBA;             // LBA of the backup GPT header
    u64  firstUsableLBA;
    u64  lastUsableLBA;
    u8   diskGUID[16];          // 128-bit GUID for the disk
    u64  partitionEntryLBA;     // LBA where partition entries start
    u32  numberOfPartitionEntries;  // number of partition entries
    u32  sizeOfPartitionEntry;      // size of each partition entry (often 128)
    u32  partitionEntryArrayCRC32;  // CRC32 of the partition entries
    u8   reserved2[420];        // 512 - 92 = 420 (fills one 512-byte sector)
};

struct GPTPartitionEntry {
    u8  partitionTypeGUID[16];
    u8  uniquePartitionGUID[16];
    u64 firstLBA;
    u64 lastLBA;
    u64 flags;
    u16 partitionName[36]; // 72 bytes (UTF-16)
};

//SECONDARY

// LBA = fdisk -l console.dd
GPTHeader Secondary_GPTHeader @ (8191 * 512);
GPTPartitionEntry Secondary_gptPartitions[128] @ (Secondary_GPTHeader.partitionEntryLBA * 512);
```

Result of GPT Analysis

- The secondary GPT header was successfully located at LBA 8191.
- The partition entries were parsed, revealing that: > Entry 2 defines a partition of type Linux File System. > This partition starts at sector 2048, consistent with previous findings from `mmls`.



Data Recovery

```
fsstat -o 2048 console.dd
# Output:
    File System Type: NTFS
    Version: Windows XP
    Cluster Size: 4096
    Total Sector Range: 0 - 6109
```

The partition was identified as NTFS, confirming that the original system was likely Windows-based.

```
dd if=console.dd of=ntfs.dd bs=512 skip=2048 count=6110

mount -oro  ntfs.dd /mnt/console1
# Output:
    file ps5.jpg
    ps5.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, ba
# SHA ps5.jpg
    200ff11c196aeeaecd7a4021aa40a47459a252b5776ecdd3104f2e5537eb75a2  ps5.jpg
```

Upon inspection of the mounted partition, a file named ps5.jpg was discovered. The file ps5.jpg is a valid JPEG image, confirming successful recovery of at least part of the original data stored prior to the reformatting attempt.

Advanced File System Analysis

To ensure a thorough investigation, advanced forensic tools were employed to enumerate files, recover orphaned data, and extract detailed metadata from the NTFS Master File Table (MFT).

```
fls -rp console.dd -o 2048
#
r/r 4-128-1:    $AttrDef
...
r/r 0-128-1:    $MFT
...
```

```
r/r 64-128-2:   ps5. # uniquely identifies the file or directory within the Master File Table.

V/V 65: $OrphanFiles
-/r * 16:       $OrphanFiles/OrphanFile-16
...
```

Presence of multiple orphaned files under $OrphanFiles, suggesting incomplete deletions or filesystem inconsistencies prior to the reformat.

To recover any residual files not linked within the filesystem, foremost was executed:

```
foremost -i console.dd -o recover/
```

Result:

- The carving process did not detect any deleted files.
- All files recovered by foremost were consistent with those already identified through filesystem analysis (ps5.jpg and xbox.jpg).
- No additional user files, fragments, or hidden data were found beyond the active filesystem entri

```
88f83bef7713f5e38aa59da9b71ec53081fe373593758879a4ce06a658ccead0  00000049.jpg
200ff11c196aeeaecd7a4021aa40a47459a252b5776ecdd3104f2e5537eb75a2  00006128.jpg
```

For detailed file metadata, the NTFS Master File Table (MFT) was extracted and analyzed using specialized tools.

```
icat -r -o 2048 console.dd 0 > MFT.bin
MFTECmd.exe -f MFT.bin --csv .\ --csvf console_mft.csv
```

- The file ps5.jpg was confirmed as an active file (InUse=True) with a creation and modification date of April 11, 2023.
- No Alternate Data Streams (ADS) or special attributes were detected. (parsing with TimelineExplorer)