Digital Forensics
Disk Forensics
Lecture 3

Volume Analysis

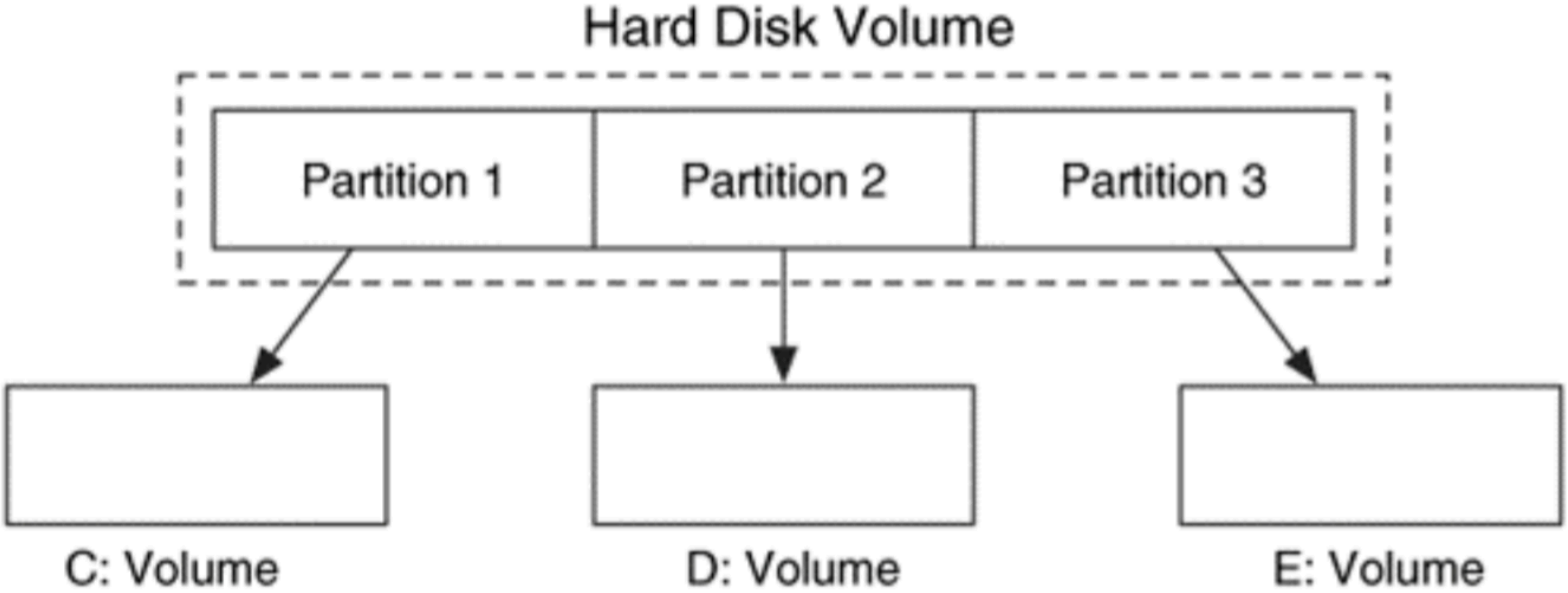
Akbar S. Namin
Texas Tech University
Spring 2018

Volume Analysis

- Volume concepts
  - A collection of addressable sectors that an OS or application can use for data storage
  - The usages:
    - To assemble multiple storage volumes into one storage volume
    - To partition volumes into independent partition
  - A hard disk is an example of a volume
- General theory of partitions
  - A collection of consecutive sectors in a volume
  - A partition is also a volume
  - Partitions are used in many scenarios including:
    - Some file systems have a maximum size that is smaller than hard disks
    - Many laptops use a special partition to store memory contents when the system is put to sleep
    - UNIX systems use different partitions for different directories to minimize the impact of file system corruption

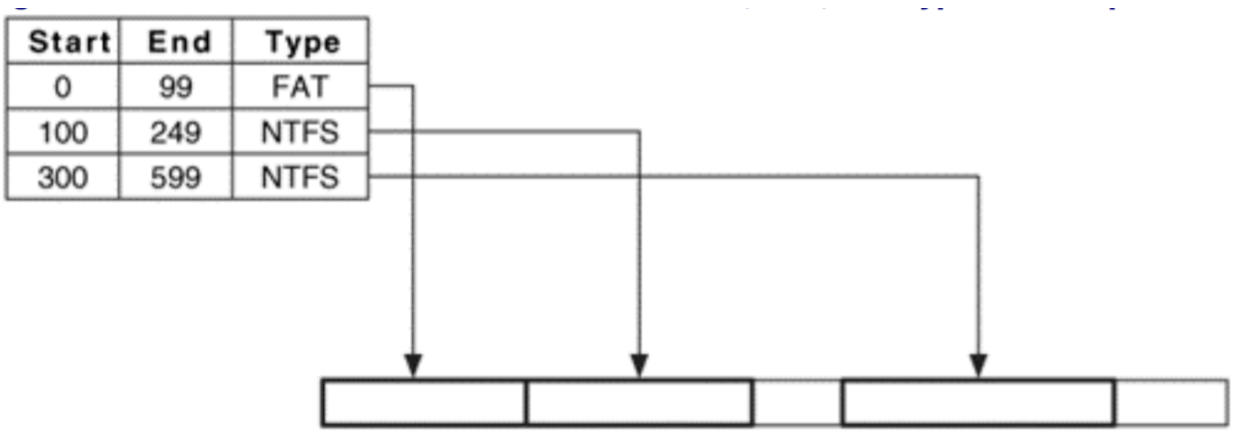
Volume Analysis

- General theory of partitions
  - An example hard disk volume is organized into three partitions, which are assigned volume names



Volume Analysis

- General theory of partitions
  - The common partitioning systems have one or more tables
  - Each table entry describes a partition
  - The data in the entry will have:
    - The starting sector of the partition,
    - The ending sector of the partition,
    - The type of partition
  - E.g. an example table with three partitions:



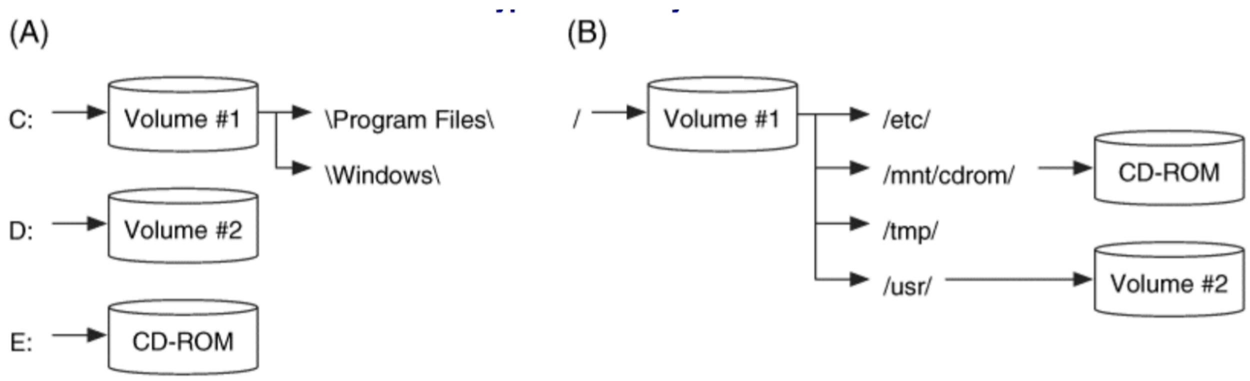
Start	End	Type
0	99	FAT
100	249	NTFS
300	599	NTFS

Volume Analysis

- General theory of partitions
  - The purpose of a partition system is to organize the layout of a volume
    - The only essential data are the starting and ending location for each partition
    - A partition system cannot serve its purpose if those values are corrupt or non-existent

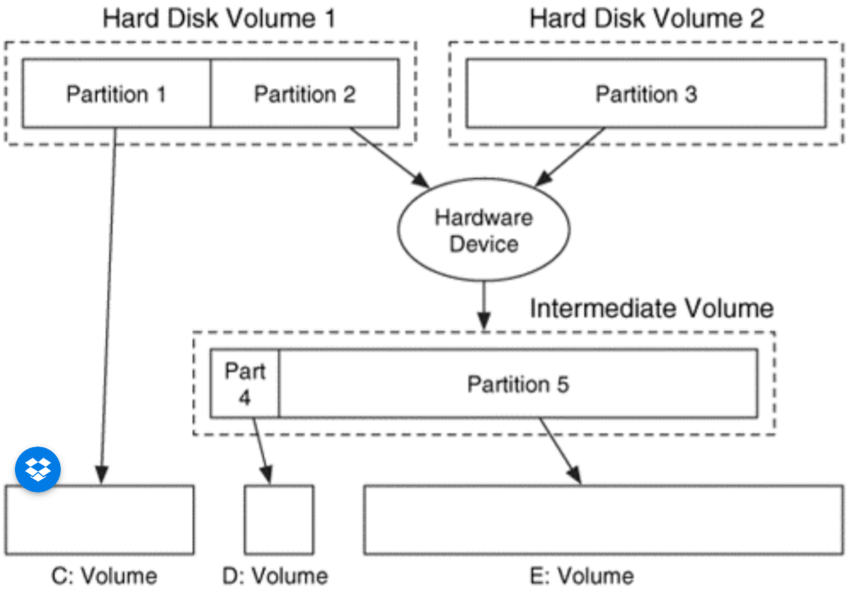
Volume Analysis

- Usage of volumes in UNIX
  - The user is presented with a series of directories that start at the root directory
  - The subdirectories of the root are either:
    - Subdirectories in the same file system
    - Or they are mounting points for new file system and volumes
  - E.g., a CD-ROM would be mounted at /mnt/cdrom in Linux
    - (A) Microsoft Windows
    - (B) typical UNIX system
  - To minimize the impact of drive corruption , UNIX partitions each disk into several volumes



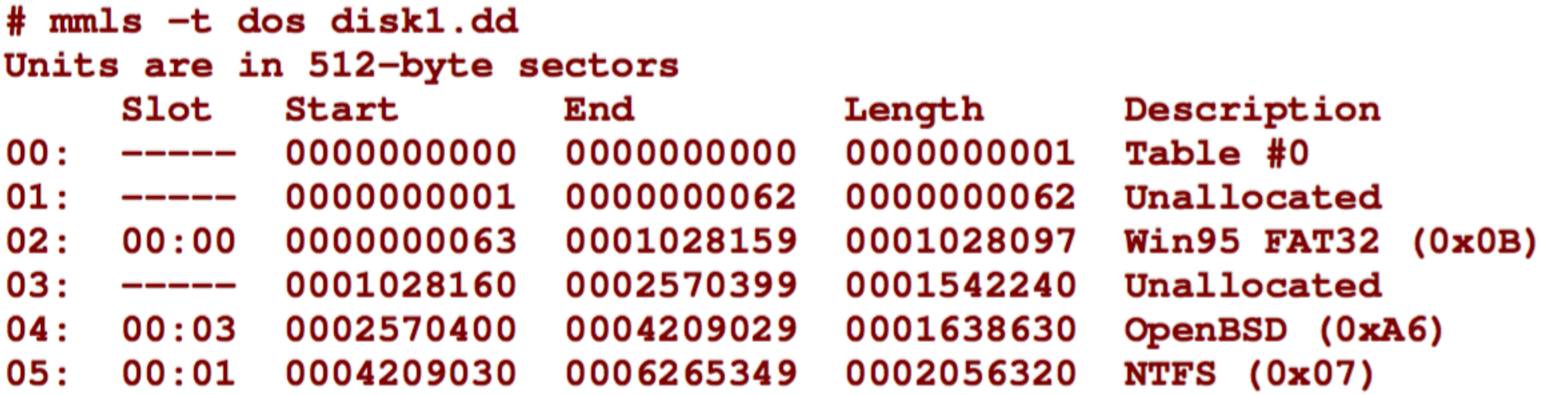
Volume Analysis

- General theory of volume assembly
  - Volume assembly techniques: to make multiple disks look like one. Why?
    - In case of disk failure (backup copy if one disk fails)
    - Make it easier to add more storage space
  - E.g. two hard disk with a total of three partitions
    - Partition 1 is assigned a volume name of C:
    - A hardware device processes partitions 2 and 3
    - The hardware device outputs one larger volume, which is organized into two partitions



Volume Analysis

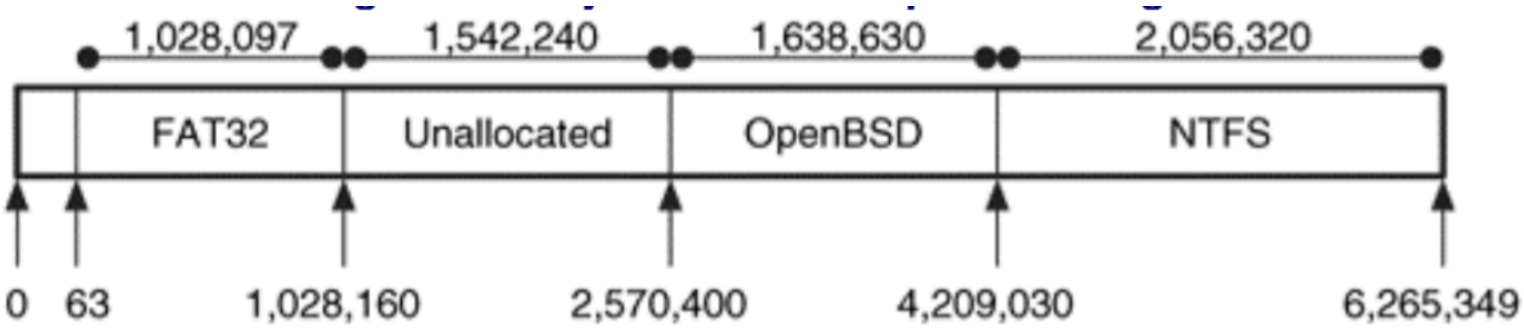
- Extracting the partition contents
  - Extracting is a simple process when the layout is known
  - The dd tool can be used with the following arguments:
    - if: the disk image to read from
    - of: the output file to save to
    - bs: the size of the block to read each time, 512 bytes (the sector size) is the default
    - skip: the number of blocks to skip before reading, each of size bs
    - count: the number of blocks to copy from the input to the output, each of size bs
  - Example:
    - Use the mmls tool from the Sleuth Kit to list the contents of the partition table.



```
# mmls -t dos disk1.dd
Units are in 512-byte sectors
     Slot    Start        End          Length       Description
00:  -----   0000000000   0000000000   0000000001   Table #0
01:  -----   0000000001   0000000062   0000000062   Unallocated
02:  00:00   0000000063   0001028159   0001028097   Win95 FAT32 (0x0B)
03:  -----   0001028160   0002570399   0001542240   Unallocated
04:  00:03   0002570400   0004209029   0001638630   OpenBSD (0xA6)
05:  00:01   0004209030   0006265349   0002056320   NTFS (0x07)
```

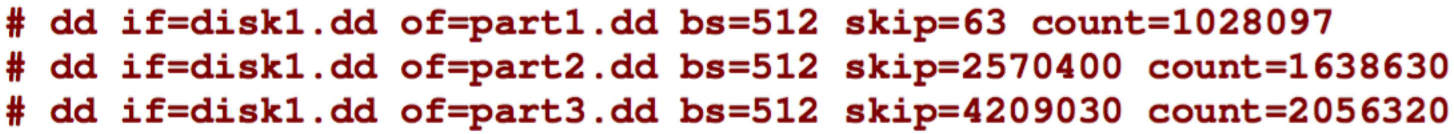
Volume Analysis

- Extracting the partition contents
  - mmls organizes the partition table entries based on their starting sector
  - Identifies the sectors that are not allocated to a partition
  - The first two lines (00 and 01) are the primary partition table
    - The unused space between the partition table and first partition
  - Line 02 is a partition with a FAT32 file system
  - Line 03 is for unallocated space on the disk
  - Line 04 is a partition for OpenBSD
  - Line 5 is a partition with an NTFS file system



Volume Analysis

- Extracting the partition contents
  - To exact the file system partitioning from the disk image:



```
# dd if=disk1.dd of=part1.dd bs=512 skip=63 count=1028097
# dd if=disk1.dd of=part2.dd bs=512 skip=2570400 count=1638630
# dd if=disk1.dd of=part3.dd bs=512 skip=4209030 count=2056320
```

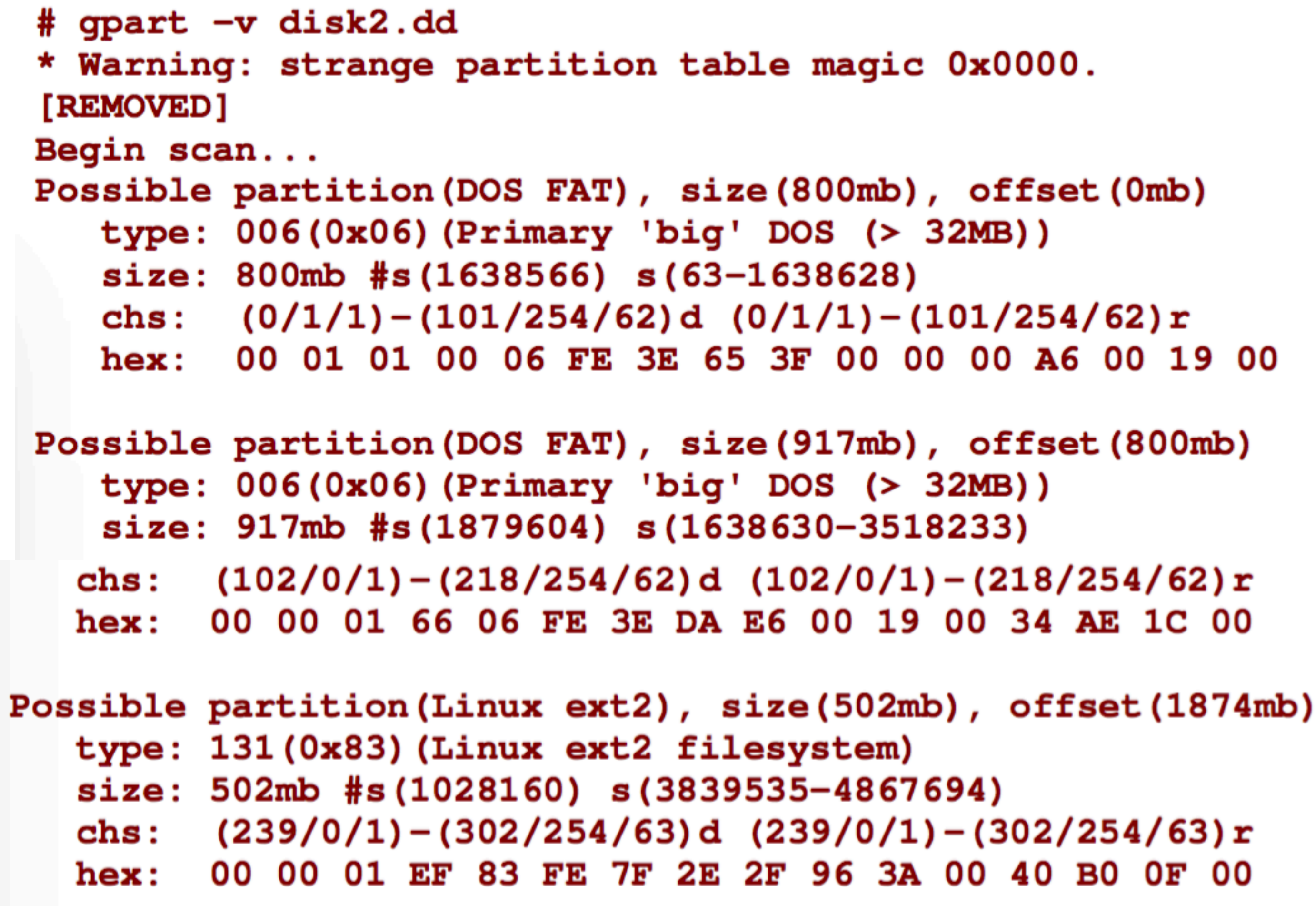
  - These commands take the disk1.dd file as input and save the output to files named part1.dd, part2.dd, and part3.dd
  - for each, blocks of 512 bytes are copied
  - The first partition is extracted by skipping 63 blocks before copying and then copying 1,028,097 blocks

Volume Analysis

- Recovering deleted partitions
  - Repartitioning a disk or clearing the partition structure is a technique used to thwart a forensics investigation
  - Fortunately, many file systems start with a data structure that has a constant “magic” or signature value
    - E.g., a FAT file system has the values 0x55 and 0xAA in bytes 510 and 511 of the first sector
  - The partition recovery tools search for these signature values and identify where a partition may have started
  - When the search tool finds a signature, additional tests can be conducted on the range of values that are valid for a given data structure
    - E.g., a FAT file system has a field that identifies how many sectors are in cluster
    - It must have a value that is a power of 2, such as, 1, 2, 4, 8, 16, 32, 64, or 128
    - Any other value would indicate that the sector was not part of a FAT file system boot sector, even though it ended with 0x55AA

Volume Analysis

- Recovering deleted partitions
  - A Linux tool that can be used for partitioning recovery is gpart
  - http://www.stud.uni-hannover.de/user/76201/gpart/



```
# gpart -v disk2.dd
* Warning: strange partition table magic 0x0000.
[REMOVED]
Begin scan...
Possible partition(DOS FAT), size(800mb), offset(0mb)
   type: 006(0x06)(Primary 'big' DOS (> 32MB))
   size: 800mb #s(1638566) s(63-1638628)
   chs:  (0/1/1)-(101/254/62)d (0/1/1)-(101/254/62)r
   hex:  00 01 01 00 06 FE 3E 65 3F 00 00 00 A6 00 19 00

Possible partition(DOS FAT), size(917mb), offset(800mb)
   type: 006(0x06)(Primary 'big' DOS (> 32MB))
   size: 917mb #s(1879604) s(1638630-3518233)
   chs:  (102/0/1)-(218/254/62)d (102/0/1)-(218/254/62)r
   hex:  00 00 01 66 06 FE 3E DA E6 00 19 00 34 AE 1C 00

Possible partition(Linux ext2), size(502mb), offset(1874mb)
   type: 131(0x83)(Linux ext2 filesystem)
   size: 502mb #s(1028160) s(3839535-4867694)
   chs:  (239/0/1)-(302/254/63)d (239/0/1)-(302/254/63)r
   hex:  00 00 01 EF 83 FE 7F 2E 2F 96 3A 00 40 B0 0F 00
```

Volume Analysis

- Recovering deleted partitions
  - The out put shows that there were likely two FAT partitions and one Ext2 partition
  - Another tool:
    - TestDisk
    - http://www.cgsecurity.org/testdisk.html

Disk Forensics

- Reference
- File System Forensic Analysis (Brian Carrier)