Digital Forensics
Lecture 02- Disk Forensics

Hard Disk Data Acquisition

Akbar S. Namin
Texas Tech University
Spring 2017

Hard Disk Data Acquisition

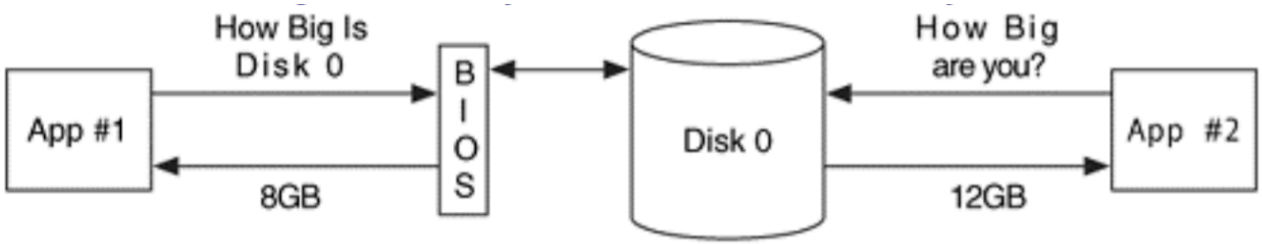
- Analysis of data found on a storage device
- It is more common to do dead analysis instead of live analysis
- General acquisition procedure
  - Copy one byte from the original storage device to a destination storage device and repeat
  - The chunks of data are transferred each time are typical a multiple 512 bytes (the size of most disk sectors)
- Every byte may contain evidence
- Acquire the files at the disk level:
  - Reason: if we acquire at the volume level and we made a copy of every sector in each partition, this would allow us to recover deleted files in each partition, but we would not be able to analyze the sectors that are not allocated to partitions.
  - E.g., a disk that has DOS partitions may not use sectors 1 – 62, and they could contain hidden data. If we acquire at the volume level, the hidden data would be lost.

Hard Disk Data Acquisition

- Acquisition tool testing
  - The National Institute of Standards and Technology (NIST) has conducted tests on common acquisition tools
  - The Computer Forensics Tool Testing (CFTT) project at NIST has developed requirements and test cases for disk-imaging tools
    - Results are on (http://www.cftt.nist.gov/disk_imaging.htm)

Hard Disk Data Acquisition

- Reading the source data
  - There are two major parts of the process
    - First, read data from a source (we focus on a typical IA32 system, x86/i386)
    - Then, write it to the destination
  - Direct versus BIOS access
    - Two methods in which data on a disk can be accessed
    - 1) the operating system or acquisition software accesses the hard disk directly, which requires that the software know the hardware details
    - 2) the operating system or acquisition software accesses the hard disk through the Basic Input/Output System (BIOS), which should know all the hardware details
    - The is a difference between these two:
      - Two applications are trying to determine the size of a disk. The BIOS is not properly configured and says that the 12GB disk is only 8GB



Hard Disk Data Acquisition

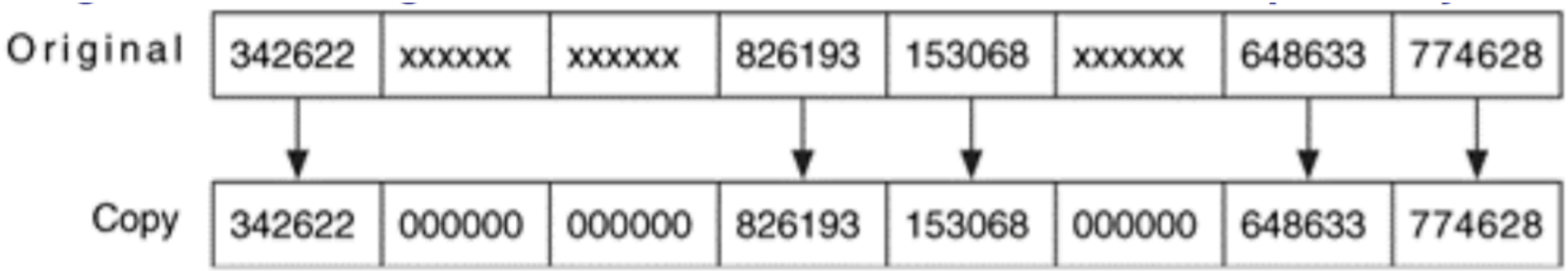
- When the BIOS is used, there is a risk that it may return incorrect information about the disk.
- The INT13h function will give access to only the first 8GB, but the disk is really 12GB
- We will not get access to the final 4GB
- Possible scenarios
  - 1) BIOS is configured for a specific hard disk geometry that is different from the one installed
  - 2) An acquisition tool uses a legacy method of requesting the size of the disk
- Two ways that an application can ask the BIOS for a disk size
  - 1) through the original INT13h function that suffers from the 8GB limit and returns the size using the disk's geometry in CHS format
  - 2) through the use of the extended INT13h functions.
- Make sure that you know how your acquisition tools access the disk, and if the tools use the BIOS, it reports the full disk

Hard Disk Data Acquisition

- Dead vs. live acquisition
  - A dead acquisition occurs when the data from a suspect system is being copied without the assistance of the suspect operating system
  - A dead acquisition can use the hardware from the suspect system as long as it is booted from a trusted CD
  - A live acquisition is one where the suspect operating system is still running and being used to copy data
    - The risk: the attacker may have modified the operating system or other software to provide false data during the acquisition
    - Attackers may install tools called rootkits into systems that they compromise and they return false information to a user
    - The rootkits hide certain files in a directory or hide running processes
    - Attackers may hide the files that they installed after compromising the system
    - Or modify the OS to replace data in certain sectors of the disk
  - Live acquisition should be avoided

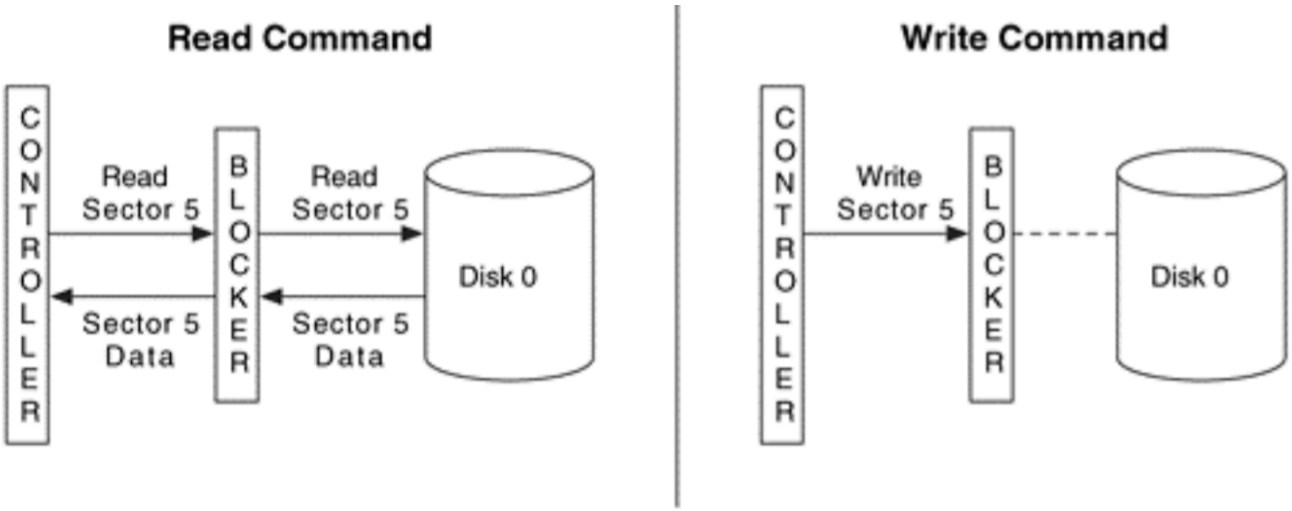
Hard Disk Data Acquisition

- Error handling
  - When reading data from a disk, the acquisition tool should be capable of handling errors.
  - errors could be caused by a physical problem
  - Dealing with a bad sector:
    - log its address and write 0s for the data that could not be read
    - Writing 0s keeps the other data in its correct location
    - If the sector were ignored instead of writing 0s, the resulting copy would be too small and most analysis tools would not work



Hard Disk Data Acquisition

- Hardware write blockers
  - Modify the original data as little as possible
  - A hardware write protector is a device that sits in the connection between a computer and a storage device
    - It monitors the commands that are being issued and prevents the computer from writing data to the storage device
  - E.g., the read request for sector 5 is passed through the write blocker, but the write command for the same is blocked before it reaches the disk

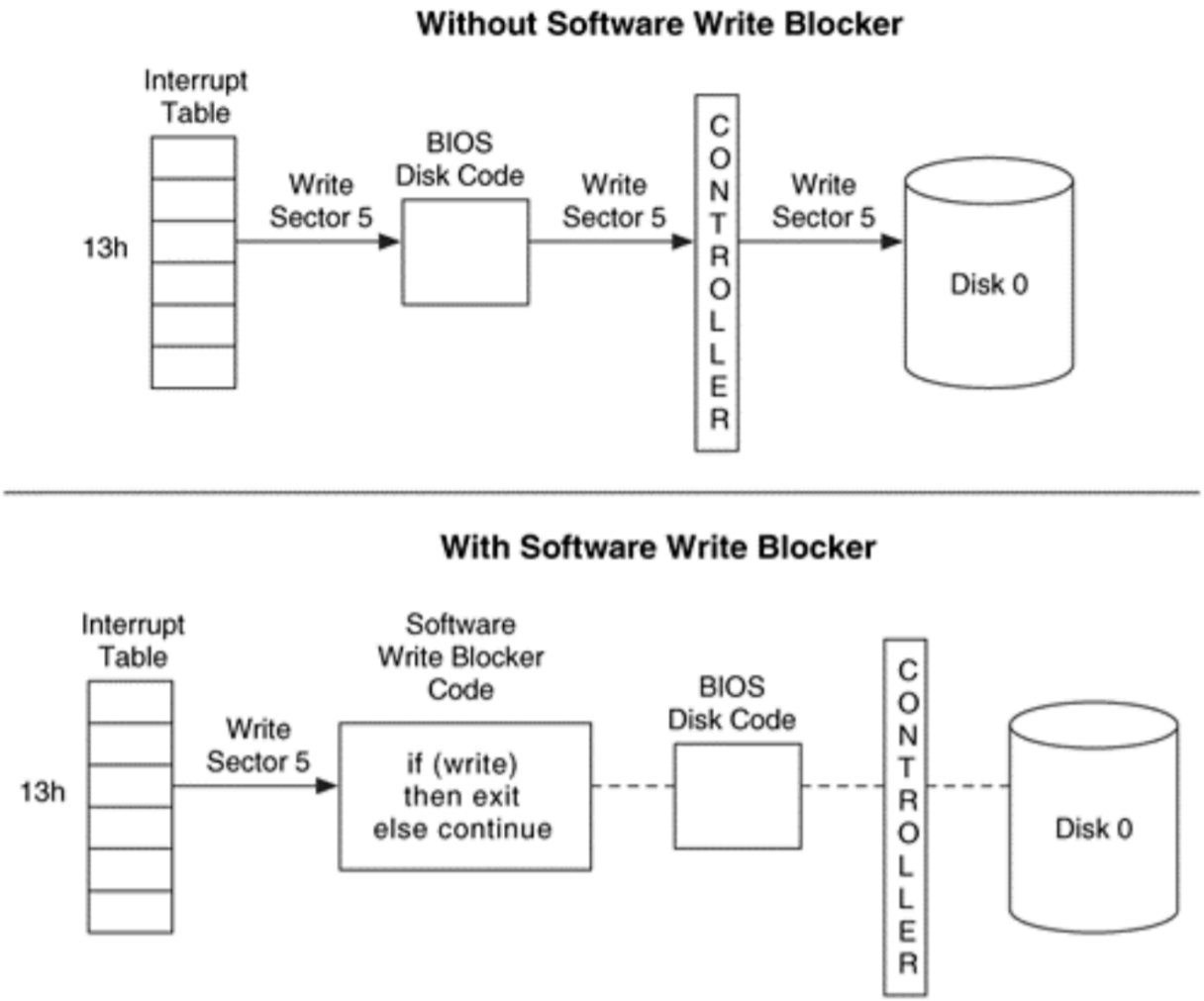


Hard Disk Data Acquisition

- Software write blockers
  - In addition to hardware write blockers, there are also software write blockers
  - They work by modifying the interrupt table
    - They are used to locate the code for a given BIOS service
    - The interrupt table has an entry for every service that the BIOS provides
    - Each entry contains the address where the service code can be found
    - E.g., the entry for INT13h will point to the code that will write or read data or from the disk
  - These blockers modify the interrupt table for interrupt 0x13 contains the address of the write blocker code instead of the BIOS code
  - When the OS calls INT13h, the writer blocker code is executed and examines which function is being requested
  - The CFIT group at NIST has developed requirements and has tested software write block devices
    - http://www.cftt.nist.gov/software_write_block.htm

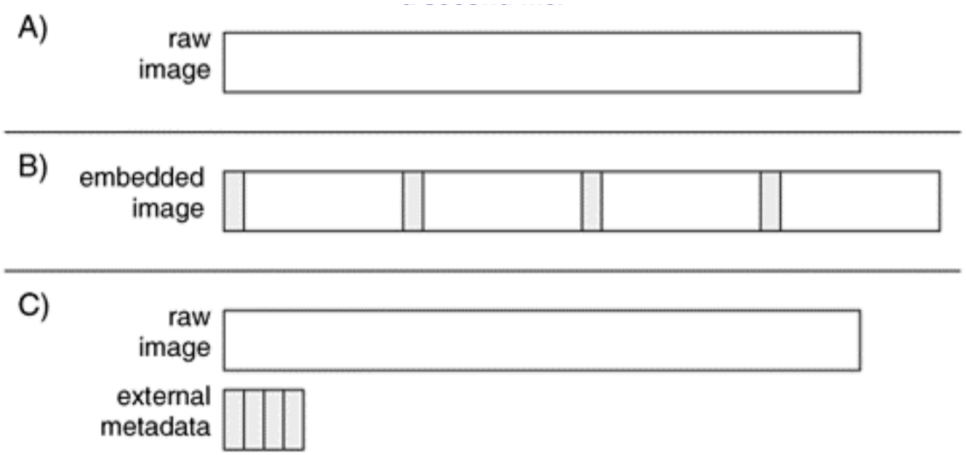
Hard Disk Data Acquisition

- Software write blockers
  - An example.



Hard Disk Data Acquisition

- Writing the Output Data
  - Once the data re read, must be written somewhere
  - Destination location
    - Write the data either directly to a disk or to a file (an image)
  - Image file format
    - We have a choice of in what format the image will be
    - A raw image contains only the data from the source device (easy to compare with the original one)
    - An embedded image contains data from the source device and additional descriptive data about the acquisition, such as hash values, dates, and times



Hard Disk Data Acquisition

- A case study using dd (the linux version)
  - One of the most simple and flexible acquisition tools
  - It copies a chunk of data from one file and writes it to another
  - It reads data from an input source, which is specified with the “if=“ flag
  - It writes the data to an output file, which is specified with the “of=“ flag
  - E.g., copy the contents of file1.dat, which is 1024 bytes, to file2.dat in 512-byte blocks: (two complete blocks read and written to the output)

```
# dd if=file1.dat of=file2.dat bs=512
2+0 records in
2+0 records out
```

  - If a full block was not used, the final two lines would ended with “+1” instead of “+0”
  - E.g., if file1.dat were 1500 bytes instead of 1024 bytes: (the resulting file will be the full 1500 bytes.)

```
# dd if=file1.dat of=file2.dat bs=512
2+1 records in
2+1 records out
```

Hard Disk Data Acquisition

- An Example on HPA (Host Protected Area disk)
- (A 57GB disk with 120,103,200 sectors. We have placed the string “here I am “ in sector 15,000, as seen here:

```
# dd if=/dev/hdb bs=512 skip=15000 count=1 | xxd
1+0 records in
1+0 records out
0000000: 6865 7265 2069 2061 6d0a 0000 0000 0000 here i am.......
```

- dd can read 1 byte at a time, it can also read 1GB at a time (performance varies)
- We then create an HPA in the final 120,091,200 sectors.
  - There are only 12,000 sectors that the OS or an application can access. This is shown below, since we cannot see the string in sector 15,000
  - No records were copied because it could not read the data

```
# dd if=/dev/hdb bs=512 skip=15000 count=1 | xxd
0+0 records in
0+0 records out
```

Hard Disk Data Acquisition

- Detecting an HPA in Linux:
  - Newer versions of Linux display a message in the dmesg log

```
# dmesg | less
[REMOVED]
hdb: Host Protected Area detected.
        current capacity is 12000 sectors (6 MB)
        native capacity is 120103200 sectors (61492 MB)
```

  - Use the hdparm tool in Linux

```
# hdparm -I /dev/hdb
[REMOVED]
        CHS current addressable sectors: 11088
        LBA    user addressable sectors: 12000
        LBA48  user addressable sectors: 12000
[REMOVED]
Commands/features:
        Enabled Supported:
           *    Host Protected Area feature set
```

Hard Disk Data Acquisition

- Detecting an HPA in Linux:
  - Use the diskstat tool from the Sleuth Kit
    - It displays the maximum native address and the maximum user address

```
# diskstat /dev/hdb
Maximum Disk Sector: 120103199
Maximum User Sector: 11999

** HPA Detected (Sectors 12000 - 120103199) **
```

  - We can use a tool called setmax in Linux and set the maximum number of sectors in the drive, which is 120,103,200

```
# setmax --max 120103200 /dev/hdb
```

Hard Disk Data Acquisition

- Output destinations
  - The output from dd can be either a new file or another storage device
  - Eg.
    - The first example copies the master ATA disk on the primary channel to a file
    - The second one copies the master ATA disk on the primary channel to the slave ATA disk on the second channel

```
# dd if=/dev/hda of=/mnt/hda.dd bs=2k
# dd if=/dev/hda of=/dev/hdd bs=2k
```

  - If the output file is not specified, the data will be written to the display
    - Useful to calculate the MD5 hash, to extract the ASCII strings, or to send the data to a remote system using a network
    - E.g,

```
# dd if=/dev/hda bs=2k | md5sum
```

Hard Disk Data Acquisition

- Cryptographic hashes
  - Used for later to prove an image's integrity
  - The md5sum utility can be used to compute a cryptographic hash of a file in accordance with dd
  - There is a tool called dcfldd , a variation of dd
  - http://sourceforge.net/projects/biatchux/
  - http://www.dc3.gov

Disk Forensics

- Reference
- File System Forensic Analysis (Brian Carrier)