Quirks Uncovered While Testing Forensic Tool

Jim Lyle

Information Technology Laboratory

ENFSC

16 October 2008

DISCLAIMER

Certain trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

Outline

- Overview of computer forensics at NIST
- Quirks uncovered
  - Write Blocking
  - Acquisition to an image file
  - Restoration from an image file
  - Other
- Questions and answers

Where is CFTT?

- US government, executive branch
- Department of Commerce (DOC)
- National Institute of Standards and Technology (NIST)
- Information Technology Lab (ITL)
- Software Diagnostics and Conformance Testing Division (SDCT)
- Computer Forensics: Tool Testing Project (CFTT)
- Also, the Office of Law Enforcement Standards (OLES) at NIST provides project input

Goals of CF at NIST/ITL

- Establish methodology for testing computer forensic tools (CFTT)
- Provide international standard reference data that tool makers and investigators can use in investigations (NSRL, CFReDS)

Project Sponsors (aka Steering Committee)

- National Institute of Justice (Major funding)
- FBI (Additional funding)
- Department of Defense, DCCI (Equipment and support)
- Homeland Security (Technical input)
- State & Local agencies (Technical input)
- Internal Revenue, IRS (Technical input)
- NIST/OLES (Additional funding & Program management)

Other Related Projects at NIST

- NSRL -- Hash (MD5, SHA1) file signature data base, updated 4 times a year (Doug White, John Tebbutt, Ben Long)
- PDAs and Cell Phones, NIST (Rick Ayers)
- SAMATE -- Software Assurance Metrics and Tool Evaluation (Paul E. Black)
- CFReDS -- Computer Forensics Reference Data Sets (Jim Lyle)

Forensic Tool Features

- … are like a Swiss army knife
  - Blade knife for cutting
  - Punch for making holes
  - Scissors for cutting paper
  - Cork screw for opening Chianti
- Forensic tools can do one or more of …
  - Image a disk (digital data acquisition)
  - Search for strings
  - Recover deleted files

Testing a Swiss Army Knife

- How should tools with a variable set of features be tested? All together or by features?
- Test by feature has a set of tests for each feature: acquisition, searching, recovery
- Examples: EnCase acquisition, iLook string search, FTK file recovery

Good News

- Forensic tools tested work
- Problems found are minor
  - Usually something is omitted
  - Nothing incriminating is created
- Investigators should be aware of the quirks

Write Blocking

- Goal: Prevent changes to a protected drive
- Host interacts with a drive by a command set through an interface
  - Read
  - Write
  - Control & info

Int 13 Extended Write

- DOS Interrupt 13 has three write cmds
  - Write (original write cmd)
  - Write long
  - Extended write (added later for large drives)
- Early write blocker versions only block write & write long

Blocking read commands

- Hardware write block devices …
  - Capture cmds sent from a host on a bus
  - Send cmds to a protected device
  - Return data to a host
- Some devices may …
  - Substitute a different cmd
  - Cache results and not issue cmd to device. If the protected device is reconfigured to report a different size, a cached size is reported incorrectly
  - Block some read cmds

Allow Reads vs Block Writes

- Block unsafe commands, allow everything else
  - + Always can read, even if new command introduced
  - - Allows newly introduced write commands
- Allow safe commands, block everything else
  - + Writes always blocked
  - - Cannot use newly introduced read commands

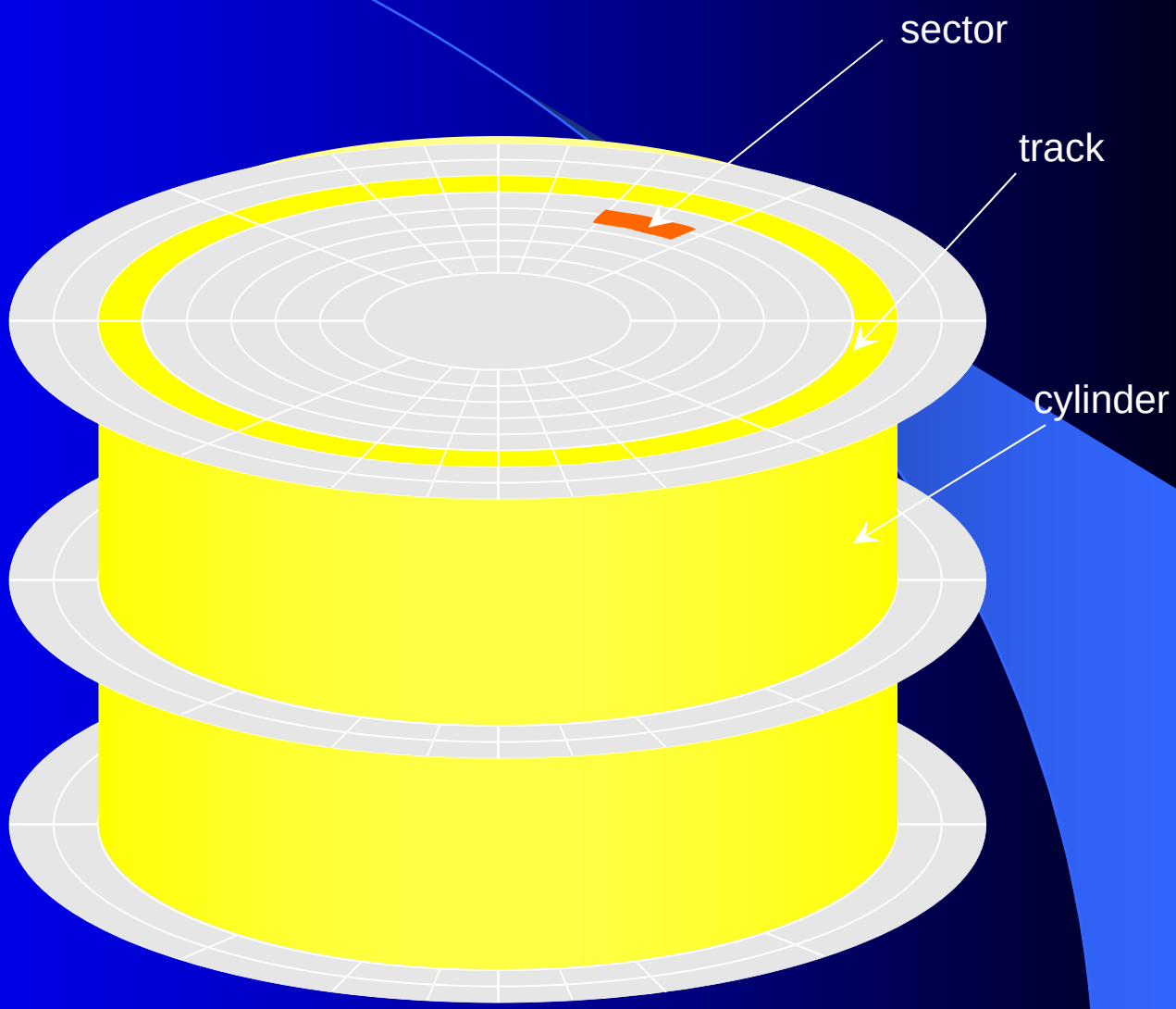
Source Acquisition

- Tool acquires either
  - entire drive (physical drive)
  - partition (logical drive)
- Evaluate the acquisition by either …
  - Hash of data acquired
  - Compare source to a restore

Core Acquisition Requirements

- All visible sectors are acquired
- All hidden sectors are acquired
- All acquired sectors are accurately acquired
- Benign fill of faulty sectors
- Error conditions

Hard Drive Organization



Hard Drive Organization

- Basic unit is a 512 byte sector
- 63 sectors are grouped as a track
- A set of tracks are grouped as a cylinder
- Sectors are addressed as:
  - Cylinder/track/sector
  - Cylinders start at 0,
  - Tracks start at 0, but
  - Sectors start at 1

Odd Sectors

- Use dd to acquire either a physical or logical drive with an odd sector count and the last sector is omitted.
- Occurs in the 2.4 kernel and earlier.
- The current 2.6 kernel does not have the problem.

BIOS Lies

- DOS based acquisition via BIOS interface
- Some BIOSs group several physical cylinders together into a logical cylinder
- There may be a fractional logical cylinder left over.
- In addition, some BIOSs may underreport the number of logical cylinders by 1 cylinder

More BIOS Lies

- Say a drive has 4003 physical cylinders but the BIOS groups 4 cylinders into one logical cylinder. The BIOS reports 1000 logical cylinders (4000 physical cylinders).
- Some tools acquire 1000 logical cylinders and miss the last 3 physical cylinders.
- If the BIOS underreports the size, some tools fail to adjust and acquire only 999 logical (3996 physical sectors).

More BIOS Lies

- Actual geometry: 3,309/16/63
  - 63x16 sectors/cylinder (1,008)
  - 3,335,472 total sectors 3,335,472x512 bytes
- BIOS wants cylinder count < 1024
- BIOS reports geometry as: 826/64/63
  - 63x16 sectors/cylinder (4,032)
  - 3,330,432 total sectors

Restoring an Image

- Testing the accuracy of a restore …
- Compare the original source sector by sector to the restored image

Missing Sectors on Restore

- Restore an image of an `IBM-DTLA-307020 with 40188960 sectors` to an identical drive the results are …
- ```
  Sectors Compared 40188960
  Sectors Differ 10395
  Diffs range 40178565-40188959
  ```
- Also the partition table gives 255 heads/cylinder and 63 sectors/track.
- That gives 16,065 (63*255) sectors/cylinder
- Note that 40,188,960 mod 16,065 is … 10,395
  ```

Next Quirk, Starting with Answer

Image an NTFS partition of 27,744,192 sectors

A	27,744,120 sectors
B	7 sectors
C	57 sectors
D	7 sectors
E	1 sector

NTFS Partition Restore

- Setup NTFS partition
  - MD5: 92b27b30bee8b0ffba8c660fa1590d49
  - 27,744,192 sectors
  - Each sector filled with sector LBA & disk ID
- Acquire partition
  - Total Sectors:27,744,191
  - 494A6ED8A827AD9B5403E0CC89379956
- Rehash (minus last sector) -- still no match

More NTFS

- Restore image to NTFS partition
- Compare to original
  - Sectors differ: 47
- Restore was in Windows XP …
- Restore again, unpower drive, no system shutdown. Compare to original
  - Sectors differ: 8
  - Diffs range: 27,744,184-27,744,191

NTFS Resolution

- Examine the eight sectors
  - Last sector not imaged
  - Other seven are a second copy of seven sectors starting at offset 27,744,120 -- Know this because each sector is tagged with LBA
- Verification:

Acquisition hash: 494a6ed8a827ad9b5403e0cc89379956

```
xena:/Users/jimmy root# dd bs=512 if=/dev/disk2s11 of=~jimmy/nt.dd
xena.local(1009)==> dd if=nt.dd bs=512 skip=27744120 count=7 of=end.dd
xena.local(1012)==> dd if=nt.dd bs=512 count=27744184 of=chunk.dd
xena.local(1013)==> cat chunk.dd end.dd | md5
494a6ed8a827ad9b5403e0cc89379956
xena.local(1022)==> md5 nt.dd
MD5 (nt.dd) = 92b27b30bee8b0ffba8c660fa1590d49
```

Faulty Sector Behaviors

- Some sectors adjacent to faulty sector missed
  - ATA interface: 8 sector window
  - USB interface: variable size window < 64
  - FW interface: variable, but different from USB
- Missed sectors filled with unknown data
- Image file gets out of sync

Imaging a Drive with Faulty Sectors

- Acquire all sectors that are not faulty,
- identify all faulty sectors, and
- in the image file replace the faulty sector content with benign fill.

Reliably Faulty Drives

- A set of known consistently faulty sectors.
- Can be imaged repeatedly with the same set of sectors reporting failure.
- Set of three reliably faulty drives:
  - MAX1 (54 faulty sectors)
  - MAX2 (398 faulty sectors)
  - WD (22 faulty sectors)

Basic Imaging Tools

- DCCIdd V 2.0
- DCFLdd V 1.3.4
- dd on Helix with Linux kernel 2.6.14
- dd on FreeBSD V 5.5
- IXimager V 2.0 February 1, 2006

Methodology

1. Create a reference drive identical to the faulty drive, but with no faulty sectors.
2. Clone the faulty drive with an imaging tool.
3. Compare the clone to the reference drive.

Results for Drive MAX1

Tool	Bus	readable sectors missed
IXimager	FW	0
Helix dd	FW	5034
DCFLdd	FW	5034
DCCIdd	ATA	306
FreeBSD dd	FW	0

The missed sectors were misidentified as faulty and filled with zeros.

DCCIdd ATA Interface

Look at first difference between the clone and reference drive.

- The first difference is a run of 8 sectors, all zeros, on the clone (10,069,088 - 10,069,095).
- First faulty sector at address 10,069,095.
- DCCIdd misidentifies seven sectors as faulty on messages to stderr.

More Runs (ATA Interface)

Next four runs …		
Bad Sector	**Run Start**	**Run End**
10069911	10069904	10069911
12023808	12023808	12023815
18652592	18652592	18652599
18656041	18656040	18656047

- All runs included at least on faulty sector.
- All runs were 8 sectors long.

DCFLdd, dd & Firewire

- Some sectors around a faulty sector misidentified as faulty and imaged as zeros.
- Unlike ATA, the length of the run of misidentified sectors including the faulty sector varied.
- First five run lengths: 168, 216, 72, 248, 112.
- Note: all are a multiple of 8.
- Faulty sector was always in last group of 8.

Results: Sectors Missed

- For IXimager and FreeBSD dd all the run lengths are one (no readable sectors missed).
- For imaging directly to the ATA interface with dd based tools the run length for a single isolated faulty sector was eight sectors (with seven sectors misidentified as faulty).
- For imaging with dd over the Firewire interface, the run lengths associated with a single, isolated faulty sector were a multiple of eight sectors (also with readable sectors misidentified as faulty).

Results: Fill Content

- IXimager filled the sectors with the string: ILookImager_Bad_Sector_No_Data
- All the tools running in the Linux environment filled the sectors with zeros (NULL bytes).
- The sectors created by dd running in FreeBSD contained data from an undetermined source.

Other Quirks

- Hash Quirks
  - Screen hash differ from log file
  - Multiple hashes: SHA ok, MD5 wrong; hardware dependent

Contacts

Jim Lyle

www.cftt.nist.gov

cftt@nist.gov

Doug White

www.nsrl.nist.gov

nsrl@nist.gov

Sue Ballou, Office of Law Enforcement Standards

Steering Committee Rep. For State/Local Law Enforcement

susan.ballou@nist.gov