Issues with Imaging Drives Containing Faulty Sectors

By

James Lyle and Mark Wozar

Presented At

The Digital Forensic Research Conference

DFRWS 2007 USA Pittsburgh, PA (Aug 13th - 15th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

http:/dfrws.org

Issues with Imaging Drives Containing Faulty Sectors

James R. Lyle and Mark Wozar

National Institute of Standards and Technology

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.

Overview

- How do imaging tools deal with faulty sectors?
- What should an imaging tool do?
- An experiment to investigate tool behavior.
- Experimental results.

CFTT Project Sponsors

- NIST/OLES (Program management)
- 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)

An Ideal Imaging Tool

- 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.

Next Steps

- Additional interfaces such as SATA and USB.
- Impact of faulty sectors on acquisition time.
- A better understanding of the run length of good sectors not acquired adjacent to a faulty sector.
- Investigation of the content used to replace the sectors not acquired in the FreeBSD environment.
- Investigation of the behavior of other imaging tools and run environments.

Contacts

Jim Lyle
cftt@nist.gov

Mark Wozar
mwozar@nist.gov

http://www.cftt.nist.gov

Sue Ballou
Office of Law Enforcement Standards
susan.ballou@nist.gov