An Alternate Methodology for Validating Hardware Write Block Devices

AAFS – 23 February 2012

Ben Livelsberger
NIST
Information Technology Laboratory
CFTT Project

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

- Federated Testing
- Validating Hardware Write Blockers - Usual Method
- NIST HWB Test Tools
- Interesting Discoveries
- Conclusion
- Sponsors

Federated Testing at NIST

- NIST is exporting its tool testing expertise and methods
- Benefits
  - A common test methodology
  - Time savings & peer-reviewed test reports
- CFTT Strength – Formal, well researched test methods
- CFTT Weakness – Limited resources for testing

Validating Hardware Write Blockers – Usual Method

- Write block drive
- Attempt write to drive
- Attempt to format drive
- Check if drive changed

Problem - OSes Implement Multiple Write Commands

- ATA drives:
  - Most OSes default to using READ & WRITE DMA commands
  - Some drives < 128 GB: implement WRITE DMA, but not WRITE DMA EXT
  - Drives > 128 GB: need EXT command set (WRITE DMA EXT) to fully access drives
  - Older drives: only implement PIO (WRITE SECTORS & WRITE SECTORS EXT)

Possible Write Commands

- T10 Technical Committee, http://www.t10.org/ - 20 SCSI write commands
- T13 Technical Committee, http://www.t13.org/ - 17 ATA write commands
- Wanted a way to test write blockers using all 17 ATA and all 20 SCSI write commands

NIST Approach – ataraw library

- ataraw 0.2.1 http://afflib.org/downloads/ataraw-0.2.1.tar.gz
  - Kyle Sanders – Masters student at Naval Post Graduate School
  - uses the Linux SG_IO ioctl, to pass SCSI/ATA command packets to Linux SCSI Generic driver
- Extended ataraw library to implement more ATA and SCSI commands (ATA specs 4-8, SCSI RBC-2)
- 4 SCSI reads, 4 SCSI writes
- 12 ATA reads, 15 ATA writes

Commands Implemented

Reads Commands (SCSI)

08h READ 6

28h READ 10

A8h READ 12

88h READ 16

7Fh READ 32

Reads Commands (ATA)

C8h READ DMA

C7h READ DMA QUEUED

20h READ SECTOR(S)

C4h READ MULTIPLE

22h READ LONG

C9h READ DMA w/o retries

23h READ LONG w/o retries

21h READ SECTOR(S) w/o retries

25h READ DMA EXT

26h READ DMA QUEUED EXT

29h READ MULTIPLE EXT

25h READ SECTOR(S) EXT

2Ah READ STREAM EXT

2Ah READ STREAM DMA EXT

Write Commands (SCSI)

A0h WRITE 6

2Ah WRITE 10

AAh WRITE 12

8Ah WRITE 16

7Fh WRITE 32

Write Commands (ATA)

30h WRITE SECTORS

34h WRITE SECTORS EXT

CAh WRITE DMA

35h WRITE DMA

3Dh WRITE DMA FUA EXT

CCh WRITE DMA QUEUED

36h WRITE DMA QUEUED EXT

C5h WRITE MULTIPLE

39h WRITE MULTIPLE EXT

CEh WRITE MULTIPLE FUA EXT

3Ah WRITE STREAM DMA EXT

3Bh WRITE STREAM EXT

CBh WRITE DMA W/O RETRIES

31h WRITE SECTORS W/O RETRIES

3Ch WRITE VERIFY

Implementation – try_read, try_write, write_verify

- try_write – send every ATA or SCSI write command to unique LBAs (based on the command's opcode) on drive
- try_read – send all the ATA or SCSI read commands to a drive
- write_verify – read sectors from a hard drive to measure which, if any, write commands were able to write to the drive

Testing a Blocker with the NIST programs

1. For each hard drive interface supported by the write block (e.g., ATA,SAS,SATA), initialize a drive to known content
2. Calculate a “before” reference hash for each drive
3. For each permutation of host-to-blocker and blocker-to-drive interfaces execute the try_read and try_write programs
4. Calculate an “after” reference hash for each drive
5. Execute write_verify for each drive. Use the write_verify output along with the reference hashes to measure whether any sectors on the test drives have changed.

Advantages of the NIST Approach

- Ability to validate your write block with multiple write commands
- Know which commands you've validated your blocker for
- Know which, if any, commands your blocker fails for
- try_write and try_read have been validated for the eSATA, FireWire, and USB interfaces in Ubuntu 11.10

Notes of Interest – defense in depth

- "/dev/sdd: Read-only file system"
- First layer of defense: the firmware logic that blocks writes and passes reads
- Second layer: advertising the protected drive as read-only
- Modern OSes try to enforce "read-only"; older versions of Linux do not

Test Results – how good is your write block?

- Tested 3 write blockers from 3 leading manufactures using the NIST tools
- For one blocker, one sector of our drives kept changing
- For the USB interface, the blocker let me write content to the drive using the WRITE 16 command
- Reason to fret?
  - defense in depth
  - All the OSes I've tested use the WRITE 10 command
  - 2008 firmware version; now it's fixed

Conclusion

- Federated Testing – exporting test materials
- The usual method for write blocker testing is incomplete
- The NIST hardware write block test tools addresses those problems
- Good product design with defense in depth
- Important for you to validate your write block devices

Project Sponsors

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

Contacts

Ben Livelsberger

livebe01@nist.gov

www.cftt.nist.gov

cftt@nist.gov

Sue Ballou, Office of Law Enforcement Standards

susan.ballou@nist.gov

NIST tools, http://www.cftt.nist.gov/###.tar.gz

Questions?