available at www.sciencedirect.com

SCIENCE DIRECT®

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

A strategy for testing hardware write block devices☆

James R. Lyle

National Institute of Standards and Technology (NIST), 100 Bureau Drive, Stop 8970, Gaithersburg, MD 20899-8970, United States

Keywords:
Tool testing
Write blocking
Digital evidence
Computer forensics
Software testing

ABSTRACT

There is a critical need in the law enforcement community to ensure the reliability of computer forensic tools. A capability is required to ensure that forensic software tools consistently produce accurate and objective test results. The goal of the Computer Forensic Tool Testing (CFTT) project at the National Institute of Standards and Technology (NIST) is to establish a methodology for testing computer forensic software tools by development of general tool specifications, test procedures, and test sets. The results provide the information necessary for toolmakers to improve tools, for users to make informed choices about acquiring and using computer forensics tools, and for interested parties to understand the tools' capabilities. Our approach for testing computer forensic tools is based on well-recognized international methodologies for conformance testing and quality testing. This paper describes requirements and test assertions that make up a strategy for testing hardware write block devices.

1. Introduction

The Computer Forensics Tool Testing (CFTT) Project at the National Institute of Standards and Technology is developing methodologies for testing software write block tools and hardware write block devices. A specification for write blocker behavior (HWB, 2004), a test plan (HWB, 2005), and test software are available on the CFTT web site, http://www.cftt.nist.gov/. The basic goal of a write blocker is to allow access to all digital data on a secondary storage device while not allowing any changes to the storage device. The basic strategy for implementing a write blocker is to place a filter between software executing on a host computer and a secondary storage device that is to be protected. The filter then monitors I/O commands sent from the application and only allows commands to the device that make no changes to the device. Such a filter can be implemented either in software or in hardware. The goal of this paper is to discuss our experience in designing test methodologies for testing hardware write block devices.

A hard drive is a device for the storage of digital data. The human user of a hard drive (or other digital storage media) usually views the drive as a place to store information as files. This simple view is not quite complete because, in reality, other information must be placed on the drive to enable retrieval of the information at a later time and describe properties of the stored information. In this paper, we refer to this as *file system meta-data*. These meta-data include objects such as partition tables, inodes, master file tables and so forth. The management of the meta-data objects is usually handled by an operating system running on a host computer with the drive attached. Both the user files and meta-data objects are located on an area of the hard drive called the *user area*. An attempt to directly access areas of the drive outside of the user area by the host computer results in an error. However,

☆ 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.

E-mail address: jlyle@nist.gov

1742-2876/$ – see front matter
doi:10.1016/j.diin.2006.06.001

once again, there is another layer of data called *device meta-data*. The device meta-data can be accessed indirectly or with vendor defined commands (not usually publicly documented). Examples include device microcode (firmware), device serial number, and so forth.

A basic strategy for testing a hardware write block device is to simply try to write to a drive protected by the device under test. However, results from such a test may be misleading unless care is taken to ensure that the test is complete. A brief overview of hard drive operational details will help identify requirements for testing write block devices.

2. Background

Before a hard drive can be used it must be physically attached to a computer. A hard drive is attached to a computer by one of several available physical interfaces. A drive is usually connected by a cable to an interface controller located either on the system motherboard or on a separate adaptor card. The most common physical interface is the ATA (AT Attachment), also called IDE (integrated drive electronics) or EIDE (enhanced IDE) interface. Other common physical interfaces include SATA (Serial ATA), SCSI (small computer system interface), IEEE 1394 (also known as FireWire or i-Link), and USB (universal serial bus). Information on the ATA and SATA command sets can be found at http://www.t13.org/ and information of SCSI, IEEE 1394 and USB command sets for block devices (i.e., hard drives) can be found at http://www.t10.org/.

All access to a drive is accomplished by commands sent from a host computer to a drive through the interface controller. However, since the low level programming required for direct access through the interface controller is difficult and tedious, each operating system usually provides other access interfaces. For example, programs running in the DOS environment can, in addition to direct access via the drive controller, use two other interfaces: DOS service interface (interrupt 0×21) or BIOS service interface (interrupt 0×13). The DOS service operates at the logical level of files and records while the BIOS service operates at the physical drive sector level. More complex operating systems, for example, Windows XP or a UNIX variant (e.g., Linux), may disallow any low level interface (through the BIOS or the controller) and only allow user programs access to a hard drive through a device driver, a component of the operating system that manages all access to a device.

Note that changes to drive meta-data may originate from the drive without action by the host.

3. Hardware based write blockers

The primary goal of a hardware write blocking device is to prevent any change to data in the user area of a hard drive while allowing access to all data on a hard drive. The write blocker should in general preserve the configuration of a protected drive. Sometimes it may be desirable to change a drive configuration to obtain access to otherwise hidden sectors, for example, such as within an HPA (host protected area). Hardware write block devices usually work by breaking the bus used to attach a hard drive to a host computer into two segments. Instead of a single bus segment between a hard drive and a host there is a bus segment between the host and the blocking device and another bus segment from the blocking device to the hard drive. The two bus segments do not have to use the same protocol. One of the first blocking devices on the market used an SCSI connection to the host computer and an ATA connection the hard drive. Once the blocking device is connected it can intercept a command from the host and select a desired course of action for the command. The most common actions are the following:

- The device forwards the command to the hard drive.
- The blocking device substitutes a different command to the hard drive. This is the case if the blocking device uses different bus protocols for communication with the host and hard drive.
- The device simulates the command without actually forwarding the command to the hard drive. For example, the blocking device may already know the size of the hard drive and rather than asking the hard drive again if a request for the size of the hard drive is sent from the host, the device may just return the answer directly to the host.
- If a command is blocked, the device may return either *success* or *failure* for the blocked operation. However, returning *failure* may sometimes cause the host computer to lock up for some commands issued by some operating systems.

Hard drive standards are not static. The standards for the ATA drives are maintained at http://www.t13.org and continue to evolve. There have been seven releases of the ATA specification, and the eighth is in development.

- ATA-1 X3T10/791D Revision 4c 1994 (ATA-1, 1994).
- ATA-2 X3T10/0948D Revision 4c March 18, 1996 (ATA-2, 1996).
- ATA-3 X3T13 2008D Revision 7b January 27, 1997 (ATA-3, 1997).
- ATA/ATAPI-4 T13/1153D Revision 18 August 19, 1998 (ATA/ATAPI-4, 1998).
- ATA/ATAPI-5 T13/1321D Revision 3 February 29, 2000 (ATA/ATAPI-5, 2000).
- ATA/ATAPI-6 T13/1410D Revision 3 October 30, 2001 (ATA/ATAPI-6, 2002).
- ATA/ATAPI-7 V1 T13/1532D Revision 4b April 21, 2004 (ATA/ATAPI-7, 2004).
- ATA/ATAPI-8 ATA Command Set Rev 3b March 21, 2006 (ATA/ATAPI-8, 2006).

Of the 256 possible command codes in the ATA protocol, what action should a blocking device take for each code? In the ATA-7 standard, of the possible command codes, about 70 are defined as general use commands that are not reserved, retired, obsolete or vendor specific. In addition, there are more than 30 retired or obsolete command codes that were defined in earlier standards. There have been 21 distinct write commands defined in the first seven ATA standards. Only four commands are defined in all seven standards: WRITE BUFFER (E8h), WRITE SECTORS with retries (30h), WRITE MULTIPLE (C5h), and WRITE DMA (CAh). Three standards introduced