Security Studies & International Affairs - Daytona Beach

College of Arts & Sciences

2014

A Study of Forensic Imaging in the Absence of Write-Blockers

Gary C. Kessler
Embry-Riddle Aeronautical University, kessleg1@erau.edu

Gregory H. Carlton
California State Polytechnic University

Scholarly Commons Citation

Kessler, G. C., & Carlton, G. H. (2014). A Study of Forensic Imaging in the Absence of Write-Blockers. *Journal of Digital Forensics, Security and Law, 9*(3). Retrieved from https://commons.erau.edu/db-security-studies/28

A STUDY OF FORENSIC IMAGING IN THE ABSENCE OF WRITE-BLOCKERS

Gary C. Kessler
Embry-Riddle Aeronautical University
Dept. of Security Studies &
International Affairs
Daytona Beach, FL 32174
garykessler@erau.edu

Gregory H. Carlton
California State Polytechnic University
Computer Information Systems Dept.
Pomona, CA 91768
ghcarlton@csupomona.edu

ABSTRACT

Best practices in digital forensics demand the use of write-blockers when creating forensic images of digital media, and this has been a core tenet of computer forensics training for decades. The practice is so ingrained that the integrity of images created without a write-blocker are immediately suspect. This paper describes a research framework that compares forensic images acquired with and without utilizing write-blockers in order to understand the extent of the differences, if any, in the resultant forensic copies. We specifically address whether differences are superficial or evidentiary, and we discuss the impact of admitting evidence acquired without write blocking. The experiments compare the changes made to a hard drive and flash drive when imaged and examined with a Windows-based forensics workstation.

Keywords: Digital forensics, computer forensics, write-blocking, forensic image, forensic data acquisition

1. INTRODUCTION

The first phase of the computer forensics process - after identifying digital devices that might have a nexus to an investigation - is data acquisition. This action includes creating a forensically correct copy of the desired media and is typically achieved by obtaining a bit-stream image of the original data (Carlton, 2007). This activity is necessary in order to adhere to the mandate of conducting a forensic examination on a copy rather than on the original evidentiary media, where the copy is presumed to be identical to the original. Digital forensic examiners typically perform the imaging process by attaching the original storage medium (i.e., a hard drive or an external storage device) to a write-blocker that is, in turn, attached to a forensic workstation, and then employing software to create the forensic image. The intent of the write-blocker is to prevent the forensic workstation's software or operating system from making any inadvertent changes to the original media, including adding, deleting, or modifying any information (Forensic Focus, 2010; Henry, 2009; Nelson, Phillips, & Steuart, 2010).

Employing a write-blocker during the imaging process is so ingrained in our teaching, education, and practice that there is no known mention within the literature addressing the treatment of a forensic copy that has been made without a write-blocker. This omission raises many scientific questions: What happens if a disk or other media is imaged without benefit of a write-blocker? Is the copy tainted? If so, what is the extent of any contamination? Procedurally, if a device is imaged without a write-blocker, should such evidence be discarded by an examiner or investigator, ignored by counsel, or challenged by the opposing party on the presumption that the image no longer represents the original media? If such a generalized objection were raised, how should a judge know whether to sustain or overrule the objection, and how should the party offering such evidence argue for the evidence's inclusion?

These questions are not entirely hypothetical. The current Scientific Working Group on Digital Evidence (SWGDE) Best Practices for Computer Forensics document is clear that write-blockers should be used "when possible" (SWGDE, 2013, p. 7) without mention of the negative impacts of not using blockers. The Federal Rules of Evidence (FRE) specifically allow duplicates to be admitted as evidence to the same extent as the original unless legitimate questions can be raised as to the veracity of the copy (FRE, 2013). Does the lack of a write-blocker undermine the veracity of a copy so as to make it ineffective as evidence?

The National Institute of Standards and Technology (NIST) has detailed specifications on how to test hardware and software write-blockers to validate their proper operation (NIST, 2003, 2004, 2005). They do not, however, describe the impact when write-blockers are not employed. Thus, lack of write blocking remains a somewhat ambiguous state.

In U.S. v. Labuda (2012), the only case we found where write blockers were specifically cited as an issue, a police forensic examiner examined a cell phone's memory card without making an image. While the defense expert challenged the police examiner's lack of process by directly examining the card, the evidence was nevertheless allowed. But the write-blocker was not a central point of the case and, therefore, this case does not appear to sufficiently answer the questions above.

The study described in this paper includes experiments to test the scientific foundation behind the requirement to use write blockers and the assertion of their necessity. The testing framework is presented below in section 2. In section 3 of this paper, we discuss the results of our examination comparing images acquired with write-blockers to those acquired without write-blockers. Lastly, in section 4, we offer our conclusions.

2. TESTING FRAMEWORK

2.1 Test Philosophy

Write blockers are used in digital forensic imaging based upon the hypothesis that changes will occur to the source media if write blockers are not employed. For testing purposes, the null hypothesis is that no changes will occur to the source media if a write blocker is not used. The authors designed a test framework in an attempt to test the null hypothesis, or to measure the amount of changes, if any, made to digital media during the process of making a forensically correct copy (i.e., a bit-stream image). The framework was necessary in order to ensure that the experiments are repeatable and reproducible. In this context, repeatable means that we can get the same result over and over, and reproducible means that other researchers can get the same results following our framework. Or, in the words of NIST (2001):

> ... repeatability is defined as the ability to get the same test results on the same testing environment (same computer, disk, mode of operation, etc.). Reproducibility is defined as the ability to get the same test results on a different testing environment (different PC, hard disk, operator, etc.). (p. 7)

The universe of possible testing environments is actually quite huge. As Lyle (2012) observes, there are many choices of (Figure 1):

- Digital media storage technology (e.g., traditional spinning disk or flash/solid-state devices)
- File system type (e.g., FAT12/16/32, NTFS, HFS/HFS+, ext2/ext3/ext4, EXFAT, etc.)
- Interface between digital media and forensic workstation (e.g., Firewire, IDE, SATA, SCSI, USB)
- Forensic workstation operating system (e.g., Windows, Mac OS X, Unix, Linux)
- Write-blocker type (e.g., the variety of software and hardware products)
- Imaging software (e.g., dcfldd, dd, EnCase, FTK Imager, X-Ways Forensics) and hardware (e.g., Forensic Duplicator, Forensic Imager, Hardcopy, Shadow)
- Analysis software (e.g., Autopsy/TSK, EnCase, FTK, ProDiscover, X-Ways Forensics)

The non-exhaustive list above represents a subset of the variables and the options available, but even these represent more than 5,000 configurations. Given the size of the test universe, we selected a small subset of media and designed a testing framework as a demonstration of possible further work. However, our media subset represents two of the most commonly used devices at the time of our study (i.e., a SATA hard disk and a USB flash drive).

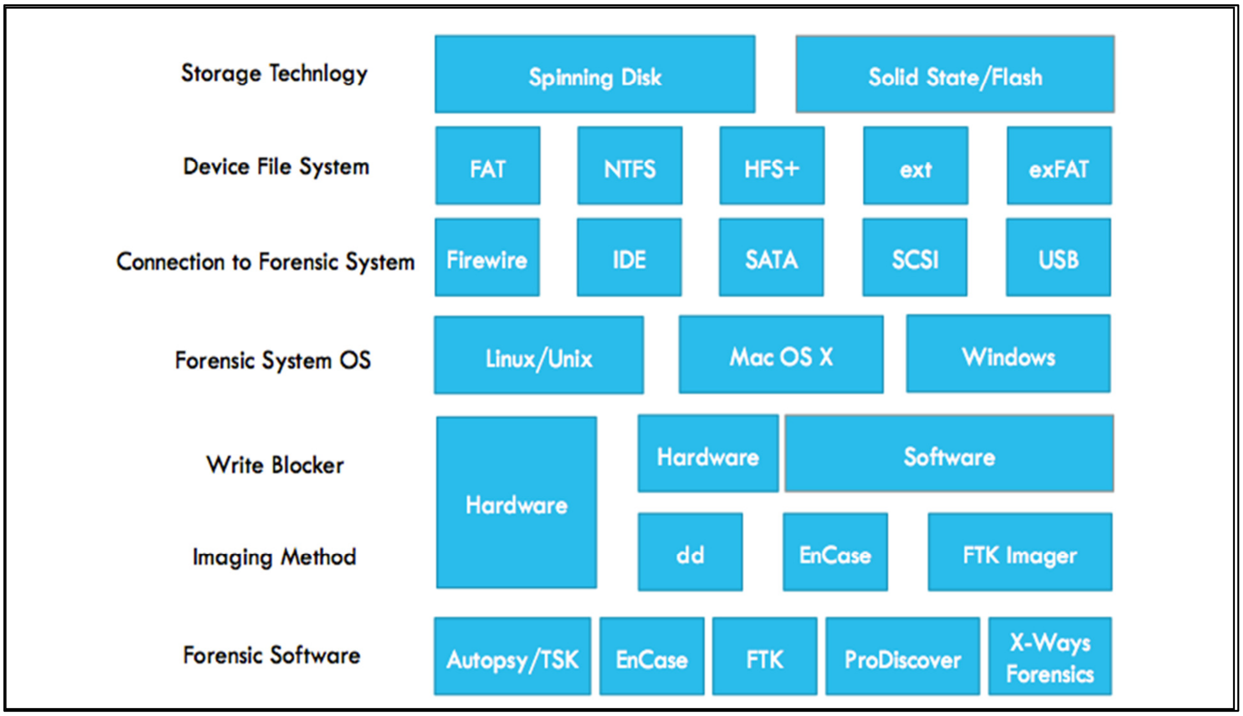

Figure 1. A subset of forensic imaging variables.
ext - Extended File System (Linux); exFAT - Extended File Allocation Table (Microsoft).
FAT - File Allocation Table (Microsoft); FTK - Forensic Toolkit (AccessData).
HFS - Hierarchical File System (Apple); IDE - Integrated Drive Electronics (Western Digital).
NTFS - New Technology File System (Microsoft); SATA - Serial Advanced Technology Attachment.
SCSI - Small Computer System Interface; TSK - The Sleuth Kit.
USB - Universal Serial Bus.

2.2 Test Design

As mentioned above, NIST (2003, 2004, 2005) has detailed functional and test plan specifications for hardware and software write blockers. The tests described here are not as thorough as the NIST tests because the point is not to validate functioning write blockers but to determine what changes occur on the target medium when write blockers are absent, either because they were never used or later found to be non-functional.

The general testing process is straightforward:

1. Prepare known data sets and write them to verified wiped storage devices.
2. Take a bit-stream image of the storage device attached to the forensic workstation with a hardware write-blocker.
3. Take a bit-stream image of the storage device attached to the forensic workstation via the Universal Serial Bus (USB) port and a software write-blocker.
4. Take a bit-stream image of the storage device attached to the forensic workstation via the USB port without use of either a hardware or software write-blocker.
5. Document hash value findings.

2.3 Test Sets

This project tested two different media types, namely, a hard disk drive (HDD) and a USB flash drive (also commonly known as a *thumb drive*). Two different test data sets were created, one for each device. Each test set had a valid file system and typical set of files in allocated and unallocated space, as an example of something that might be found in the field.

The first media item was an 80 GB SATA Seagate ST380815AS hard drive. The drive was wiped to an all-zero state using the built-in wiping function of EnCase. The wiped hard drive was then connected to a computer in order to install the Windows 7 Professional (32-bit version) operating system using the NTFS file system; default settings were accepted and all Windows drivers were installed. Microsoft Office 2010 was then installed; normal Office documents, including a mix of Word documents and Excel spreadsheets, were then written to the drive. Finally, the test team visited several typical Web sites such as asrock.com, bing.com, google.com, msn.com, newegg.com, windows.microsoft.com, and yahoo.com. This collection of information was designated HDD Data Set 0.

The second media item was an 8 GB Kingston USB flash drive. The flash drive was attached to the forensic workstation and a full format performed from the Windows operating system, employing the FAT32 file system. A variety of files and documents were then copied to the drive, including a mix of Word documents, Excel documents, Microsoft OneNote files, Adobe Portable Document Format (PDF) files, PowerPoint files, and several image files. The flash drive was then properly ejected from the forensic workstation. This collection of information was designated USB Data Set 0.

2.4 Image Creation

Three bit-stream forensic images were acquired from each of the media items. In general, Image 0 (which served as the control set) was acquired using a hardware write-blocker, Image 1 was acquired using software write blocking, and Image 2 was acquired without a write-blocker. All images were created using EnCase software.

Hardware write blocking employed Tableau Forensic Bridges, described in more detail below. The Tableau hardware write-block capabilities were validated prior to our tests.

Software write blocking was accomplished by setting the Windows Registry key controlling the write status of the USB port, namely
HKEY_LOCAL_MACHINE\SYSTEM\

CurrentControlSet \Control\StorageDevicePolicies\WriteProtect. Setting this Registry key's value to 0x0 allows writing to the USB port and a value of 0x1 disables writing. After making any change to the Registry key's value, the system was rebooted and the appropriate functionality verified.

3. TESTS AND RESULTS

3.1 Imaging and Hashes

The SATA hard drive was connected to the forensic workstation's USB port using a Tableau T35e Forensic SATA/IDE Bridge. The T35e can operate with the write-block mode either ON or OFF; the write-blocking functionality was confirmed prior to running the tests.

Image 0 was obtained by turning ON the write-block function of the T35e bridge. EnCase was used to acquire the image, producing both Message Digest #5 (MD5) and Secure Hash Algorithm 1 (SHA-1) hashes (Table 1).

Image 1 was obtained by turning OFF the write-block function on the bridge and, instead, setting the Windows registry key to write-block the USB port. After changing the Registry key, the forensic workstation was rebooted and the USB write-block functionality was tested prior to attaching the drive to the workstation. The image was acquired and hash values calculated using EnCase.

Image 2 was obtained by turning OFF the write-block function of the bridge and leaving the Windows registry key related to the USB port's write capability at its normal setting. After changing the keys, the forensic workstation was rebooted and the USB write-enable capability was tested. The image was acquired and hash values calculated using EnCase.

Table 1
HDD Image Hash Values

HDD Image 0 (hardware write-blocker)	
MD5	F8BE2078382A68ADC792B734FFB480CD
SHA-1	23812B61A41E01F61AF0DA048B80AD25F1D4C8BA
HDD Image 1 (software write-blocker)	
MD5	F8BE2078382A68ADC792B734FFB480CD
SHA-1	23812B61A41E01F61AF0DA048B80AD25F1D4C8BA
HDD Image 2 (no write-blocker)	
MD5	CF4539F0E98E67F368E5EB1D3119EB03
SHA-1	31CE3038803A8B4BE39FD197905123664C5C2C5C

Image 0 of the USB flash drive was obtained by connecting the device to the forensic workstation's USB port using a Tableau T8 Forensic USB Bridge in write-blocker mode. The write-blocking functionality was confirmed prior to running the tests. EnCase was used to acquire the image, producing both MD5 and SHA-1 hashes (Table 2).

Image 1 was obtained by plugging the flash drive directly into the USB port of the forensic workstation, after setting the Windows registry key to write-block the USB port. After changing the Registry key, the forensic workstation was rebooted and the USB write-block functionality was tested prior to plugging the flash drive into the workstation. The image was acquired and hash values calculated using EnCase.

Image 2 was obtained by plugging the flash drive directly into the USB port of the forensic workstation, leaving the Windows registry key related to the USB port's write capability at its normal setting. After changing the key, the forensic workstation was rebooted and the USB write-enable capability was tested prior to plugging the flash drive into the workstation. The image was acquired and hash values calculated using EnCase.

Table 2
USB Image Hash Values

USB Image 0 (hardware write-blocker)	
MD5	1571CFD2F1B5E9F00FAAFCC266C85EE0
SHA-1	7EE9467068D99795A43699B7E5CBDC53A7EA55EC
USB Image 1 (software write-blocker)	
MD5	1571CFD2F1B5E9F00FAAFCC266C85EE0
SHA-1	7EE9467068D99795A43699B7E5CBDC53A7EA55EC
USB Image 2 (no write-blocker)	
MD5	1571CFD2F1B5E9F00FAAFCC266C85EE0
SHA-1	7EE9467068D99795A43699B7E5CBDC53A7EA55EC

3.2 HDD Results

As the information in Table 1 shows, the hardware and software write-blocking function worked equally well in this test, in that the hash values of both Image 0 and Image 1 of the HDD matched. The hash value of Image 2 (no write blocking) differed, indicating that there were differences between that image and the write-blocked images.

Subsequent examination of the image files showed a number of differences between Image 2 and Images 0/1. First, Image 2 contained an extra file, *C\System Volume Information\MountPointManagerRemoteDatabase* that was not present on Images 0/1. Second, 358 individual files hashes were different on Image 2 compared to Images 0/1. It is noteworthy that all but one of the changed hashes were assigned to folders that generally do not have hashes; the one modified file was *D\$Extend\$UsnJrnl·$J*, which is a file that records when and what modifications are made to files and folders within Windows.

3.3 USB Flash Drive Results

As the information in Table 2 shows, all of the images of the flash drive yielded the same hash value. The individual file hashes were also examined to verify that there were no discrepancies.

3.4 Test Universe and Summary

Table 3 summarizes the test scenario in terms of forensic variables and parameters (based on Figure 1) and the results reported above

Table 3
Test Universe and Result Summary

	Hard Disk Drive			**Flash Drive**		
Device Interface	SATA			USB		
File System	NTFS			FAT32		
Workstation OS	Windows 7					
Forensic Software	Encase					
	Image 0	*Image 1*	*Image 2*	*Image 0*	*Image 1*	*Image 2*
Connection	SATA Bridge to USB			USB Bridge to USB	USB port	
Write-block	Hardware	Software	None	Hardware	Software	None
Hash (MD5)	F8BE…80CD		CF45…EB03	1571…5EE0		

As noted earlier in this paper, there are so many choices and options in the test parameters that our study cannot be used to draw overly generalized conclusions (Lyle, 2012). The test cases do, however, provide a framework for further research.

4. CONCLUSION

This study examined two imaging scenarios testing the results of imaging without a write blocker. In one case, no changes occurred to evidentiary media; in the second case, only minimal changes were made but not to user files where probative information - both incriminating and exculpatory - is likely to be found. Since the study looked at only two out of thousands of possible scenarios, additional, deeper studies with other devices and configurations are certainly in order so that a better understanding can be gained about the broad impact of inoperable or failed write-blockers.

It is also interesting to address the concept of repeatability in conducting forensic data acquisitions in the absence of write-blockers. While two forensic images acquired from the same original media without use of a write-blocker will likely yield different hash values (i.e., digital signatures), this difference alone does not preclude a scientifically repeatable finding. Even with the different media hash values and differences in system attributes (such as last accessed date), repeatable findings remain intact for the content of stored files, file slack, the overwhelming majority of unallocated space, and unused space. In fact, the individual hash values of stored files remain identical when the images are compared. Therefore, a forensic examiner may present evidence of data contained within a stored file or the file slack associated with the file, and another forensic examiner will be able to duplicate these findings even if both examiners created images from the original media without using write-blockers. The important lesson to learn is that differences in media hash values do not, by themselves, imply contamination of data.

The intent of this study is *not* to suggest that the use of write-blockers is unnecessary but rather to test the assertion that the absence of effective write blocking causes insurmountable challenges to the veracity - and value - of digital evidence. While our test results show that some changes are made to digital media when write-blocking is not employed (or, presumably, when a write-blocker fails), they also show that the original evidence is not tainted beyond use and, certainly, no additional content has been added to the medium. Thus, the results suggest, if a party objects to the introduction of a forensic image solely because of the lack of write-blocking, the burden of proof is on the objector to show how the value of the evidence has been diminished.

This study is intended to help inform the dialogue about what is and is not best practices in digital forensics. There are many cases when write-blocking cannot be effectively used when analyzing or examining digital media - such as is sometimes the case with mobile devices, or when collecting volatile data or random access memory (RAM) - and practitioners need to understand the true impact of the absence of write blocking so that we can better defend our processes and examinations.

ACKNOWLEDGEMENTS

The authors would like to thank a senior project team of undergraduate students from Cal Poly Pomona - Willen Lui, David Hughes, Joseph Mickelson, and Gurpreet Dhillon - for their work on this project. We would also like to thank external reviewers for their valuable comments.

REFERENCES

Carlton, G.H. (2007). A Protocol for the Forensic Data Acquisition of Personal Computer Workstations. UMI 3251043. Ann Arbor, MI, ProQuest.

Federal Rules of Evidence (FRE). (2013, December 1). The Committee of the Judiciary, House of Representatives.

Washington, D.C.: U.S. Government Printing Office. Retrieved from http://judiciary.house.gov/?a=Files.Serve&File_id=5334E54F-12CC-44B1-A0BC-697E8E29BD15

Forensic Focus. (2010, May 11). Connecting a USB device without a write-blocker. Discussion thread. Retrieved from http://www.forensicfocus.com/Forums/viewtopic/t=5809/

Henry, P. (2009, September 12). Best Practices in Digital Evidence Collection. SANS DFIR. Retrieved from http://digital-forensics.sans.org/blog/2009/09/12/best-practices-in-digital-evidence-collection/

Lyle, J. (2012, November 30). Computer Forensics Tool Testing. In Forensics@NIST 2012. Retrieved from http://www.nist.gov/oles/upload/5-Lyle_James-CFTT.pdf

National Institute of Standards and Technology (NIST). (2001, November 7). General Test Methodology for Computer Forensics Tools, version 1.9. U.S. Department of Commerce. Retrieved from http://www.cftt.nist.gov/Test Methodology 7.doc

National Institute of Standards and Technology (NIST). (2003, September 1). Software Write Block Tool Specification & Test Plan, version 3.0. U.S. Department of Commerce. Retrieved from http://www.cftt.nist.gov/SWB-STP-V3_1a.pdf

National Institute of Standards and Technology (NIST). (2004, May 19). Hardware Write Blocker Device (HWB) Specification, version 2.0. U.S. Department of Commerce. Retrieved from http://www.cftt.nist.gov/HWB-v2-post-19-may-04.pdf

National Institute of Standards and Technology (NIST). (2005, March 21). Hardware Write Blocker (HWB) Assertions and Test Plan, draft 1 of version 1.0. U.S. Department of Commerce. Retrieved from http://www.cftt.nist.gov/HWB-ATP-19.pdf

Nelson, B., Phillips, A., & Steuart, C. (2009). *Guide to Computer Forensics and Investigations*, 4th ed. Boston: Course Technology.

Scientific Working Group on Digital Evidence (SWGDE). (2013, September 14). Best Practices for Computer Forensics, version 3.0. Retrieved from https://swgde.org/documents/Current Documents/2013-09-14 SWGDE Best Practices for Computer Forensics V3-0

U.S. v. Labuda. (2012, April 11). Case #2:10-20066, U.S. District Court (TN-W). Retrieved from http://infosecusa.com/cases/us-v-labuda