Secure State Deletion: Testing the efficacy and integrity of secure deletion tools onSolid State Drives

Michael Freeman
Edith Cowan University

Andrew Woodward
Edith Cowan University

DOI: 10.4225/75/57b2875c40cce
7th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2009.

Secure State Deletion: Testing the efficacy and integrity of secure deletion tools on Solid State Drives

Michael Freeman and Andrew Woodward
secau – Security Research Centre
School of Computer and Security Science
Edith Cowan University
mfreema0@student.ecu.edu.au
a.woodward@ecu.edu.au

Abstract

The research aimed to determine the efficacy and integrity of several hard-drive disk deletion tools on solid state drives (SSDs). SSDs contain new technologies such as wear-levelling and device under provisioning to provide efficient functionality and speed for data management, but the same technologies may also provide obstacles to ensuring that all information is fully removed from the drive. Furthermore SSDs stores files in 4KB pages, yet data can only be deleted in 512KB blocks. This function uses the disk controller to remove all the pages from the block a file is being deleted from, storing the pages in a disk controlled cache. Once the whole block has been reset, the valid data is retrieved from the cache and replaced on an available block. The reset block is added to the SSDs free space. The specific purpose of this paper was to discover if any data was recovered, especially from the disk controlled cache while testing various tools and methods for their effectiveness of securely wiping data off SSDs. All tools except the GNU core utility DD left some file information which was recovered, though none of the recovered files was loadable. Additionally, the paper introduces the concept of the TRIM functionality and provides a baseline further research into this feature. Finally, a comparison of methods for securely deleting Solid State Drives is provided.

Keywords

Solid state drive, SSD, dd, acquisition, secure erasure, scalpel, TRIM, eraser, forensics, file carving

INTRODUCTION

Solid state drives (SSD) are gaining market share as the technology becomes more stable and as prices fall (Monroe & Unsworth, 2009). An additional driver for increased uptake of these drives is the release of Windows 7, which contains native support for solid state drives (Silver *et al*, 2009). Unlike common hard-drives which use an electrical stepper motor, a platter of cylindrical disks and read/write heads positioned on arms between the platters, SSDs use NAND flash memory (Shimpi, 2009b). As a result of this electrical rather than mechanical storage, these drives produce no sound and very little heat. One of the biggest differentiators from a digital forensics point of view is that they store data differently than mechanical drives, as is illustrated in this paper.

Since SSDs store data differently to common platter-based hard drives, they also delete data using different methods. Instead of new data being written over the top of old data which has been flagged as free space on a platter-based HDD, a SSD needs to reset the cells that contain the data to be deleted. Only then does the disk controller index these cells as free space. SSDs utilise a disk controller, which uses techniques such as wear-levelling and a cache to delete data. The principle purpose of this research is to discover if any data or files are recoverable from a SSD after it has been securely deleted, possibly still residing in the cache. A secondary purpose is to test a number of secure deletion techniques and tools on a SSD to see how effective they are at securely wiping information from a SSD. Thirdly, the paper introduces the concept of the TRIM functionality and creates a baseline for further research into this feature.

DATA STORAGE AND DELETION ON SOLID STATE DRIVES

As was mentioned in the introduction of this paper, solid state dives store and handle data very differently than mechanical drives. This mostly has to do with the fact that mechanical drives are optimised to for read and write speeds based on a rotating mechanism and the associated geometry. This rotation introduces the concept and subsequent delay known as seek time, and access methods and algorithms are optimised to minimise this time. With solid state drives, data can be effectively be accessed with a seek time approaching zero, as there is no delay whilst the required address on the platter is located through rotation and movement of the read / write head. The following sections outlines data is stored and deleted on solid state drives.

How Solid State Drives Store Data

NAND flash memory at its basic construct is one transistor per cell. A charge is applied to the cell to change its state from a ‘1’ to a ‘0’ (Gerbarg, 2009). The smallest groups of cells recognized are called pages, typically 4 kilobytes (KB) in size. Files are typically written to an SSD in pages. Pages are then grouped into blocks, commonly 128 pages to a block (512KB). Blocks are the smallest structure that can be deleted on a Solid State Drive (SDD). Typically 1024 blocks constitute a plane (512 megabytes (MB) and multiple planes reside on a single NAND flash die. Finally, multiple die may be embedded inside a single Integrated Circuit. (Shimpi, 2009a)

SSDs are also available as Single Level Cells (SLC) or Multi-Level Cells (MLC). A SLC stores one bit or two states per cell, representing a 1 or 0. MLCs typically store two bits or four states per cell (mp3Car.com, 2009).

The NAND flash memory is allocated and indexed by the disk controller. Controllers translate instructions from the host operating system and file allocation table to the physical NAND media, such as standard ATA commands of read /write into flash read / writes (mp3car). ATA views a SSD as a series of Logical Block Addresses (LBAs) (Gerbarg, 2009). The controller determines how the LBAs map to the NAND flash pages (Shimpi, 2009c). Additionally the disk controller can handle 6 to 12 NAND die’s simultaneously, which means the controller can read / write to more than one NAND die at a time (mp3Car.com, 2009)

Since SSDs has a MTFB (mean time between failure), typically of 10,000 -100,000 writes per cell (mp3car), it is important that all the available NAND Flash memory is evenly utilized. By writing to a certain section repeatedly will possibly reduce the life of the device. Hence SSDs use wear-levelling, which intends to write data to every available page before a single page is re-used or more importantly, deleted. (“Intel Solid State Drives”, n.d) Wear-levelling is an algorithm run by the SSD controller that evenly spreads data across all the pages of the NAND flash, for the purpose of increasing the overall endurance of the whole drive beyond the endurance of a single NAND cell. The SSD controller keeps a track of all free-space within the drive and will not actually delete any data until all available free pages have been utilized, even if the files have been deleted within the file system. This is what gives SSDs their amazingly quick write and delete speeds.

Additionally, SSD’s commonly under provision the amount of storage media available to the host operating system. An important difference between HDDs and SSDs, is that a 32 gigabyte (GB) SSD contain 32GB of NAND flash cells or 34,359,738,370 (32x1024^3) bytes. Whereas platter-based HDD makers believe 32GB means 32,000,000,000 bytes (or approximately 29.8GB). Operating systems still recognise a SSD as having the same amount of available space as a HDD (29.8GB in the case of a 32GB SSD) and the SSD controller assumes control of the additional space as available spare space (2.2GB in the case of a 32GB SSD) (Shimpi, 2009e). The SSD controller utilises this additional free space to save files, reducing the need for the deletion of blocks and keeping the drive at peak performance.

Data Deletion on solid state drives

A Solid State Drive (SSD) controller only begins to delete data when it starts to run out of free space or pages which do not contain any data, either referenced or deleted by the host operating system. Pages that no longer contain data referenced by the host operating system are marked as invalid by the disk controller (Gerbarg, 2009). This is when the amazing write speeds of SSDs decrease, as the SSD needs to create free space before writing the new data to the storage medium.

As mentioned before, files are saved on a SSD in pages, typically 4KB in size. Pages are then grouped into blocks. While data can be written to individual pages, data can only be deleted in blocks (Shimpi, 2009b). Multiple files (stored in pages) can be contained in a block, yet a block is the smallest element that can be deleted, thus special techniques have been developed to overcome this problem.

SSDs utilize a cache when they need to create free space. The location of the cache is typically on a die within the SSD (mp3Car.com, 2009). As new data is written to the SSD, the disk controller needs to create more free space or pages. To do this, the new data is written to free pages. The data from the block containing the most invalid page/s to be deleted is placed into the cache and the all the cells in the block are reset to ‘1’ (Gerbarg, 2009). Then in the cache, the deleted pages are removed from the block, leaving only the valid pages. Finally, the block of data is transferred from the cache to an available block. The newly reset block is added to the controllers free space (Shimpi, 2009d).

As stated in the introduction, the main purpose of this research is to discover if any data or files are recoverable from a SSD after it has been securely deleted, possibly still residing in the cache. A secondary purpose is to test a number of secure deletion techniques and tools on a SSDD to see how effective they are at secure deleting information from a SSD. A third purpose of this paper is to introduce the Trim functionality and create a baseline for further research into this feature.

The TRIM Function

Another technique utilised by SSD controllers make the resetting of blocks more efficient is the TRIM Functionality. TRIM works differently from the method described above as the actual TRIM command is an ATA command made by the host operating system, which is recognised by the disk controller. Hence when a file is deleted within the host operating system, a TRIM command is sent to the disk controller with the associated LBAs (Logical Block Addresses) for the deleted file, informing the controller those LBAs are no longer needed and maybe cleaned as per the process above. The SSD then resets those blocks which become additional free space. The repercussions of the TRIM functionality are that the write performance of the SSD should not decrease to the same extent (Shampi, 2009f).

While TRIM functionality is becoming more prevalent, now available in operating systems such as Linux and Windows 7, SSD manufacturers are slowly introducing firmware and controllers that will utilize such a feature. At the time of testing for this paper, the only manufacturer producing SSD's with controllers supporting TRIM functionality were Intel badged SSD's. Since then other manufacturers, such as OCZ, Crucial and SuperTalent have released updated firmware with TRIM functionality for some of their SSDs.

Unfortunately, access to a SSD with TRIM functionality was not available for this research.

Secure File Deletion

Secure deletion or disk wiping is a method used to actually remove data from a physical storage medium. When a file is deleted by an Operating System, information of that file such as where it is located on the physical device in the file allocation table is removed. But the data is still physically written to storage medium, until the same physical medium is written over with new information (Mallery, 2006). The same situation applies to SSDs.

Secure Deletion or disk wiping tools write all 0's, 1's or pseudo-random data (Mallery, 2006) to all or most of the drive depending on the tool used. Some tools write data to the whole drive, while others target certain sections of the drive such as unallocated or free space. These are sectors or blocks which are not currently allocated to any data by the operating system or file allocation table, which may contain recoverable data (Mallery, 2006). A secondary purpose of this research is to test a number of secure deletion techniques and tools on a SSD to see how effective they are at securely deleting information from a SSD.

METHODOLOGY

This research included two tests, with only a subtle difference between the two tests. In the first test, a SSD was filled with data to the maximum size as allowed by the operating system. Some data was deleted, and more added. In the second test, the same SSD was filled to the maximum size allowed by the operating system. Then a large amount of data was deleted from the SSD and a significant amount of data was added to the SSD, so that the total amount of data written in the experiment to the drive was larger than the total advertised size of the SSD.

For test one, the DD application was used on the SSD and all bits were written to zero and then was filled to the SSDs referenced capacity with a range of files, both in file type and size. A number of files were then deleted from the drive and removed from the Trash. Additionally, a single large file was added to the drive.

The drive was then wiped using the process of a specific secure deletion or wiping tool. Then again using the DD command, a bitwise image was recovered off the drive. Finally, the recovered image was input into Scalpel, which returned any recovered files. This process was repeated for each specific Wiping Tool.

For test two, this process was repeated again, with the only significant change being the amount of data saved and deleted from the SSD. Since SSDs utilise wear-levelling and under provision the available storage media to the operating system, the disk controller keeps a percentage of the total storage area un-index by the host operating system to keep the SSD running at peak performance. In this test the SSD was filled to the capacity of the drive, and then a significant amount of data was deleted and more data added to the drive to fill all the NAND flash memory to over the advertised full size of the drive.

The purpose for deleting and adding additional data to the SSD for each test was to induce the use of the cache for creating free space on the drive. The intent for the two separate tests was to discover if there were any significant differences in the recovery of files between filling the SSD with data to the referenced capacity and filling the SSD with data over the advertised size of the drive.

TEST METHODOLOGY

The test environment consisted of a single quad core computer, with 4GB of random access memory, running both Windows 7 Professional and Ubuntu 8.10 32 bit versions. A 32GB PQI SATA II 2.5 inch Solid State Drive was used as the target drive for the experiments and testing. The model number of the drive was S525 and the firmware version was 02.10104. The computer was restarted (cold booted) when moving from one operating environment to the other.

Windows 7 was utilised for formatting the SSD with NTFS and transferring / deleting files on the SSD. Once the files had been deleted from the SSD, they were also removed from the Recycle Bin. Ubuntu was utilised for zeroing the SSD between tests using the DD command, recovering bitwise images of the SSD after the wiping tools and finally running the file carving tool scalpel. The SSD was automatically mounted on both Operating Systems as SSD_Test.

A range of secure deletion and wiping tools were utilised during the tests. The four tools employed a range of methods to wipe the data and ran on either the Windows 7 or Ubuntu 8.04 Operating Systems. These tools were:

DD (GNU core utility)

The DD command is a terminal run application which is part of the GNU core utilities, installed on most distributions of Linux. This application can be used to create bitwise images, copy files and convert raw data into either zeros or random data. **(**MacKenzie, 2009)

The following command was used in an Ubuntu 8.04 terminal window to securely wipe the SSD for this research (MacKenzie, 2009):

```
dd if=/dev/zero of=/dev/sdb bs=1M
```

`if=/dev/zero` - means to overwrite the device with null bytes.

`of=/dev/sdb` – means the device to overwrite will be the Solid State Drive sdb.

`bs=1M` – block size of 1 Binary Mb or 1,048,576 bytes.

Eraser (version 5.8.7)

Eraser is a security tool created for Windows. Eraser is able to securely wipe individual files and folders, as well as securely wipe unallocated or free space as well as slack space. This space includes sections of a drive that do not have a file allocated in the file system, but may still contain recoverable data (Mallery, 2006).

For this research, the secure wiping of unused space was utilised. After all the files had been added to the SSD, they were deleted by selecting all files, right clicking and selecting the delete option. The drive SSD_Test then displayed as empty. Finally Eraser was tasked to securely wipe all the free space from SSD_Test, the SSD.

Wipe

Wipe is a Linux based secure file wiping tool that writes random data. Wipe typically uses more than one pass to remove files but for this research only one pass was used, since the other tools only used one pass.

Wipe was installed on Ubuntu 8.04 using the following command in a terminal window:

```
sudo apt-get install wipe
```

And the following command was also given in the terminal window

```
sudo wipe –q –Q 1 /dev/sdb
```

`-q` = quick wipe option, default four passes of random data (man2html, 2003)

`-Q 1` = Set number of pass for quick wiping as 1 (man2html, 2003)

`/dev/sdb` = SSD to be wiped

SDelete

SDelete is a command line application for use in Windows, made available by Microsoft from their TechNet site **(**Russinovich, 2006).It is a secure delete application that can be used to wipe existing files as well as any file data in unallocated space, such as non-referenced but recoverable data. SDelete implements the US Department of Defence Standard (DOD 5220.22-M) for cleaning and sanitizing data. An known issue with SDelete is that while it securely deletes file data, it misses the file names located in free space. (Russinovich, 2006**)**

For this research, the secure wiping of unused space was utilised. After all the files had been added to the SSD, they were deleted by selecting all files, right clicking and selecting the delete option. The drive SSD_Test then displayed as empty.

SDelete was downloaded and the unzipped contents were place in s folder on the C drive called SDelete. Next a command prompt was opened and navigated to the C:\SDelete folder. The following command was then input into the command prompt: ("Data Clean Up: Secure Delete", n.d**)**

```
sdelete –z d:
```

`-z` = wipe free space

`d:` = the mounted location of SSD_Test in Windows 7

Scalpel

Scalpel is a fast file carver based on the application foremost 0.69. It is file-system independent, able to carve files that match a database of file type headers and footers and extract those files from Windows, Linux and raw partitions ("Scalpel: A Frugal, High Performance File Carver", n.d).

Scalpel was installed on the Ubuntu 8. 04 Operating System using the following command in a terminal window:

```
        sudo apt-get install scalpel
```

Scalpel was run on each image using the following command in a terminal window (wipingtool was substituted for the wiping tool used before the image was created):

```
sudo scalpel /wipingtool.img –o /home/me/Desktop/wipingtool/
```

-o = location of Scalpel output..

Each time scalpel was run, a number of folders were created in the output location containing the recovered files.

Test One

The 32GB SSD, once formatted with NTFS, showed 29.8GB of available space. The SSD was filled with a variety of files ranging in file-type and size. These are shown in Table 1 in summary form:

Table1: File types and size added to SSD_Test1

File Type	Extension	Number of files	Total Size	Smallest File	Largest File
Video Files		74	21.8GB	1.6MB	1.4GB
	.mpg	10			
	.wmv	4			
	.avi	60			
Word Files	.doc	109	23.3MB	20KB	8.9MB
Zip Files	.zip	5	1.38MB	48KB	590KB
Text Files	.txt	11	48KB	1KB	6KB
Powerpoint Files	.ppt	14	31MB	166KB	21.129MB
Image Files		318	196MB	2KB	5.397MB
	.bmp	1			
	.gif	1			
	.jpg	316			
PDF Files	.pdf	103	715MB	8KB	648.93MB
Music Files		1068	6.91GB	28KB	45.13MB
	.wma	99			
	.mpg4	244			
	.mp3	725			
Excel Files	.xls	7	336KB	14KB	102KB

A number of files were then deleted from the SSD and also from the Trash Can again summarised in (Table 2).

Table 2: Files deleted from SSD_Test

Filetype	**Size**		
.avi	1.15GB	263MB	
.pdf	27.5KB	15MB	80KB
.mp3	14MB		
.wma	5.9MB		
.txt	1KB	1KB	
.doc	22KB	32KB	

A 1.1GB .avi file was then saved to the SSD. After each tool was run on the SSD, a bitwise forensic image was created of the contents of SSD_Test using the Linux dd application with the following command (wipingtool was changed with the name of the tool used):

```
sudo dd if=/dev/sdb of=/home/me/Desktop/wipingtool.img
```

The resulting image was then carved by Scalpel to discover if any files were recoverable from the device.

Test Two

In test two the same tools, methods and SSD were utilised. Additionally the same original 29.8GB of data were added to the drive. The following files were than deleted from the solid state drive, totalling over 3.6GB.

Table 3: Files deleted from SSD_Test

Filetype	**Size**		
.avi	1.15GB	263MB	
.pdf	27.5KB	15MB	80KB
.mp3	14MB	45.1MB	44MB
.wma	5.9MB		
.txt	1KB	1KB	
.doc	22KB	32KB	
.jpg	5.4MB	3.9MB	
.avi	698MB	1.36MB	

And the following files were finally added to the SSD, totalling 3.39GB. This amounted, when added to the original data added to the drive, total 33.19GB, more than the advertised total size of the SSD of 32GB.

Table 4: Additional files added to SSD_Test

Filetype	**Size**
.avi	1.11GB
.avi	703MB
.wmv	1.59GB

Once again, after each tool was run on the SSD, a bitwise image was created of the contents of SSD_Test using the Linux DD application with the following command (wipingtool was changed with the name of the tool used):

```
sudo dd if=/dev/sdb of=/home/me/Desktop/wipingtool_2.img
```

The resulting image was then carved by Scalpel to discover if any files were recoverable.

RESULTS

For both tests One and Two, a number of files were carved by Scalpel from each tool except DD. Scalpel was unable to carve any files from the DD images from either experiment. The results of these experiments are shown below in Tables 5 and 6.

Table 5: Files found as a result of carving using the scalpel tool in Test One. Also indicated was whether the file could be opened

Tool	File Type	No. Found	Loadable	Total Size
dd		0	0	0MB
eraser	bmp 4-0	20	0	1.90MB
	fws 10-0	1000	0	3.72GB
	fws 10-1	893	0	3.32GB
	mpg 9-0	1	0	6MB
	png 3-0	8	0	84.4MB
	ra 18-0	6	0	5.72MB
	ra 19-0	2	0	1.9MB
	tif 5-0	6	0	1.11GB
	tif 6-0	8	0	1.48GB
	wpc 13-0	1000	0	957MB
	wpc 13-1	914	0	871MB
	zip 20-0	8	0	840KB
sdelete	bmp 4-0	7	0	683KB
	mpg 9-0	10	0	26.5MB
wipe	fws 10-0	1000	0	3.42GB
	fws 10-1	895	0	3.33GB
	png 3-0	9	0	83.5MB
	ra 18-0	7	0	6.67MB
	ra 19-0	6	0	5.72MB
	tif 5-0	7	0	1.3GB
	tif 6-0	13	0	2.42GB
	wpc 13-0	1000	0	957MB
	wpc 13-1	868	0	827MB
	zip 20-0	11	0	641KB

Significantly, as shown in both tables, while a number of files were carved, none of the carved files for any tools, in either test were loadable, hence opened or viewed. And while the numbers found and total size of each file type carved was different between the two tests, there were no noteworthy differences between the files carved from each tool between test one (Table 6) and test two (Table 7).

DISCUSSION AND CONCLUSION

It appears that files cannot be recovered after a SSD has been securely wiped, even from the cache function used by SSDs to delete data to create free space.

Furthermore, the most effective way to securely wipe a Solid State Drive so that no data can be carved or recovered from the device is to use the DD application. As shown in both tables (table 6 & 7), no files were carved from the DD image in either test. While DD can also be utilised in Windows environments, the second most effective tool was SDelete, which works easily and effectively in the Windows Operating System environments. Scalpel was able to carve only 17 files totalling just over 27MB from the SDelete image in test one (table 6) and 30 files totalling just over 40MB in test two (table 7).

Last were Eraser and Wipe. From the Eraser image in test one Scalpel carved 3866 files, totalling 13925MB. From the test two image Scalpel carved 384 files, totalling just under 11121MB. From the Wipe image in test one Scalpel carved 3816 files, totalling just over 12350MB. From the test two image Scalpel carved 3844 files, totalling over 9908MB. While no carved files were totally recovered, a large amount of data was carved from each image.

Table 6: Files found as a result of carving using the scalpel tool in Test One. Also indicated was whether the file could be opened

Tool	File Type	No. Found	Loadable	Total Size
dd		0	0	0
eraser	bmp 4-0	20	0	1.9MB
	fws 10-0	1,000	0	3.72GB
	fws 10-1	851	0	3.16GB
	mpg 9-0	2	0	23MB
	png 3-0	11	0	109MB
	ra 18-0	12	0	11.4MB
	ra 19-0	8	0	7.62MB
	tif 5-0	7	0	1.3GB
	tif 6-0	5	0	953MB
	wpc 13-0	1000	0	953MB
	wpc 13-1	924	0	881MB
	zip 20-0	9	0	1.02MB
sdelete	bmp 4-0	8	0	781KB
	mpg 8-0	8	0	5.87MB
	mpg 9-0	14	0	34.8MB
wipe	fws 10-0	1000	0	3.72GB
	fws 10-1	900	0	3.35GB
	png 3-0	6	0	59.2MB
	ra 18-0	12	0	11.4MB
	ra 19-0	2	0	1.90MB
	tif 5-0	5	0	953MB
	tif 6-0	8	0	1.48GB
	wpc 13-0	1000	0	953MB
	wpc 13-1	900	0	858MB
	zip 20-0	11	0	847KB

Solid State Drives (SSD) store data on NAND flash cells compared to platter-based Hard Drives (HDD), which means the way data is read, written and deleted from a SSD is also different. SSDs utilise a SSD controller, which allows data to be read and written to multiple die simultaneously and coordinates the removal of invalid data from NAND flash memory only when more free space is required. Furthermore, SSDs use techniques such as wear-levelling and device under provisioning to ensure all NAND flash cells are used and enough free space is available.

The main purpose of this paper was to discover if files could be recovered from securely deleted SSDs. While files could be carved by Scalpel from the SSD after secure deletion by most tools, none of the carved files were loadable, or fully recovered. Furthermore, the secondary purpose was to evaluate the efficiency of several secure deletion tools and methods. The most effective was the GNU core utility DD, which removed all the data from the drive, while second was the Windows application SDelete. Finally Wipe and Eraser were the least efficient, with thousands of carved files each.

Future research could be made into how much information was recoverable after being securely deleted or wiped by each tool using more intrusive methods such as viewing each recoverable file in a hex-editor. Some meaningful information may possibly be recovered using such methods. Such research could further verify which tools and methods are the most effective for securely wiping Solid State Drives.

Additional further research could also be conducted into the TRIM function. This research has now created baselines which can be used to test SSDs with TRIM functionality, including discovering if TRIM functionality affects or improves the secure deletion of information off Solid State Drives.

REFERENCES

Data Clean Up: Secure Delete [n.d]. Georgia Institute of Technology. Retrieved September 12, 2009 from http://datacleanup.gatech.edu/pdf_docs/secure_delete_how_to-sdelete1.pdf

Gerbarg, L. (2009, August 4). From write() down to the flash chips. /dev/why!?! Retrieved October 12, 2009 from http://devwhy.blogspot.com/2009/08/from-write-down-to-flash-chips.html

Intel Solid State Drives [n.d]. Retrieved October 26, 2009 from http://www3.intel.com/cd/channel/reseller/asmo-na/eng/products/nand/feature/index.htm

MacKenzie, D. (2009, September 29). GNU Coreutils. Free Software Foundation, Inc. Retrieved October 26, 2009 from http://www.gnu.org/software/coreutils/manual/coreutils .pdf

Mallery, J. (2006, June 12). Secure File Deletion: Fact or Fiction. SANS Institue. Retrieved October 12, 2009 from http://www.sans.org/reading_room/whitepapers/incident/secure_file_deletion_fact_or_ fiction_631?show=631.php&cat=incident

man2html (2003, February 18). WIPE. Retrieved October 24, 2009 from http://abaababa.ouvaton.org/wipe /wipe.1. html

Monroe, J. & Unsworth, J. (2009). Dataquest Insight: Evolving Technologies enhance new generations of PC-grade SSDs. Retrieved 20th October from Gartner Reports

Mp3Car.com (2009, August). Everything you need to know about Solid State Disks(SSD). Retrieved October 12, 2009 from http://www.mp3car.com/vbulletin/general-hardware-discussion/134989-everything-you-need-know-about-solid-state-disks-ssd.html

Russinovich, M. (2006, November 1). Microsoft TechNet – SDelete v1.51. Microsoft Corporation. Retrieved October 24, 2009 from http://technet.microsoft.com/en-us/sysinternals/bb897443.aspx

Scalpel: A Frugal, High Performance File Carver. [n.d]. Retrieved October 26, 2009 from http://www.digitalforensicssolutions.com/Scalpel/

Shimpi, A. (2009a, March 18). The Anatomy of an SSD. In The SSD Anthology: Understanding SSDs and New Drives from OCZ. Retrieved September 12, 2009 from http://www.anandtech.com/storage/showdoc.aspx?i=3531&p=5

Shimpi, A. (2009b, August 30). A Quick Flash Refresher. In The SSD Relapse: Understanding and Choosing the Best SSD. Retrieved September 12, 2009 from http://www.anandtech.com/storage/showdoc .aspx?i=3631&p=2

Shimpi, A. (2009c, August 30). Live Long and Prosper: The Logical Page. In The SSD Relapse: Understanding and Choosing the Best SSD. Retrieved September 12, 2009 from http://www.anandtech.com/storage/showdoc .aspx?i=3631&p=3

Shimpi, A. (2009d, August 30). The Cleaning Lady and Write Amplification. In The SSD Relapse: Understanding and Choosing the Best SSD. Retrieved September 12, 2009 from http://www.anandtech.com/storage/showdoc .aspx?i=3631&p=3

Shimpi, A. (2009e, August 30). Understanding Spare Area (or Why My 80GB Drive Appears as 74.5GB). In The SSD Relapse: Understanding and Choosing the Best SSD. Retrieved September 12, 2009 from http://www.anandtech.com/storage/showdoc .aspx?i=3631&p=7

Shimpi, A. (2009f, August 30). The Instruction That Changes (almost) Everything: TRIM. In The SSD Relapse: Understanding and Choosing the Best SSD. Retrieved September 12, 2009 from http://www.anandtech.com/storage/showdoc .aspx?i=3631&p=8

Silver, M. A., MacDonald, M., Kleynhans, S., Skorupa, J. & Orans, L. (2009). Reasons to care about Windows 7, and reasons not to. Retrieved 20th October 2009 from Gartner Reports

COPYRIGHT