1-1-2011

An evaluation of data erasing tools

Thomas Martin
Khalifa University (KUSTAR), United Arab Emirates

Andrew Jones
Edith Cowan University

DOI: 10.4225/75/57b2c01440cef
9th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, 5th -7th December 2011

AN EVALUATION OF DATA ERASING TOOLS

Thomas Martin[1] and Andrew Jones[1,2]
[1]Information Security Research Group
Khalifa University (KUSTAR), United Arab Emirates
[2]School of Computer and Security Science, Edith Cowan University,Perth, WA
thomas.martin@kustar.ac.ae, andrew.jones@kustar.ac.ae

Abstract

The permanent removal of data from computer disks has always been problematic. This has been due, in part, to the lack of availability of tools, and in part due to the misperception by the user that when a file is deleted it is destroyed and cannot be recovered and that when a disk is formatted, the data is destroyed. In this paper, we examine a number of the commonly available tools to determine how effectively they function and whether they achieve the aim of the effective destruction of data

Keywords

Disk file erasure tools

INTRODUCTION

Today, computer users have the ability and the need to store vast quantities of digital data. Terabyte computer drives are small and affordable and increasingly found in both the commercial and the home environment. Individuals and organizations are quick to fill these drives with images, movies, personal, corporate and customer data. But devices and the drives they contain are regularly sold, lost or stolen. Data often has a limited useful lifespan, after which its possession is a liability. The actions to remove data that are performed by the operating system are limited to freeing up space on the disk, not ensuring the data cannot be recovered.

A variety of products exist that claim to perform the secure deletion of data. These can either target individual files/folders or entire drives. A number of standards exist, the most common being US DoD 5220.22-M(E), and most of the standards use multiple overwrites to ensure that all data is overwritten and as a result cannot be recovered. An experiment was created to put 12 of the better known products to the test. The aim of the experiment was primarily to see if the data on the disk was erased correctly, but also to compare performance and determine if there were any identifying artefacts left by the individual processes.

PREVIOUS WORK

The study of secure data deletion was first started in (M.Slusarczuk et al, 1987). In (Gutmann, 1996), Gutmann argued that imprecise writing of data can mean that remnants of data remain upon overwriting. Normal disk circuitry is set up to ignore such variations, but more sophisticated equipment (such as magnetic force microscopy) could recover data even after it had been overwritten several times. He proposed a system of overwrites to make any such recovery impossible. Depending on which encoding is used, or if this is even known, the process takes between 10 and 35 overwrites.

(Wright & Kleiman, 2008) called into question some of the conclusions of Gutmann's work. From their experiments, they found that even a single overwrite can make recovery difficult. While individual bits may be recovered, the collection of large amounts of data is infeasible. This is due to the track density of disks, the history of multiple overwrites typical in most disks that are not pristine, and a number of other factors.

There are many different standards that deal with the secure deletion of data. These include: (Defense, 2006), (Richard Kissel, 2006), (DSD, 2008), (CESG), (CSEC, 2006), (NZSIT, 2008).

METHODOLOGY

The first step in the process was to create the baseline dataset - a collection of files that would be present on the disk before each eraser tool was used. The disk that was to be used in the experiment was 80GB in size (80,026,361,856 bytes). A variety of files to fill this disk was obtained from a number of locations, including Linux distributions (http://www.linux.com/directory/Distributions), documents from Project Gutenberg

(http://www.gutenberg.org/wiki/Main_Page), audio and video podcasts (http://revision3.com/), etc. A breakdown of the file types is given in Table 1.

Table 6 Contents of dataset by file type

	Number of files	Size of files (bytes)	% files	% size
.zip	34,175	7,840,054,030	23.95%	10.69%
.txt	32,658	12,485,041,529	22.89%	17.03%
.html/htm	32,556	997,986,644	22.82%	1.36%
.jpg	29,273	2,233,242,253	20.52%	3.05%
.png	9,671	332,134,802	6.78%	0.45%
.gif	3,171	57,848,611	2.22%	0.08%
.mp3	353	6,036,455,322	0.25%	8.23%
.jpeg	172	39,837,374	0.12%	0.05%
.pdf	153	24,977,994	0.11%	0.03%
.avi	43	4,468,614,994	0.03%	6.10%
.iso	13	34,656,677,888	0.01%	47.27%
.mov	5	3,083,433,409	0.00%	4.21%
.img	1	948,244,480	0.00%	1.29%
rest	434	105,952,541	0.30%	0.14%
Total	142,678	73,310,501,871	100.00%	100.00%

Tools

Prior to each individual experiment, the drive was erased using the Wipe Drive feature of FTK 3.2, the files were copied to the drive and an image taken (the disk was also imaged after the erasure for manual inspection and verification). Several tools were used in the analysis of each eraser.

- FTK
- Md5deep
- Scalpel
- Cmp
- Grep, scripts, batch files, custom written code and OS commands

FTK (http://accessdata.com/products/computer-forensics/ftk) is a commonly used forensics tool. As well as providing an initial view of hard disk images, it displays all files remaining, and provides searching and carving features.

Md5deep (http://md5deep.sourceforge.net/) is a command-line tool that allows MD5 digests to be taken recursively of all files within a given target directory, including the contents of all subdirectories. The .iso images were mounted and digests of the contained files taken, but the same was not necessary for the .zip files as these were already part of the dataset.

Scalpel (http://www.digitalforensicssolutions.com/Scalpel/) is a configurable tool for searching images for files and file fragments based on known file header/footer strings. For this experiment, Scalpel was configured to search for the following filetypes: gif, jpg, mpg, doc, dbx, idx, mbx, htm, pdf, pgd, wav and zip. Scalpel can search for other filetypes, but the list was constrained to decrease the number of false-positives (otherwise each execution would have created 100s of GB in false files based on coincidental matches). Notice that some of the searched for filetypes were not present in the dataset. These were included to give an idea on the rates of false-positives and to determine if any of the erasers were deliberately placing fake files on the disk as part of the erasing process. Version 1.6 of Scalpel was used in this experiment.

Cmp (http://www.gnu.org/software/diffutils/) is a command-line tool to compare two files and list all the differences byte-by-byte.

Grep is a command-line tool for searching files.

Several scripts and batch files were used over the course of the experiment. During each experiment, one batch file periodically executed the "tasklist" command to obtain the memory and CPU usage of the erasing tool. Nonzero.exe is a custom C++ program specifically written for this experiment. Many erasing tools will overwrite the data with a series of \x00 values (as the final pass where there are multiple overwrites). While this may seem to cover the complete drive, the human eye cannot hope to verify this. Nonzero scans the raw image data for any sequence of non-zero byte values (or any that deviates from a known pattern).

Each tool was evaluated in the exact same manner by repeating the same steps:

1. The dataset image was restored to the disk
2. A batch file was configured to capture the memory and CPU usage of the tool
3. The erasing tool was run using the 3-pass DoD standard (if available) and all other default settings on the entire drive/all files contained in the drive
4. Once the erasing tool completing its cycle, the disk was disconnected
5. The disk was connected via a write-blocker (Tableau Ultrablock II) and imaged (using AccessData FTK imager 2.7.0.33)
6. The image was imported to FTK for manual analysis (mainly looking for any obvious directory names, file names or recovered files)
7. The list of all files that FTK recovered was exported and their MD5 digest compared with the original dataset
8. The raw image was exported and scalpel run to carve known filetypes
9. The MD5 digests of all carved files were taken and compared to the dataset
10. If the drive appeared to be all \x00 (or a simple repeating pattern), nonzero.exe was run to confirm this/find any changes in the pattern

To elaborate on steps 7 and 9, significant use was made of md5deep. The MD5 digests of the dataset were stored in a text file, with each line containing the digest then the complete file path and name. This file was sorted. In steps 7 and 9, files were generated that may have come from the original dataset or may be random files. FTK automatically creates hash digests of all files identified, this was exported to a text file (again with the MD5 digest first and the file sorted). For the Scalpel generated files, md5deep was again used (and the file with the digests sorted). A perl script was then run to look for matches in the FTK/Scalpel generated files digests with the digests from the original dataset (matching is much easier and more efficient when the files are sorted by digests). As well as comparing the files using the digests, a sample of the carved files were manually inspected to check if any files were partially recovered, or if there were any discrepancies between the carved files and the originals.

ERASERS

In Table 2, a summary is provided of the 12 erasers selected for the experiment. The details given are the eraser name, type (i.e. disk or file eraser), version number and overwriting standard used. The final two columns relate to the driving force behind the creation of the tool. A distinction is made between not-for-profit (NFP) tools created by hobbyists/open source community and tools that are commercial (the tool itself is free but used to promote full version products or related services).

Table 7 Summary of tested Erasing tools

	Name	Type	Version	Standard	License	Other interests/products
1	Active @ KillDisk	Disk	5.2.3	Single pass	Commercial	Data recovery software + services, CD/DVD burner tools
2	BCWipe	File	4.01.23	DoD 5220.22	Commercial	Wipers for Unix, whole disk erasers, BestCrypt encryption tools
3	Blancco	Disk	4.1	DoD 5220.22	Commercial	Data erasure tools for PCs, servers, mobile devices and flash media
4	CBLData Shredder	Disk	1.0e	DoD 5220.22	Commercial	Data recovery from different media: hard drive, RAID, tape, laptop, zip/floppy drive, flash cards, etc.
5	DBAN	Disk	2.2.6	DoD 5220.22	NFP	Open source project on Sourceforge
6	DPWiper	File	1.1	DoD 5220.22	NFP	Personal project
7	Eraser	File	6.0.8.2273	DoD 5220.22	Commercial	Browser history eraser, multi-application server package, SPAM blocker and a .NET web farm clustering synchronization tool
8	File Eraser	File	5.7	DoD 5220.22	NFP	Startup manager tool
9	File Shredder	File	2.0	DoD 5220.22	NFP	Anti-virus, anti-forensics, privacy protection/password management tools.
10	Freeraser	File	1.0.0.23	DoD 5220.22	Commercial	Portable applications for USB drives
11	HardDrive Eraser	Disk	2.0	DoD 5220.22	NFP	Personal project
12	Wipetool	Disk	2.35 build 1178	Unknown	Commercial	Hard disk diagnostics and recovery utilities

ANALYSIS

Table 3 has a summary of the analysis done using FTK and Scalpel. The rest of this section comprises of a discussion of the meaning and significance of these results.

Table 8 Results from analysis of Erased disks

Tool	FTK Folder names	FTK Filenames	FTK files recovered	FTK file matches	Overwrite pattern	Scalpel Files	MD5 matches	Partial matches
Active @ Kill Disk	None	None	None	None	All zeros	None	None	None
BCWipe	Some	None	28,326	None	Random	170	1	None
Blancco	None	None	(15)	None	"B5" repeating	None	None	None
CBLData Shredder	None	None	None	None	Random	60	None	None
DBAN	None	None	None	None	All zeros	None	None	None
DPWiper	None	None	91	6	Random	126	None	None
Eraser	Most	None	3,255	6	Random	40	None	None
Fileeraser	None	None	3,075	6	Random	6333	94	None
File Shredder	Most	None	145,695	6	Random	59	None	None
Freeraser	Most	Most	145,687	6	Random	12	None	None
Hard Drive Eraser	None	None	58	None	1kb repeating	34	None	9
Wipetool	None	None	None	None	All zeros	None	None	None

Proven Erasure

Some tools overwrote the disk with a random pattern, while others used a fixed, repeating pattern. Where it seemed to be the latter, one of the custom-written tools was used to verify this. In the case of confirmation, no further analysis would be strictly necessary. It is impossible to recover any files from direct reading of an image that is 100% \x00 bytes (however, all steps were taken on all images for consistency).

The images resulting from Active @ Kill Disk, DBAN and Wipetool were all confirmed to consist entirely of \x00 bytes. The Blancco image consisted almost entirely of \xB5 bytes, the remaining 0.09% did not yield any interesting material, and none of the 15 files recovered by FTK matched any in the dataset.

Successful Erasure

No information about the original dataset was successfully recovered from the CBLDataShredder image. It did not have a repeating pattern that could be verified, but neither FTK nor Scalpel recovered any of the original data. Since none of the 60 files carved by Scalpel matched the dataset and a visual inspection did not reveal any matches, they can be assumed to be random coincidence, which is to be expected given the size of the drive. No argument is being made regarding the merit of fixed overwrites over random (or vice versa) in the last pass, there is merely a difference in what can be proven.

System Volume Information Files

FTK recovered the same 6 files from each images of DPWiper, Eraser, File Shredder and Freeraser (see Table 4 for the DPWiper files). Before discussing the significance of these results, it is necessary to cover some relevant aspects of probability. Given the problem of studying disks that (for the most part) are overwritten with random data, and looking for matches with the original files (of which there are great many), the possibility has to be considered that the original files may be coincidentally recreated. It can be shown that the length of the file is the important parameter in determining the likelihood of accidental matching. If this length is significantly greater than 10 bytes, then the probability of this occurring rapidly approaches zero. This means that the matches found are almost certainly recovery of the original files rather than coincidental recreations (the smallest file is 15 bytes).

Having shown that the files were recovered due to a failing in the erasure, the question is now how it happened. The answer is most likely related to the location of the recovered files, they were all originally present in the System Volume Information folder (not just in the case of DPWiper, but with the other three erasers as well). This folder is used by the Operating System to recover system files in the case of errors or crashes. It is typically unavailable to the user or running applications. This explains why erasing programs would have a problem

erasing these files completely (note that all four were file erasers rather than disk erasers). It also somewhat reduces the impact this might have on users relying on file erasers. Not every filetype may be saved in the System Volume Information folder. Batch files, icons and readme files were recovered, which are not generally used to store sensitive information. However, for secure use it would be important to know exactly what was saved in the System Volume Information folder (or to disable this functionality) before one could confidently use any of DPWiper, Eraser, File Shredder and Freeraser.

Table 9 Files recovered from DPWiper Image by FTK

	FTK name	FTK Location	Dataset	Offset	Size (bytes)
1	A0016112.INF	[root]/System Volume Information/_restore{DD0CF2F7-77D2-4945-B346-6B5613DA5B5D}/RP56/	E:\Dataset\Documents\ AUTORUN.INF	C001C720	88
2	A0016113.bat	[root]/System Volume Information/_restore{DD0CF2F7-77D2-4945-B346-6B5613DA5B5D}/RP56/	E:\Dataset\Documents\ re_unzip.bat	C001CB28	117
3	A0016114.LCK	[root]/System Volume Information/_restore{DD0CF2F7-77D2-4945-B346-6B5613DA5B5D}/RP56/	E:\Dataset\Documents\ ETEXT02\8lied10.zip.LCK	C001CF28	15
4	A0016115.me	[root]/System Volume Information/_restore{DD0CF2F7-77D2-4945-B346-6B5613DA5B5D}/RP56/	E:\Dataset\Documents\ ETEXT94\read116.me	BBED3E00	11,163
5	A0016116.me	[root]/System Volume Information/_restore{DD0CF2F7-77D2-4945-B346-6B5613DA5B5D}/RP56/	E:\Dataset\Documents\ ETEXT94\read728.me	BBED6E00	9,789
6	A0016117.ICO	[root]/System Volume Information/_restore{DD0CF2F7-77D2-4945-B346-6B5613DA5B5D}/RP56/	E:\Dataset\Documents\ IMAGES\FAVICON.ICO	BBEE7E00	1,150

In terms of information about the files and folders, only DPWiper managed to successfully erase both. The entire directory structure appeared to be intact in the Eraser and File Shredder images, and the Freeraser image had the original file names as well.

Arbitrary Files

Finally, there are the worst offenders, the tools that permitted the recovery of apparently arbitrary files. First is BCWipe. As well as FTK recovering the directory structure, Scalpel carved 17 files. This number on its own is not significant, however, the subsequent comparison of the digests showed that one of the 17 files matched with a file from the dataset, specifically, E:\Dataset\Documents\IMAGES\BUTTONBKG.JPG. This is a small, white rectangle image, only 313 bytes in size. It is difficult to imagine a situation where such a file would be significant, but the recovery of a single file, byte-for-byte perfectly recreated, casts into doubt the entire value of the erasing tool.

Second is Fileeraser. FTK recovered the same six files as from the tools in the previous section. Scalpel carved 6333 files, of which there were initially no matches from the dataset. However, manual analysis showed that while there were a great many invalid files (.zip and .mpg), there were several legible files (mainly .html). From examining the contents of the html files, it became clear that they were from the .iso images (specifically the Slackware installation disk). The .iso images from the dataset were mounted using VirtualCloneDrive (http://www.slysoft.com/en/virtual-clonedrive.html) and MD5 digests were taken of all the files. When these digests were added to the MD5 list, it was found that of the 6333 files Scalpel carved, an astonishing 94 matched the dataset, 92 .html files and 2 .jpg. The files are relatively small in size, a few kilobytes, but easily long enough to rule out coincidence.

Lastly is Hard Drive Eraser. The image appeared to be a continuous repetition of a 1KB pattern. Scalpel carved 34 files from the image, none of which matched the files from the dataset. As mentioned previously, random samples from the carved files were checked in all experiments to see if there could be partial matches. This was quite obviously the case with the files carved from the HardDriveEraser image. Enough text was recovered from

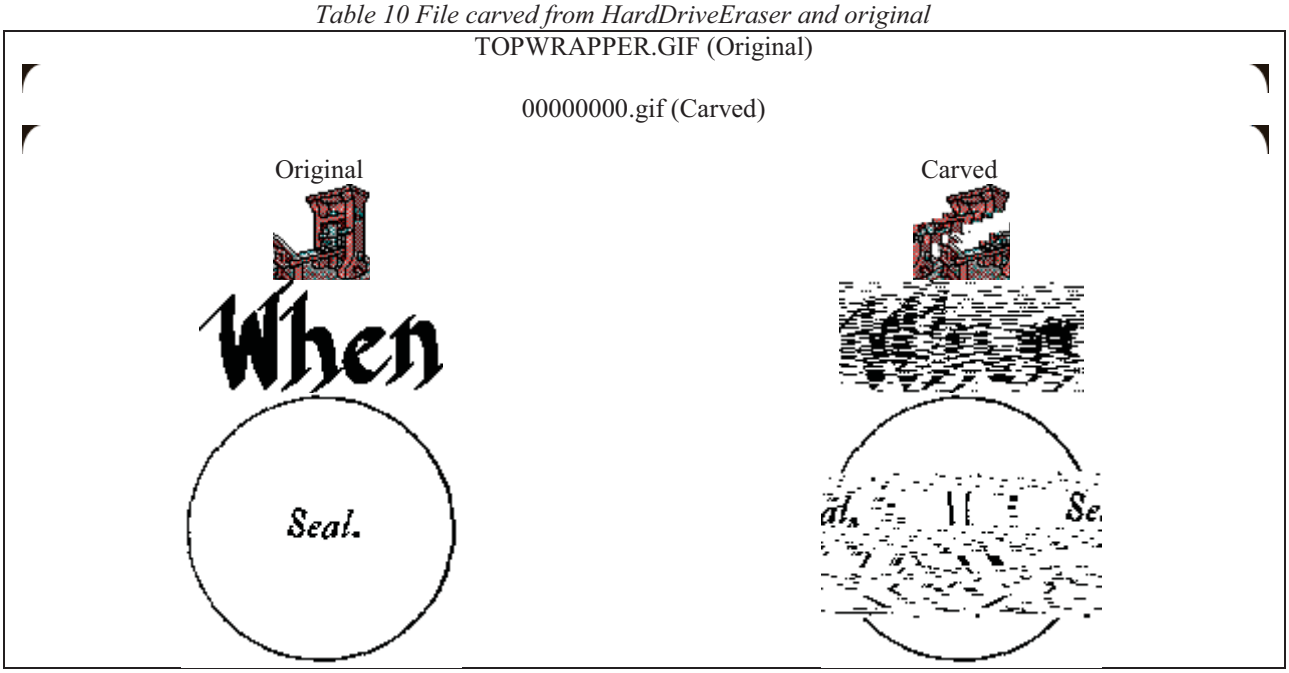
"00000026.htm" to perform a search of the dataset using the grep command. This returned a matching file that appeared to be the same: "I:\Dataset\Documents\ETEXT96\articles.html". However, the fact that the hashes had not found this match proved that the files could not be the same. Using the cmp command, it was possible to identify minor differences between the two files. The bytes at positions 95 and 96 had been changed from \x7274 to \x0200. The same appeared to have occurred with the image file "00000000.gif" and "TOPWRAPPER.GIF", shown in Table 5. The bytes had been changed in the exact same position, but to \x0400 instead of \x0200. The two files came from similar areas of the dataset (I:\Dataset\Documents\ETEXT96\ and I:\Dataset\Documents\IMAGES\), and all files were carved from the same general area of the image. The image comprised of 51 files: HardDriveEraser.001 to HardDriveEraser.051, with HardDriveEraser.003 being the only one that Scalpel carved any files from. The entire carved set comprised of 4 .gif files, 2 .jpgs, 7 .htm files, 3 .zip files and 18 .mpg files. The other image files were badly corrupted. None of the .zip files or the .mpg files were valid (none could be opened), so no attempt at matching was made. Eventually, all 7 .htm files were paired with almost identical originals from the dataset. Long strings were taken from the carved files and the "grep" command was used to recursively search through the dataset. Each had been changed in particular positions in similar ways.

In order to further investigate the erasing failings and the apparent overwriting pattern, the nonzero code was used, with some modification. Instead of looking at each byte of the image, the modified code (cmpblock.exe) looked for any sequences of the image that was not an exact copy of the 1KB block. Besides the source of the carved files, there was evidence of 8 further failings in the erasure. 8 of the non-matching sections had long sequences of ASCII text. Using the grep command it was possible to find the dataset file they came from (each came from a Project Gutenberg text file). There are two reasons why scalpel did not find this. The first is that only the middle sections of the texts were in the partially erased image. The second is that scalpel does not carve text files as they do not have a standard header/footer.

Table 10 File carved from HardDriveEraser and original

TOPWRAPPER.GIF (Original)

00000000.gif (Carved)



RESULTS AND DISCUSSION

In evaluating the merits of data erasers, the most important of criteria is simply the question "Did the program actually erase the data?" All other considerations are secondary to that. Of the 12 tools tested, 7 were shown to have failings in this regard. The most serious were BCWipe, Hard Drive Eraser and worst of all was Fileeraser with a total of 94 recovered files. The failings of the other four, DPWipe, Eraser, File Shredder and Freeraser were less significant. The same 6 files were recovered from each. While the failings of the former group could have been from anywhere in the dataset, the failings of the latter could only have been for files of relevance to the system. However, the average user that has to rely on complete erasure may not be aware of this possible blind spot.

The remaining 5 tools that correctly erased all traces of the dataset were all disk erasers. The minimum expectation of a file eraser is that the data for that specific file be overwritten, but there is also the consideration that no other data be affected. This may limit what the eraser can do without inadvertently corrupting other data. However, when an entire partition is chosen for erasure, all data can and must be destroyed. It allows for a wider range of techniques which are easier to implement.

The consideration of the removal or obfuscation of file and directory names paints a similar picture. All disk erasing tools succeeded whereas only 2 of the 6 file erasers achieved the same. So while it is possible for a file eraser to successfully remove file and directory names, it is not trivial and not functionality that can be expected.

#	Tool	Disk/File eraser	Ability to erase System Volume Information	Ability to erase all files	Running Time (h:m)	Memory Usage (average)	CPU Usage (total h:m:s)
1	Active @ KillDisk	Disk	Success	Success	0:57	17,093 K	0:02:45
2	BCWipe	File	Success	Fail	21:51	819,054 K	1:13:57
3	Blancco	Disk	Success	Success	3:19	N/A	N/A
4	CBLDataShredder	Disk	Success	Success	69:01	4,136 K	7:23:50
5	DBAN	Disk	Success	Success	3:57	N/A	N/A
6	DPWiper	File	Fail	Success	5:00	7,984 K	0:56:11
7	Eraser	File	Fail	Success	8:20	43,095 K	1:35:04
8	Fileeraser	File	Fail	Fail	7:10	25,866 K	0:27:07
9	FileShredder	File	Fail	Success	7:41	64,915 K	5:31:43
10	Freeraser	File	Fail	Success	19:50	4,633 K	2:20:00
11	HardDriveEraser	Disk	Success	Fail	1:11	6,209 K	0:26:18
12	Wipetool	Disk	Success	Success	0:53	4,668 K	0:00:08

Table 6 shows how the different programs compared in terms of running time, average memory usage and total CPU usage. There is considerable variation in the times taken and resources consumed. Some tools took in the region of one hour, others took days. The most notably resource-expensive tools were BCWipe with over 800MB memory consumed, and CBLDataShredder, taking almost three days to run. There is considerable variation in the running time, which is unexpected given most tools were configured to perform the same number of overwrites (3). The ones that did not were Active @ Kill Disk, which was 1 pass, and WipeTool which did not state the number of passes (although the similar running time would suggest that it was also 1 pass). The speed at which HardDriveEraser completed (closer to the 1 pass Active @ Kill Disk than any of the 3 passes) would suggest the program did not operate as designed (a possible error occurred to cause it to behave erratically). Further investigation of this anomaly will be undertaken.

Of the twelve tools analyzed, five are personal/non-profit projects (DBAN, DPWiper, FileEraser, FileShredder and HardDriveEraser) and the other seven are commercial products. Of the tools that completely erased all data from the drives, only one was non-commercial: DBAN. This might suggest that commercial products are more effective, but we believe this has more to do with the difference between disk and file erasers (there were only two non-commercial disk erasers: DBAN and HardDriverEraser). If one considers the tools from which arbitrary files were recovered, there was one commercial tool (BCWipe) as well as two non-commercial (FileEraser and HardDriveEraser).

Recommendations

From the experiments performed, we can make the following recommendations. For erasing entire drives, Active @ KillDisk and WipeTool performed the best. However, they did not implement the US DoD 5220.22-M(E) standard. If this is required (for compliance or personal preference), and a boot disk is feasible, Blancco and DBAN are recommended (CBLDataShredder was inefficient and HardDiskEraser did not wipe the drive).

File erasers should not be used where there must be guarantees of data erasure, i.e. do not use a file eraser where whole disks need to be erased, but only where it is imperative that some files remain on the drive. Even then, it would be preferable to backup the required files, erase the disk with a disk eraser and restore the files. In the

event a file eraser must be used, DPWiper is recommended. Only those files that had been stored in the "System Volume Information" folder were recovered, and FTK was not able to retrieve the file/directory names.

CONCLUSIONS

This paper has presented the rigorous method developed to test the features of various data erasing tools, as well as the results from our experiments. The number of failings was unexpectedly high. These tools give the impression that with multiple overwrites they will remove the data from all but the most sophisticated analysis. However, it was found that in several cases, data remains in plain view.

The most obvious area of future work would be in the "System Volume Information" folder. Any user interested in protecting their privacy with a file eraser needs to be aware of what files are automatically stored in this location.

Future experiments would benefit from the following:

- Repeating the experiments on multiple disks
- Increase the number of tools tested
- The use of updated versions of previously tested tools
- Automate a way to count how many file/directory names have been correctly recovered

This work has highlighted some serious failings in existing data erasing tools. Users should be able to trust that files cannot be recovered once they have been erased. This research has shown which tools meet this fundamental requirement.

Acknowledgments

Special thanks go to Hazza Al-Tenaiji for his work on the initial experiments.

REFERENCES

CESG. British HMG IS5.

CSEC. (2006). *Clearing and Declassifying Electronic Data Storage Devices* (ITSG-06). Communications Security Establishment Canada.

Defense, D. o. (2006). *National Industrial Security Program (NISP) 5220.22 M*.

DSD. (2008). *Australian Government Information Security Manual*. Defence Signals Directorate.

Gutmann, P. (1996). *Secure Deletion of Data from Magnetic and Solid-State Memory*. Department of Computer Science, University of Auckland.

Slusarczuk, M. et al, (1987). *Emergency Destruction of Information Storing Media*, Institute for Defense Analyses

NZSIT. (2008). *NZSIT 402*. New Zealand Government Communications Security Bureau.

Richard Kissel, M. S. (2006). *NIST Special Publication 800-88 Guidelines for Media Sanitization*. U.S. Department of Commerce.

Wright, C., & Kleiman, D. &. (2008). *Overwriting Hard Drive Data: The Great Wiping Controversy*. ICISS .