2013

A Forensic Analysis And Comparison Of Solid State Drive Data Retention With Trim Enabled File Systems

Alastair Nisbet
Auckland University of University, anisbet@aut.ac.nz

Scott Lawrence
Auckland University of Technology, scottielawrence@me.com

Matthew Ruff
Auckland University of Technology, matthewruff@me.com

DOI: 10.4225/75/57b3d766fb873
Originally published in the Proceedings of the 11th Australian Digital Forensics Conference. Held on the 2nd-4th December, 2013 at Edith Cowan University, Perth, Western Australia

A FORENSIC ANALYSIS AND COMPARISON OF SOLID STATE DRIVE DATA RETENTION WITH TRIM ENABLED FILE SYSTEMS

Alastair Nisbet, Scott Lawrence, Matthew Ruff
Auckland University of Technology, Digital Research Laboratories
Auckland, New Zealand
anisbet@aut.ac.nz, scottielawrence@me.com, matthewruff@me.com

Abstract
Solid State Drives offer significant advantages over traditional hard disk drives. No moving parts, superior resistance to shock, reduced heat generation and increased battery life for laptops. However, they are susceptible to cell failure within the chips. To counter this, wear levelling is used so that cells are utilised for data at approximately the same rate. An improvement to the original wear levelling routine is TRIM, which further enhances the lifetime of the cells by allowing the garbage collection process as one operation rather than an on-going process. The advantages of TRIM for the user is that it increases efficiency of the drive's wear levelling algorithms, meaning quicker access times and longer lifetimes. The basic wear levelling routines have caused significant difficulties for forensic investigators as data is moved to different random locations without user input. Whilst this problem has been examined in past research, the implementation of TRIM has not had much attention. This research examines SSD drives across three TRIM enabled file systems, Windows, Linux and MAC OS X operating systems. The results show that TRIM leaves far less data for forensic investigators than drives without TRIM enabled.

Keywords
Solid state drives, forensic analysis, ant-forensics, data erasure

INTRODUCTION

Solid State memory has been available in various formats for almost quarter of a century. With the recent reduction in cost and increase in size, Solid State Drives (SSDs) are now a popular choice for laptop and desktop computers. Advantages of SSD drives are lighter weight, faster access times and less heat generation. Coupled with significant resistance to damage from physical shock and low power usage, this makes them especially attractive for use in laptop computers. However, SSDs deal with deletion of data in a very different manner than traditional Hard Disk Drives (HDDs). Generally when data is deleted from a hard drive, the data is retained until new data is written onto the same location. If no new data is written over the deleted data, then the forensic investigator can recover the deleted data, albeit often in fragments. A computer that has been unused for years can still have data recovered from its hard drive (Moulton, 2008). To counter this, users may deliberately overwrite deleted data using software that will write zeroes to the deleted locations.

SSDs suffer from wear on cells which significantly reduces the lifetime of the drive. If blocks on a drive are continuously written to and erased, then the drive may lose significant amounts of space due to failure of individual cells which makes the entire block of cells unusable (Perdue, 2008). To counter this, SSDs perform wear levelling by taking data from a well-used block on the drive and writing it to an under-used block. When the regular "Garbage Collection" is performed, the old area where the data was moved from is written with zeroes. This means that users do not need to proactively zero-delete data locations, but rather it is done as a matter of course by the drive. While this prolongs the lifetime of the drive, it significantly reduces the data that can be retrieved from the drive by the forensic investigator. To further increase the lifetime of the drive, TRIM can be enabled that forces the operating system to notify the drive that data has been deleted from a location and the drive can then mark its location as invalid(Williams, 2010). This occurs shortly after the data is deleted, and the SSD Garbage Collection routine will now skip the collection from these locations, saving wear on the drive and speeding up the garbage collection process.

The implementation of TRIM hands some power back to the operating system to determine how deletion of data will be handled by the drive. This is beneficial to the forensic investigator because deleted data will remain on the drive as invalid data, rather than written over by zeroes. The process of data deletion on SSDs was examined in 2009 to discover how much data was retained when the drive was wiped(Freeman and Woodward, 2009). In their research, a TRIM enabled drive was not available and so TRIM was not examined. This research takes this investigation further and looks at the data retained by SSDs utilising the TRIM capability. Three file systems are investigated: NTFS, Ext4 and HFS+.

STATE OF THE ART

The design of an SSD is made up of non-volatile NAND based flash memory chips which provide sufficient density, quicker access time and reduced latency for use as a primary storage device. A key component of an SSD is the controller which bridges the NAND memory components to a host computer (Shimpi, 2009). The controller is a processer embedded in the drive that executes firmware-level code and functions. Primarily the key functions performed by the controller include read and write caching, garbage collection and wear levelling(Rent, 2010). The makeup of NAND flash chips on the SSD consists of cells, pages, blocks and planes designed in a hierarchical structure that have their own individual and unique physical properties and characteristics(Vidas and King, 2011). The standard page size on SSDs is 4KB and these pages are grouped together into blocks which commonly consist of 128 pages per block. This results in a 512KB block. Many of the SSD's issues stem from the way the drive reads and writes. As a block is the smallest structure that can be erased in NAND-flash, 128 pages must be erased in order to erase a particular block. Single NAND-flash cells can contain either one or two bits of data, either Single Level Cell (SLC) or Multi Level Cell (MLC). Both SLC and MLC are physically and fundamentally the same, it is only how data is stored and read which separates the two(Moulton, 2008).

SSDs are susceptible to much shorter life spans then typical HDDs. The standard life expectancy of an SSD is estimated at 5 years, however, writing and erasing data on SSDs results in wear to the flash cells which limits the lifetime of the drive(Moulton, 2008). To prolong the drives longevity the practice of wear levelling was introduced (Perdue, 2008). Wear levelling requires all cells on the drive to be written to at least once before writing over the same cell again. As each cell has limitations to the number of re-writes and erasures before its lifespan diminishes, typically around the 3,000/5,000 cycle figure, spreading the erasure and writes across all cells ensures no single erase block is likely to prematurely fail. However, when the drive becomes full, speed can be significantly affected(Shimpi, 2009).

Flash cells operate under a method of delete-before-write which requires a cell to be completely erased or zeroed-out before a further write can be committed. To alleviate the impact created by the time consuming erase operation, the garbage collection process was introduced(Shimpi, 2009). Garbage collection is a background operation that accumulates data blocks that have been previously marked for deletion, performs a whole block erasure on each 'garbage' block, and returns the reclaimed space for subsequent rewrite operations. The garbage collection process is a combination of algorithms designed to deal with performance degradation that occurs on these drives over time(Williams, 2010). However garbage collection is primarily a background process and to preserve drive performance while in operation an additional solution was required and hence the introduction of TRIM (Mehling, 2009).

The support of the TRIM instruction has been implemented in most contemporary operating systems; NTFS on Windows 7, Ext4 used with the Linux Kernel version 2.6.33 or later, and Mac OS X 10.6.8 or later using HFS+. TRIM, as opposed to standard garbage collection, interacts with the operating system which marks the blocks as deleted(Intel, 2013). The controller on the drive is still responsible for when garbage collection is initiated but TRIM addresses the need for more "scratch" space through a simple command sent when a delete operation is performed where the OS sends a

TRIM command to the SSD with a list of the required blocks marked for deletion. When a file is deleted, the operating system sends a trim command to the SSD controller. The block marked for deletion is then copied to cache, the deleted pages are wiped and a new block is written with new pages (Shimpi, 2009).

While TRIM does not alleviate all performance issues, the use of TRIM has proven to be a solution to help maintain an SSDs high performance even with heavy usage. Due to the design and technology used in traditional Hard Disk Drives, it has been an arguably simple process in many cases to acquire deleted data and provided a relatively easy avenue of investigation for Computer Forensic Professionals. As SSDs operate differently from the traditional Hard Disk Drive, this has resulted in concern as to what the future is for the traditional tools and techniques for both the Computer Forensic and Data Recovery industries (Bell and Boddington, 2010).

DATA COLLECTION

The experiments focused around three main test cases. These were the three file system / operating system platforms selected. All three platforms support and implement the TRIM instruction when used with a TRIM supported SSD. Within each test case multiple test scenarios were created based on workload and drive usage. In each combination tests were run with TRIM enabled and disabled. The effectiveness of TRIM as an anti-forensic tool to purge deleted data was also tested at the device level which is covered in our Anti-Forensics test case.

A fourth test case, the Anti-Forensics test case was added to look at the effectiveness of TRIM in key areas. One was the effectiveness of TRIM in combination with the tool hdparm as an anti-forensics measure as outlined in Vidas and King's research (Vidas and King, 2011). Another was to provide a device level picture of how TRIM is executed when the instruction is passed to the drive. The experiment was designed to enable the system to provide data for both of these areas. These experiments were performed directly on the Sandforce SSD with the help of the Linux kernel and the hdparm and dcfldd tools. The experimental setup eliminated any design and implementation choices around TRIM on various platforms that may be identified in the three initial test cases.

The objectives for the experiments were as follows:

Test SSD Data Retention on NTFS, Ext4, and HFS+ file systems under the following conditions:

- Idle workload
- Activity workload
- Low drive usage
- High drive usage
- TRIM enabled and disabled

Test the effectiveness of TRIM as an Anti-Forensics measure under the following conditions:

- Manual passing of the TRIM command at the device level

These test cases cover three file system / operating system platforms; NTFS on Windows 7 (SP1), Ext4 on Ubuntu 11.10 using kernel version 3.0.0, and HFS+ on Mac OS X 10.7. We used the 64-Bit version of each operating system. These operating systems all support the TRIM instruction under certain conditions. Each operating system is installed using the default setup options including the creation of partitions.

TESTING

Two scenarios were selected to provide two different workloads applied to the drives during the tests. The first was an idle work scenario and consisted of the test taking place with the operating system performing no other work. The second was an activity test scenario which simulated a set of user-initiated tasks and the use of software applications by a user. The idle workload involved allowing the drive to sit powered on, with the operating system loaded but with no applications running beyond the default background processes. It was left to sit for one hour before extraction began. The idle workload in our tests is as close as possible to zero drive activity without powering down.

The steps in the process were then:

1. The SSD was wiped using a custom designed two-phase process.

   - A Linux Bash script was created to TRIM the entire drive resulting in an SSD wipe.

   - A second phase involved writing a HEX pattern of zeros "00" to the entire drive.

2. The SSD was imaged with the platform image for the next test due to be run.

3. The test computer was selected; either Test One, Test Two, or Test Mac Mini.

4. The SSD was placed in a caddy tray and inserted into the removable drive bay.

5. Depending on the test computer selected the BIOS settings on start-up were manipulated and this disabled the internal drive if one was present.

6. The SSD was then booted as normal

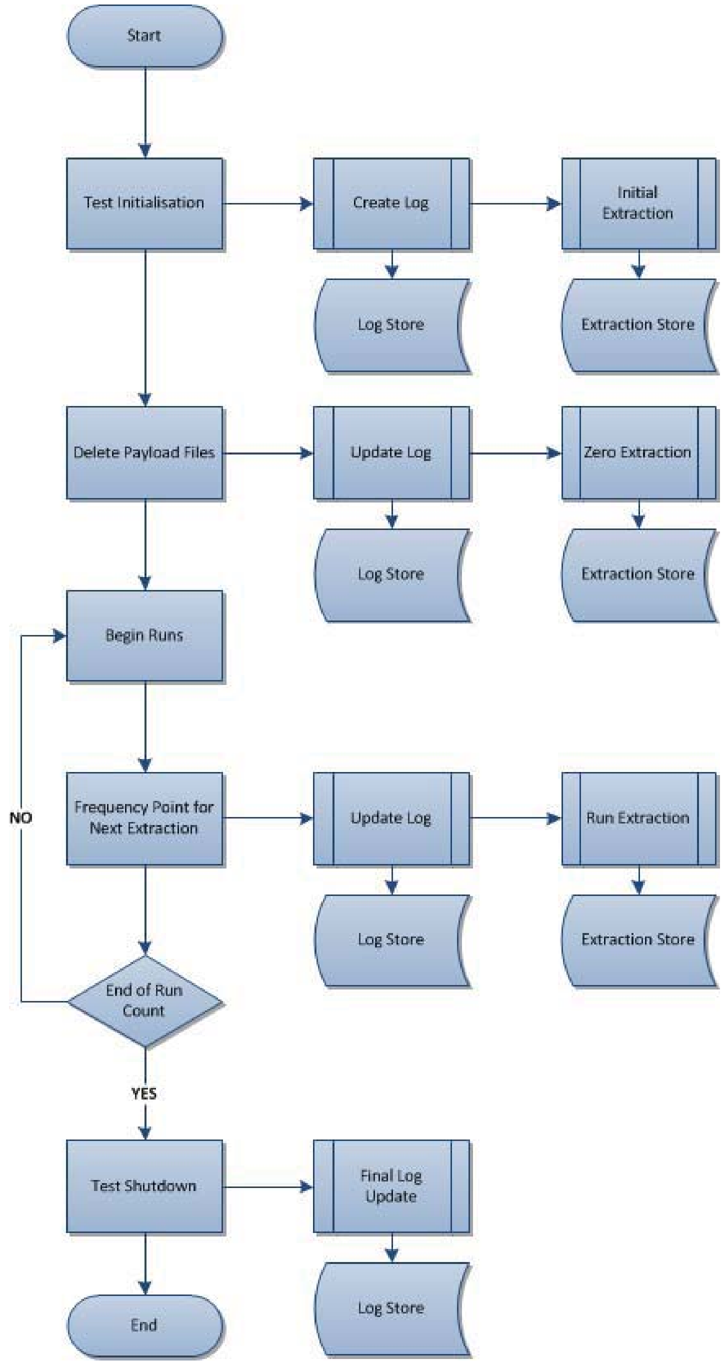

Figure 1 - Process diagram of platform testing process

The workload consisted of several common software applications scripted to perform a range of different actions. This was designed to simulate a typical user initiated workload on the drive. VLC Media Player, Mozilla Firefox, Mozilla Thunderbird and Libre Office were selected. These four applications cover the typical tasks of a user including music and video playback, web browsing, email and document creation. The activity workload was automated for one hour using custom scripts. A separate script was created to loop the one hour workload as many times as was required for longer tests.

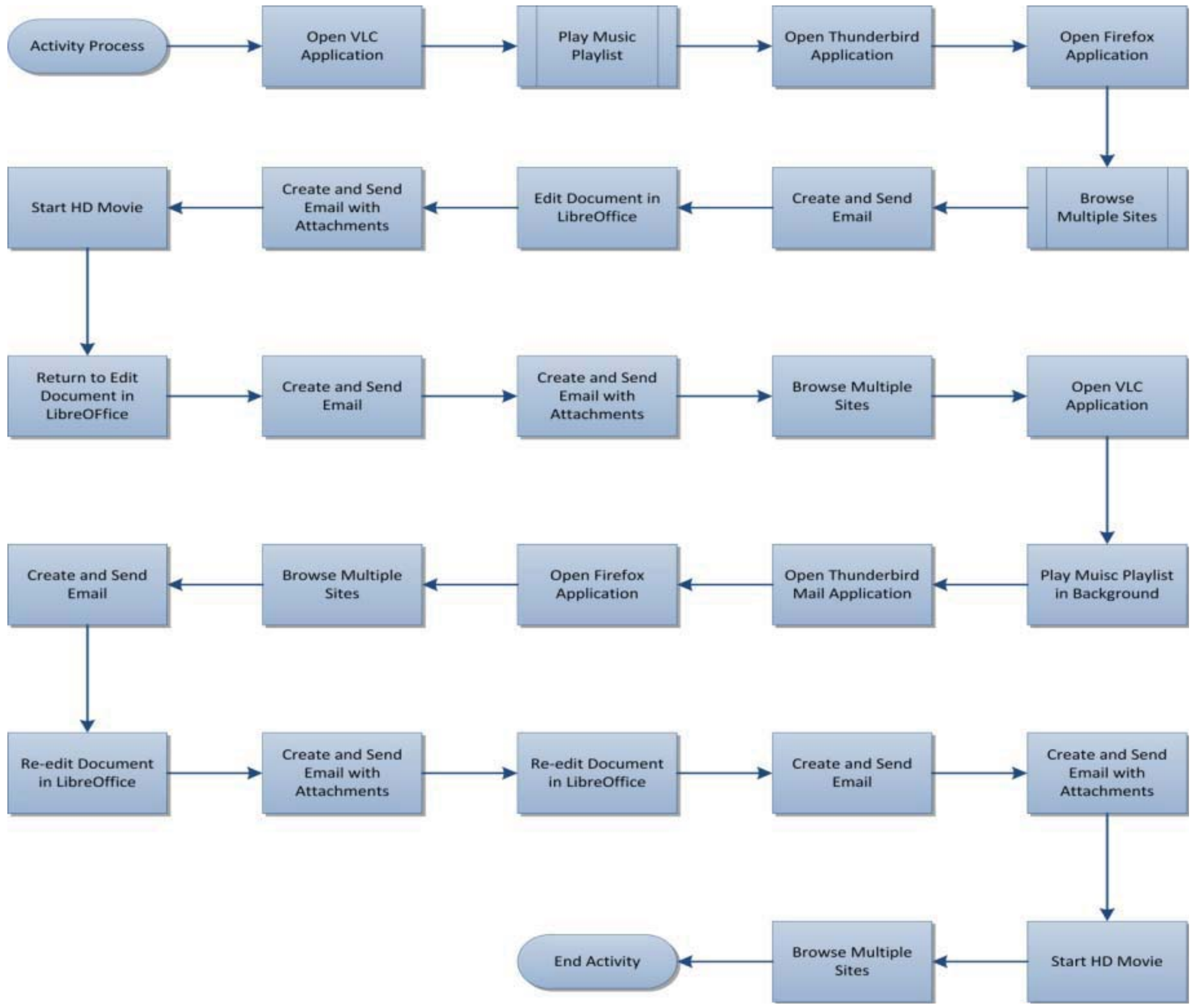

Figure 2 - Activity workload scenario process

Two usage scenarios were selected. Low drive usage represented a default install of the operating system and any required drivers and updates. The only applications installed were those required for the "Activity" work scenarios. The second scenario was high drive usage. In this scenario free space was reduced to within 5% - 10% of the total SSD size using dummy files. A run of the tests was completed with TRIM disabled to provide a control set of data. In this NOTRIM scenario only the SSDs native garbage collection routines and any operating system writes to the extraction areas were seen. TRIM effectiveness itself, at the device level is covered in the Anti-Forensics test case.

ANTI FORENSICS

This test case repeated the experiment conducted by Vidas and King (2011). They found that manual TRIM, using the software tool hdparm could be used as an anti-forensics measure(Vidas and King, 2011). Shu and Obr point out that since ATA-8 supports a TRIM of up to 65,536 blocks (@ 512Bytes in a single command it would be possible to TRIM an entire drive in seconds(Shu and Obr, 2007). The drive would then proceed to sanitize itself, wiping all data.

The experiment was recreated using the hdparm software tool and custom scripts. Two scenarios were tested; a drive filled entirely with a repeating text pattern and a drive filled entirely with random data. Due to the compression and de-duplication features implemented within the SandForce based SSD used for the experiments, it was important to test using a dataset that would force the controller to write every Byte sent. This test scenario would fill the drive with randomly generated data before a whole drive TRIM was initiated. The drive was then analysed and the amount of remaining data calculated.

RESULTS AND DISCUSSION

The results from the 3 test runs were then examined to identify the percentage of Bytes retained. Tables 1-3 compare the retained data across the 3 test scenarios.

Table 1: Percent of payload bytes retained after initial payload deletion

Table - Percent of Bytes retained after initial payload deletion during each test.												
	smallone.small			largeone.large			random.small			random.large		
Test Scenario	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+
Idle, Low Usage, TRIM	100.00 %	100.00 %	100.00 %	18.71 %	100.00 %	49.52 %	100.00 %	100.00 %	100.00 %	22.52 %	100.00 %	16.27 %
Idle, Low Usage, NOTRIM	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Idle, High Usage, TRIM	100.00 %	100.00 %	100.00 %	11.83 %	100.00 %	20.05 %	100.00 %	100.00 %	100.00 %	6.73 %	100.00 %	6.16 %
Idle, High Usage, NOTRIM	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Activity, Low Usage, TRIM	100.00 %	100.00 %	100.00 %	10.96 %	100.00 %	25.48 %	100.00 %	100.00 %	100.00 %	10.28 %	100.00 %	41.64 %
Activity, Low Usage, NOTRIM	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Activity, High Usage, TRIM	100.00 %	100.00 %	100.00 %	18.90 %	100.00 %	42.99 %	100.00 %	100.00 %	100.00 %	10.58 %	100.00 %	24.26 %
Activity, High Usage, NOTRIM	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %

Table 2: Percent of payload bytes retained 1 hour after payload deletion

Table - Percent of Bytes retained after one hour from first extraction during each test.												
	smallone.small			largeone.large			random.small			random.large		
Test Scenario	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+
Idle, Low Usage, TRIM	0.00 %	100.00 %	0.00 %	0.16 %	100.00 %	0.15 %	0.39 %	100.00 %	0.39 %	0.55 %	100.00 %	0.58 %
Idle, Low Usage, NOTRIM	24.45 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Idle, High Usage, TRIM	0.00 %	100.00 %	0.00 %	0.08 %	100.00 %	0.13 %	0.39 %	100.00 %	0.39 %	0.47 %	100.00 %	0.50 %
Idle, High Usage, NOTRIM	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Activity, Low Usage, TRIM	0.00 %	100.00 %	0.00 %	0.14 %	100.00 %	0.16 %	0.39 %	100.00 %	0.39 %	0.58 %	100.00 %	0.50 %
Activity, Low Usage, NOTRIM	100.00 %	100.00 %	36.00 %	100.00 %	100.00 %	96.01 %	100.00 %	100.00 %	35.25 %	100.00 %	100.00 %	100.00 %
Activity, High Usage, TRIM	0.00 %	100.00 %	0.20 %	0.19 %	100.00 %	0.15 %	0.39 %	100.00 %	0.39 %	0.52 %	100.00 %	0.58 %
Activity, High Usage, NOTRIM	100.00 %	59.57 %	4.38 %	100.00 %	98.98 %	95.60 %	100.00 %	0.39 %	35.26 %	100.00 %	0.39 %	100.00 %

Table 3: Percent of payload bytes retained 5 hours after deletion

Table - Percent of Bytes retained after last extraction during each test.												
	smallone.small			largeone.large			random.small			random.large		
Test Scenario	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+	NTFS	Ext4	HFS+
Idle, Low Usage, TRIM	0.00 %	100.00 %	0.00 %	0.16 %	100.00 %	0.15 %	0.39 %	100.00 %	0.39 %	0.55 %	100.00 %	0.58 %
Idle, Low Usage, NOTRIM	24.45 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Idle, High Usage, TRIM	0.00 %	100.00 %	0.00 %	0.08 %	100.00 %	0.13 %	0.39 %	100.00 %	0.39 %	0.47 %	100.00 %	0.50 %
Idle, High Usage, NOTRIM	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Activity, Low Usage, TRIM	0.00 %	100.00 %	0.00 %	0.14 %	100.00 %	0.16 %	0.39 %	100.00 %	0.39 %	0.58 %	100.00 %	0.50 %
Activity, Low Usage, NOTRIM	0.37 %	0.36 %	1.28 %	100.00 %	99.40 %	87.34 %	1.01 %	100.00 %	0.38 %	100.00 %	100.00 %	100.00 %
Activity, High Usage, TRIM	0.38 %	100.00 %	0.22 %	0.19 %	100.00 %	0.16 %	0.39 %	100.00 %	0.39 %	0.52 %	100.00 %	0.58 %
Activity, High Usage, NOTRIM	4.30 %	59.57 %	0.23 %	100.00 %	97.56 %	87.25 %	0.58 %	0.40 %	6.37 %	100.00 %	100.00 %	100.00 %

In tables 1-3, the difference in results with the combinations of platform test cases and test scenarios that made up the experiments is clearly shown. Whilst NTFS had some previous testing, very little focus has been focussed on the Ext4 and HFS+ file systems. A primary objective for these experiments was to see how much, if any, data would be retained on the file systems with TRIM enabled/disabled respectively. A further result of interest was whether time had an effect on data retention when TRIM was enabled. It was found that this was not the case. The figures from the extraction of data after the first hour mark show minimal change between that and the final extraction after the five hour mark. This shows that regardless of the period of time a drive may be active or idle, once the TRIM command is sent to the drive, any erasing usually takes place within minutes.

The tests used four different payload types. These were two small files, one being filled with random data and another filled with a repeating string of text. The same was created for two larger sized files. After the initial payload deletion, it is noticeable how aggressive TRIM is. How and when TRIM operates seems to be dependent on several factors. It appears the SSD decides very quickly to erase cells when a TRIM command for that sector is issued. This does not however mean that the TRIM instructions are always acted on.

Although the two smaller payload files had 100% of data recovered, the larger files returned much smaller amounts of data. The smaller files are deleted and extraction is almost instant due to the file size. The larger files take a longer period of time to perform those tasks and therefore TRIM has more time to perform its operation before later blocks are extracted. It should be noted that the larger files had only small amounts of data remaining, but that data was effectively the beginning of the payload file itself. Again this shows the aggressiveness of TRIM. TRIM can effectively wipe the remaining data in a matter of seconds thus providing only a small segment of a file as remaining data. This is seen in the results which identify smaller files that are deleted and extracted in a shorter period of time compared to larger files where the process is longer with only smaller amounts remaining. The results after one hour show that any chance of recovering sizeable amounts of data in these instances is almost zero. A forensic investigator would therefore have to perform an extraction very quickly after the TRIM command is passed. At the initial payload deletion extraction mark, the results identify that the garbage collection process has not initiated. Unlike the TRIM results which show data loss immediately, garbage collection is seen as static and making no attempt to "clean up" any of the now apparently free space once occupied by the payload files.

At the one hour extraction mark, the results identify considerable reductions in recoverable data across most of the scenarios and platforms. With TRIM enabled, less than 0.5% is recoverable across the NTFS and HFS+ file systems. Ext4 however uses a batch discard to deal with TRIM instructions. Larger batches of TRIM ranges are combined and sent to the drive at later intervals. This was proposed as a performance advantage over sending TRIM instructions with every file delete(Shimpi, 2009). Batched Discard creates a scenario where even though TRIM is enabled the drive is not immediately aware of the file deletion and therefore free space it can erase. This would indicate why NTFS and HFS+ TRIM enabled file systems appear to delete more aggressively in the results. It should also be noted that the garbage collection process acting independently of any file system instruction and like that of the Ext4 NOTRIM scenario, purged more deleted data than Ext4 with TRIM enabled. In many aspects, HFS+ and NTFS share similarities in results. Ext4, due to the batch discard, results in a different outcome.

At the final extraction mark, all four TRIM enabled scenarios returned between 0% and 0.38% bytes of data retained for the smallone.small payload file in contrast to the NOTRIM enabled scenarios which returned values in the region of 0.23% bytes retained to 100%. Using the results of the smallone.small payload file highlights the differences in data retention between NTFS which appears to be more volatile in its patterns.

Under the TRIM enabled scenarios with the largeone.large payload, the end result identifies lower retention of data, primarily leaving only the first few segments of the file. Both idle and activity scenarios with TRIM enabled saw the same results in the extent of the data deleted. Arguably, HSF+ showed a more aggressive garbage collection process against NTFS which retained 100% of the file with NOTRIM. The random.small payload file results show the size of blocks that TRIM is able to erase. This again is seen in the results for the random.large payload file which sees byte retention degrading in a consecutive pattern of size.

The results obtained during our Anti-Forensics test case experiments indicate that TRIM alone is not a reliable measure for proper sanitization of a SSD. The repeating pattern test scenario indicates a 99.8% zeroing of the drive using a whole drive TRIM method. On the 120GB SSD used for these tests, this is still 185MB to 214MB in certain instances.

The random filled drive scenario results indicate a lower amount of retained data after a whole drive TRIM method. It would appear that the SSD controller in our test drive, due to de-duplication and compression features has likely played a part in the higher retained data figures in the text pattern scenario. This would lead us to believe that the figures presented are a more accurate representation of the true effectiveness of TRIM as an Anti-Forensics measure. Assuming this is the best case outcome for the whole drive TRIM method it is still not a reliable or recommended process for securely wiping the SSD. We still see 80MB to 97MB of data retained on our 120GB SSD.

CONCLUSION

The comparison of data retention across TRIM enabled file systems has shown that, in the case of TRIM, if TRIM is intending on performing a TRIM operation on marked blocks, in a matter of minutes that data is purged and unrecoverable. Garbage collection, being drive initiated, is not as rapid with its operation. The garbage collection process operates on its own method, algorithms and time delay. Both NTFS and HSF+ have shown this in their results. Ext4 however operates uniquely as to how and when the TRIM command is sent due to the batch discard implementation.

The Batch Discard implementation in the Ext4 file system on Linux creates an opportunity for an improved chance of data recovery when compared to NTFS and HFS+ file systems in a TRIM enabled setting. In all three file systems tested, TRIM is capable of destroying deleted data when the SSD is sent the TRIM command. In a NOTRIM scenario there is a more aggressive garbage collection process when high drive usage and drive activity are taking place across all three platforms. While some literature has indicated that manual TRIM to the entire SSD is a quick and effective means of performing anti-forensics, these experiments have shown that some minimal data is retained. Whilst this method is quick, we suggest the use of the ATA Secure Erase standard built into most SSDs.

REFERENCES

Bell, G. and R. Boddington (2010). "Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery." The Journal of Digital Forensics, Security and Law 5(3).

Freeman, M. and A. Woodward (2009). Secure State Deletion: Testing the efficacy and integrity of secure deletion tools on Solid State Drives. Australian Digital Forensics Conference, Perth, Australia, Edith Cowan University.

Intel. (2013). "Solid State Drives and Caching." Retrieved May 16th, 2013, from http://www.intel.com/p/en_US/support/ssdc?wapkw=solid+state+drives+and+caching.

Mehling, H. (2009). "Solid State Drives Take Out the Garbage." Retrieved 12th June, 2013, from http://www.enterprisestorageforum.com/technology/features/article.php/3850436/Solid-State-Drives-Take-Out-the-Garbage.htm.

Moulton, S. (2008). Solid State Drives: Ruining Forensics. Las Vegas, USA, Defcon 16.

Perdue, K. (2008). "Wear Leveling." Retrieved 18th June, 2013, from www.eettaiwan.com/STATIC/.../EETOL_2008IIC_Spansion_AN_13.pdf.

Rent, M. T. (2010, 18th May 2013). "SSD Controller." from http://www.storagereview.com/ssd_controller.

Shimpi, A. L. (2009, 3rd May 2013). "The Anatomy of an SSD." from http://www.anandtech.com/show/2738/5.

Shimpi, A. L. (2009). "The SSD Anthology: Understanding SSDs and New Drives from OCZ." Retrieved 12th May, 2013, from http://www.anandtech.com/show/2738/6.

Shu, F. and N. Obr (2007) "Data Set Management Commands Proposal for ATA8-ACS2."

Vidas, T. and C. King (2011). "Empirical Analysis of Solid State Drive Disk Data Retention when used with Contemporary Operating Systems." The International Journal of Digital Forensics & Incident Response 8: 8.

Williams, R. (2010). "Too TRIM? When SSD Data Recovery is Impossible." Retrieved May 13th, 2013, from http://techgage.com/article/too_trim_when_ssd_data_recovery_is_impossible/