Belkasoft
http://belkasoft.com

SSD Forensics 2014

Recovering Evidence from SSD Drives: Understanding TRIM, Garbage Collection and Exclusions

Yuri Gubanov, Oleg Afonin
Belkasoft Research
research@belkasoft.com

We published an article on SSD forensics in 2012. SSD self-corrosion, TRIM and garbage collection were little known and poorly understood phenomena at that time, while encrypting and compressing SSD controllers were relatively uncommon. In 2014, many changes happened. We processed numerous cases involving the use of SSD drives and gathered a lot of statistical data. We now know more about many exclusions from SSD self-corrosion that allow forensic specialists to obtain more information from SSD drives.

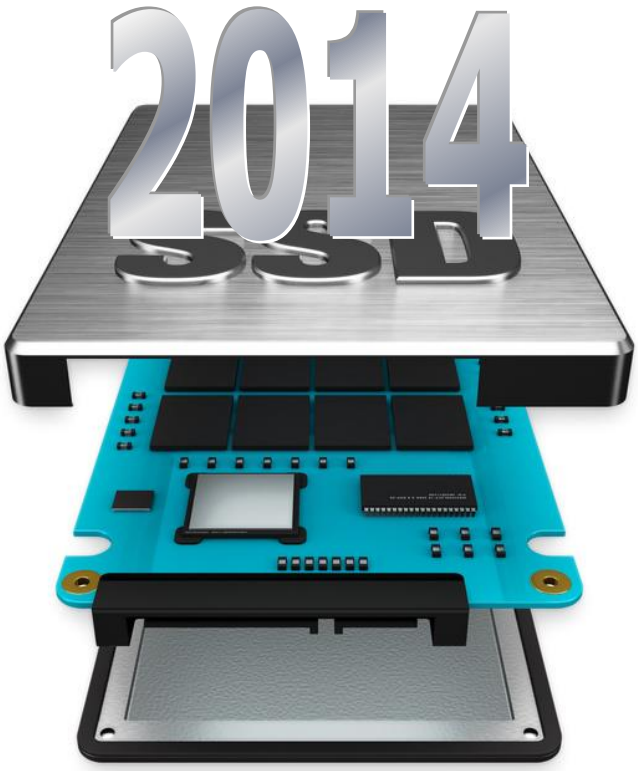
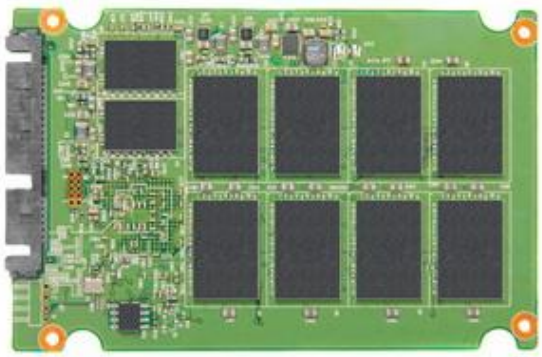


Table of Contents

Introduction



Several years ago, Solid State drives (SSD) introduced a challenge to digital forensic specialists. Forensic acquisition of computers equipped with SSD storage became very different compared to acquisition of traditional hard drives. Instead of straightforward and predictable recovery of evidence, we are in the waters of stochastic forensics with SSD drives, where nothing can be assumed as a given.

With even the most recent publications not going beyond introducing the TRIM command and making a conclusion on SSD self-corrosion, it has been common knowledge – and a common misconception, – that deleted evidence cannot be extracted from TRIM-enabled SSD drives, due to the operation of background garbage collection.

However, there are so many exceptions that they themselves become a rule. TRIM does not engage in most RAID environments or on external SSD drives attached as a USB enclosure or connected via a FireWire port. TRIM does not function in a NAS. Older versions of Windows do not support TRIM. In Windows, TRIM is not engaged on file systems other than NTFS. There are specific considerations for encrypted volumes stored on SSD drives, as various crypto containers implement vastly different methods of handling SSD TRIM commands. And what about slack space (which has a new meaning on an SSD) and data stored in NTFS MFT attributes?

Different SSD drives handle after-TRIM reads differently. Firmware bugs are common in SSD drives, greatly affecting evidence recoverability. Finally, the TRIM command is not issued (and garbage collection does not occur) in the case of data corruption, for example, if the boot sector or partition tables are physically wiped. Self-encrypting SSD drives require a different approach altogether, while SSD drives using compressing controllers cannot be practically imaged with off-chip acquisition hardware. Our new research covers many areas where evidence is still recoverable - even on today's TRIM-enabled SSD drives.

SSD Self-Corrosion

In case you haven't read our 2012 paper on SSD forensics, let's stop briefly on why SSD forensics is different.

The operating principle of SSD media (as opposed to magnetic or traditional flash-based storage) allows access to existing information (files and folders) stored on the disk. Deleted files and data that a suspect attempted to destroy (by e.g. formatting the disk, even if "Quick Format" was engaged) may be lost forever in a matter of minutes. And even shutting the

affected computer down immediately after a destructive command has been issued, does not stop the destruction. Once the power is back on, the SSD drive will continue wiping its content clear all by itself, **even if installed into a write-blocking imaging device**. If a self-destruction process has already started, there is no practical way of stopping it unless we're talking of some extremely important evidence, in which case the disk accompanied with a court order can be sent to the manufacturer for low-level, hardware-specific recovery.

The evidence self-destruction process is triggered with the TRIM command issued by the operating system to the SSD controller at the time the user deletes a file, formats the disk or deletes a partition. The TRIM operation is fully integrated with partition- and volume-level commands. This includes formatting the disk or deleting partitions; file system commands responsible for truncating and compressing data, and System Restore (Volume Snapshot) operations.

Note that the data destruction process is **only** triggered by the TRIM command, which must be issued by the operating system. However, in many cases the TRIM command is NOT issued. In this paper, we concentrate on these exclusions, allowing investigators to gain better understanding of situations when deleted data can still be recovered from an SSD drive. However, before we begin that part, let's see how SSD drives of 2014 are different from SSD drives made in 2012.

Checking TRIM Status

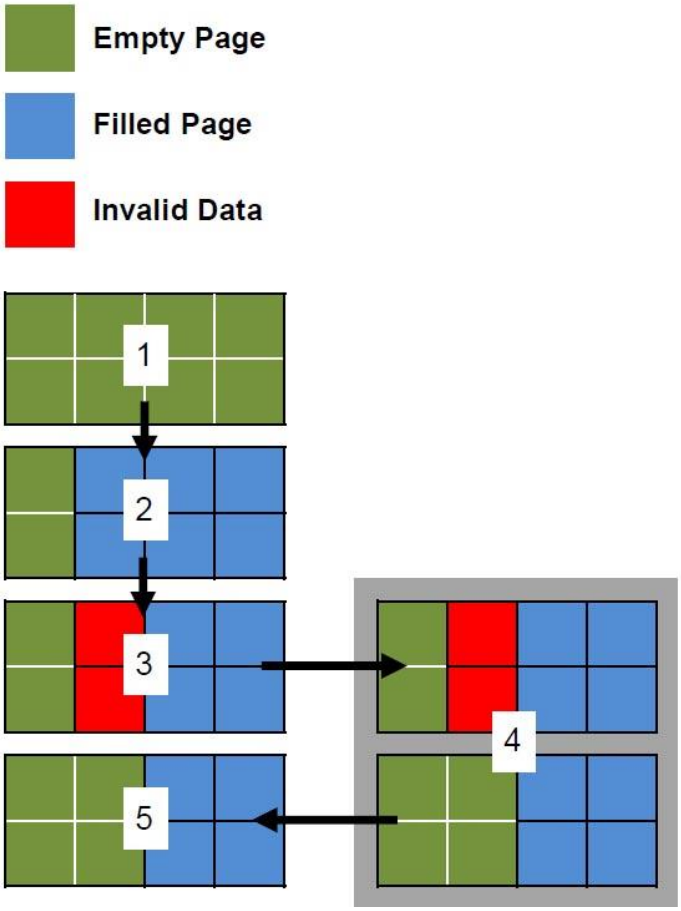
When analyzing a live system, it is easy to check a TRIM status for a particular SSD device by issuing the following command in a terminal window:

fsutil behavior query disabledeletenotify

You'll get one of the following results:
DisableDeleteNotify = 1 meaning that Windows TRIM commands are **disabled**
DisableDeleteNotify = 0 meaning that Windows TRIM commands are **enabled**

fsutil is a standard tool in Windows 7, 8, and 8.1.

On a side note, it is possible to enable TRIM with "*fsutil behavior set disabledeletenotify 0*" or disable TRIM with "*fsutil behavior set disabledeletenotify 1*".



1.) SSD pages contain no data

2.) User writes data to SSD pages

3.) User deletes some data. Pages are marked as 'not in use' by the host OS, but data remains on SSD.

4.) TRIM command tells SSD controller that pages contain invalid data. Pages with invalid data are cleaned.

5.) Data is written back to SSD memory cells. The invalid data has been cleaned and data is able to be written to the pages at full speed.

Figure 1. TRIM, image taken from http://www.corsair.com/us/blog/how-to-check-that-trim-is-active/

Note that using this command only makes sense if analyzing the SSD which is still installed in its original computer (e.g. during a live box analysis). If the SSD drive is moved to a different system, the results of this command are no longer relevant.

SSD Technology: 2014

Back in 2012, practically all SSD drives were already equipped with background garbage collection technology and recognized the TRIM command. This did not changed in 2014.

Two years ago, SSD compression already existed in SandForce SSD Controllers (http://en.wikipedia.org/wiki/SandForce). However, relatively few models were equipped with encrypting or compressing controllers. As SandForce remained the only compressing controller, it was easy to determine whether it was the case.

(http://www.enterprisestorageforum.com/technology/features/article.php/3930601/Real-Time-Data-Compressions-Impact-on--SSD-Throughput-Capability-.htm).

In 2013, Intel used a custom-firmware controlled version of a SandForce controller to implement data compression in 3xx and 5xx series SSDs (http://www.intel.com/support/ssdc/hpssd/sb/CS-034537.htm), claiming reduced write amplification and increased endurance of a SSD as the inherent benefits (http://www.intel.de/content/dam/www/public/us/en/documents/technology-briefs/ssd-520-tech-brief.pdf).

Marvell controllers are still non-compressing (http://blog.goplextor.com/?p=3313), and so are most other controllers on the market including the new budget option, Phison.

Why so much fuzz about data compression in SSD drives? Because the use of any technology altering binary data before it ends up in the flash chips makes its recovery with third-party off-chip hardware much more difficult. Regardless of whether compression is present or not, we have not seen many successful implementations of SSD off-chip acquisition products so far, TEEL Tech (http://www.teeltech.com/mobile-device-forensics-training/advanced-bga-chip-off-forensics/) being one of rare exceptions.

Let's conclude this chapter with a quote from PC World:

> *"The bottom line is that SSDs still are a capacity game: people buy the largest amount of storage they can within their budget, and they ignore the rest".*
> *http://www.pcworld.com/article/2087480/ssd-prices-face-uncertain-future-in-2014.html*

In other words, SSD's get bigger and cheaper, inevitably demanding some cost-saving measures which, in turn, may affect how deleted data are handled on these SSD drives in a way described later in the Reality Steps In: Why Real SSDs are Often Recoverable chapter.

SSD Manufacturers

In recent years, we've seen a lot of new SSD "manufacturers" entering the arena. These companies don't normally build their own hardware or design their own firmware. Instead, they simply spec out the disks to a real manufacturer that assembles the drives based on one or another platform (typically, SandForce or Phison) and one or another type, make and size of flash memory**.** In the context of SSD forensics, these drives are of interest exactly because they all feature a limited choice of chipsets and a limited number of firmware revisions. In fact, just two chipset makers, SandForce and Phison, enabled dozens of "manufacturers" make hundreds of nearly indistinguishable SSD models.

So who are the real makers of SSD drives?

According to Samsung, we have the following picture:

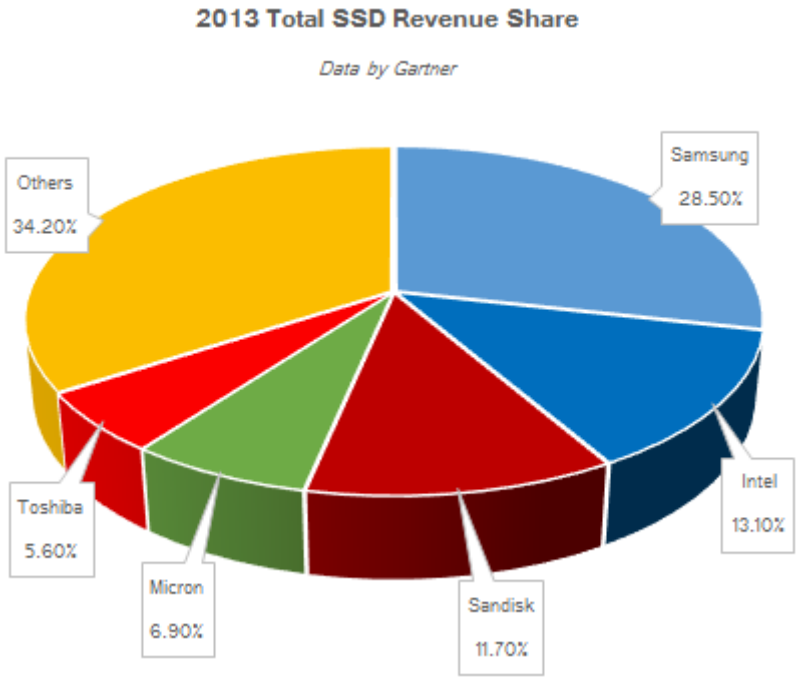


Figure 2. Source: http://www.kitguru.net/components/ssd-drives/anton-shilov/samsung-remains-the-worlds-largest-maker-of-ssds-gartner/

Hardware for SSD Forensics (and Why It Has Not Arrived)

Little has changed since 2012 in regards to SSD-specific acquisition hardware. Commonly available SATA-compliant write-blocking forensic acquisition hardware is used predominantly to image SSD drives, with BGA flash chip acquisition kits rare as hen's teeth.

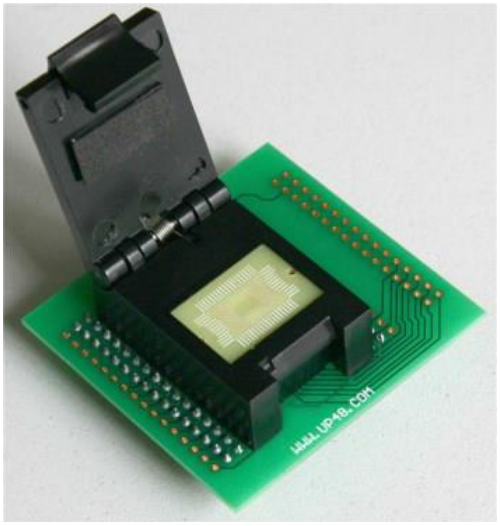


Figure 3. TEEL Tech BGA Acquisition Toolkit

Why so few chip-off solutions for SSD drives compared to the number of companies doing mobile chip-off? It's hard to say for sure, but it's possible that most digital forensic specialists are happy with what they can extract via the SATA link (while there is no similar interface in most mobile devices). Besides, internal data structures in today's SSD drives are extremely complex. Constant remapping and shuffling of data during performance and lifespan optimization routines make actual data content stored on the flash chips inside SSD drives heavily fragmented. We're not talking about logical fragmentation on file system level (which already is a problem as SSD drives are never logically defragmented), but rather physical fragmentation that makes an SSD controller scatter data blocks belonging to a contiguous file to various physical addresses on numerous physical flash chips. In particular, massive parallel writes are what make SSD drives so much faster than traditional magnetic drives (as opposed to sheer writing speed of single flash chips).

One more word regarding SSD acquisition hardware: write-blocking devices. Note that write-blocking imaging hardware does not stop SSD self-corrosion. If the TRIM command has been issued, the SSD drive will continue erasing released data blocks at its own pace. Whether or not some remnants of deleted data can be acquired from the SSD drive depends as much on acquisition technique (and speed), as on particular implementation of a particular SSD controller.

Deterministic Read After Trim

So let's say we know that the suspect erased important evidence or formatted the disk just minutes before arrest. The SSD drive has been obtained and available for imaging. What exactly should an investigator expect to obtain from this SSD drive?

Reported experience while recovering information from SSD drives varies greatly among SSD users.

"I ran a test on my SSD drive, deleting 1000 files and running a data recovery tool 5 minutes after. The tool discovered several hundred files, but an attempt to recover returned a bunch of empty files filled with zeroes", said one Belkasoft customer.

"We used Belkasoft Evidence Center to analyze an SSD drive obtained from the suspect's laptop. We were able to recover 80% of deleted files in several hours after they've been deleted", said another Belkasoft user.

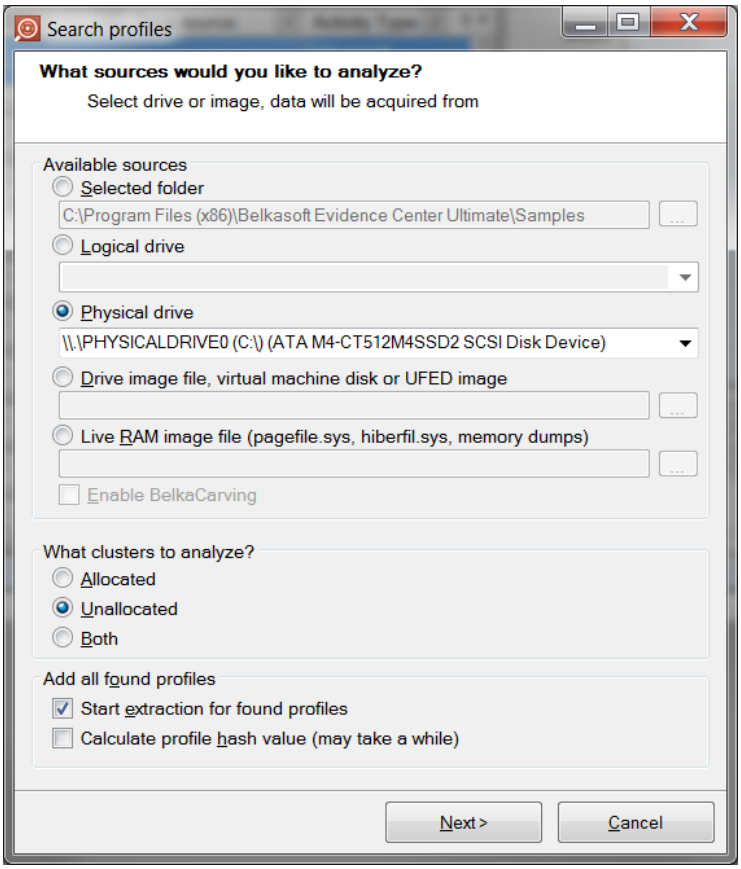


Figure 4. Carving options in Belkasoft Evidence Center: for the experiment we set Unallocated clusters only and SSD drive connected as physical drive 0.

Why such a big inconsistency in user experiences? The answer lies in the way the different SSD drives handle trimmed data pages.

Some SSD drives implement what is called Deterministic Read After Trim (DRAT) and Deterministic Zeroes After Trim (DZAT), returning all-zeroes immediately after the TRIM command released a certain data block, while some other drives do not implement this protocol and will return the original data until it's physically erased with the garbage collection algorithm.

Deterministic Read After Trim and Deterministic Zeroes After Trim have been part of the SATA specification for a long time. Linux users can verify that their SSD drives are using DRAT or DZAT

by issuing the `hdparm -I` command returning whether the drive supports TRIM and does "Deterministic Read After Trim".

Example:

```
$ sudo hdparm -I /dev/sda | grep -i trim
           *    Data Set Management TRIM supported (limit 1 block)
           *    Deterministic read data after TRIM
```

However, the adoption of DRAT has been steadily increasing among SSD manufacturers. Two years ago we often saw reports on SSD drives with and without DRAT support. In 2014, the majority of new models came equipped with DRAT or DZAT.

There are three different types of TRIM defined in the SATA protocol and implemented in different SSD drives.

- Non-deterministic TRIM: each read command after a Trim may return different data.
- Deterministic Trim (DRAT): all read commands after a TRIM shall return the same data, or become determinate. Note that this level of TRIM does not necessarily return all-zeroes when trimmed pages are accessed. Instead, DRAT guarantees that the data returned when accessing a trimmed page will be the same (“determined”) before and after the affected page has been processed by the garbage collection algorithm and until the page is written new data. As a result, the data returned by SSD drives supporting DRAT as opposed to DZAT can be all zeroes or other words of data, or it could be the original pre-trim data stored in that logical page. The essential point here is that the values read from a trimmed logical page do not change since the moment the TRIM command has been issued and before the moment new data get written into that logical page.
- Deterministic Read Zero after Trim (DZAT): all read commands after a TRIM shall return zeroes until the page is written new data.

As we can see, in some cases the SSD will return non-original data (all zeroes, all ones, or some other non-original data) not because the physical blocks have been cleaned immediately following the TRIM command, but because the SSD controller tells that there is no valid data held at the trimmed address on a logical level previously associated with the trimmed physical block.

If, however, one could possibly read the data directly from the physical blocks mapped to the logical blocks that have been trimmed, then the original data could be obtained from those physical blocks until the blocks are physically erased by the garbage collector. Apparently, there is no way to address the physical data blocks via the standard ATA command set, however, the disk manufacturer could most probably do this in their own lab. As a result, sending the

trimmed SSD disk for recovery to the manufacturer may be a viable proposition if some extremely important evidence is concerned.

Notably, DRAT is not implemented in Windows, as NTFS does not allow applications reading the trimmed data.

Acquiring Evidence from SSD Drives

So far the only practical way of obtaining evidence from an SSD drive remains the traditional imaging (with dedicated hardware/software combination), followed by an analysis with an evidence discovery tool (such as Belkasoft Evidence Center, http://forensic.belkasoft.com/en/bec/en/evidence_center.asp).

We now know more about the expected outcome when analyzing an SSD drive. There are generally two scenarios: either the SSD only contains existing data (files and folders, traces of deleted data in MFT attributes, unallocated space carrying no information), or the SSD contains the full information (destroyed evidence still available in unallocated disk space).Today, we can predict which scenario is going to happen by investigating conditions in which the SSD drive has been used.

Scenario 1: Existing Files Only

In this scenario, the SSD may contain some files and folders, but free disk space will be truly empty (as in "filled with zero data"). As a result, carving free disk space will return no information or only traces of information, while carving the entire disk space will only return data contained in existing files. So, is file carving useless on SSD drives? No way! Carving is the only practical way of locating moved or hidden evidence (e.g. renamed history files or documents stored in the Windows\System32 folder and renamed to .SYS or .DLL).

Practically speaking, the same acquisition and analysis methods should be applied to an SSD drive as if we were analyzing a traditional magnetic disk. Granted, we'll recover no or little destroyed evidence, but any evidence contained in existing files including e.g., deleted records from SQLite databases (used, for example, in Skype histories) can still be recovered (http://forensic.belkasoft.com/en/recover-destroyed-sqlite-evidence-skype-and-iphone-logs).

Scenario 2: Full Disk Content

In the second scenario, the SSD disk will still contain the complete set of information – just like traditional magnetic disks. Obviously, all the usual techniques should be applied at the analysis stage including file carving.

Why would an SSD drive NOT destroy evidence as a result of routine garbage collection? The garbage collection algorithm erasing the content of released data blocks does not occur if the TRIM command has not been issued, or if the TRIM protocol is not supported by any link of the chain. Let's see in which cases this could happen.

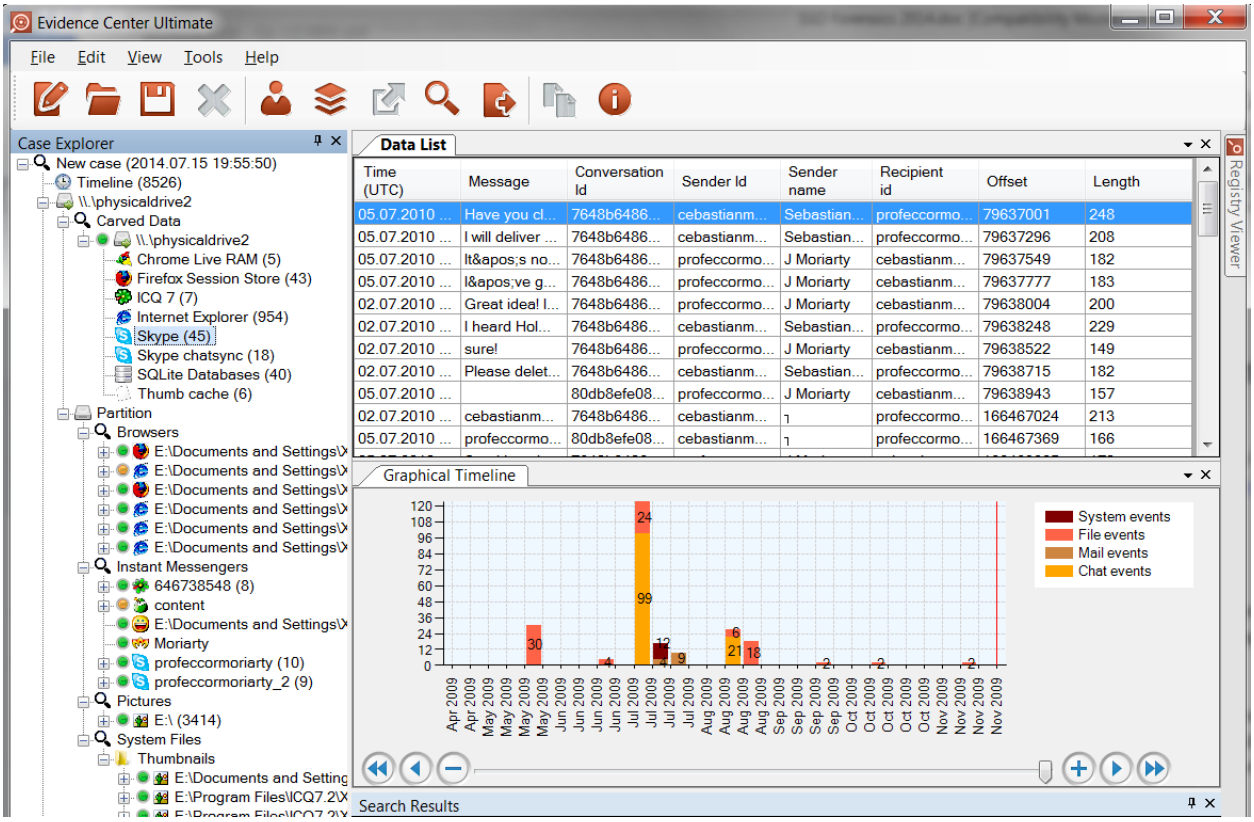


Figure 5. More than 1000 items were carved out of unallocated sectors of SSD hard drive, particularly, Internet Explorer history, Skype conversations, SQLite databases, system files and other forensically important types of data

Operating System Support

TRIM is a property of the operating system as much as it is the property of an SSD device. Older file systems do not support TRIM. Wikipedia http://en.wikipedia.org/wiki/Trim_(computing) has a comprehensive table detailing the operating system support for the TRIM command.

Operating System	Supported since	Notes
DragonFly BSD	2011-05 May 2011	
FreeBSD	2010-078.1 - July 2010	Support was added at the block device layer in 8.1. File system support was added in FreeBSD 8.3 and FreeBSD 9, beginning with UFS. ZFS trimming support was added in FreeBSD 9.2. FreeBSD 10 will support trimming on software RAID configurations.

Linux	2008-12-252.6.28–25 December 2008	Initial support for discard operations was added for FTL NAND flash devices in 2.6.28. Support for the ATA Trim command was added in 2.6.33. Not all file systems make use of Trim. Among the file systems that can issue Trim requests automatically are Ext4, Btrfs, FAT, GFS2 and XFS. However, this is disabled by default, due to performance concerns, but it can be enabled by setting the "discard" mount option. Ext3, NILFS2 and OCFS2 offer ioctls to perform offline trimming. The Trim specification calls for supporting a list of trim ranges, but as of kernel 3.0 trim is only invoked with a single range that is slower.
Mac OS X	2011-06-2310.6.8 – 23 June 2011	Although the AHCI block device driver gained the ability to display whether a device supports the Trim operation in 10.6.6 (10J3210), the functionality itself remained inaccessible until 10.6.8, when the Trim operation was exposed via the IOStorageFamily and file system (HFS+) support was added. Some online forums state that Mac OS X only supports Trim for Apple-branded SSDs; third-party utilities are available to enable it for other brands.
Microsoft Windows	2009-10NT 6.1 (Windows 7 and Windows Server 2008 R2) – October 2009	Windows 7 only supports trim for ordinary (SATA) drives and does not support this command for PCI-Express SSDs that are different type of device, even if the device itself would accept the command. It is confirmed that with native Microsoft drivers the Trim command works in AHCI and legacy IDE / ATA Mode.
OpenSolaris	2010-07 July 2010	
Android	2013-74.3 - 24 July 2013	

Old Versions of Windows

As shown in the table above, TRIM support was only added to Windows 7. Obviously, TRIM is supported in Windows 8 and 8.1. In Windows Vista and earlier, the TRIM protocol is not supported, and the TRIM command is not issued. As a result, when analyzing an SSD drive obtained from a system featuring one of the older versions of Windows, it is possible to obtain the full content of the device.

- **Possible exception:** TRIM-like performance can be enabled via certain third-party solutions (e.g. Intel SSD Optimizer, a part of Intel SSD Toolbox).

MacOS X

Mac OS X started supporting the TRIM command for Apple supplied SSD drives since version 10.6.8. Older builds of Mac OS X do not support TRIM.

Notably, user-installed SSD drives not supplied by Apple itself are excluded from TRIM support.

Old or Basic SSD Hardware

Not all SSD drives support TRIM and/or background garbage collection. Older SSD drives as well as SSD-like flash media used in basic tablets and sub-notes (such as certain models of ASUS Eee) do not support the TRIM command. For example, Intel started manufacturing TRIM-enabled SSD drives with drive lithography of 34nm (G2); their 50nm SSDs do not have TRIM support.

In reality, few SSD drives without TRIM survived that long. Many entry-level sub-notebooks use flash-based storage often mislabeled as "SSD" that does not feature garbage collection or supports the TRIM protocol.

(Windows) File Systems Other than NTFS

TRIM is a feature of the file system as much as the property of an SSD drive. At this time, Windows only supports TRIM on NTFS-formatted partitions. Volumes formatted with FAT, FAT32 and exFAT are excluded.

Notably, some (older) SSD drives used trickery to work around the lack of TRIM support by trying to interpret the file system, attempting to erase dirty blocks not referenced from the file system. This approach, when enabled, only works for the FAT file system since it's a published spec.
(http://www.snia.org/sites/default/files2/sdc_archives/2009_presentations/thursday/NealChristiansen_ATA_TrimDeleteNotification_Windows7.pdf)

External drives, USB enclosures and NAS

The TRIM command is fully supported over the SATA interface, including the eSATA extension, as well as SCSI via the UNMAP command. If an SSD drive is used in a USB enclosure or installed in most models of NAS devices, the TRIM command will not be communicated via the unsupported interface.

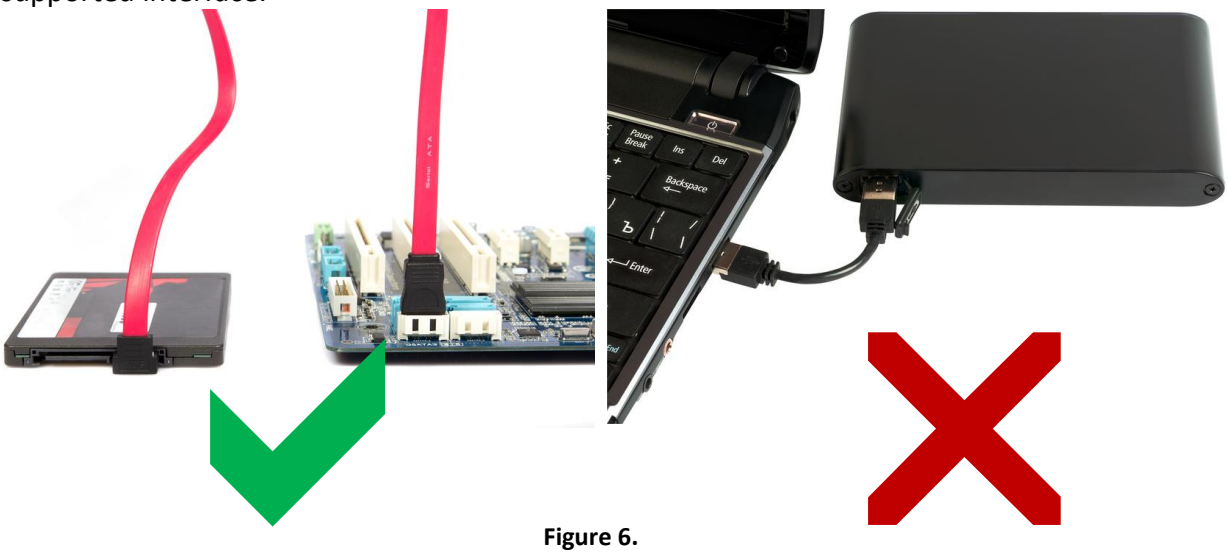


Figure 6.

There is a notable exception. Some NAS manufacturers start recognizing the demand for units with ultra-high performance, low power consumption and noise free operation provided by SSD drives, slowly adopting TRIM in some of their models. At the time of this writing, of all manufacturers only Synology appears to support TRIM in a few select models of NAS devices and SSD drives.

Here is a quote from Synology Web site (https://www.synology.com/en-uk/support/faq/591):

> *SSD TRIM improves the read and write performance of volumes created on SSDs, increasing efficiency as well as extending the lifetime of your SSDs. See the list below for verified SSD with TRIM support.*
>
> - *You may customize a schedule to choose when the system will perform TRIM.*
> - *SSD TRIM is not available when an SHA cluster exists.*
> - *TRIM cannot be enabled on iSCSI LUN.*
> - *The TRIM feature under RAID 5 and 6 configurations can only be enabled on the SSDs with DZAT (Deterministic Read Zero after TRIM) support. Please contact your SSD manufacturers for details on DZAT support.*

PCI-Express and PCIe SSDs

Interestingly, the TRIM command is not natively supported by any version of Windows for many high-performance SSD drives occupying the PCI Express slot. Do not confuse PCI Express SSD's with SATA drives carrying M.2 or mSATA interfaces.

- **Possible exception:** TRIM-like performance can be enabled via certain third-party solutions (e.g., Intel SSD Optimizer, a part of Intel SSD Toolbox).

RAID

The TRIM command is not yet supported over RAID configurations (with few rare exceptions). SSD drives working as part of a RAID array can be analyzed.

A notable exception from this rule would be the modern RAID 0 setup using a compatible chipset (such as Intel H67, Z77, Z87, H87, Z68) accompanied with the correct drivers (the latest RST driver from Intel allegedly works) and a recent version of BIOS. In these configurations, TRIM can be enabled.

Corrupted Data

Surprisingly, SSD drives with corrupted system areas (damaged partition tables, skewed file systems, etc.) are easier to recover than healthy ones. The TRIM command is not issued over corrupted areas because files are not properly deleted. They simply become invisible or inaccessible to the operating systems. Many commercially available data recovery tools (e.g., Intel® Solid-State Drive Toolbox with Intel® SSD Optimizer, OCZ SSD Toolbox) can reliably extract information from logically corrupted SSD drives.

Bugs in SSD Firmware

Firmware used in SSD drives may contain bugs, often affecting the TRIM functionality and/or messing up garbage collection. Just to show an example, OCZ Agility 3 120 GB shipped with buggy firmware v. 2.09, in which TRIM did not work. Firmware v. 2.15 fixed TRIM behavior, while v. 2.22 introduced issues with data loss on wake-up after sleep, then firmware v. 2.25 fixed that but disrupted TRIM operation again (information taken from http://www.overclock.net/t/1330730/ocz-firmware-2-25-trim-doesnt-work-bug-regression-bad-ocz-experience). A particular SSD drive may or may not be recoverable depending on which bugs were present in its firmware.

Bugs in SSD Over-Provisioning

SSD over-provisioning is one of the many wear-leveling mechanisms intended for increasing SSD life span. Some areas on the disk are reserved on the controller level, meaning that a 120 GB SSD drive carries more than 120 GB of physical memory. These extra data blocks are called over-provisioning area (OP area), and can be used by SSD controllers when a fresh block is

required for a write operation. A dirty block will then enter the OP pool, and will be erased by the garbage collection mechanism during the drive's idle time.

Speaking of SSD over-provisioning, firmware bugs can affect TRIM behavior in other ways, for example, revealing trimmed data after a reboot/power off. Solid-state drives are remapping constantly after TRIM to allocate addresses out of the OP pool. As a result, the SSD reports a trimmed data block as writeable (already erased) immediately after TRIM. Obviously, the drive did not have the time to actually clean old data from that block. Instead, it simply maps a physical block from the OP pool to the address referred to by the trimmed logical block. What happens to the data stored in the old block? For a while, it contains the original data (in many cases it's compressed data, depending on the SSD controller). However, as that data block is mapped out of the addressable logical space, the original data is no longer accessible or addressable.

Sounds complex? You bet. That's why even seasoned SSD manufacturers may not get it right at the first try (e.g. as discussed on OCZ Technology Forum at http://www.ocztechnologyforum.com/forum/showthread.php?96382-Deterministic-Read-After-Trim). Issues like this cause issues when, after deleting data and rebooting the PC, some users would see the old data back as if it was never deleted. Apparently, because of the mapping issue the new pointers would not work as they should, due to a bug in the drive's firmware. OCZ released a firmware fix to correct this behavior, but similar (or other) bugs may still affect other drives.

SSD Shadiness: Manufacturers Bait-and-Switch

When choosing an SSD drive, customers tend to read online reviews. Normally, when the new drive gets released, it is getting reviewed by various sources soon after it becomes available. The reviews get published, and customers often base their choice on them.

But what if a manufacturer silently changes the drive's specs without changing the model number? In this case, an SSD drive that used to have great reviews suddenly becomes much less attractive. This is exactly what happened with some manufacturers. According to ExtremeTech (http://www.extremetech.com/extreme/184253-ssd-shadiness-kingston-and-pny-caught-bait-and-switching-cheaper-components-after-good-reviews), two well-known SSD manufacturers, Kingston and PNY, were caught bait-and-switching cheaper components after getting the good reviews. In this case, the two manufacturers were launching their SSDs with one hardware specification, and then quietly changed the hardware configuration after reviews have gone out.

So what's in there for us? Well, the forensic-friendly SandForce controller was found in the second revision of PNY Optima drives. Instead of the original Silicon Motion controller, the new batch of PNY Optima drives had a different, SandForce-based controller known for its less-than-

perfect implementation of garbage collection leaving data on the disk for a long time after it's been deleted.

Small Files: Slack Space

Remnants of deleted evidence can be acquired from so-called slack space as well as from MFT attributes.

In the word of SSD, the term "slack space" receives a new meaning. Rather than being a matter of file and cluster size alignment, "slack space" in SSD drives deals with the different sizes of minimum writeable and minimum erasable blocks on a physical level.

Micron, the manufacturer of NAND chips used in many SSD drives, published a comprehensive article on SSD structure:
https://www.micron.com/~/media/Documents/Products/Technical%20Marketing%20Brief/ssd_effect_data_placement_writes_tech_brief.pdf

In SSD terms, *Page* is the smallest unit of storage that can be written to. The typical page size of today's SSD is 4 KB or 8 KB.

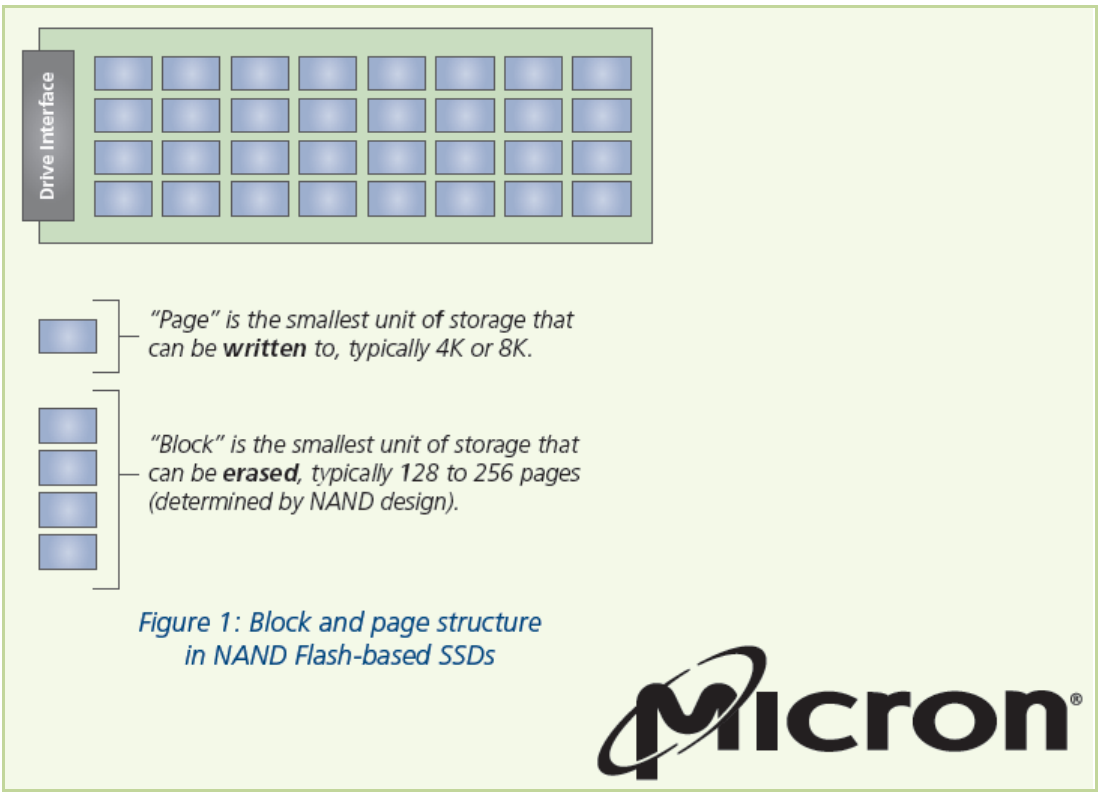


Figure 7.

Block, on the other hand, is the smallest unit of storage that can be erased. Depending on the design of a particular SSD drive, a single block may contain 128 to 256 pages.

As a result, if a file is deleted and its size is less than the size of a single SSD data block, OR if a particular SSD data block contain *pages* that still remain allocated, that particular block is NOT erased by the garbage collection algorithm. In practical terms, this means that files or file fragments (chunks) sized less than 512 KB or less than 2 MB depending on SSD model, may not be affected by the TRIM command, and may still be forensically recoverable.

However, the implementation of the Deterministic Read After Trim (DRAT) protocol by many recent SSD drives makes trimmed pages inaccessible via standard SATA commands. If a particular SSD drive implements DRAT or DZAT (Deterministic Read Zero After Trim), the actual data may physically reside on the drive for a long time, yet it will be unavailable to forensic specialists via standard acquisition techniques. Sending the SSD drive to the manufacturer might be the only way of obtaining this information on a physical level.

Small Files: MFT Attributes

Most hard drives used in Windows systems are using NTFS as their file system. NTFS stores information about the files and directories in the Master File Table (MFT). MFT contains information about all files and directories listed in the file system. In other words, each file or directory has at least one record in MFT.

In terms of computer forensics, one particular feature of MFT is of great interest. Unique to NTFS is the ability to store small files directly in the file system. The entire content of a small file can be stored as an attribute inside an MFT record, greatly improving reading performance and decreasing wasted disk space (“slack” space), referenced in the previous chapter.

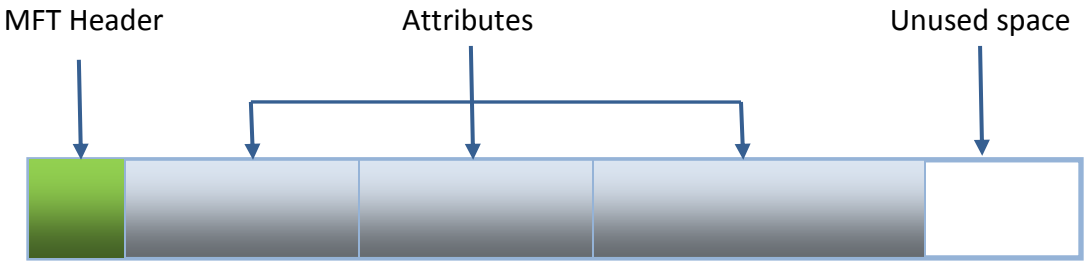


Figure 8. MFT record

As a result, small files being deleted are not going anywhere. Their entire content continues residing in the file system. The MFT records are not emptied, and are not affected by the TRIM command. This in turn allows investigators recovering such resident files by carving the file system.

How small does a file have to be to fit inside an MFT record? Very small. The maximum size of a resident file cannot exceed 982 bytes. Obviously, this severely limits the value of resident files for the purpose of digital forensics.

Encrypted Volumes

Somewhat counter-intuitively, information deleted from certain types of encrypted volumes (some configurations of BitLocker, TrueCrypt, PGP and other containers) may be easier to recover as it may not be affected by the TRIM command. Files deleted from such encrypted volumes stored on an SSD drive can be recovered (unless they were specifically wiped by the user) if the investigator knows either the original password or binary decryption keys for the volume. Encrypted containers are a big topic, so we'll cover it in a dedicated chapter.

TRIM on encrypted volumes is a huge topic well worth a dedicated article or even a series of articles. With the large number of crypto containers floating around and all the different security considerations and available configuration options, determining whether TRIM was enabled on a particular encrypted volume is less than straightforward. Let's try assembling a brief summary on some of the most popular encryption options.

Apple FileVault 2

Introduced with Apple OS X "Lion", FileVault 2 enables whole-disk encryption. More precisely, FileVault 2 enables whole-volume encryption only on HFS+ volumes (Encrypted HFS). Apple chose to enable TRIM with FileVault 2 volumes on drives. It has the expected security implication of free sectors/blocks being revealed.

Microsoft BitLocker

Microsoft has its own built-in version of volume-level encryption called BitLocker. Microsoft made the same choice as Apple, enabling TRIM on BitLocker volumes located on SSD drives. As usual for Microsoft Windows, the TRIM command is only available on NTFS volumes.

TrueCrypt

TrueCrypt supports TRIM pass-through on encrypted volumes located on SSD drives. The company issued several security warnings in relation to wear-levelling security issues and the TRIM command revealing information about which blocks are in use and which are not. (http://www.truecrypt.org/docs/trim-operation and http://www.truecrypt.org/docs/wear-leveling)

PGP Whole Disk Encryption

By default, PGP whole-disk encryption does not enable TRIM on encrypted volumes. However, considering wear-leveling issues of SSD drives, Symantec introduced an option to enable TRIM on SSD volumes via a command line option: --fast (http://www.symantec.com/connect/forums/pgp-and-ssd-wear-leveling).

If an encrypted volume of a fixed size is created, the default behavior is also to encrypt the entire content of a file representing the encrypted volume, which disables the effect of the TRIM command for the contents of the encrypted volume.

More research is required to investigate these options. At this time one thing is clear: in many configurations, including default ones, files deleted from encrypted volumes will not be affected by the TRIM command. Which brings us to the question of the correct acquisition of PCs with encrypted volumes.

Forensic Acquisition: The Right Way to Do

The right way to acquire a PC with a crypto container can be described with the following sentence: “If it’s running, don’t turn it off. If it’s off, don’t turn it on.” Indeed, the original decryption keys are cached in the computer’s memory, and can be extracted from a LiveRAM dump obtained from a running computer by performing a FireWire attack. These keys can be contained in page files and hibernation files. Tools such as Passware can extract decryption files from memory dumps and page/hibernation files, decrypting the content of encrypted volumes.

Reality Steps In: Why Real SSDs are Often Recoverable

In reality, things may look different from what was just described above in such great technical detail. In our lab, we’ve seen hundreds of SSD drives acquired from a variety of computers. Surprisingly, Belkasoft Evidence Center was able to successfully carve deleted data from the majority of SSD drives taken from inexpensive laptops and sub-notebooks such as ASUS Eee or ASUS Zenbook. Why is it so? There are several reasons to that, mainly “cost savings” and “miniaturization”, but sometimes it’s simply over-engineering.

1. Inexpensive laptops often use flash-based storage, calling that an SSD in their marketing ploy. In fact, in most cases it's just a slow, inexpensive and fairly small flash-based storage having nothing to do with real SSD drives.
2. Ultrabooks and sub-notes have no space to fit a full-size SSD drive. They used to use SSD drives in PCIe form factor (as opposed to M.2 or mSATA) which did not support the SATA protocol. Even if these drives are compatible with the TRIM protocol, Windows does not support TRIM on non-ATA devices. As a result, TRIM is not enabled on these drives.
3. SSD drives are extremely complex devices requiring extremely complex firmware to operate. Many SSD drives were released with buggy firmware effectively disabling the effects of TRIM and garbage collection. If the user has not upgraded their SSD firmware to a working version, the original data may reside on an SSD drive for a long time.
4. The fairly small (and inexpensive) SSD drives used in many entry-level notebooks lack support for DRAT/DZAT. As a result, deleted (and trimmed) data remain accessible for a long time, and can be successfully carved from a promptly captured disk image.
5. On the other end of the spectrum are the very high-end, over-engineered devices. For example, Acer advertises its Aspire S7-392 as having a RAID 0 SSD. According to Acer marketing, "RAID 0 solid state drives are up to 2X faster than conventional SSDs. Access your files and transfer photos and movies quicker than ever!" (http://www.acer.com/aspires7/en_US/). This looks like over-engineering. As TRIM is not enabled on RAID SSD's in any version of Windows, this ultra-fast non-conventional storage system may slow down drastically over time (which is exactly why TRIM was invented in the first place). For us, this means that any data deleted from these storage systems could remain there for at least as long as it would have remained on a traditional magnetic disk. Of course, the use of the right chipset (such as Intel H67, Z77, Z87, H87, Z68) accompanied with the correct drivers (the latest RST driver from Intel allegedly works) can in turn enable TRIM back. However, we are yet to see how this works in reality. (http://www.anandtech.com/show/6477/trim-raid0-ssd-arrays-work-with-intel-6series-motherboards-too)

Conclusion

SSD forensics remains different. SSDs self-destroy court evidence, making it difficult to extract deleted files and destroyed information (e.g., from formatted disks) is close to impossible. Numerous exceptions still exist, allowing forensic specialists to access destroyed evidence on SSD drives used in certain configurations.

There has been little progress in SSD development since the publication of our last article on SSD forensics in 2012. The factor defining the playing field remains delivering bigger size for less money. That aside, compressing SSD controllers appear to become the norm, making off-chip acquisition unpractical and killing all sorts of DIY SSD acquisition hardware.

More SSD drives appear to follow the Deterministic Read After Trim (DRAT) approach defined in the SATA standard a long time ago. This in turn means that a quick format is likely to instantly render deleted evidence inaccessible to standard read operations, even if the drive is acquired with a forensic write-blocking imaging hardware immediately after.

SSD drives are getting more complex, adding over-provisioning support and using compression for better performance and wear leveling. However, because of the increased complexity, even seasoned manufacturers released SSD drives with buggy firmware, causing improper operation of TRIM and garbage collection functionality. Considering just how complex today's SSD drives have become, it's surprising these things do work, even occasionally.

The playfield is constantly changing, but what we know now about SSD forensics gives hope.

About the Authors

Yuri Gubanov is a renowned computer forensics expert. He is a frequent speaker at industry-known conferences such as CEIC, HTCIA, FT-Day, DE-Day, ICDDF, TechnoForensics and others. Yuri is the Founder and CEO of Belkasoft, the manufacturer of computer forensic software empowering police departments in more than 60 countries. With years of experience in digital forensics and security domain, Yuri led forensic training courses for multiple law enforcement departments in several countries.

You can reach Yuri Gubanov at yug@belkasoft.com or add him to your LinkedIn network at http://linkedin.com/in/yurigubanov.

Oleg Afonin is an expert in digital forensics and Belkasoft marketing director. You can reach Oleg Afonin at research@belkasoft.com.