University of Piraeus

Department of Digital Systems

Forensics research in Solid State Drives

Master Thesis

Of the Student: Dekleris Dimitrios

(A.M. 1507)

Supervisor: Christos Xenakis

Athens , June 2017

Contents:

ABSTRACT

In our days more and more the SSDs are entering our life. The advantage of providing us faster with information, the quicker elaboration of data, the small size and the cost decreasing year after year, makes it tempting for someone to buy it. They definitely work different from the HDDs and so we are facing new challenges in the sector of forensics with these disks.

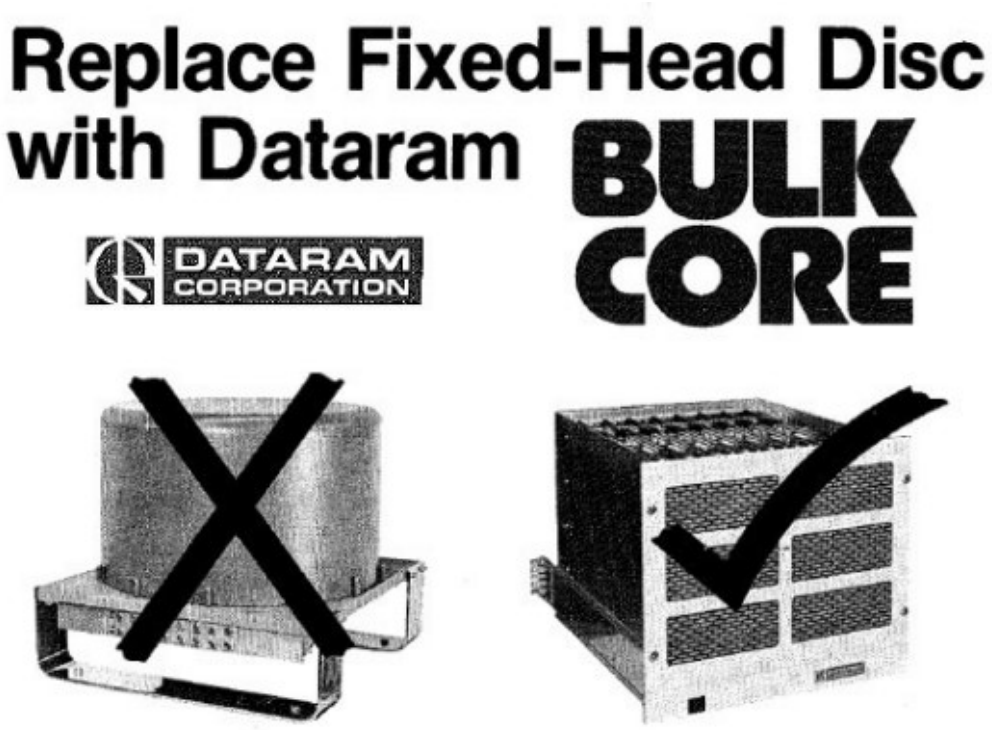
The purpose of this paper is to give the reader one more part in the foggy picture of what is going on nowadays in the field of forensics in SSDs. Reading this paper you will get a lot of information and knowledge around the SSDs and what happens in the forensic part. In order to achieve that, after providing our reader with the proper information, we are checking some possible scenarios of finding evidence in our disk even after a quick format, the results will surprise you!

The History of SSD

It has been about 67 years since the first idea or the ancestor, someone could say , of the Solid State Drive walked on earth. Year 1950, two contiguous technologies appear in the market, the CCROS (charged capacitor read only storage) and magnetic core memory. They were born in an era of vacuum tube computers and lasted until the low cost drum storage components appeared.

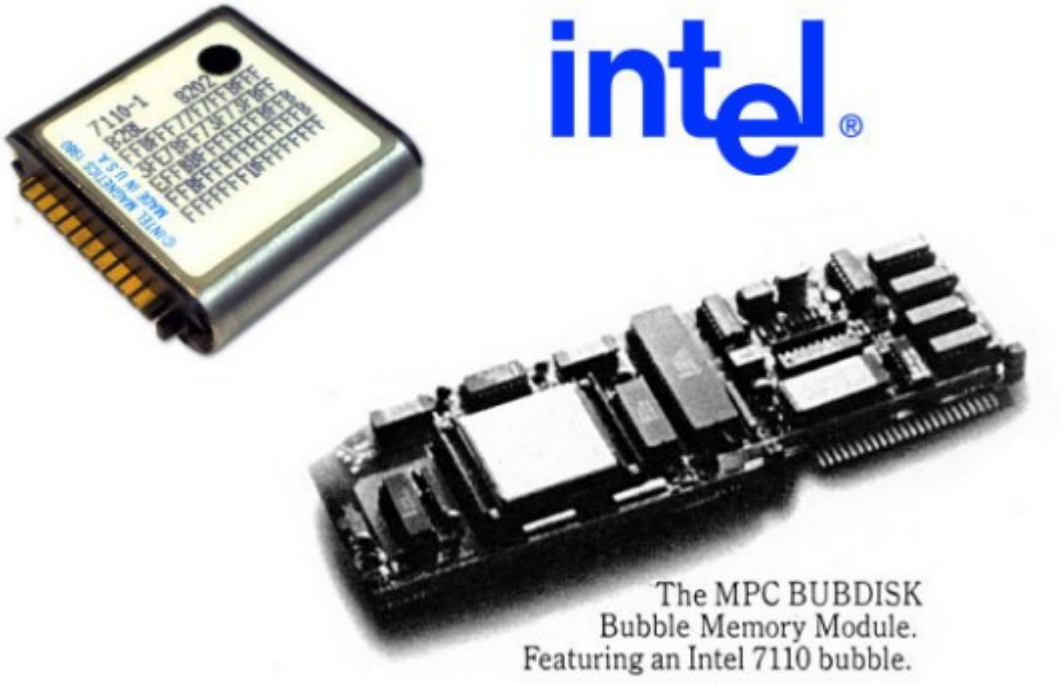
During the 70's we meet the first milestone in the history of SSDs. Thanks to the evolution of technology, we can see a huge stream of implementation at supercomputer's semiconductor memory (such as IBM and Amdahl). In the late 70's General Instruments gave to the market the silicon nitride EAROM (electrically alterable ROM) which were working somehow like today's NAND flash memories, but through practice it was proved that this type of product could not survive the test for about 10 years life and it was abandoned from the companies.

The first SSD actually appears in 1976 by Dataram, which introduces to the world the "BULK Core" and it looked like this!

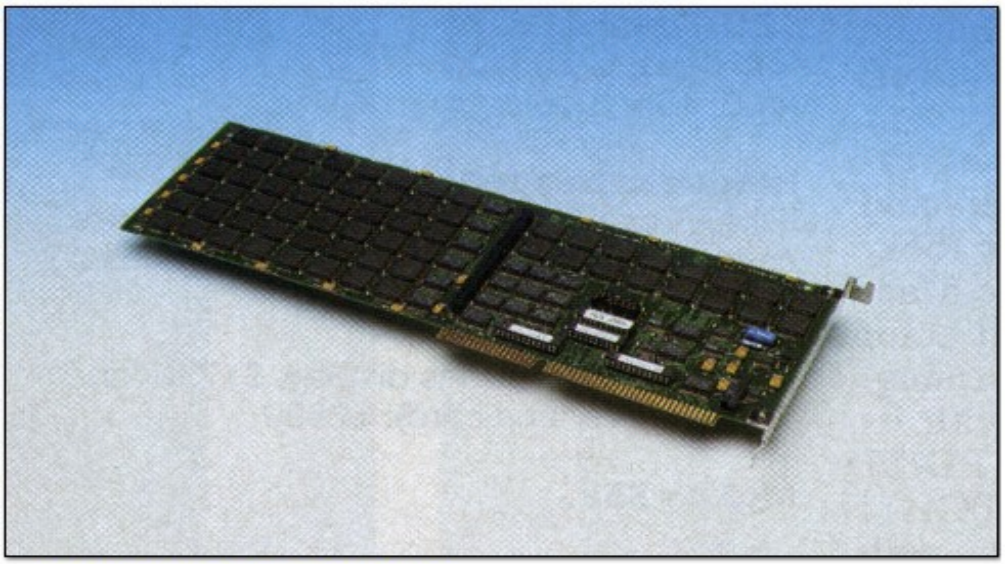


The need for faster but cheaper SSDs was the reason for evolution, imagine that the Bulk Core system was providing a massive storage of about 2MB for minicomputers, the data access time was ranged from 0.75 ms to 2 ms while today SSDs have 0.06ms, also consider that the cost for 256KB of storage, back these days, would be something around 9.700$ and nowadays would be about 40.000$. The Bulk Core was compatible with Digital Equipment Corporation and Data general.
In 1978 the company named Texas Memory Systems produced a bigger capacity SSD for the needs of oil companies for the process of seismic data acquisition.

In 1983 the bubble memory appears. The magnetic bubble memory had properties very alike the modern flash memories, such as not loosing data when the power was off. Although the bubble memory was around the market since the ‘60s, it wasn’t until 1979 that consumers started using this technology and mostly in 1982 when a chip appeared in a few portable computers. The apple II SSD, called the MPC Bubdisk used that chip and its cost was about (back these days) 895$ for 128KB of data!



Passing from 90’s to year 2000 and after, in 1995 the Israeli firm M-Systems sets the first template of today’s SSD form, the flash fast disk (FFD-350 series). In 3.5 inch form, was provided with capacities from 16MB to 896MB.



DIGIPRO FLASHDISK (1990)

M-SYSTEMS FFD (1995)

M-SYSTEMS
FFD LINEUP (2005)

And we come to year 2000 and after where the first SSDs come to place during the rising of the notebooks! Their capacity increased with the rise of notebook's capability and finally we got a 2.5 inch SSD replacing an original HDD disk.

SAMSUNG
FLASH SSD (2006)

SANDISK
SATA 5000 (2007)

Nowadays the capacity of a 2.5 inch SSD reaches the 1TB and keeps upgrading with time, so after this, it is very interesting to have a look within the differences from HDD to SSD and how the SSD works.

What is an SSD and how does it work?

So what is actually the SSD? We could say that a Solid State Drive (or Disk if you like) is a solid state device which in order to storage information uses integrated circuit assemblies for memory, to store data persistently! SSDs in comparison with the HDDs, have no mechanical moving components and as an advantage of this, they are more resistant in physical shock, more silent, they consume less energy than HDDs and they have lower latency and access time to data, which makes them faster. But let's see all these differences and advantages with more details.

First we will have a small look at how an HDD works. The hard drive stores his data on a series of magnetic spinning disks, which are called platters. It also consists of an actuator arm with read and write heads attached to it. This arm, positions the read and write heads over the correct area of the platter in order to read and write some information. Now because the drive heads must stop in specific areas of the disk, in order to read and write data, consider also the fact that the disk is constantly spinning, there is some "wait time" before we can have access to the data. Added to this, the drive may need to read from multiple locations in order to load a file or launch a program, which means it may have to wait for the platters to spin into the right position multiple times before it could complete the command we gave him. Furthermore, if the drive is asleep or in a low state of power, it could take some extra seconds for the disk to come in an operational state.

From the beginning of modern computers, we could clearly see that the hard drives could not possibly match the speed at which a CPU was operating. Imagine that the latency in the HDD is measured in ms (milliseconds) while for a typical CPU in ns (nanoseconds). One ms is equal to 10^6 ns and usually takes 10-15 ms for a hard drive to find data and begin reading them. The hard drive industry tried to minimize that problem by introducing the market smaller platters or on disk memory caches and faster spindle speed to counteract this problem but many times we realize that is the nature the one that puts the limits in our universe and although today we have disks with speed up to 10.000-15.000 RPM the problem still remains and the CPU remains much more faster.

The table below lays out the memory hierarchy:

LEVEL	ACCESS TIME	TYPICAL SIZE
Registers	"instantaneous"	under 1KB
Level 1 Cache	1-3 ns	64KB per core
Level 2 Cache	3-10 ns	256KB per core
Level 3 Cache	10-20 ns	2-20 MB per chip
Main Memory	30-60 ns	4-32 GB per system
Hard Disk	3,000,000-10,000,000 ns	over 1TB

And here comes the SSD! Solid State Drives took their name actually because they don't rely on spinning disks or moving parts, instead the information is stored in a pool of flash memory (NAND flash memory). Before we proceed, it would be wise to have a small look at what is a flash memory.

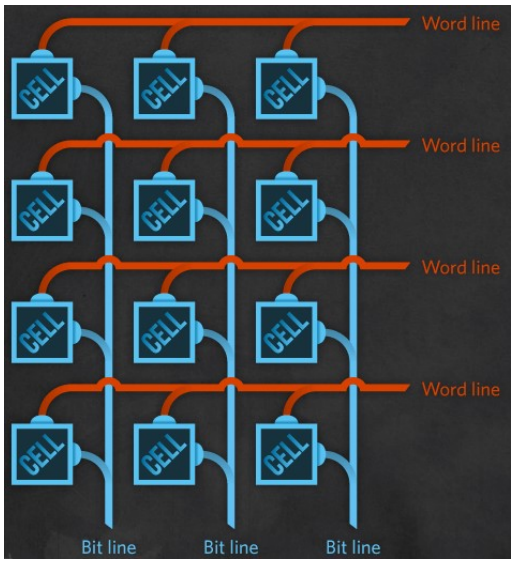
In computer dialect we have a difference between storage and memory. Storage holds all the stuff of digital information (movies, pictures, music etc) and can retain all these even if the power is off, while on the other hand, memory or RAM (random access memory) holds the program a computer is running as well as data but requires power to do so. Both storage and RAM boast their capacity according to the number of bytes they can hold. A small scale just to realize the difference of size from storage and RAM, storage nowadays could be up to some Terabytes while RAM comes up to some Gigabytes.

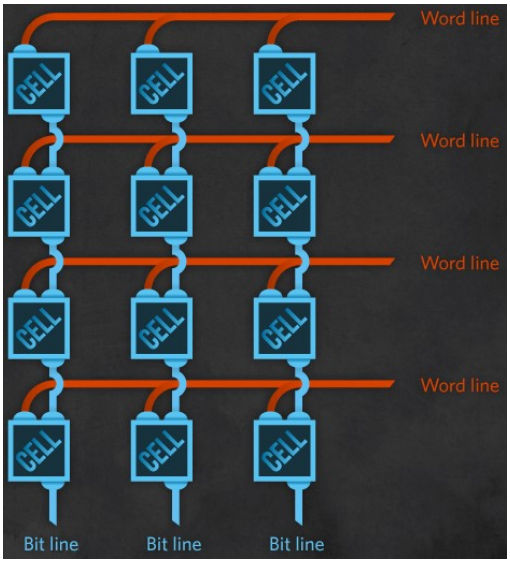
Now we also have the devices referred as "flash memory", which blur the line between storage and memory! A device with flash memory can still hold lots of data (in the scale of TB), even if the power is off. But unlike hard drives, which contain moving parts and spinning platters they have nothing mechanical in them!

They are made from transistors and other components that someone could find in a computer chip and for that they can proudly enjoy the label of "*solid state*" and take advantage of semiconductor properties.

There are two different types of flash memory. The NOR and the NAND memory, they both contain transistors in a grid but the wiring between the cells is different. In NOR flash the cells are wired in parallel while in NAND flash in a series. Because NOR cells contain more wires they become bigger and more complex. NAND cells require less wires thus can be packed in a chip in greater density. As a result we have NAND flash memories to be less expensive and with the advantage to read and write data more rapidly.

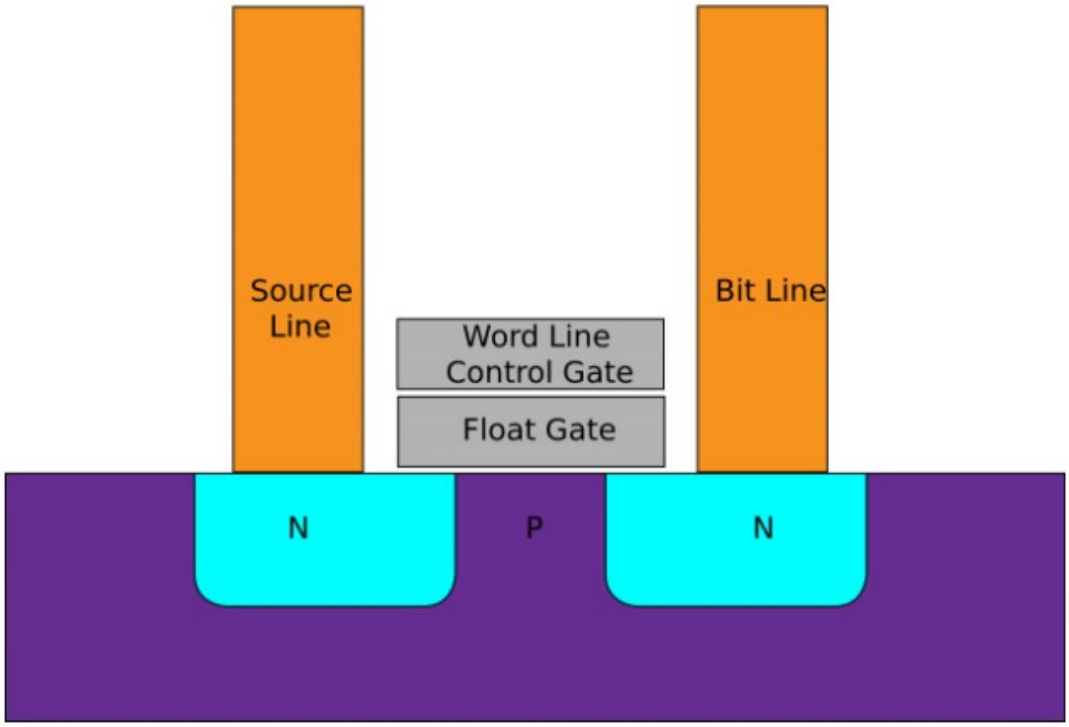
The above make the NAND flash an ideal storage technology and explains why they have dominated in the market in the construction of SSDs. NOR flash is ideal for having lower density, high speed and read only applications, such as those in code storage.





(Above you can see a simplified diagram between a chunk of NOR and NAND flash)

NAND itself is made from what we call “floating gate transistors”. Unlike the transistor designs used in DRAM, which have to be refreshed many times per second, NAND flash is designed to retain its charge state even if the power is off. This makes NAND a type of “non volatile” memory.



Above we have the picture of a simple flash cell design. The electrons are stored in the grey area with the name “float gate” which then reads as charged = “0” while not charged = “1”. In NAND flash memory the “0” means the data is stored in the cell while “1” the opposite, it has a different logic as we typically all know.

The NAND memory of a SSD stores the data differently. It has transistors arranged in a grid of rows and columns, so if a chain of transistors conducts current, that means we have a zero “0” , if not we have a “1”. In the beginning all the transistors are setted to 1, as save operations begin, the current is blocked in some transistors, turning them to 0 and this occurs because of how the transistors are arranged. At each section of column and row, a cell is formed with two transistors.

The first is called the “floating gate” and the second “control gate”. When we have the current reaching the control gate, the electrons flow upon the floating gate, creating a net positive charge that blocks the current flow. By applying the correct voltage to the transistors, we have a unique pattern of 1s and 0s emerging. NAND flash comes in three different styles based on how many 1s and 0s can be stored in each cell. Thus we have SLC (single level cell) NAND, that stores one bit per cell (either one or zero), TLC (triple level cell) NAND and MLC (multy level cell) NAND that stores two bits per cell. MLC flash gives us higher capacity but wears out more quickly, still is less expensive per gigabyte that SLC, as a result it is this technology to be preferred in the construction of the most SSDs.

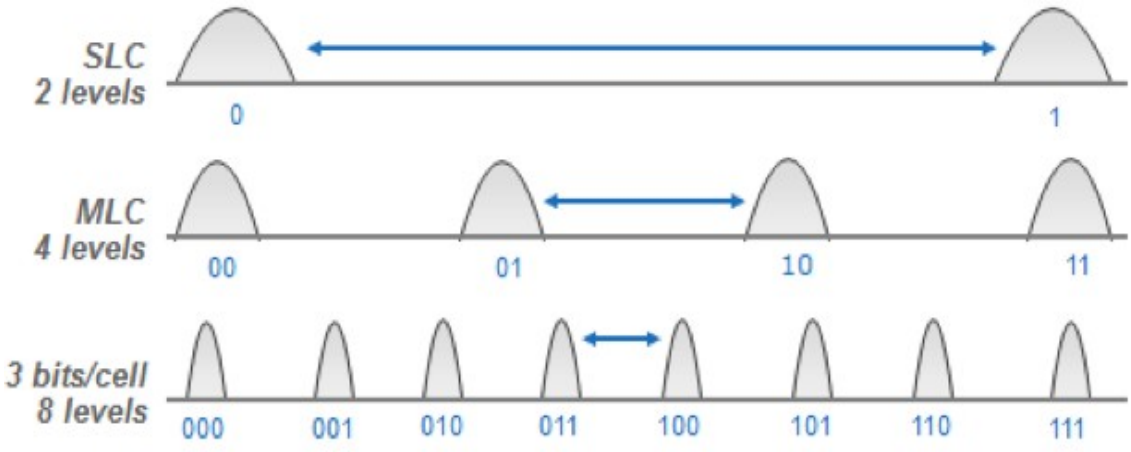
As we mentioned before the NAND is organized in a grid. The whole grid layout is referred as “a block”, while the individual rows that make up the grid are called “a page”. A common page size comes from 2KB to 4KB, 8KB or 16KB with 128 or 256 pages to form the block. Therefore the block size typically varies from 256KB to 4MB.

The advantage of this system is obvious! Because the SSDs have no mechanical parts, they operate at speeds far beyond a typical HDD. In order to see that better, here comes a chart that shows the access latency for typical storage (in microseconds).

	SLC	MLC	TLC	HDD	RAM
P/E cycles	100k	10k	5k	*	*
Bits per cell	1	2	3	*	*
Seek latency (µs)	*	*	*	9000	*
Read latency (µs)	25	50	100	2000-7000	0.04-0.1
Write latency (µs)	250	900	1500	2000-7000	0.04-0.1
Erase latency (µs)	1500	3000	5000	*	*
Notes	* metric is not applicable for that type of memory				

As we can observe, NAND is nowhere as fast as the RAM but it is many times faster than the HDD. While write latencies are significantly slower for NAND than read latencies, they still outrun traditional spinning media.

From the above chart, there are some things we must notice. First we can see that by adding more bits per cell of NAND, it has a serious impact on the memory’s performance. It is better for writes as opposed to reads, for example the TLC latency is four (4x) times worse compared with SLC NAND for reads, but six (6x) times worse for writes. The erase latencies are also impacted. Well, the impact is not proportional, for example the TLC is nearly twice as slow as the MLC, despite of being able to hold 50% more data (three bits per cell than two).



TLC NAND voltages

The reason why the TLC NAND is slower than SLC or MLC has to do with how much data moves in and out of the cell. For instance, in the SLC NAND the controller has to know only if the bit is “0” or “1”, while in the MLC we may have four values (00 01 10 11) and as you can predict in the TLC we have eight possible values. In order to read the proper value out of the cell, it requires from the memory controller to use a very precise amount of voltage to find out whether any specific cell is charged or not.

	SLC	MLC	TLC
Bits per cell	1	2	3
P/E Cycles	100,000	3,000	1,000
Read Time	25 µs	50 µs	~75 µs
Program Time	200-300 µs	600-900 µs	~900-1350 µs
Erase Time	1.5-2 ms	3 ms	4.5 ms

Reads Writes and Erasure

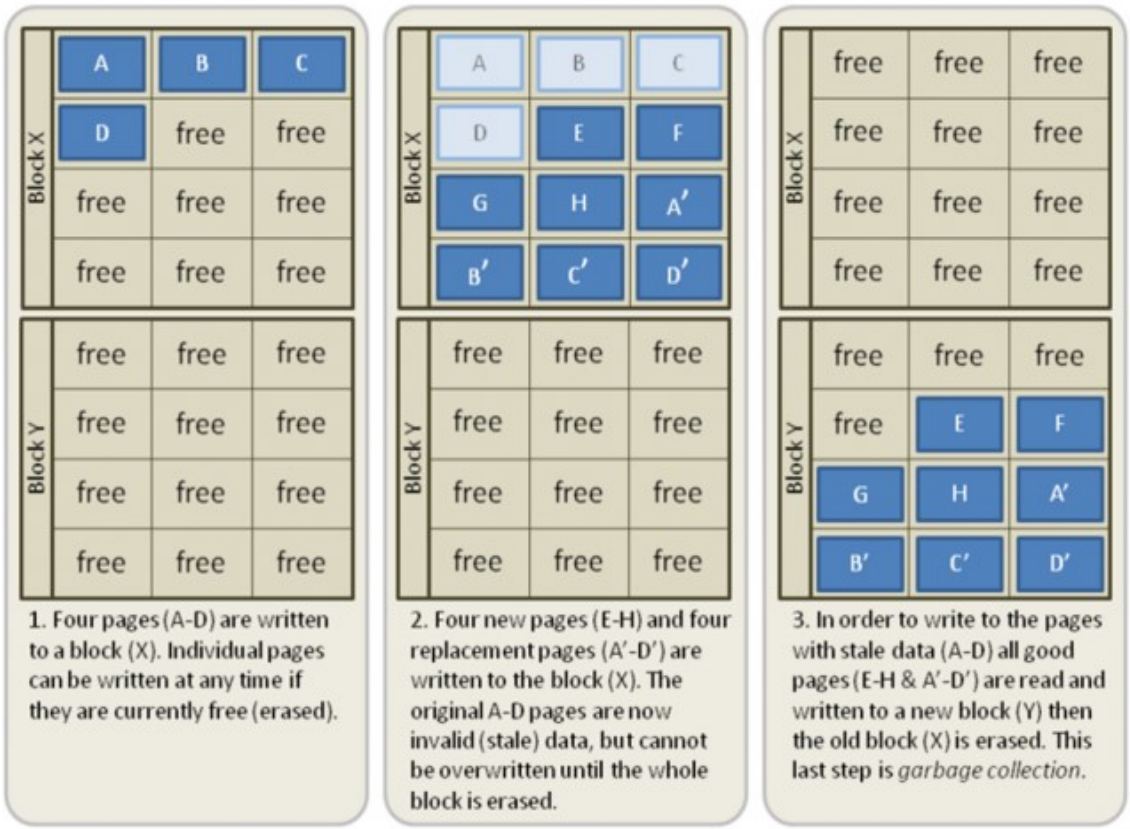
Garbage Collection

One functional disadvantage of the SSDs is that although they can read and write data very fast (especially when the drive is empty), the time for overwriting is much bigger and in this case they are slower. This happens because while an SSD reads data at the page level (from individual rows inside the NAND memory grid) and also writes at the same level, considering the fact that the surrounding cells are empty, it only erases data at the block level!

This is because the action for erasing data in a NAND flash requires a high amount of voltage. Theoretically we could erase NAND at the page level, but the amount of energy (voltage) required for this action "harms" the individual cells around the cells that our targeted data are being rewritten. Using the path of erasing data at the block level mitigates this problem.

Furthermore, the only way for the SSD to update an already existing page, is to copy the whole content of the block into the memory, erase the block and after that to write the content of the old block plus the updated page. In the case where the drive is full and there are no available pages empty, the SSD scans for blocks that are marked for deletion (and not having been deleted yet), erases them and then writes the data to the freshly erased page. This is the reason why a SSD becomes slower as it ages, a new and mostly empty drive has a lot of blocks that are available for writing data immediately, while a mostly full drive is more likely to be forced to do the whole procedure (find and erase pages) and thus be slower.

Something you must be familiar with or have heard, when interacting with SSDs, are the two words - expression: "**garbage collection**". Garbage Collection is the background procedure that allows our drive to soothe the performance impact of the program/erase cycle, implementing certain tasks in the background. Below you can see this procedure in the image.



We must note that in the above example, the drive takes advantage of the fact that it can write very fast to the empty pages by writing new values for the first four blocks (A' to D'). It has also written two new blocks, E and H. Now the blocks from A to D are marked for the garbage collector to take them, they are marked as stale, which means that they contain information that the drive has marked as out of date. Now during an idle time, the SSD will move the fresh pages to a new block erase the old block and finally mark it as free space! This automatically gives the drive the advantage, for the next time it will need to perform a write, to do it directly to the now empty block X rather than performing the program/erase cycle thus being faster!

In order to prevent the undesirable effect of aging our SSD, modern SSDs run complex routines called garbage collection. They do that because is beneficial to always keep as large a reserve as possible of empty blocks ready for writing. Garbage collection involves having the controller search through the inventory of written pages for the ones that have been marked as "stale" (meaning the data they contain must be modified by the OS). But, because in order to change a page's state is impossible unless you first erase it, the changes are written to new pages and the old pages are marked as stale.

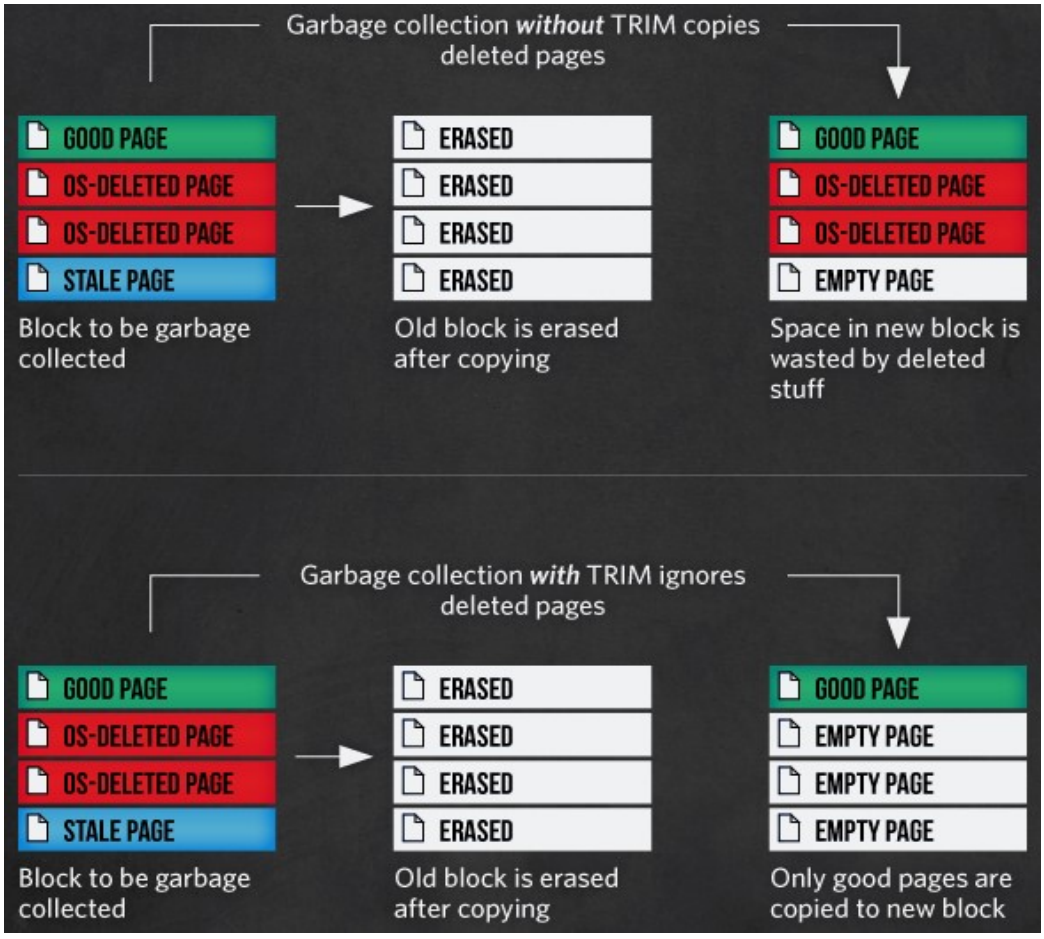
Garbage collection looks out for blocks with a mix of good and stale pages and after it duplicates all the good pages into new blocks, leaving behind to the old block the stale ones, finally erases the old block and marks it as ready for use.

Trim

The second thing you must have heard in the topics about SSDs is the word "**TRIM**". The TRIM is the command that allows the operating system to inform the SSD that it can skip the procedure of rewriting certain data the next time it performs a block erase. Thanks to that command, we have a lower total amount of data being written to the drive, thus increasing the SSD's life (there is also another technique for that, we will analyze it later on, it is called wear leveling). Both the reads and the writes damage the flash, but writes do much more damage than reads, thankfully block level longevity has not proven to be an issue in the modern NAND flash memories.

Trim is an ATA command which the operating system causes to be sent when we have the deletion of a file. This command actually provides us with the bridge to communicate from the file level to the block level, supplying the operating system with a way to inform the SSD that we have deleted files and mark those file's pages as stale.

With the help of the TRIM, the drive is no longer forced to save pages which belong to deleted files. Trim does not exclude the need for garbage collection, it actually works with the garbage collection by making it more efficient to find the marked pages as stale. Let's see an example below.



Without the Trim, the garbage collection is not informed about the deleted files and continues moving the pages containing deleted data along with the “good” pages, increasing this way the write amplification. So Trim actually informs the controller that it can stop collecting pages with deleted content, so they can be left behind for erasure with the rest of the block.

Trim makes a difference in the reduction of the write amplification and extends the life and the performance of the SSD, so always use it if you can. But for our own purposes of the research we are going to see what happens if we have deactivated the TRIM command (all these later on).

Wear Leveling

As we said before, we will talk now about the technique that expands the life limit of a SSD, the “**wear leveling**” also about the concept of “**write amplification**”. Because SSDs write data in pages but erase them in blocks (as mentioned before), the amount of data being written to our SSD happens to be larger than the actual update. For example, if we make a change of a 4KB file, the entire block that contains this file must be updated and rewritten. Now depending on the number of the pages per block and the size of our pages, we might end up writing a 4MB size of data to update a 4KB size file. The procedure of garbage collection mitigates the impact of write amplification, as the TRIM command does so. Keeping a significant space of the drive free also reduces the impact.

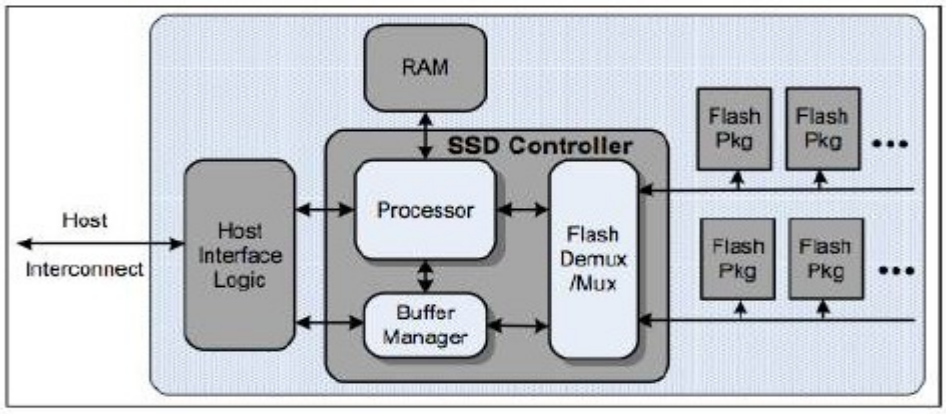
Wear leveling refers to the technique that ensures that certain NAND blocks are not written or erased more often than others inside our SSD. Although wear leveling proliferates the life expectancy and endurance of the drive, by writing equally in the NAND, it could sometimes increase the write amplification. To be clearer with the last phrase, it is often necessary to program and erase blocks (in order to achieve the wanted balance of writes) even though their content is not changed by the user. It is supposed that a proper wear leveling algorithm seeks to balance this affair.

Nowadays many manufactures use the wear leveling technique to counteract the degradation of the NAND flash. As we mentioned, distributing the data writes across all the blocks of the SSD, ensures us that the flash memory wears evenly, but of course even with that, there is a point in time that the drive will decay over time. A NAND flash memory of the single level cell variety usually delivers 50.000 program/erase cycles, while the flash of the multi level cell delivers 5.000 cycles.

Because of that, many of you may find in the market (as a solution from the data centers) the combination of HDD and SSD. The approach is to use the SSD in a laptop and the HDD as an external storage space, for music movies and other files, with that trick you have the ultrafast access to data from your SSD (on the one hand) and the inexpensive and high capacity HDD (on the other).

The SSD Controller

The last part that comes to complete the picture of a functional SSD, is the SSD controller.



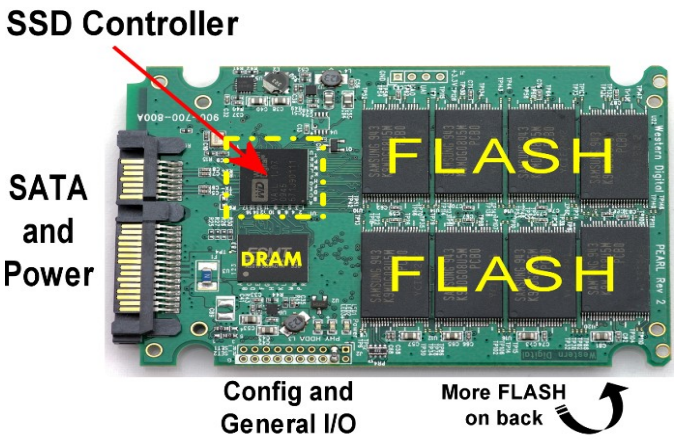
SSD Logic Components

Although we could say that the mechanical challenges inside a HDD are pretty serious, considering the fact that it must balance multiple read-writes heads nanometers above platters that spin up to 10.000 RPM, however the controller is a class by itself.

Controllers usually have a DDR3 memory pool to help them managing the NAND itself. A lot of drives incorporate single level cell caches that act like buffers, increasing with that the drive performance by dedicating fast NAND to read/write cycles. The NAND flash memory is actually connected to the controller through parallel memory channels and the drive controller is performing some of the same load balancing work, such as a high end storage array, wear leveling, garbage collection and SLC cache management.

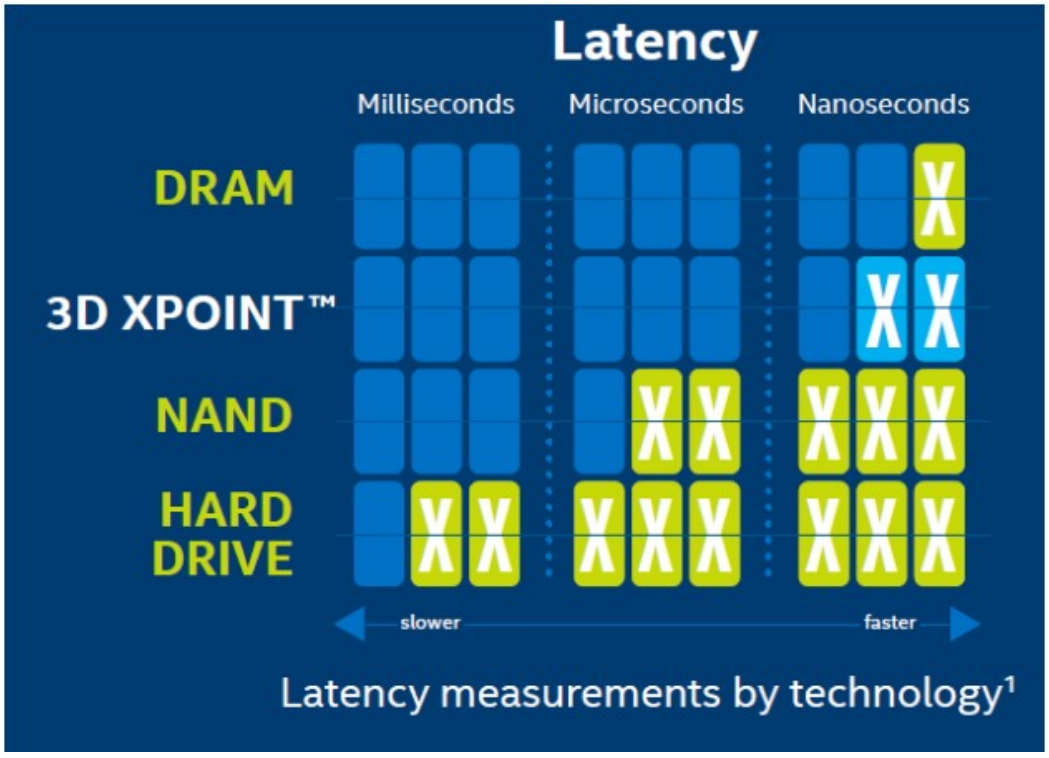
There are also SSDs that use data compression algorithms to reduce the total number of writes thus improving the drive's lifetime. The SSD controller is responsible also for the error correction and as the time passes, the algorithms for

controlling single bit errors have become increasingly complex. Of course we cannot go into too much detail about the controllers, because the producing companies do not let their secrets to leak, such as the recipe to create a controller!



The future that never ends. The new type of hard drive called SSD, has advantages and disadvantages, it is much faster than HDD but also problematic as the time passes. Of course the cost of this new technology is significant but also is the size. The latest tech gives as the opportunity of making things smaller but also more fragile.

A new era has already begun. Only a few weeks ago Intel announced the new disk that will replace the SSD (as we know it today) because of its speed, the “Intel Optane SSD”!



This image is the latest information we have about the optane ssd. When it was firstly announced by Intel, in 2015 , they claimed that it would be about 1.000 times faster than a NAND flash, 10 times thicker than DRAM and would have 1.000 times

better endurance than NAND, without saying faster at “what” or what kind of “NAND”.

But today is much more clear, the 3D Xpoint has the one thousandth of the latency of NAND flash and about 10 times the latency of DRAM and at the same amount the density of DRAM (10 times).

Hardware for SSD forensics

Everyone who is familiar with forensics in HDDs will definitely know the write-blocker device and the usability of this tool. Mainly in order to image SSD drives, available SATA compliant write blocking forensic acquisition hardware is used but the question is why so little chip-off solutions for the SSDs compared to the number of companies that do mobile chip-off? I believe that the possible answer to that question is hidden behind the word “controller”!

Imagine we could easily chip-off the controller from SSD so that we could avoid the Garbage Collection and after that have a full and safe access to the pool of data named flash memory! We know for sure that the controller does not apply the data uniformly inside the flash (meaning that with the constant remapping and the shuffling of data for the performance and the lifespan routines of the disk, the data inside the flash memory are heavily fragmented) they scatter them and also that these data will be encrypted. So, although if we could gain access to these precious data, we couldn’t take advantage of them or read them at all.

What happens after TRIM

There are some SSDs in the market that operate what we call DRAT (deterministic Read After Trim) and DZAT (deterministic Zeroes After Trim), returning all zeroes just after the Trim command releases a certain data block, but there are also other drivers that do not support or follow this protocol and could return all the original data (until of course they are erased by the garbage collection).

DRAT and DZAT have been a part of the SATA designation a long time now. In Linux environment there is a way to verify that SSD drives use DRAT or DZAT with the command “hdparm –I” which returns us whether the drive supports TRIM and if he does apply DRAT.

«$ sudo hdparm -I /dev/sda | grep -i trim»

Data Set Management TRIM supported (limit 1 block)
Deterministic read data after TRIM

We have three different types of TRIM designated in the SATA protocol which are implemented in various SSD drives.

- Non - deterministic TRIM: where every read command after TRIM can return different data.
- Deterministic Trim (DRAT): where all the read commands after TRIM shall return the same data or become determined.
- Deterministic Read Zero after Trim (DZAT): where all the read commands after the TRIM must return zeroes until the page is written with new data.

We must note here that the DRAT type does not return necessary zeroes when trim pages are accessed, instead it guarantees us that the data which return, will all be the same (determined), before and after the "marked" pages pass from the process of garbage collection and until they are written with new information. Also that the DRAT is not implemented in Windows because NTFS doesn't allow applications reading the trim data.

As we can figure out there are cases that the SSD will give us not original data (all zeroes, all ones or other non original data) this is not because the blocks have been cleared immediately after the trim command, but because the controller tells that there are no valid data held at the trimmed address on a logical level previously associated with the trimmed physical block.

If we could (somehow) had the ability to read the data directly from the physical blocks mapped to the logical blocks that have been trimmed, we could also obtain the original data from the physical blocks before of course they are erased by the garbage collection procedure. Obviously there is no way to address the physical blocks through the standard ATA command set, nevertheless the manufacturer of the disk could probably achieve that in the lab. So sending the SSD for forensics investigation to the manufacturer may be a solution to the problem.

Possible scenarios for acquiring evidence in SSD

We could distinguish two possible occasions for the state of the disk we would like to examine. The first could be that our disk contains only existing files, the second one could be having the full disk content.

In the first one, the SSD contains some files and folders but the free disk space will be really empty (like filled with zero data). As a result searching the free disk space would return us no information or just traces of information, while carving the entire disk space would return data contained in existing files, but this does not mean that

the carving method is useless. It helps us to locate moved or hidden evidence. Any evidence contained in existing files including for example deleted records from SQL databases can still be recovered.

In the second occasion our disk contains the complete set of information. If there is a possibility for the SSD not to destroy all the evidence (due to the routine of garbage collection), that would be if the TRIM command has not been issued or if the TRIM protocol is not supported. Below are some cases that this could happen.

Operating System Support

The TRIM command is a part of the operating system as long as this is the property of the SSD device itself. There are older file systems that do not support TRIM. In Windows 7 it is the first time we meet the TRIM support, of course it is supported in Windows 8, 8.1 and Windows 10. In Vista and earlier the protocol is not supported and the command is not issued. As a result, when we analyze the SSD in order to complete a forensics investigation, with an old version of Windows it is possible to obtain the total content of the disk.

There is only a small possible exception, where the TRIM like performance can be activated by third party solutions like a part of Intel SSD toolbox called “Intel SSD optimizer”.

Mac OS X have started supporting the TRIM command since the version 10.6.8 for the Apple, older builds of Mac OS X do not support it.

Older or Basic Hardware

We can still find in the market SSDs that do not support TRIM and/or background garbage collection. SSD like flash media and older SSDs used in basic tablets and sub notes do not support it. Intel started manufacturing TRIM enable SSDs with drive lithography of 34nm (G2) and their 50nm SSDs do not have TRIM support. Many entry level sub notebooks often misunderstood as SSDs did not have the feature of garbage collection or supported the TRIM protocol.

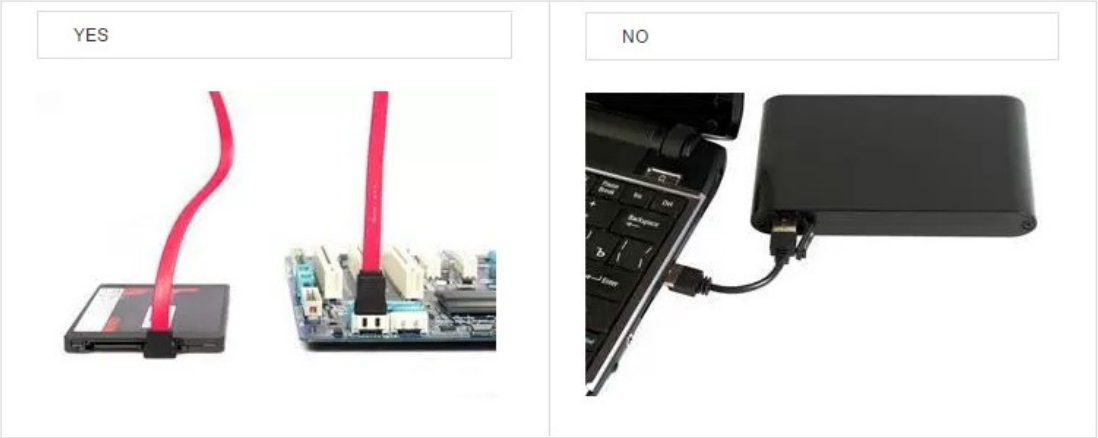
File Systems (NTFS)

TRIM is a feature of the file system which belongs to the SSD drive. Windows only support TRIM on NTFS formatted partition and volumes formatted like FAT, FAT32 and exFAT are not included.

Note that some older SSD drives used a swindle to work around the lack of TRIM support by trying to interpret the file system, attempting to erase the “Dirty blocks” not referenced from the file system. This technique when enabled it worked only for the FAT file system and it is a published spec.

External drives, USB enclosures and NAS

Through the SATA interface the TRIM command is fully supported, including the eSATA extensions and SCSI through the unmap command. It is believed that if an SSD drive is connected through a usb enclosure or installed in most models of NAS devices (Network Attached Devices), the TRIM command will not work at all because it will not be communicated via the unsupported interface.



Of course there is an exception to the last phrase since the manufactures of NAS devices are starting to recognize the usability of devices with ultra high performance, noise free operations and low power consumption provided by SSDs and they slowly start to adopt the TRIM in some new models.

Corrupted Data

SSD drives with damaged partition tables, file system etc (corrupted system areas), are more possible to recover than the healthy ones. That is because the TRIM command is not applied over corrupted areas because the files there are not deleted properly. As a result these files become invisible and inaccessible to the operating system and many data recovery tools can reliably extract information from logically corrupted SSD drives.

Bugs in SSD firmware

Firmware that is used in SSD drives may sometimes contain bugs which often affect the TRIM functionality and mess up the garbage collection. For example it is reported that the OCZ Agility 3 with 120GB space was shipped with buggy firmware

version 2.09, in which TRIM did not worked. The version 2.22 introduced issues with data loss on wake up procedure after sleep, something that the version 2.25 fixed but the last one disrupted TRIM operation again. So actually a particular SSD may or may not recover, depending on what bugs are applied to its firmware.

Bugs in SSD over provisioning

One of the many wear leveling techniques that intent to expand SSDs lifetime is the SSD over provisioning. Some areas of the disk are reserved on the controller level, which actually means that for example a 250GB SSD contains more than 250GB of physical memory. These extra data blocks are called OP area (over provisioning) and they are used by the controller when a fresh block is required for a write operation. A "dirty" block would then enter the OP pool and would be erased by the garbage collection mechanism during an idle time of the drive.

Firmware bugs can affect the TRIM behavior in other ways also, for example revealing trimmed data after a reboot or a power off. SSDs remap very often after the TRIM to allocate addresses out of the OP pool. As a result of this action, the SSD gives us a report of some trimmed data blocks as "writeable" (actually just erased) immediately after TRIM, obviously the drive did not have the time actually to clean the old data from these blocks. Instead it just maps some physical blocks from the OP pool to the address referred to, by the TRIM logical block!

Now let's take a look on what happens to the old block. For a while it still contains the original data (in many cases they will be compressed, depends on the feature of the controller). Nevertheless as the data block is mapped out of the addressable logical space, we can access or find the original data since they are inaccessible.

Issues like this may cause problems like, after deleting data and immediately rebooting the computer , some users could see the old data again as they were never deleted. Obviously because of the mapping issue the new pointers could not work as they were supposed to, due to a bug in the driver's firmware.

SSD shadiness

It has been published on internet that about two years ago, a couple of very well known SSD manufactures (Kingston and PNY), after releasing to the market a very good model of SSD and after this one got very good reviews, they were caught bait and switching some components of the device with cheaper ones! In that occasion the two manufactures sold their SSDs with one hardware specification and silently changed the hardware configuration after the reviews were gone out.

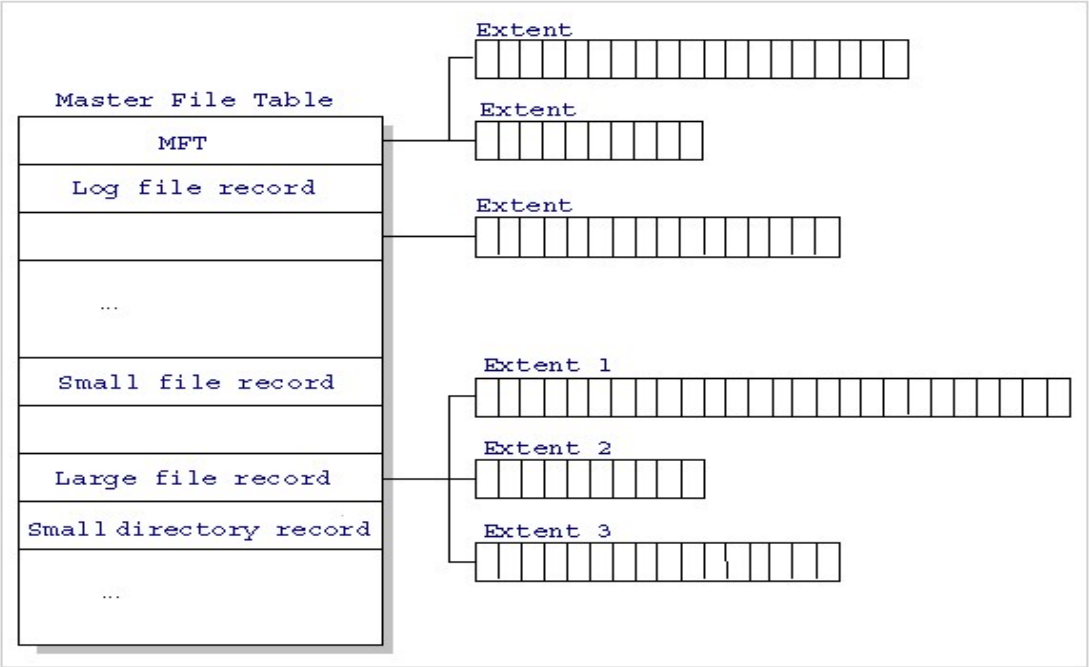
But what all that could mean for a person who is familiar with forensics? The forensic friendly SandForce controller was found on the revision of PNY Optima

drives. Instead of the original silicon motion controller, the new batch of PNY Optima drives had a different, SandForce based controller know for its less than perfect implementation of garbage collection, as a result leaving data on the disk for a long time after these were deleted!

Small files, Slack Space

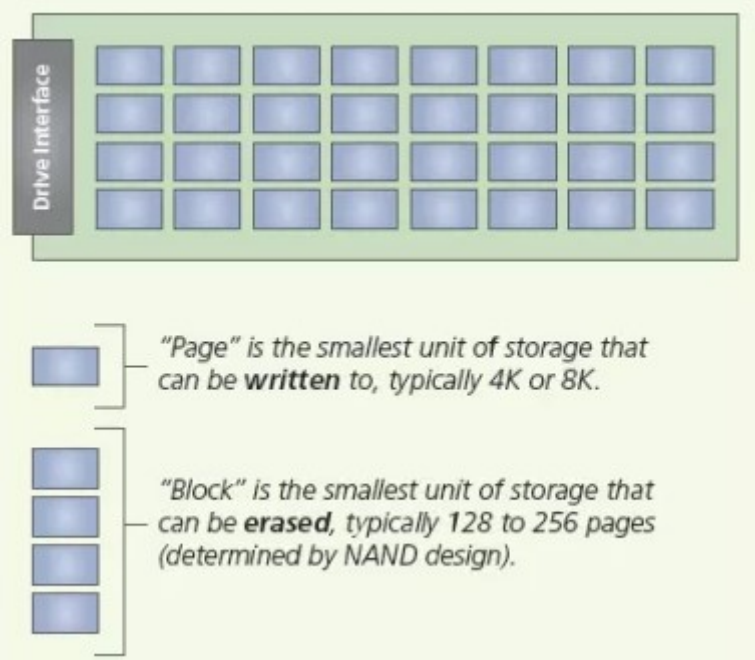
Another possible way finding evidence could be through the remnants of deleted evidence from so called slack spaces as well as from MFT attributes (Master File Table).

MFT Structure



In the new world of SSDs the word “slack space” gets a different meaning. Instead of being a matter of file and cluster size alignment, the new definition deals with the different sizes of minimum writeable and erasable blocks on the physical level.

As we have mentioned before, page is the smallest unit of storage that can be written in SSDs (typically around 4KB and 8KB) and the block on the other hand is the smallest unit of storage that can be erased, depending on the design of a particular SSD drive one single block may contain from 128 to 256 pages.



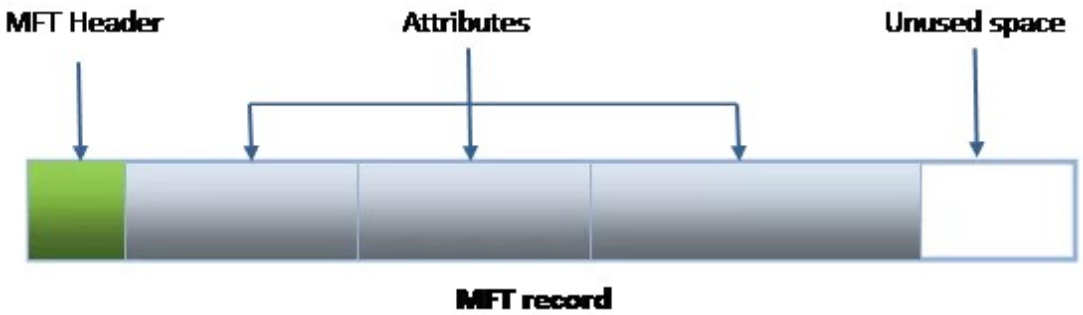
As a result we could have the case that if a file is deleted and its size is less than the size of a single data block, or in the case a data block which contains pages that remain allocated, the particular block is not erased by the garbage collector. In practical terms this means that files or file fragments with size less than 512KB or less than 2MB (depending on the SSD model), may not be affected by the TRIM command and can be able for forensic recover.

However, the existence of DRAT protocol (we mentioned before) adopted by many SSD manufactures, makes trimmed pages inaccessible through standard SATA commands. With the presence of DRAT or DZAT protocols in the SSDs, the actual data may stay on the drive for a long time but unavailable to forensic specialists with standard acquisition techniques.

MFT Attributes

It is well known to every user of windows, that this operating system is using NTFS as file system. This type of file system stores information about the files and the directories in the MFT (master file table). The MFT contains information about all the files and the directories that are listed in the file system and actually every file or directory has at least one record in the MFT.

In computer forensics terms, a specific feature of the MFT stimulates our interest. NTFS has a unique ability to store small files directly to the file system. The entire substance of a small file can be stored as an attribute inside a MFT record, improving greatly the reading performance and decreasing the slack space (wasted disk space).



As a result, the small files being deleted are actually not, because the entire content continues to exist in the file system, the MFT records are not emptied and also not affected by the TRIM command. This is what allows the investigators to recover resident files by carving the file system. The maximum size of this small file that will continue to live inside an MFT record cannot exceed the size of 982 bytes and in turn this fact severely limits of course the value of resident files for the purpose of digital forensics.

Encrypted Volumes

There is a possibility to recover deleted information, from certain types of encrypted volumes (provided that the investigator is familiar with the either the original password or the binary encryption keys of the volume), as it may not be affected by the TRIM command (some configurations of Bitlocker, Truecrypt , Pgp and others). Files being deleted from encrypted volumes on SSDs drive can be recovered unless the user has wiped them out specifically.

On the way down, we will see what is known about this matter in some of the most popular encrypted programs.

- Apple FileVault: The Company (Apple) introduced it with the ability to provide whole disk encryption. To be more accurate, file-vault enables whole volume encryption only on HFS volumes (encrypted volumes). Apple chose to enable TRIM with file vault volumes on drives.
- Microsoft Bit-locker: This is Microsoft's choice for the corresponding Apple's program. They followed the same option of enabling TRIM on Bit locker volumes located on SSDs. As usual for Microsoft Windows, the TRIM command is only available on NTFS volumes.
- TrueCrypt also supports TRIM on located encrypted volumes in SSDs. They have issued several security warnings in relation to wear leveling security issues where the TRIM command was revealing information about which blocks are in use and which are not.
- PGP whole disk encryption, on encrypted volumes has no TRIM command enabled. But, when they were facing wear leveling issues on SSDs, they

introduced an option in order to enable the TRIM on SSD volumes through a command line option.

So after all these information, we come to the conclusion that if an encrypted volume is created, the behavior is to encrypt also the content of the file representing the encrypted volume. This disables the affection from the TRIM command, for the contents of this encrypted volume!

We should take a closer look when investigating these options. But one thing is for sure, in many configurations (including the default ones), files that are deleted from the encrypted volumes of a disk may not be affected by the TRIM command.

In reality SSD forensics remains difficult. The SSDs do destroy court evidence, making it extremely hard to recover deleted files and destroyed information (especially when the disk has been formatted). Still numerous exceptions exist and some specialists may be able to gain access in destroyed evidence under certain circumstances.

The modern idea in production of SSDs nowadays is to increase the capacity and also reduce the cost for the market. Also trying to achieve a smaller size the compression of the controllers turns to become a standard, making chip off acquisition not efficient and an unpractical hardware technique to mess with.

Most SSDs appear to follow the DRAT approach, which means that a quick format may instantly render deleted files inaccessible to standard read operations, even if the drive is acquired with write blocker device used right after.

SSDs are becoming more and more complex, adding over provisioning support and use compression for better performances and wear leveling. However because of this increase of complexity, some manufactures have released SSDs with buggy firmware, which drives to improper operation of the TRIM and garbage collection. Sometimes considering how complex the whole function of the SSD has become makes you wonder how these things manage to work!

Experimental Part

In this part of the research we are going to see if some of the above cases are true or not. In order to achieve that we will use two computers with different software in each one, the first is a Dell laptop with 8GB RAM and Intel Core i5-5250U CPU 1.6GHz with Windows 10 and the second one is a table computer with 8GB RAM and AMD Phenom II X4 810 CPU 2.6GHz with Windows 7.

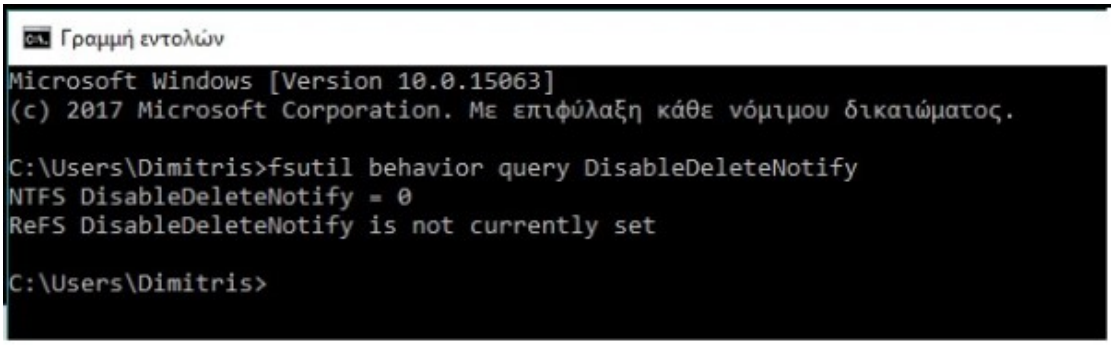
In both computers we have installed two different **open source** forensic programs, the Autopsy (computer software that deploys many tools used in the The Sleuth Kit) which can analyze Windows, disks and file systems like NTFS, FAT, UFS1/2 Ext2/3 and the Pc Inspector which is specially designed for the recovery of multimedia files from a camera memory or for micro SD cards, both capable for retrieving files after format or deleted ones in hard disks.

We are using a Kingston A400 SSD with 120GB space for our experiment, 2.5'' with read speed 500 Mb/s and write speed 320 Mb/s (SATA III 6Gbit/s), the idea is to fill up the hole disk with one image (we have taken a picture with a mobile camera at the size of 3.14MB) and repeatedly written it to the disk as many times as possible. After that we are performing a quick format to the disk and by the end of this procedure we will start searching with our two open source programs.

We are going to search for 4 different scenarios:

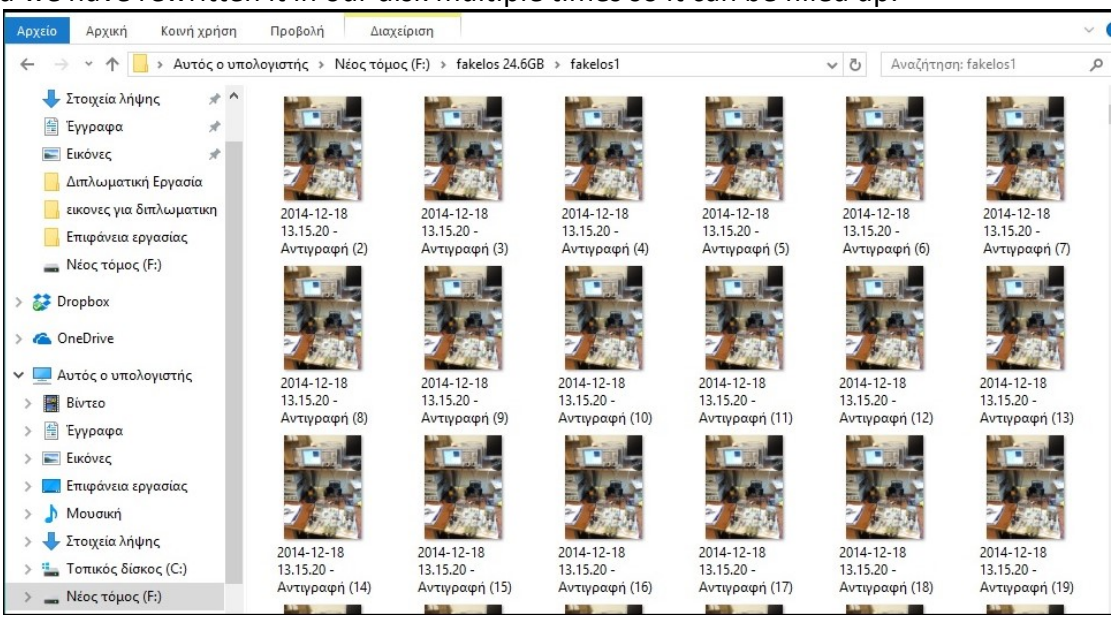
- Number one scenario, we will connect the SSD to an external case and this case will connect to our computers through usb gate. As we have written before it is believed that by this way the TRIM command is not supported or not transmitted to the controller of the SSD, as a result we can retrieve all our deleted files! This one will be tested in Windows7 and in Windows10.
- Number two scenario, will include the same procedure (fill up the disk, quick format, search) but this time only in Windows7 and with a SATA cable, in order to check if this time the TRIM command will activate the Garbage Collection, transmitting the command through the SATA connection.
- Finally the third scenario, will be the same procedure but this time we will have installed inside our disk software of Windows7 (32-bit) and after that it will be filled up with our image, then a quick format will occur and we will see the results of the two programs.

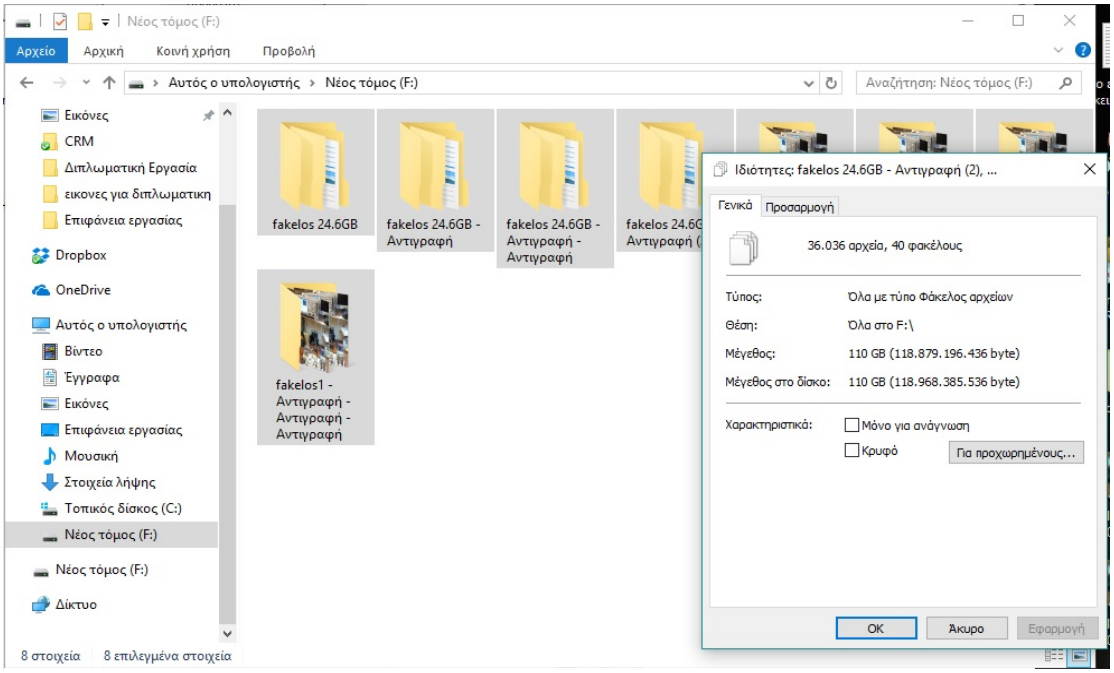
In all the above cases, we will always have the TRIM command enabled, thus we will check for it every time before we start a procedure. For a windows user to check if TRIM is active-enabled the only thing he has to do is to open to command line (cmd) and type the following command: **fsutil behavior query DisableDeleteNotify**
As you can see below in our Windows10 the TRIM command is enabled. We must notice that if the result of the command is zero (0) the trim is enabled, if one (1) it is not.



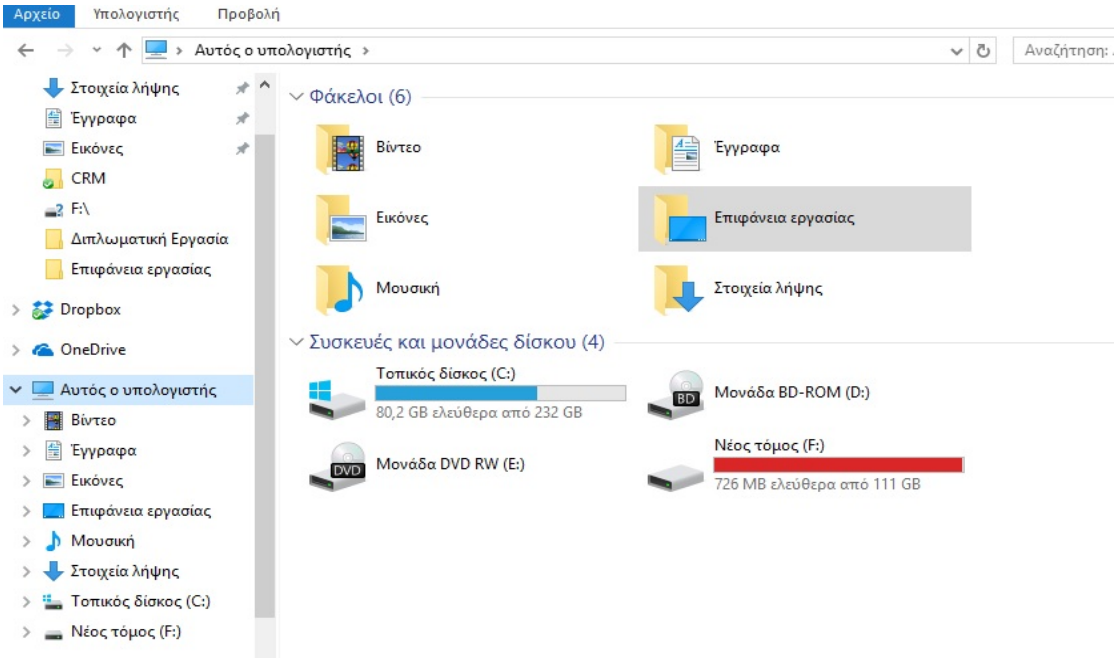
SCENARIO Number 1A (Windows10/usb):

In this case we have taken a picture from a mobile phone at the size of 3.14 MB and we have rewritten it in our disk multiple times so it can be filled up.

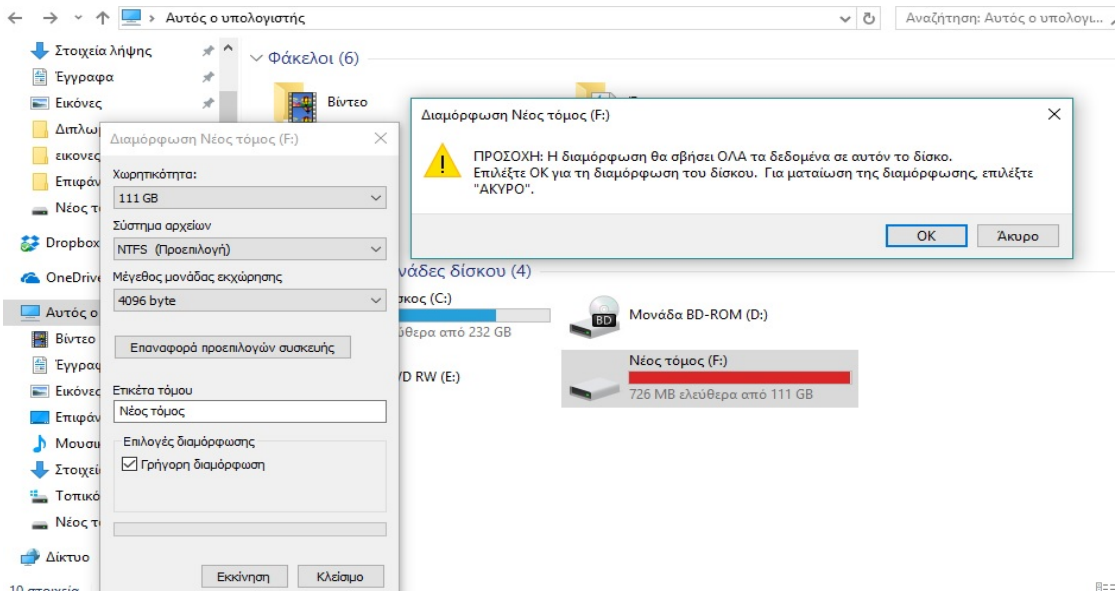




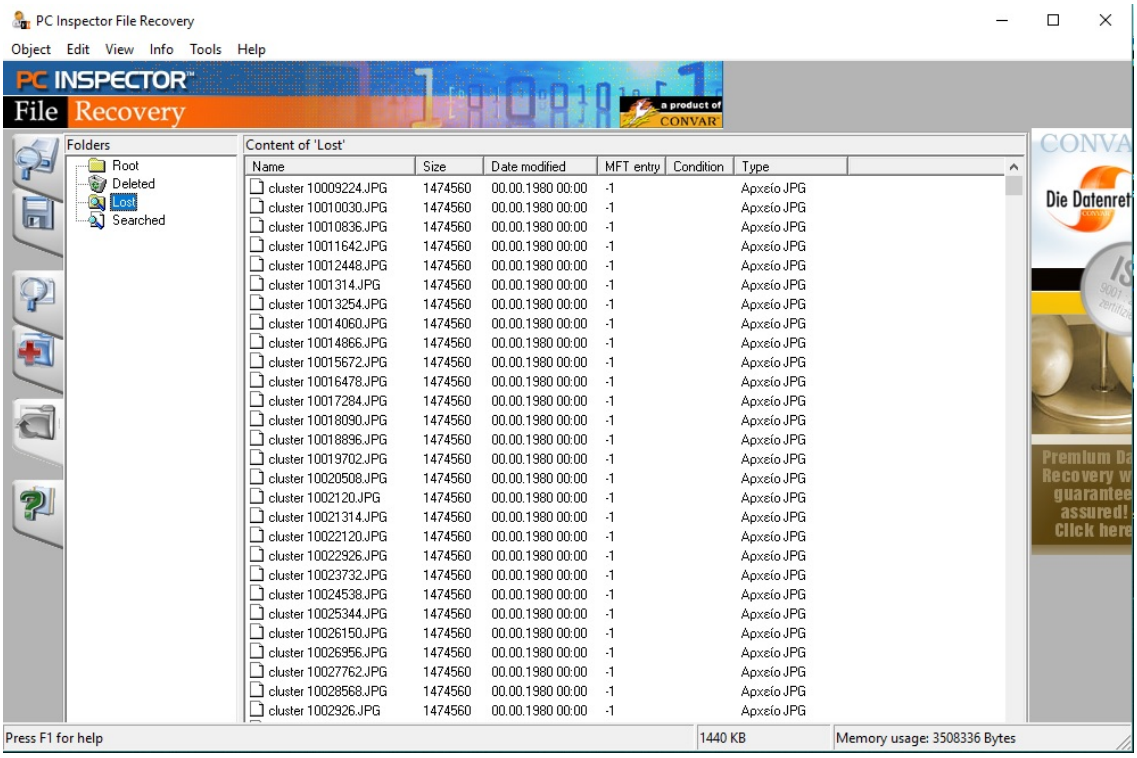
As we can see the hole disk is filled up to 110Gb out of 111Gb that has free and available space, using 36.036 files to achieve that.

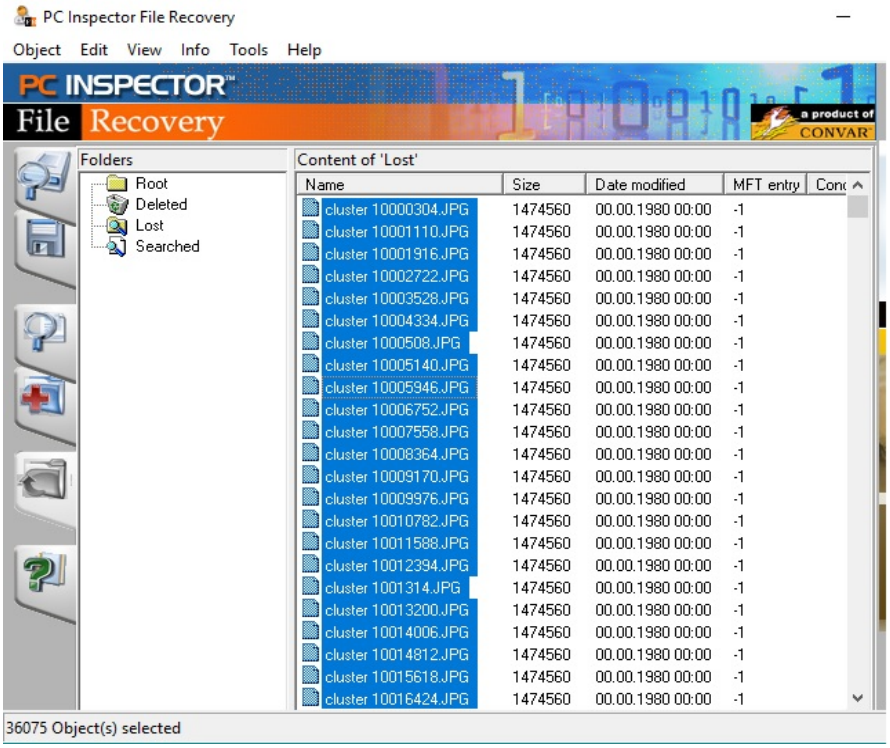


After having a quick format to our SSD we are launching the Autopsy program and after that the Pc Inspector. The procedure for the Autopsy lasted about 6 hours and for the Pc Inspector less than an hour. The results are surprising!

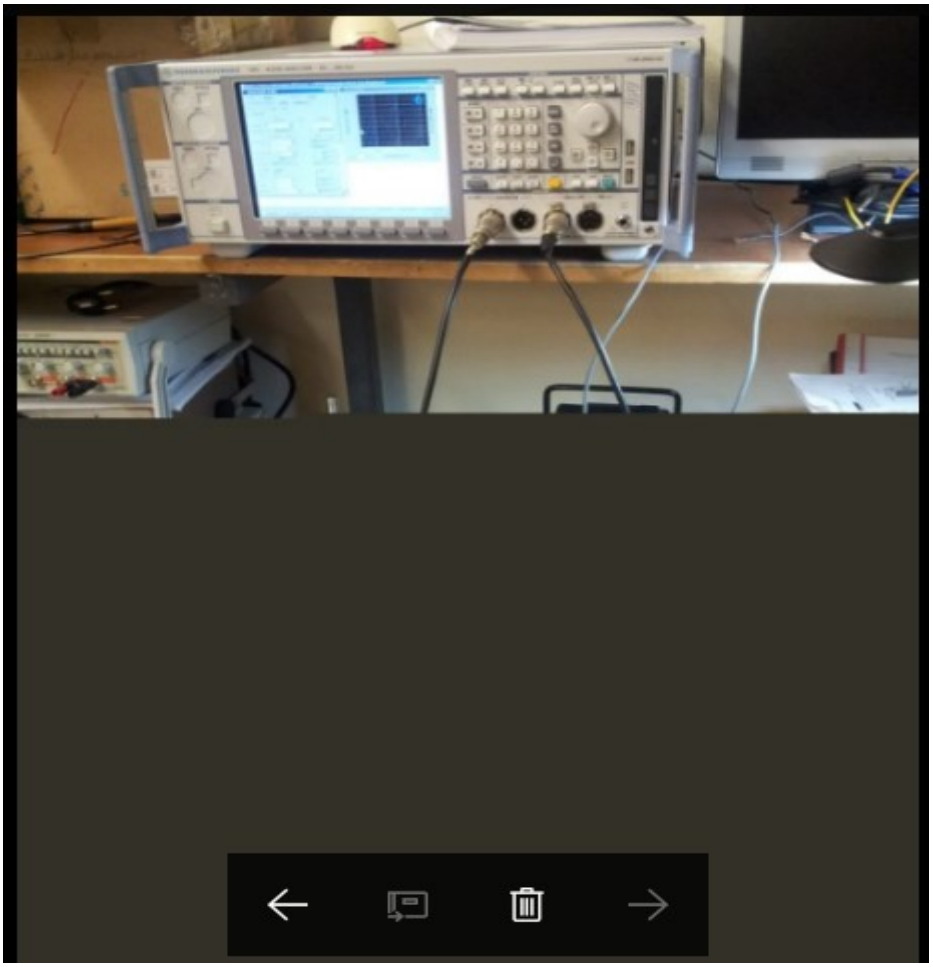


The Pc Inspector could not retrieve any files at all, at least in his sector “deleted files” the list was empty but we could see in the “lost files” sector that it had retrieved all our files, but in clusters as you can see below!

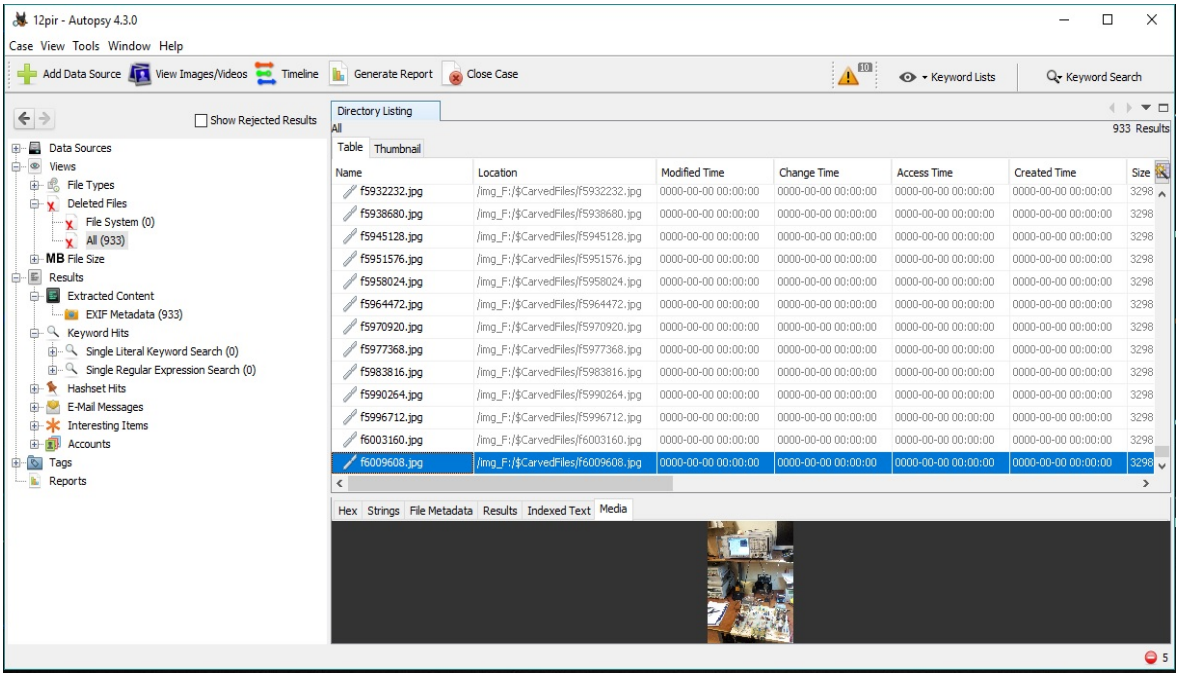


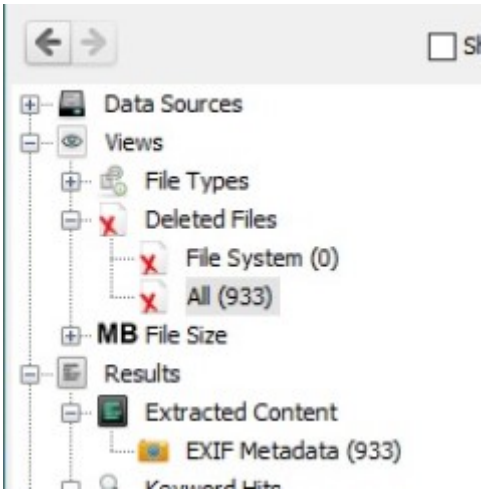


The clusters retrieved were about 36.075 in number as much as we have written in the disk (they are not 36.036 because the procedure of rewriting and carving the files was done a second time and that time more files were written to the disk). It is very interesting that the cluster was looking like this:



On the other hand the Autopsy program although it was started first, it could retrieve the amount of 933 files (from the total amount of 36.036) but the images were complete as you can see below!

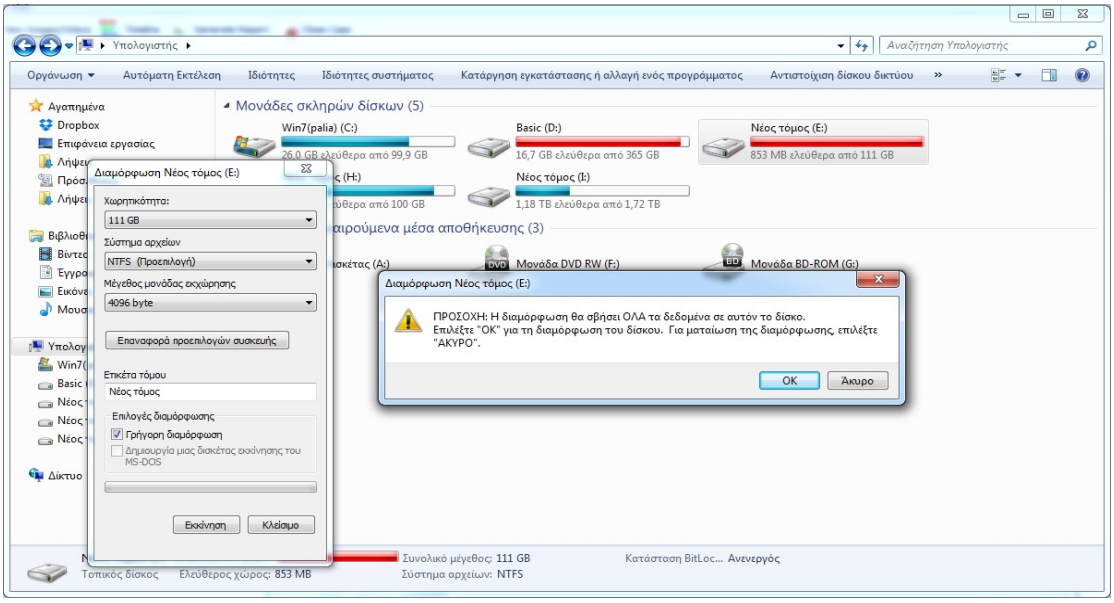


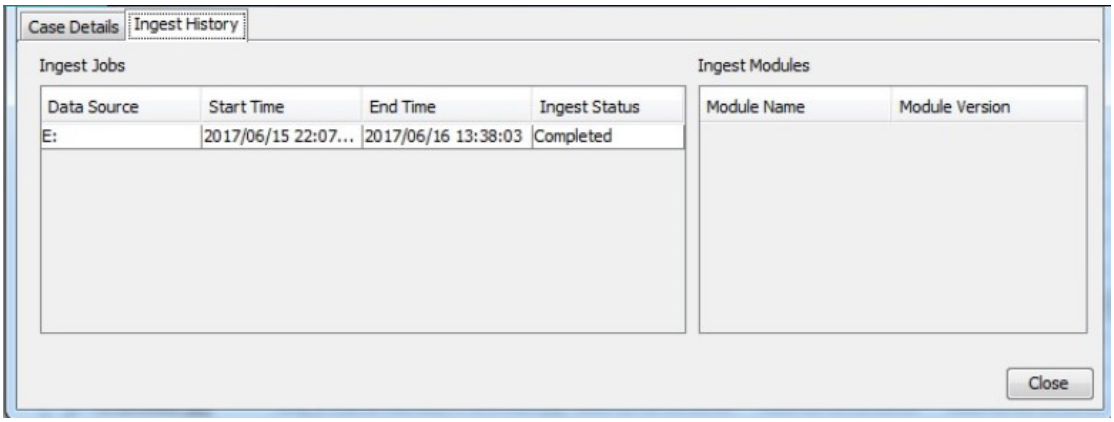


SCENARIO Number 1B (Windows7/usb):

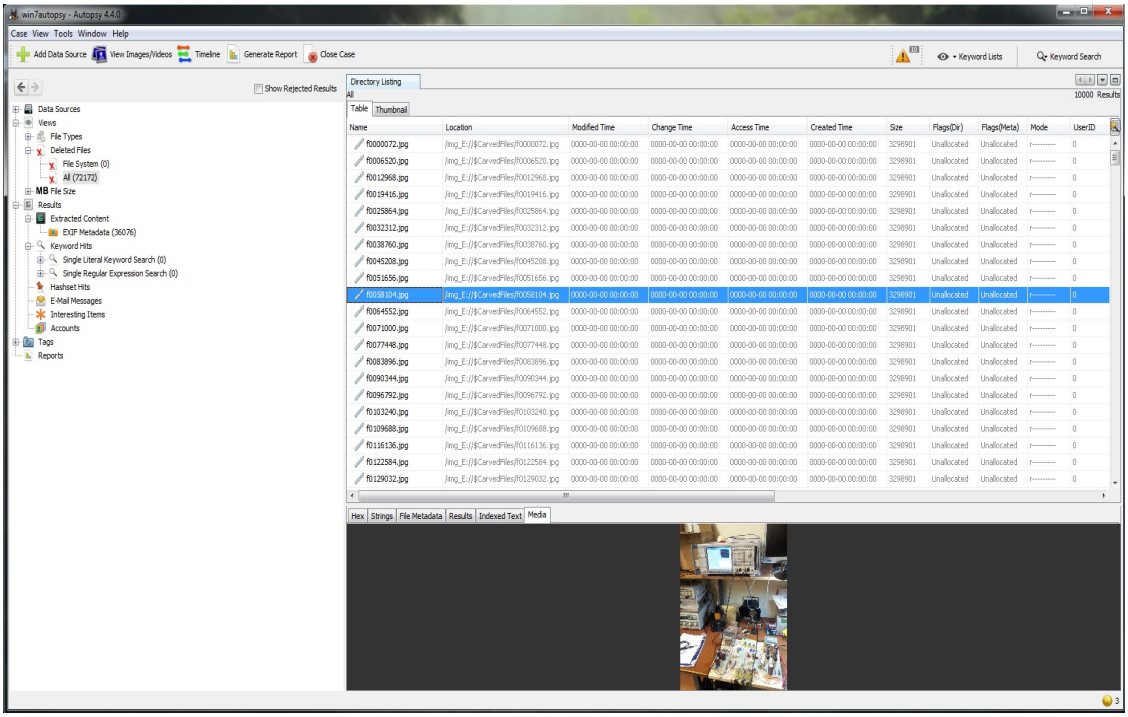
In the same scenario but with Windows7 this time things were a little bit different. The procedure was the same, checking first the TRIM command if enabled and then filling up the disk with the same image, a quick format and after that we started searching. Take a look what happened.

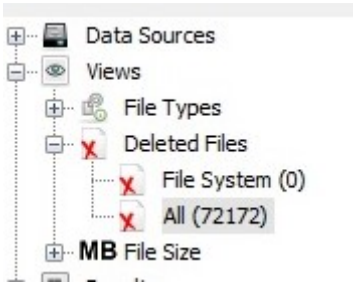
The procedure this time for the Autopsy software lasted about 13 and a half hours and in the Pc Inspector we had the same result as the scenario 1A (also the same time of procedure).

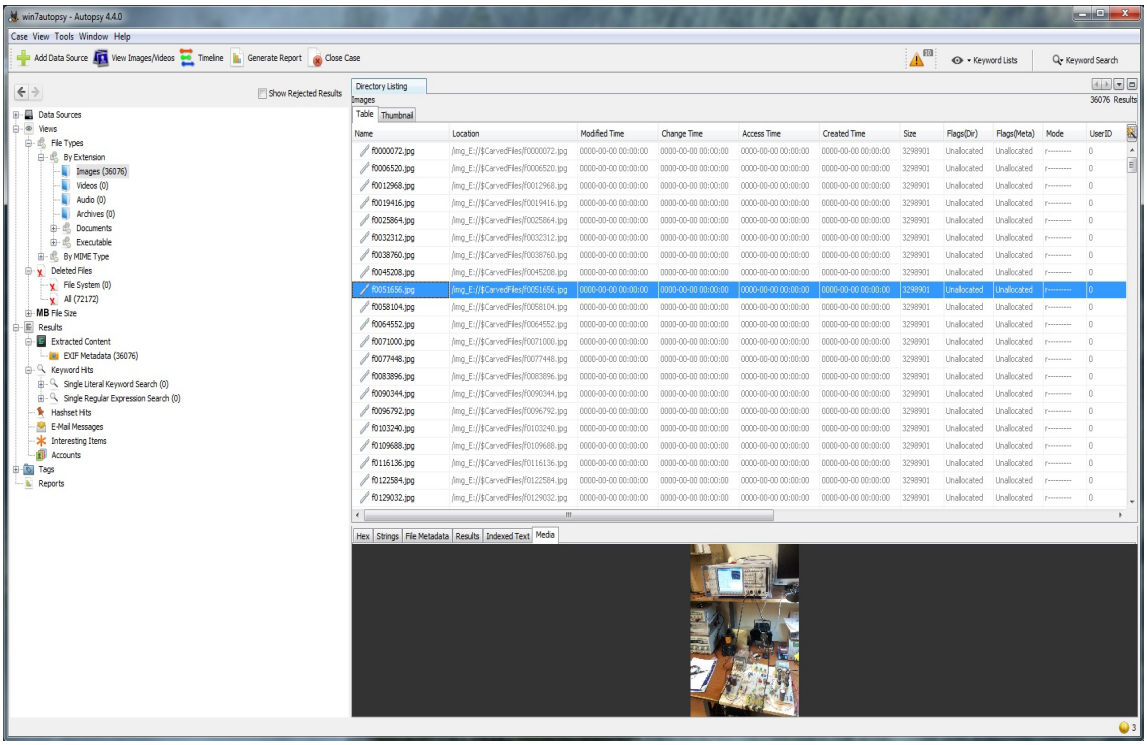


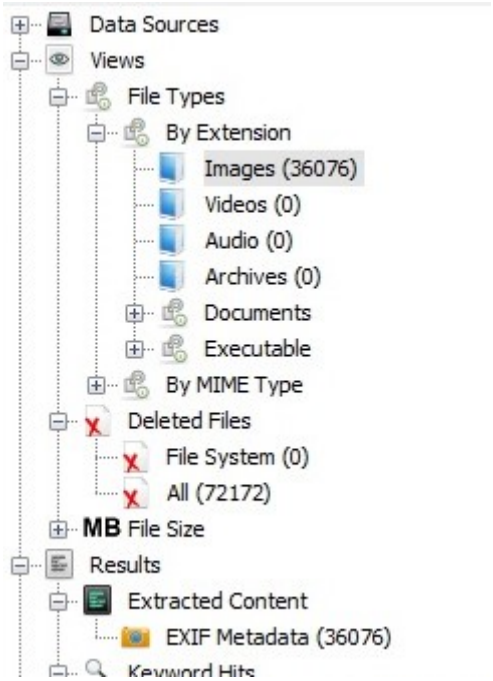


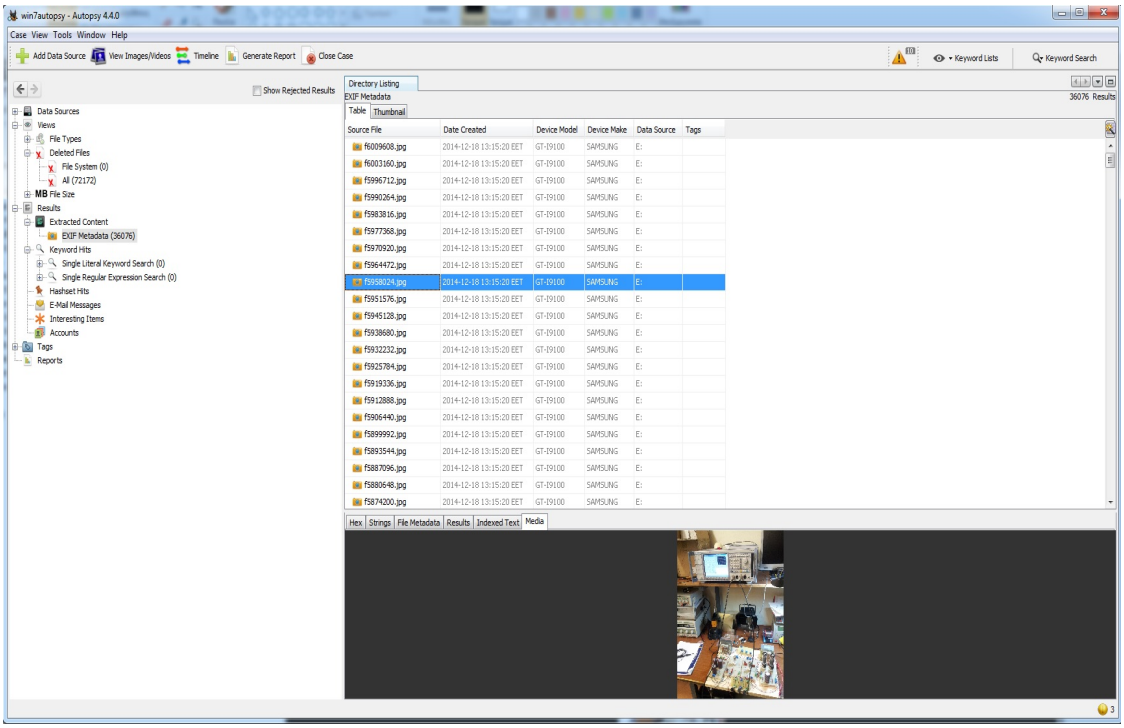
So this time the Autopsy not only found all the files from our disk but also found that the disk was rewritten twice, giving us proof about this. The Autopsy in the sector Deleted Files/All has the number of 72.172 , which if divided by 2 the result is 36.086 the actual number of files written on our disk!


win7autopsy - Autopsy 4.4.0
Case View Tools Window Help
Add Data Source
View Images/Videos
Timeline
Generate Report
Close Case
Keyword Lists
Keyword Search
Show Rejected Results
Directory Listing
Data Sources
Views
File Types
Deleted Files
File System (0)
All (72172)
MB File Size
Results
Extracted Content
EXIF Metadata (36076)
Keyword Hits
Single Literal Keyword Search (0)
Single Regular Expression Search (0)
Hashset Hits
E-Mail Messages
Interesting Items
Accounts
Tags
Reports
Table
Thumbnail
Hex
Strings
File Metadata
Results
Indexed Text
Media

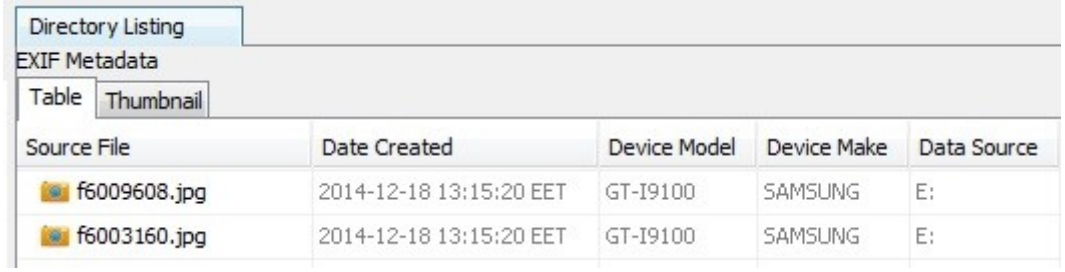

Data Sources
Views
File Types
Deleted Files
File System (0)
All (72172)
MB File Size


win7autopsy - Autopsy 4.4.0
Case View Tools Window Help
Add Data Source
View Images/Videos
Timeline
Generate Report
Close Case
Keyword Lists
Keyword Search
Show Rejected Results
Directory Listing
Images
36076 Results
Data Sources
Views
File Types
By Extension
Images (36076)
Videos (0)
Audio (0)
Archives (0)
Documents
Executable
By MIME Type
Deleted Files
File System (0)
All (72172)
MB File Size
Results
Extracted Content
EXIF Metadata (36076)
Keyword Hits
Single Literal Keyword Search (0)
Single Regular Expression Search (0)
Hashset Hits
E-Mail Messages
Interesting Items
Accounts
Tags
Reports
Table
Thumbnail
Hex
Strings
File Metadata
Results
Indexed Text
Media

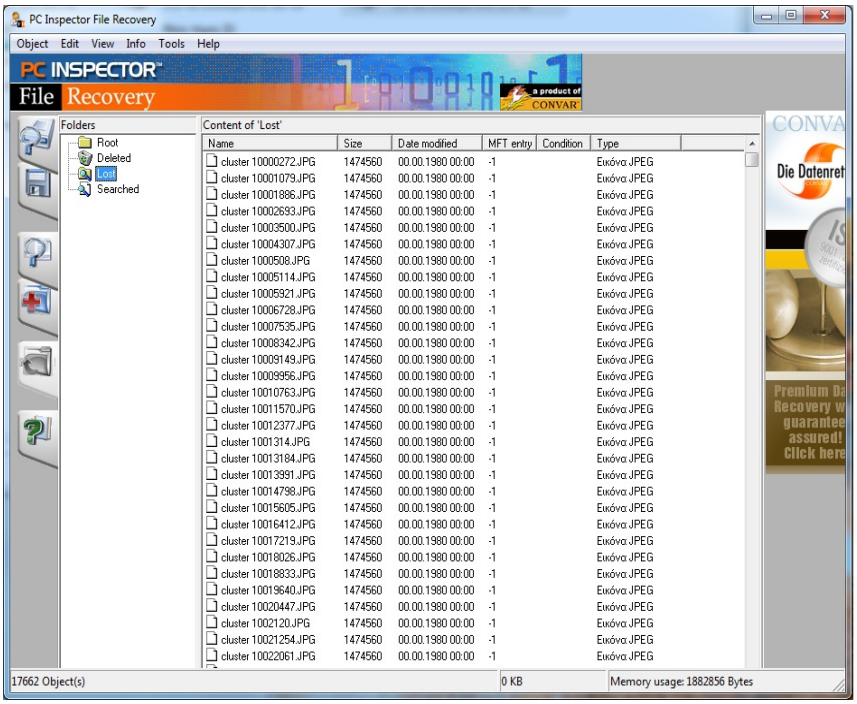


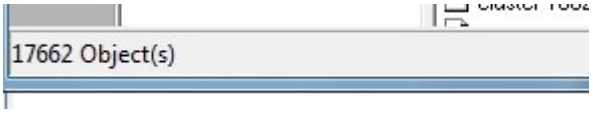


In the metadata we could also find information about the images carved this time, from what mobile was the picture taken and the actual date and time.



Now for our second open source software the results of deleted files were again zero , but in the sector of “lost files” , Pc Inspector had found all the files in clusters and also that these were written twice.

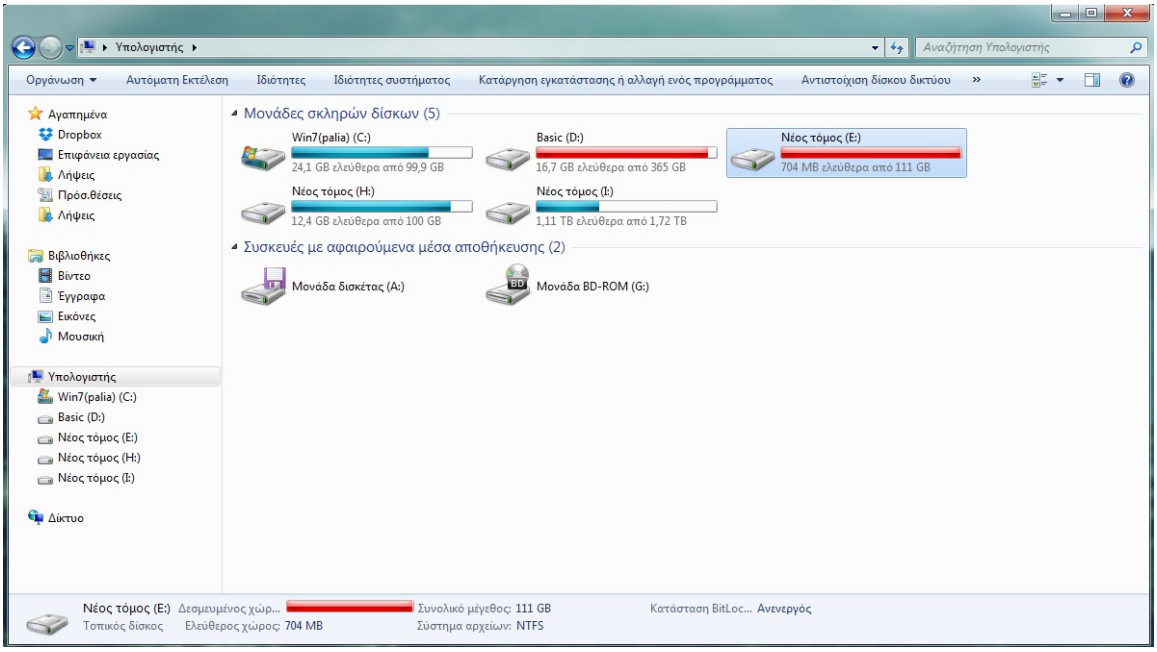


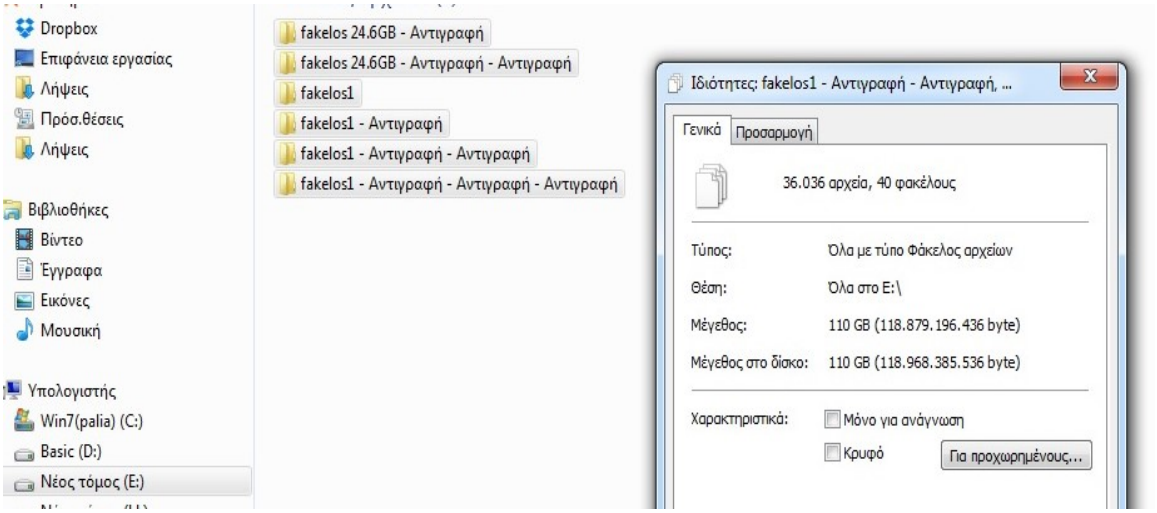


The images retrieved from that list were again half as before.

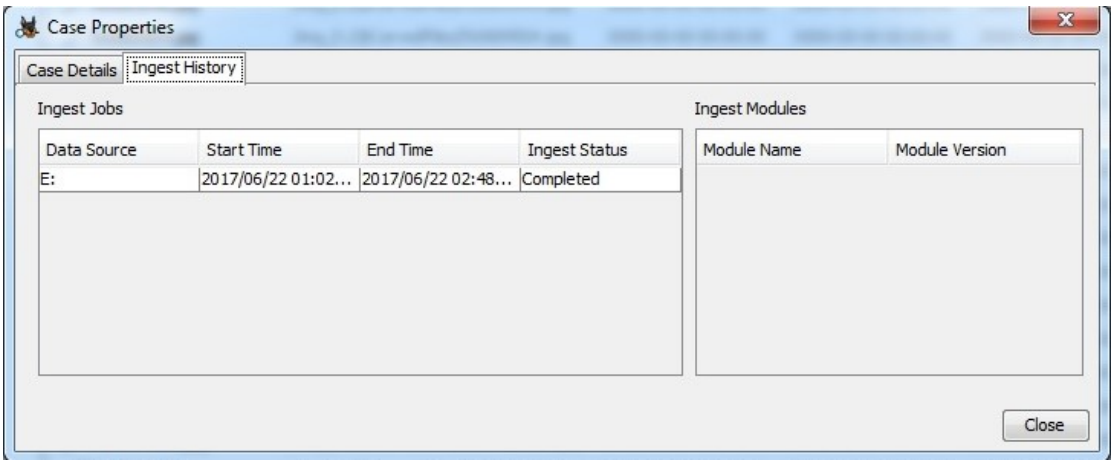
SCENARIO Number 2 (Windows7/SATA):

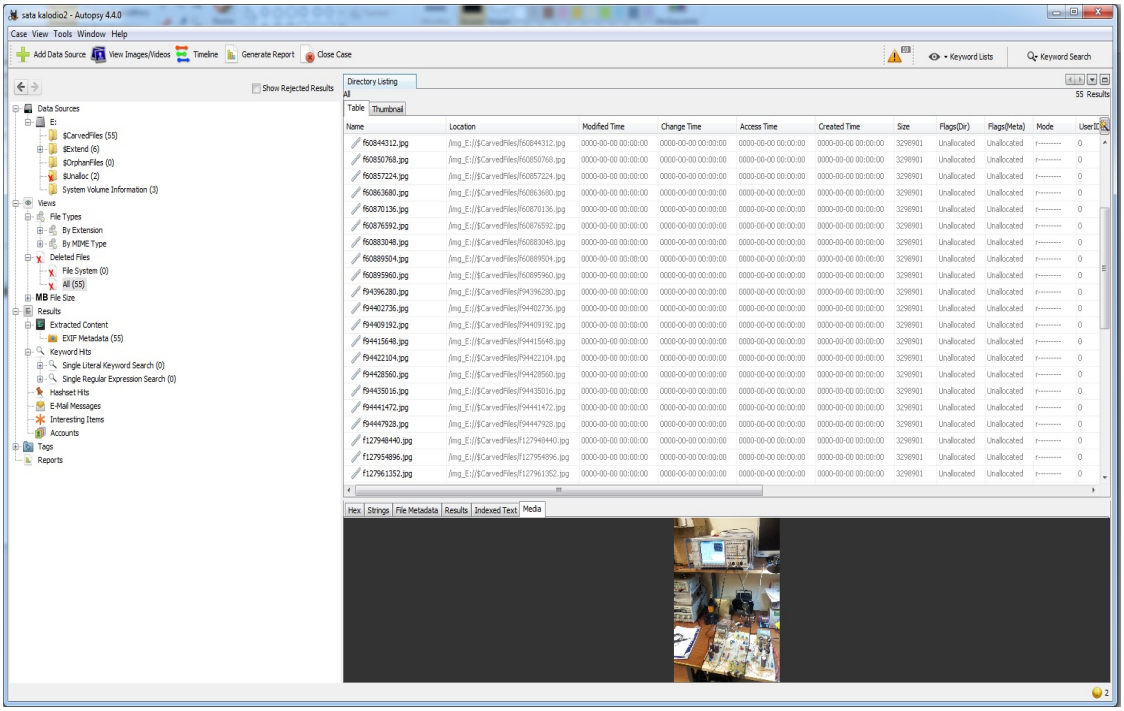
In this case, we connected our SSD with a SATA cable in our table computer, with Windows7. The procedure again the same, filling up the whole disk with the same image, quick format and then running our programs for evidence!



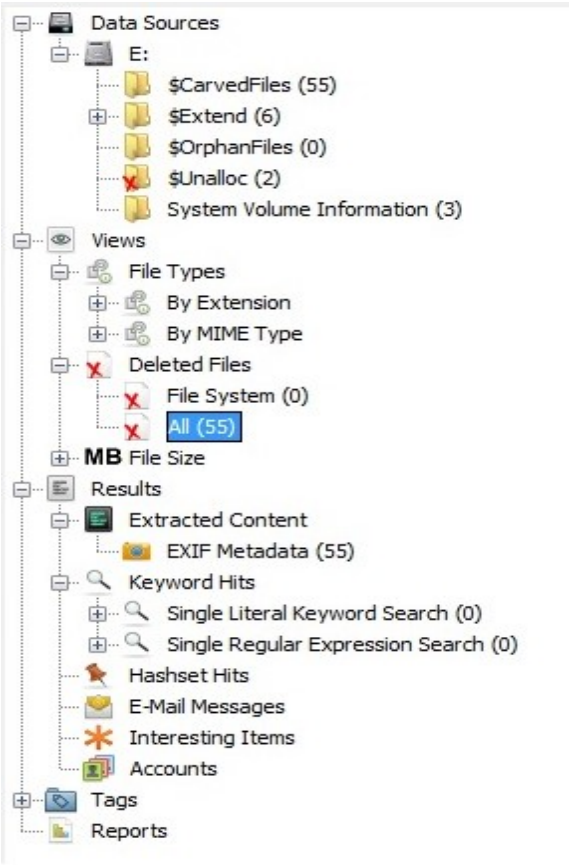


As we can see all the files are 36.036 giving us a 110GB of data. After the quick format we started with the Autopsy program and this time the procedure lasted about two hours and fifty minutes, also the results are surprising!

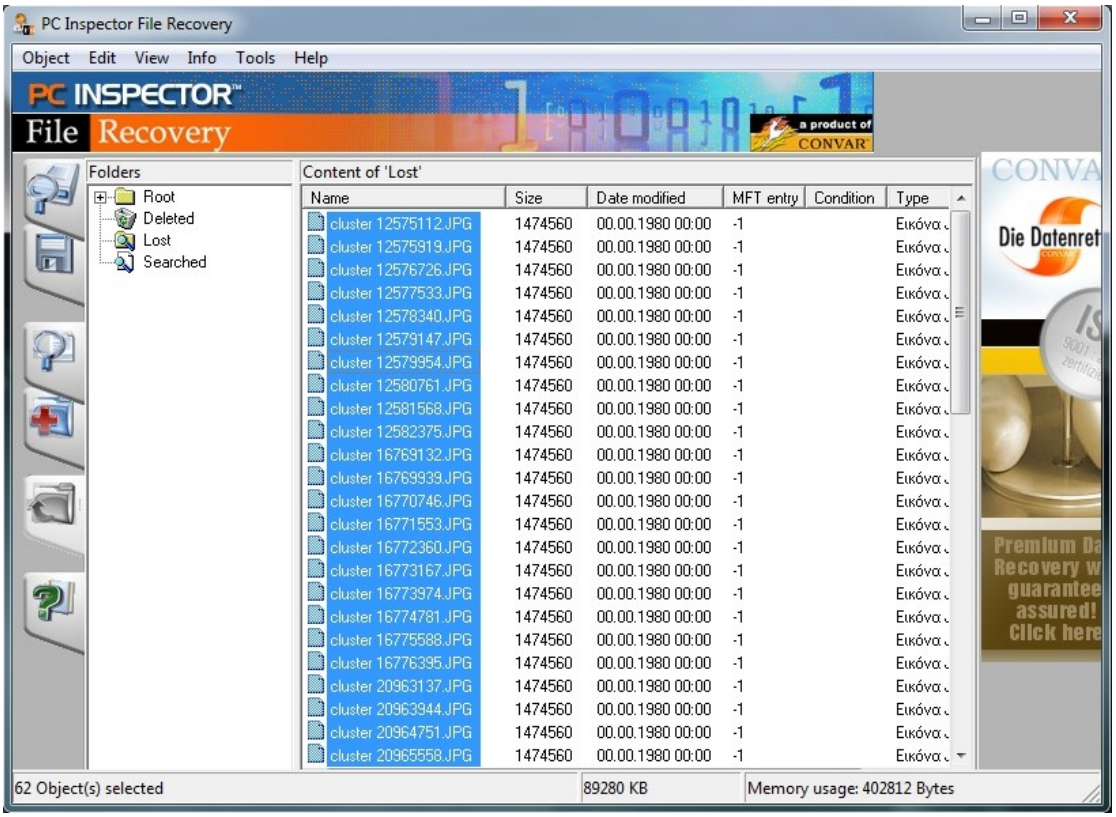




This time seems that the garbage collection was activated and actually very fast, since the procedure started a minute after the quick format and as you can see from the results, our program retrieved only 55 files out of 36.036!! Probably during the carving and since TRIM was working the controller was informed to start the garbage collection and only a few files came to our hands!

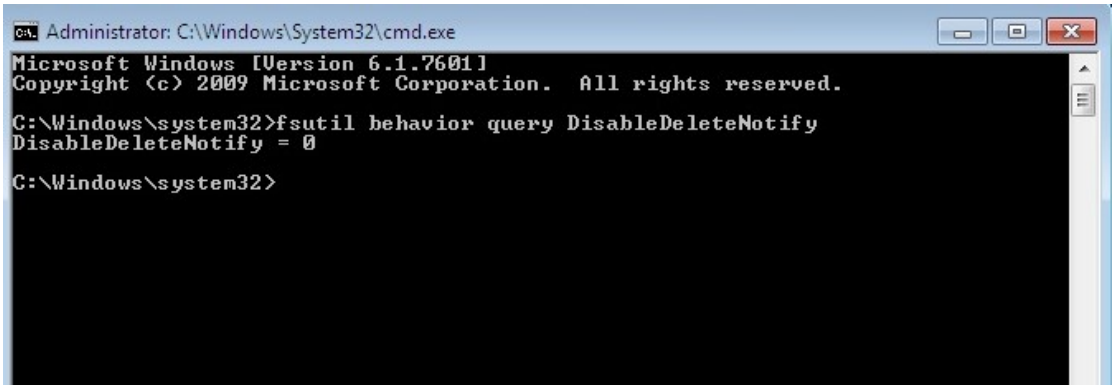


Our second software for forensics the Pc Inspector also found “nothing” but a few files in the sector “lost files” in the amount of 62 files out of 36.036! In the “deleted files” sector the result was zero.

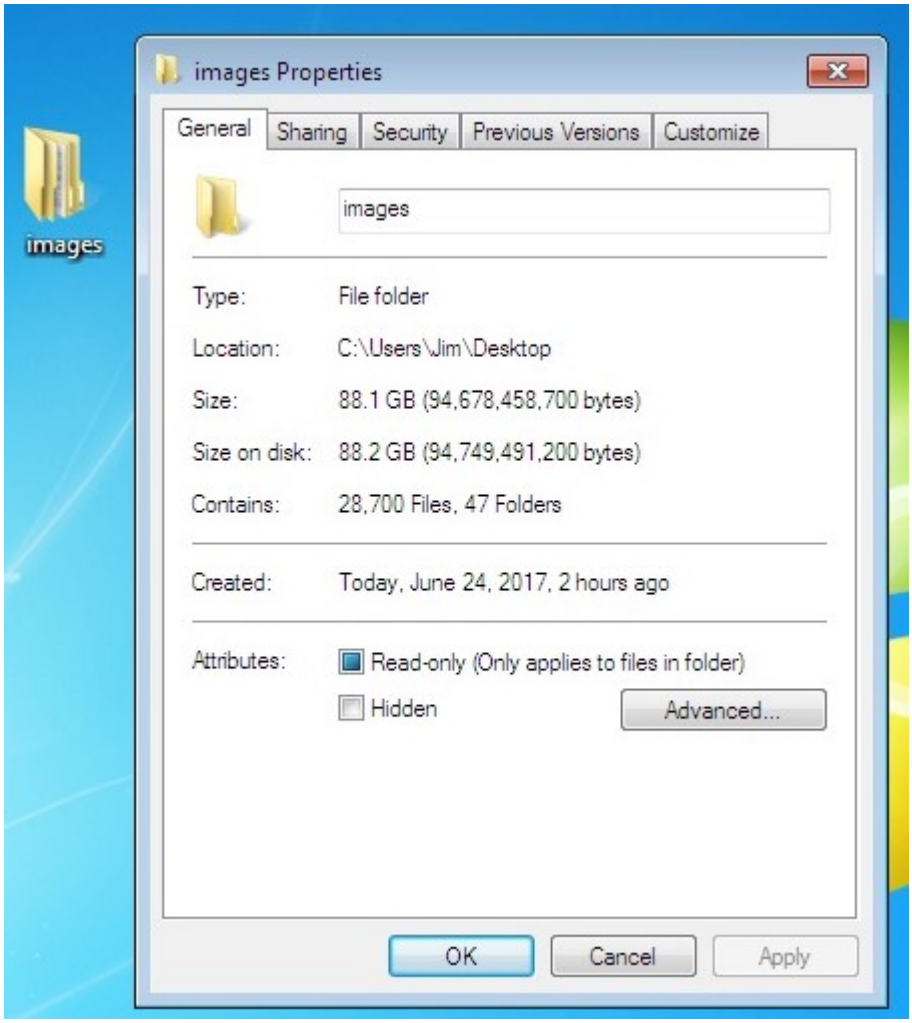


SCENARIO Number 3 (Windows7/SATA/Windows installed in the SSD):

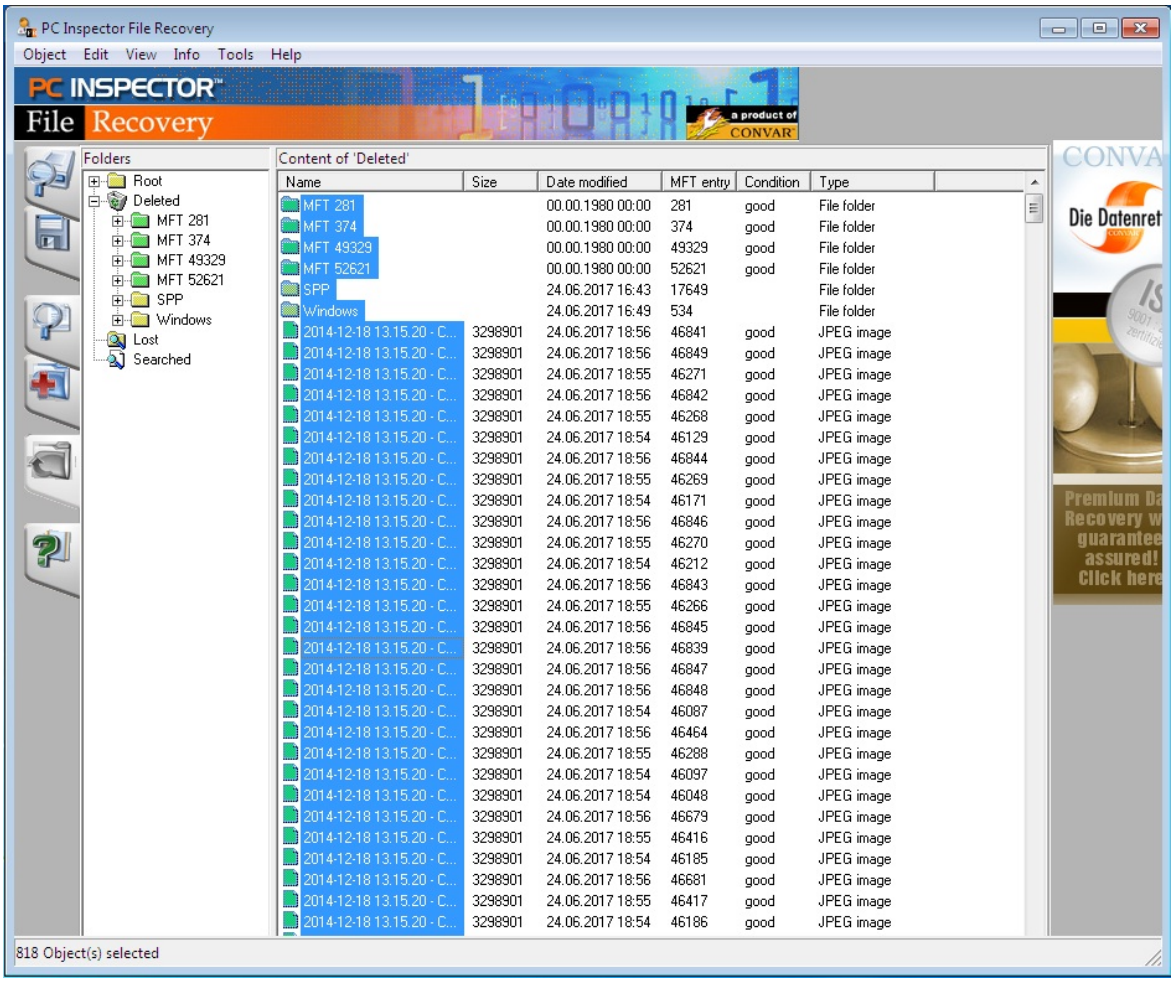
In our final scenario the plan is a bit different now. We have installed Windows7 (32-bit) in our SSD, we have checked for the TRIM if enabled (the answer was yes) and we fulfilled it with the same image as many times as possible leaving a few GB free for the envelope of the case created by our program Autopsy (during the carving procedure the Autopsy creates a big enough file with all the results of the case inside).



This time the files were simply “deleted”, no quick format was made in this case, but after that the same procedure was on. First the Pc Inspector started the work and after about a few minutes we had our first results.



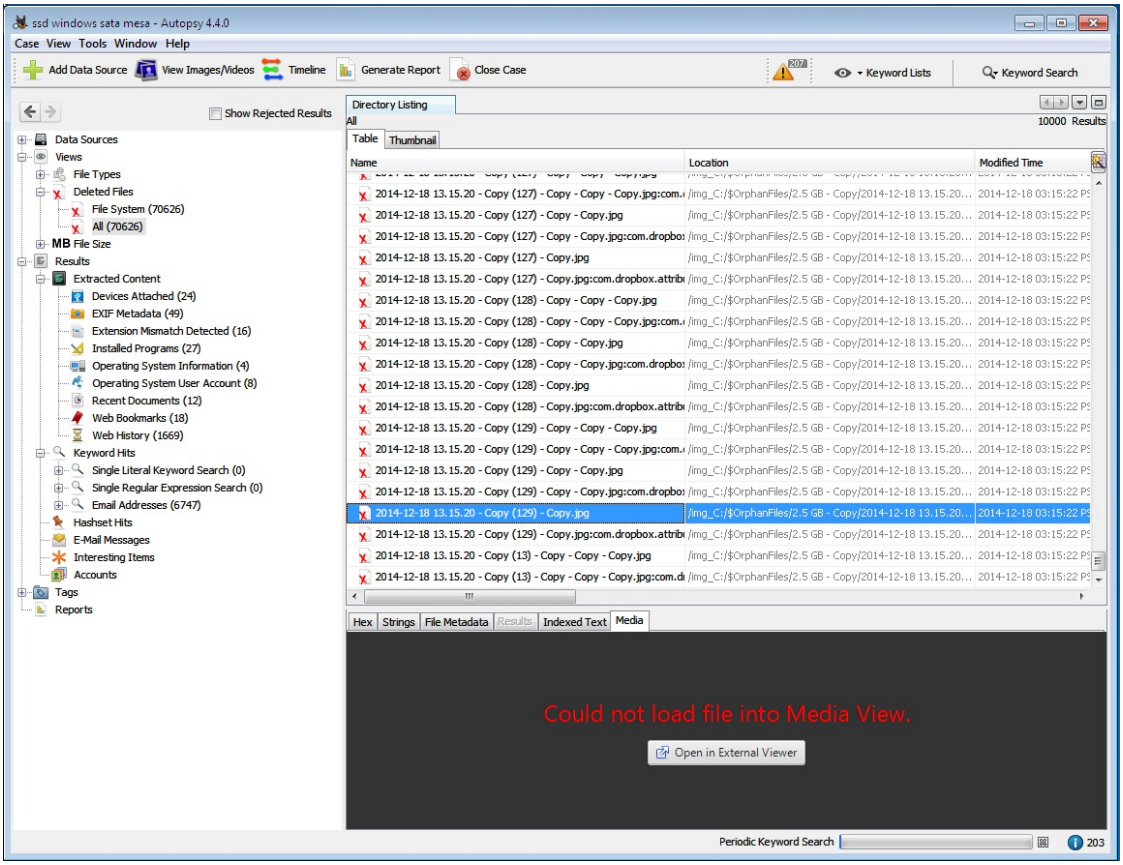
The amount of files this time was about 28.700 in number due to the space occupied by our software installed on the same disk. Below you will see the result of the Pc Inspector after carving the SSD.

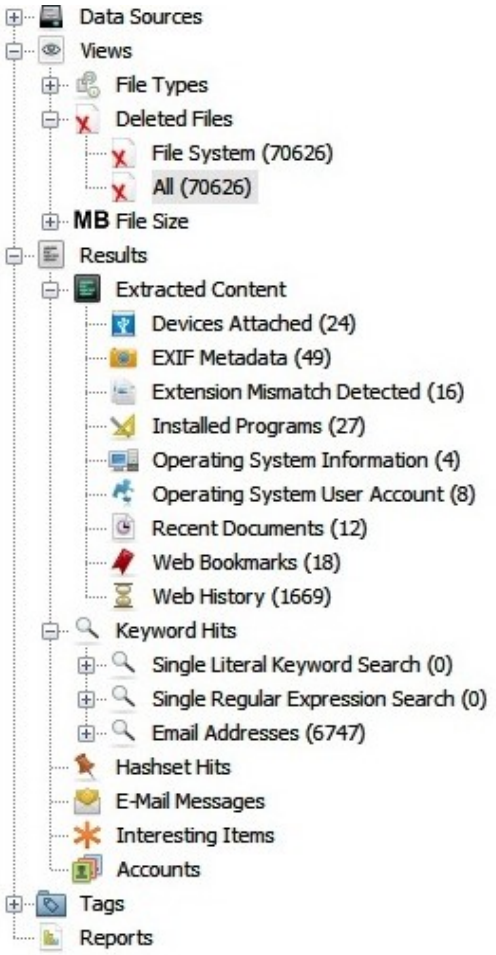


Our program managed to carve only 818 file with some of them being our image, also these ones could not be opened by any program to be seen in our pc. It seems that TRIM worked and nothing was left behind after the garbage collection!

Next in turn was the Autopsy, with the procedure lasting about 13 hours and 40 minutes take a look what happened! Our program found many information from the disk for everything else but our picture file! Well actually it found that in our disk there were written 70.626 files and these were later deleted (here actually is the two times of writing the files and formatting the disk after that) but in the end these files could not be retrieved and seen by any program!

The garbage collection also worked before we could have any useful information or evidence in our hands giving us nothing in the end.


ssd windows sata mesa - Autopsy 4.4.0
Case View Tools Window Help
Add Data Source
View Images/Videos
Timeline
Generate Report
Close Case
Keyword Lists
Keyword Search
Show Rejected Results
Directory Listing
10000 Results
Table
Thumbnail
Name
Location
Modified Time
Data Sources
Views
File Types
Deleted Files
File System (70626)
All (70626)
MB File Size
Results
Extracted Content
Devices Attached (24)
EXIF Metadata (49)
Extension Mismatch Detected (16)
Installed Programs (27)
Operating System Information (4)
Operating System User Account (8)
Recent Documents (12)
Web Bookmarks (18)
Web History (1669)
Keyword Hits
Single Literal Keyword Search (0)
Single Regular Expression Search (0)
Email Addresses (6747)
Hashset Hits
E-Mail Messages
Interesting Items
Accounts
Tags
Reports
Hex
Strings
File Metadata
Results
Indexed Text
Media
Could not load file into Media View.
Open in External Viewer
Periodic Keyword Search
203


Data Sources
Views
File Types
Deleted Files
File System (70626)
All (70626)
MB File Size
Results
Extracted Content
Devices Attached (24)
EXIF Metadata (49)
Extension Mismatch Detected (16)
Installed Programs (27)
Operating System Information (4)
Operating System User Account (8)
Recent Documents (12)
Web Bookmarks (18)
Web History (1669)
Keyword Hits
Single Literal Keyword Search (0)
Single Regular Expression Search (0)
Email Addresses (6747)
Hashset Hits
E-Mail Messages
Interesting Items
Accounts
Tags
Reports

Conclusion

Finally we have reached an end to this research. The results were truly surprising since what was made as hypothesis mostly came true. On the one hand, TRIM although enabled will not pass to the controller of our SSD when connected through usb port to our computer, as a result the garbage collection never starts. On the other hand when our disk is connected to our pc through SATA cable or with software installed on it the garbage collection starts working immediately!

The only dull spot in our survey was in the first scenario with the Autopsy program carving only a thousand of files instead all of them, perhaps time matters perhaps not (talking about when the garbage collection will start working).

The truth is that in the end there are some methods that evidence can be retrieved by an SSD disk although this seems impossible theoretically. Thus there is still pretty room for extra research in this matter.

Bibliography

1. http://www.digitalrecordsforensics.org/drf_links.cfm
2. https://www.extremetech.com/extreme/210492-extremetech-explains-how-do-ssds-work
3. https://en.wikipedia.org/wiki/Solid-state_drive
4. https://www.geek.com/games/how-its-made-crucial-ssds-come-from-very-clean-rooms-1536568/
5. http://computer.howstuffworks.com/solid-state-drive.htm
6. http://www.forensicfocus.com
7. https://github.com/CyberShadow/trimcheck
8. https://www.techrepublic.com/article/disk-wiping-and-data-forensics-separating-myth-from-science/
9. https://www.techrepublic.com/article/why-forensics-investigators-must-handle-solid-state-drives-with-care/
10. https://www.techrepublic.com/article/all-flash-arrays-the-smart-persons-guide/
11. https://www.defcon.org/html/defcon-24/dc-24-speakers.html#Kopchak
12. http://www.zdnet.com/article/how-to-really-erase-any-drive-even-ssds-in-2016/
13. http://www.infinul.com/blog/adventures-with-ssd-forensics/
14. http://www.thessdreview.com/daily-news/latest-buzz/garbage-collection-and-trim-in-ssds-explained-an-ssd-primer/3/
15. https://arstechnica.com/gadgets/2015/04/ask-ars-my-ssd-does-garbage-collection-so-i-dont-need-trim-right/
16. https://serverfault.com/questions/733489/how-to-disable-automatic-garbage-collection-on-an-ssd
17. http://www.pcworld.com/article/2088341/how-to-restore-your-ssd-to-peak-performance.html
18. http://www.digitalforensicsassociation.org/opensource-tools/