



INTERNATIONAL
HELLENIC
UNIVERSITY

Digital Triage In Forensics Investigation

Nikolaos Bakirtzis

SID: 3301120004

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

NOVEMBER 2013

THESSALONIKI – GREECE



INTERNATIONAL
HELLENIC
UNIVERSITY

Digital Triage In Forensics Investigation

Nikolaos Bakirtzis

SID: 3301120004

Supervisor:	Prof. Vasilios Katos
Supervising Committee	Assoc. Prof.
Members:	Assist. Prof.

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

NOVEMBER 2013

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in ICT Systems at the International Hellenic University.

The digital triage in forensics investigation can really make the digital investigation a success or a disaster depending on numerous factors. There are numerous triage tools freely available online but there is no mature framework for practically testing and evaluating them. In the following pages we will analyze four open source triage tools and identify the advantages and drawbacks of each of them. We will also test their compliance to published forensic principles (ACPO).

The results prove that due to high complexity and variety of system configurations, triage tools should become more adaptable, in dynamic and manual manner, depending on the case and context, instead of sustaining a monolithic functionality.

After identifying the problem, an effort was made to create a program, that has the ability to search a whole computer, or any partition or file chosen, for files with any possible extension, that are installed or created by the user. This was possible by comparing the MD5 hashes of the files. In this way the investigator can search, in a very short time, the computer under examination, for installed and created files or programs, altered programs, possible malware and harmful programs.

This program can have even greater usability if it is incorporated into other digital triage programs or if it is enhanced with more advanced functionality.

Special thanks is given to my supervisor Prof. Vasilios Katos that showed me the way to where address my research and solved any issues raised giving insightful feedback.

Nikolaos Bakirtzis

05-11-2013

Contents

ABSTRACT.....	III
CONTENTS.....	IV
1 INTRODUCTION.....	1
2 LITERATURE REVIEW	7
2.1 DIGITAL FORENSIC PROCESS.....	8
2.1.1 <i>Techniques</i>	11
2.1.2 <i>Volatile data</i>	11
2.1.3 <i>Analysis tools</i>	12
2.1.4 <i>Certifications</i>	12
2.2 DIGITAL TRIAGE IN FORENSICS INVESTIGATION.....	12
2.3 TESTING TRIAGE TOOLS	17
2.4 BULK EXTRACTOR.....	17
2.4.1 <i>Advantages</i>	18
2.4.2 <i>Drawbacks</i>	18
2.5 TRIAGEIR V.0.79	19
2.5.1 <i>Advantages</i>	19
2.5.2 <i>Drawbacks</i>	20
2.6 TR3SECURE.....	20
2.6.1 <i>Advantages</i>	21
2.6.2 <i>Drawbacks</i>	21
2.7 KLUDGE 3.20110223	21
2.7.1 <i>Advantages</i>	22
2.7.2 <i>Drawbacks</i>	22
2.8 SYNOPSIS	23
2.8.1 <i>Bulk Extractor</i>	23
2.8.2 <i>TriageIR, TR3Secure and Kludge</i>	25
2.9 SUGGESTIONS.....	28
2.9.1 <i>Bulk Extractor</i>	29

2.9.2	<i>TriageIR 0.79</i>	29
2.9.3	<i>TR3Secure</i>	29
2.9.4	<i>Kludge 3.20110223</i>	29
2.10	FUTURE WORK ON THE FOUR TOOLS	30
3	PROBLEM DEFINITION	31
3.1	BULK EXTRACTOR.....	32
3.2	TRIAGEIR V.0.79	32
3.3	TR3SECURE.....	32
3.4	KLUDGE 3.20110223.....	33
4	CONTRIBUTION	35
4.1	FIRST THOUGHTS	35
4.2	THE NEW TARGET.....	37
4.3	THE DEVELOPING PROCESS	38
4.4	THE SCANNER PROGRAM - HOW IT WORKS	39
4.4.1	<i>Help mode</i>	40
4.4.2	<i>Scan mode with no parameters</i>	42
4.4.3	<i>Scan mode with parameters</i>	47
4.4.4	<i>Compare mode with parameters</i>	52
4.5	THE SCANNER_VERSION_2 PROGRAM	62
4.6	THE SCANNER_VERSION_3 PROGRAM	63
4.7	ADVANTAGES.....	70
4.8	DRAWBACKS.....	71
4.9	EVALUATION	71
4.10	SUGGESTIONS	71
4.11	FUTURE WORK – ENHANCEMENTS	72
4.12	CONCLUSION	73
5	PERSONAL REFLECTION	75
6	CONCLUSIONS	78
6.1	DIGITAL FORENSICS IS DIFFERENT.....	78
6.1.1	<i>The challenge of data diversity</i>	79
6.1.2	<i>Data scale</i>	79
6.1.3	<i>Temporal diversity: the never-ending upgrade cycle</i>	79

6.1.4	<i>Human capital demands and limitations</i>	80
6.1.5	<i>The CSI effect</i>	80
6.1.6	<i>The cost of development and the role of government</i>	80
6.2	LESSONS LEARNED DEVELOPING DIGITAL FORENSICS TOOLS	81
6.2.1	<i>Platform and language</i>	81
6.2.2	<i>Parallelism and high performance computing</i>	81
6.2.3	<i>All-in-one tools vs. single-use tools</i>	81
6.2.4	<i>Evidence container file formats</i>	82
6.3	CONCLUSION.....	82
	REFERENCES	85
	APPENDIX	91

1 Introduction

Computers and computing devices are more and more a part of our lives. Not only most of us have a computer, such as desktops and laptops but we also have smart phones, tablets and GPSs in our cars. If we don't have a phone available, we can use our laptop or tablet and communicate with others using instant messaging, email, Twitter, or Skype applications. Furthermore computers become more and more a part of our lives, so does crime involving those devices. Whether it's "cyberbullying" or "cyberstalking," identity theft, or intrusions and data breaches, all result in some form of data theft. A wide number of real-world physical crimes are now being committed through the use of computers, and as such, get renamed by prepending "cyber" to the description. As we transfer lot of the things that we did in the real world to the online world, we became targets of cybercrime. [19]

What makes this activity even more insidious and sophisticated is that we don't recognize it for what it is, because conceptually, the online world is simply so foreign to us. If someone breaks a storefront window to steal a television, there's noise, alarm, broken glass, and someone running away with the stolen goods. Cybercrime isn't like this; something isn't stolen and then absent, but is also copied. Additionally, the crime does result in something that is stolen and removed from us, but we may not understand that immediately, because we're talking about 1s and 0s in cyberspace, not for example a car in our garage. These malicious activities also increase in complexity. In many cases, the fact that a crime has happened is not obvious until someone notices a significant decrease in a bank account, which shows that the perpetrator has already gained access to systems, gathered the data needed, accessed that bank account, and left with the money. The incidents are not detected until well after they've occurred. In other cases, the malicious activity continues and even escalates after we become aware of it, because we're unable to defend from the attack. [19]

Computers, as mentioned above, can be used for committing a crime, contain evidence of a crime or be targets of the crime. Specifying the role and nature of the elec-

tronic evidence, how to analyze a crime scene that contains electronic evidence and how the responder should act to these circumstances is important. [1]

Triage is a term deriving from medicine. It is defined as “the sorting and allocation of treatment to patients and victims in battle and disaster according to priorities that target to maximize the number of survivors”. [52] In incident response the term triage is specified as the step where an analyst when receiving a report about an incident, assesses the danger, prioritizes the incident, relates it to other incidents and decides whether this report is true. [9] Triage is a way of prioritizing tasks and allocating the resources which are limited. [23]

Digital triage give to us: The knowledge what items to be taken from the scene. This helps us manage the flow of items that will be examined. This speeds the whole process. [23] This is done by executing triage software in the suspect system.

From the above definition it is obvious that the success of the investigation depends on the first actions and triage of the first responder. Correct priorities and handling of the live system may reveal an encrypted partition or a remote IP. [77]

Historically, the computer related crime concerned only a small number of victims and investigators. Nowadays this situation is changing and the impact of digital evidence within conventional investigations is very common. In addition, any investigation in the public or private sector usually involves the seizure, preservation and examination of electronic evidence. [1]

The former pull-the-plug approach is obsolete and overlooks the volatile data that will be lost. Today, investigators face the fact of sophisticated data encryption, hacking tools and malicious software that exist only in memory. [1]

The most widespread strategies to collect potential evidence are two: a) use personnel with limited forensic training and seize everything or b) use skilled experts with selective acquisition. In the first approach, there will be potential damage of digital evidence during the process. The second approach has advantages in serious crime or major incidents, but removes valuable skills from the forensic staff which reduces the throughput and capacity respond in the laboratory. [37]

Examiners need to constantly upgrade their skills, tools, and knowledge to keep up with the new technologies. The solution is not just to unplug the computer and evaluate it later. Examiners must know how to capture an image of the running memory and perform volatile memory analysis using various tools. [93] In figure 1 there is a compari-

son between live response with Sys-Internal tools vs. memory analysis on a static memory dump. [93]

		Sys-Internal vs. Memory Analysis Tools									
		<i>Network Connections</i>	<i>Open Ports and Sockets</i>	<i>Running Processes</i>	<i>Hidden Running Processes</i>	<i>Terminated Processes</i>	<i>Loaded DLLs</i>	<i>Open Files</i>	<i>OS Kernel Modules</i>	<i>Process Dumps</i>	<i>Strings</i>
Live Response											
PsList			X								
ListDLLs						X					
Handle							X				
Netstat	X	X									
Fport		X									
Userdump								X			
Strings									X		
PsLoggedOn											X
Memory Analysis											
Volatility	X	X	X	X	X	X	X	X			X
PTFinder			X	X	X						

Figure 1: Live response with Sys-Internal tools vs. memory analysis on a static memory dump. [93]

Electronic evidence is very fragile. It can be altered, damaged, or destroyed by wrong handling and improper examination. This is why special actions should be taken to document, collect, preserve and examine the digital evidence. Failure to do so can make it useless or drive to a faulty conclusion. [1]

A usual forensic lifecycle for a hard drive can be the following:

- Remove the disk from the computer.
- Connect this disk to the examiners computer with a writeblocker that prevents any alterations.
- Make an image of the disk that represents all the contents of the disk.
- Make cryptographic hash values for each and every digital object and for the disk.
- Search for malware.
- View files.

- Search metadata.
- Extract metadata.
- Identify and bookmark privacy concerns.
- Create replicates and examine them in an emulator.
- Convert digital replicates to interoperable files known as digital facsimiles.
- Analyze metadata.
- Create log files during the examination process.
- Make a forensic report as documentation. [48]

All the above were given me the incentive for this dissertation which contains the following: In the Literature Review there is an explanation and analysis of what digital triage in forensics investigation is about and what is the current picture. Digital triage categories, characteristics, scopes, goals, objectives and achievements till nowadays. References to four existing tools that are free and things to consider when choosing the right tool. There is research for these four open source triage tools to help us understand and study the most important issues concerning these digital triage processes. There is description of the effort needed and the practical challenges an analyst may encounter when employing them. Also these tools are evaluated depending on the requirements of the ACPO principles, a practice guide developed for digital forensics. [77] In bibliography they analyzed and they also proposed ways of improving these tools.

In the Problem Definition there is an analysis of what existing tools provide, what they don't offer and why this is important. Why the lack of a proper and advanced tool is so important. There is gap analysis and drawbacks of the presented tools. Also possible contributions of a new tool.

In the Contribution is presented the thought to make a program that could search the computer targeted and find added files, altered files, infected files or even malware constructed by the suspect. As a start the target was to make a program that would search the computer under investigation for the files we want to find. To have the ability to choose the type of files we want, the domain of the computer to search in, the file the results would be saved in. The results would show the name and the path of the files found. The next target was to find all the MD5 hashes of the files we want. This could help to find which files were modified in comparison to the original ones. This could be achieved by producing the MD5 hashes [50] of the files found and comparing them with the MD5 hashes of the original files. In the next stage all the above was necessary not to

be hard coded but to offer the examiner the ability to insert the desired values as parameters during the execution. During the development many features were added and the program was upgraded and became more complicated until it reached its final form. My target was to ease the forensics analyst and provide him an overview of the hard disc with a glimpse. Demonstrate and present the contents of the hard disc in a way that helps the investigator have a more concrete image of the contents of the disc. Help him take the right decisions fast.

In the Personal Reflection there is the personal opinion about the subjects analyzed, the problems, the solutions and the contribution offered. What went wrong, what to avoid the next time, where to improve and upgrade in the future.

In the Conclusions there is the summary of the work of the dissertation. Evaluation of achievements and final thoughts.

This dissertation is structured as follows:

Chapter 2 contains the literature review. An extensive research of what has been published by accredited scholars and researchers on the topic has been made. In chapter 2.1 digital forensic process is analyzed including techniques, volatile data, analysis tools and certifications that can be obtained. In chapter 2.2 digital triage in forensics investigation is explained. In chapter 2.3 we talk about testing triage tools. In chapter 2.4 the tool Bulk Extractor is tested retrieving its advantages and drawbacks. In chapter 2.5 the tool TriageIR is tested highlighting its advantages and drawbacks. In chapter 2.6 the tool TR3Secure is tested mentioning its advantages and drawbacks. In chapter 2.7 the tool Kludge is tested acknowledging its advantages and drawbacks. In chapter 2.8 there is a synopsis of the testing process and in chapter 2.9 there are suggestions for each tool. In chapter 2.10 there are some proposals on future work on the four tools.

Chapter 3 contains the definition of the problem in digital forensics. In chapters 3.1, 3.2, 3.3, 3.4 problems are specified for each and every one of the four tools.

Chapter 4 contains the contribution that this dissertation has made to the topic. In chapter 4.1 there are the first thoughts about what program to develop. In chapter 4.2 the new target chosen is analyzed. In chapter 4.3 the developing process of the program is explained. In chapter 4.4 the functionality of the scanner program is described. In chapter 4.5 the functionality of the scanner_version_2 program is described. In chapter 4.6 the functionality of the scanner_version_3 program is described. In chapter 4.7 there are the advantages of the program. In chapter 4.8 there are the drawbacks of the pro-

gram. In chapter 4.9 there is the evaluation of the program. In chapter 4.10 some suggestions are made for better use of the program. In chapter 4.11 future work and enhancements for the program are included. In chapter 4.12 there is the conclusion about the program.

Chapter 5 contains the personal reflection and the personal thoughts about the topic of the dissertation.

Chapter 6 contains the conclusions after ending this dissertation and creating the scanner program. In chapter 6.1 is explained why digital forensics is different from any other topic. In chapter 6.2 are mentioned the lessons learned from developing digital forensics tools. In chapter 6.3 there are the overall conclusions and the final thoughts.

In the appendix there is the source code of the scanner.php, scanner_version_2.php and scanner_version_3.php programs.

2 Literature Review

A technical definition for digital forensics is the following: the tools and techniques that are used to preserve, recover and analyze digital evidence and data on digital devices or being transmitted by them. [3]

Digital forensics is a part of forensic science including the recovery and examination of evidence found in digital devices that are related to computer crime. [71] [18] Digital forensics investigations apply in many cases. The most usual is to check a hypothesis before the criminal or civil courts. Forensics are also used in the private sector; for example during internal investigations or intrusion cases. [24] There are many types of forensics: computer forensics, network forensics, forensic data analysis and mobile device forensics. [24]

Digital forensics are also used to verify evidence of the suspects, check alibis or statements, determine intent, identify sources, or authenticate documents. [85] Investigations have a wider scope of the other forensic analysis also including complex timelines or hypotheses. [17]

Digital forensic process begins with the collection, duplication, and authentication of all the data before the examination begins. These first three phases have the biggest costs. Duplication is a standard practice for all laboratories but it takes great time which has become a big problem. One solution is the pre-examination techniques known as digital triage. These techniques help the prioritization and lead the examination. [16]

Digital triage in forensics investigation is a rapid growing sector with serious purpose, numerous implementations, multiple objectives and many achievements. There are many tools free and licensed and in the following paragraphs we are going to analyze four well known free tools Bulk Extractor [12], TriageIR [90], TR3Secure [89] and Kludge [46].

Computer forensics, also known as computer forensic science [60] is a branch of digital forensic science that refers to legal evidence stored in computers and other digital media. The aim of computer forensics is to analyze digital data forensically with the

target to identify, preserve, recover, analyze and present important information about these data. [21]

Since 1980 the computer crime has developed and grown radically. [49] Nowadays computer forensics investigates a great range of crimes, such as child pornography, fraud, cyber stalking, murder and rape. Computer forensics is also used in civil proceedings for information gathering. [21]

These techniques and knowledge are used to extract useful conclusions from a digital artifact; like a computer system, storage medium and electronic document. [99] The target of forensic examination can be from simple information extraction to rebuilding a timeline of events. [39]

In court the computer forensic evidence are required to have information that are authentic, reliably obtained, and admissible. [2] Many countries have developed guidelines and practices for evidence recovery. In United Kingdom, analysts use the Association of Chief Police Officers (ACPO) guidelines which help ensure the authenticity and integrity of evidence. [21]

Computer forensics process usually include four stages:

- Acquisition.
- Analysis.
- Evaluation.
- Presentation. [22]

Mobile device forensics is a branch of digital forensics, used in the recovery of digital evidence or data from a mobile device, PDA devices, GPS devices and tablet computers. [54]

The forensic examination of mobile devices is a relatively new field, starting from the early 2000s. The big spread of phones caused a bottleneck in the forensic examination of the mobile devices. [80]

2.1 Digital forensic process

Computer forensic examination most times uses the process of standard digital examination. [27] Examinations are executed basically on static data and not so much on

"live" systems. This has changed from the way early forensic practices used to do. In that time the lack of special programs drove the examiners to work mainly on live data. In figure 2 we see the traditional process models. In figure 3 we see the computer forensics field triage process model (CFFTP). [74]

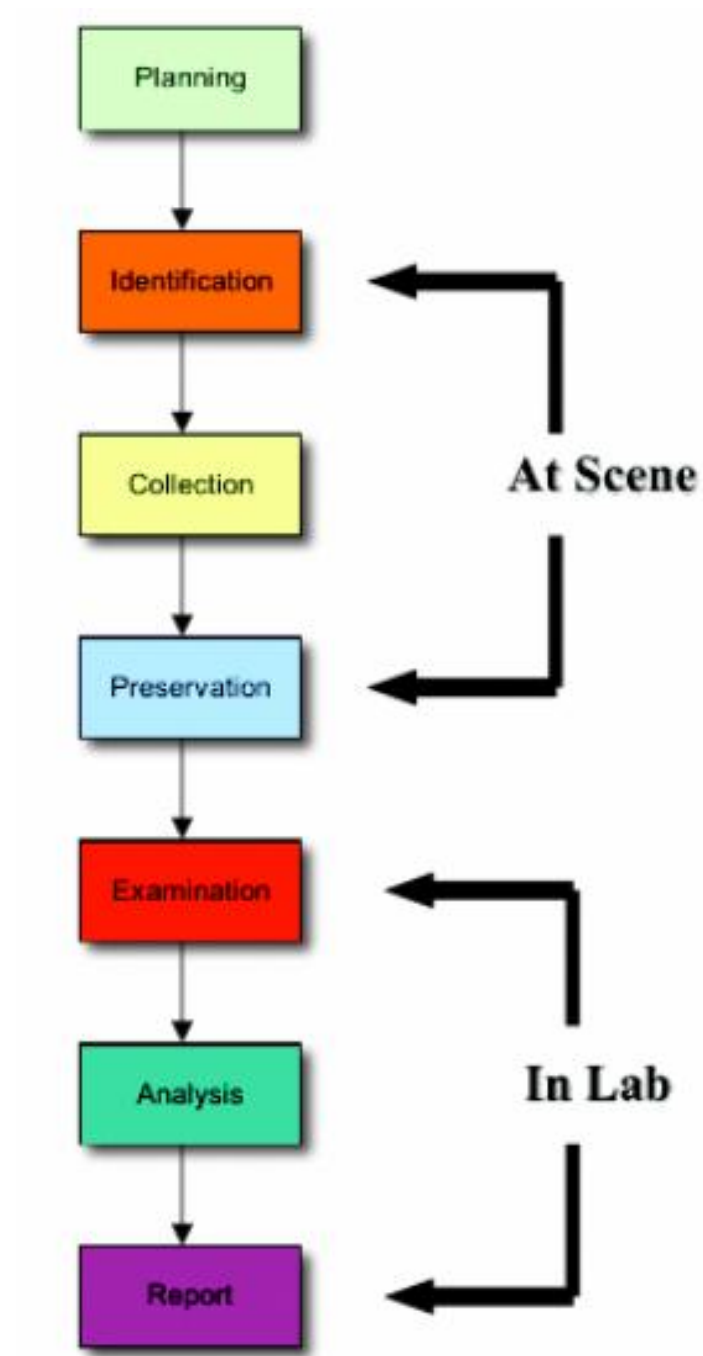


Figure 2 – Traditional Process Models [74]

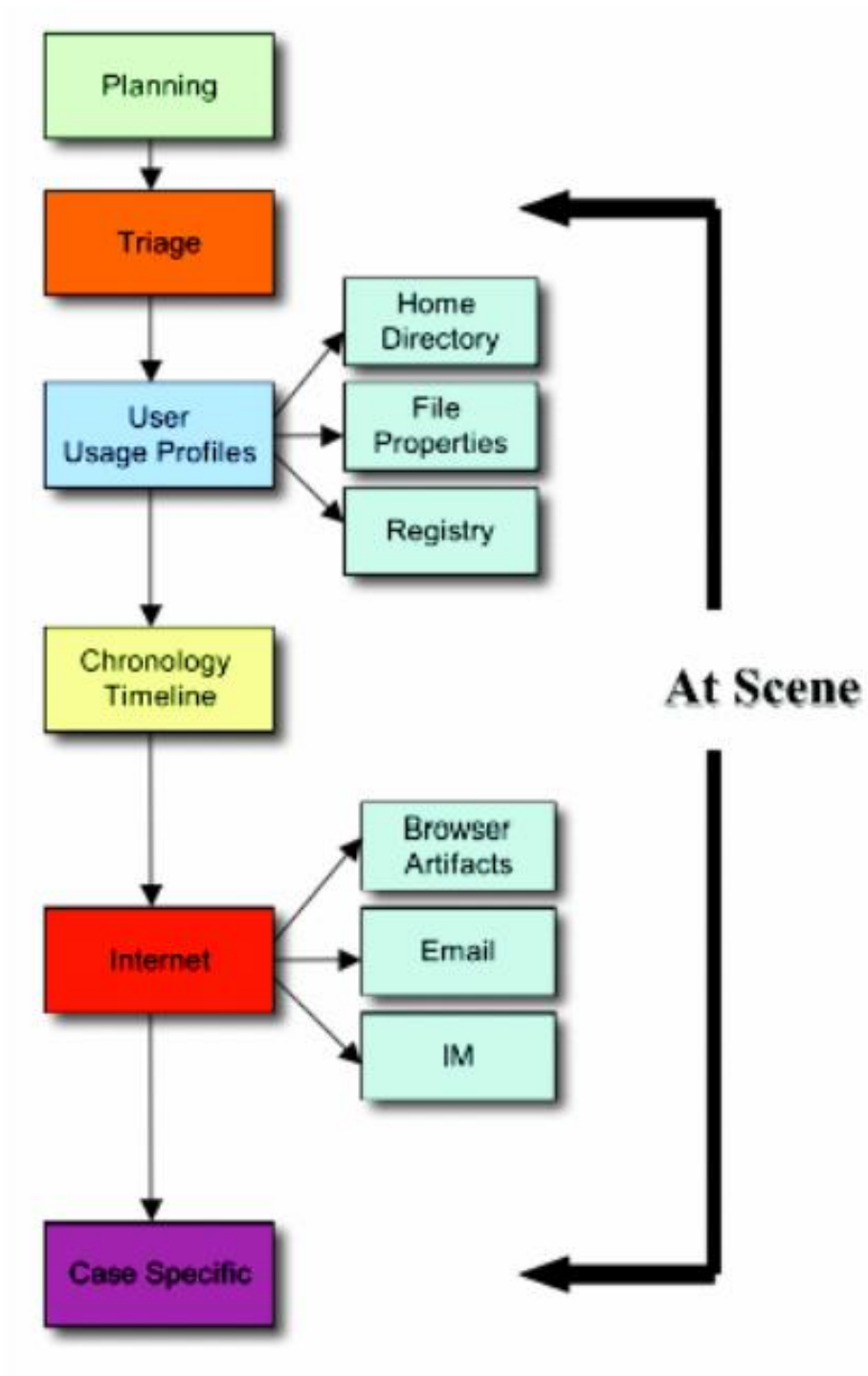


Figure 3 - CFFTPM Phases

2.1.1 Techniques

The techniques that are used in a computer forensics examination are the following:

1) Cross-drive analysis: Correlation of data being on different hard drives. [35] [98]

2) Live analysis: Examination of computers by using the installed operating system. This is achieved with custom forensics or sysadmin tools to gather all the information needed. This is crucial when we face encrypting file systems. [21]

3) Deleted files: The recovery of deleted archives. [64] Operating systems don't erase the physical data written on the hard drives. So analysts can reconstruct the information thought to be deleted. [21]

4) Stochastic forensics: This technique exploits stochastic properties of the computer to search for activities that don't have digital artifacts. The main use is to distinguish data theft.

5) Steganography: A method to hide information is by using steganography. This is the technique of hiding information inside a digital image or picture. This is mainly being used by criminals to hide pornographic content. Examiners by comparing hash values can understand if the image is the original. [26]

2.1.2 Volatile data

During the forensic process, when we shut down a computer, the data that are stored in RAM are disappeared. This is why we have to save them first. [20] Between other practices "live analysis" includes recovering RAM information before seizing a computer. [21]

After the seizure of the computer, RAM retains some electric current and this can help the examiner by using cold boot attack to regain some information. RAM retains this electric current for more time in very low temperatures. For example storing the RAM below $-60\text{ }^{\circ}\text{C}$ helps to achieve a successful recovery. But this is quite difficult to achieve while being at the scene. [41]

In order to extract volatile data correct, we must be at the lab, so as to take notice of the chain of evidence and to make our work easier. Some additional techniques to trans-

for a live computer is the mouse jiggler and the uninterruptible power supply (UPS) that provides the power. [21]

An easy way to save the data stored in RAM is to copy these data to a disc. [36]

2.1.3 Analysis tools

There is a wide number of open source and commercial tools for computer forensics examination. The forensic investigation gathers information from the media and the Windows registry. In addition tries to crack passwords and searches for keywords that have a relation to the crime under investigation. Finally retrieves e-mail addresses and texts and also pictures and video. [27]

2.1.4 Certifications

To achieve a forensics certification there is the ISFCE Certified Computer Examiner and the IACRB Certified Computer Forensics Examiner.

IACIS, the International Association of Computer Investigative Specialists, provides the Certified Computer Forensic Examiner, CFCE, course.

Nowadays most forensic software companies provide certifications on the programs they offer. An example is the Guardiancesoftware offering “EnCE” certification and AcCessdata offering “ACE” certification. [21]

2.2 Digital triage in forensics investigation

When an incident occurs, digital forensics processes are used to examine the incident, to collect and examine the digital evidence so as to evaluate the incident, identify a perpetrator and prove if a cyber-crime has been committed. [77] An incident can be caused by a human penetrator or by something irrelevant. Traces could be left any-

where. Here come the digital forensics investigation to search the digital crime scene and retrieve the aforementioned traces and evidence. [77]

Organizations nowadays have a specific IT security strategy to cover all the IT security and the other activities taking place in the organization. Until recently only law enforcement agencies were using computer forensics. But now many organizations use computer forensics to be protect by or examine cases for fraud, money laundering, pornography and harassment. [40]

A model depicting the forensic lifecycle could be the following:

- Identification.
- Authorization.
- Preparation.
- Securing and Evaluating the Scene.
- Documenting the Scene.
- Evidence Collection.
- Packaging, Transportation and Storage.
- Initial Inspection.
- Forensic Imaging and Copying.
- Forensic Examination and Analysis.
- Presentation and Report. [48]

In figure 4 we depict the IT Security fundamentals and in figure 5 we depict the Digital Forensics Investigation Fundamentals.

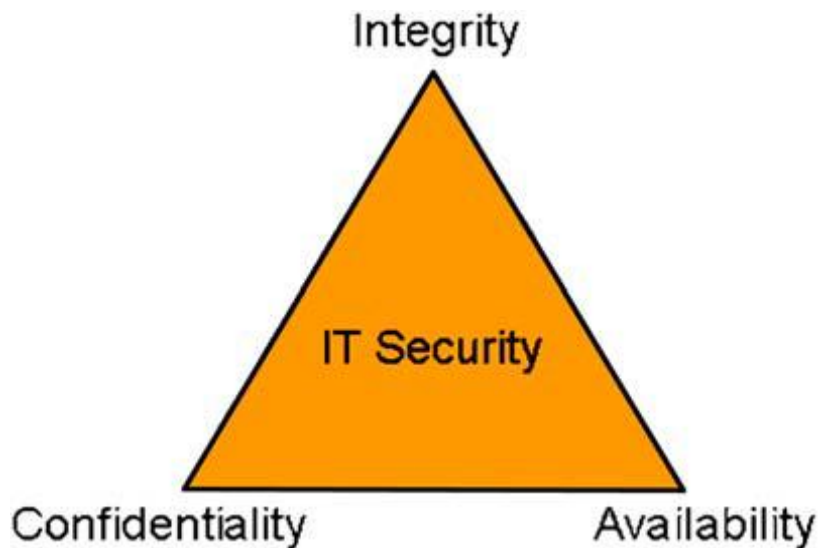


Fig. 4 – IT Security fundamentals. [73]

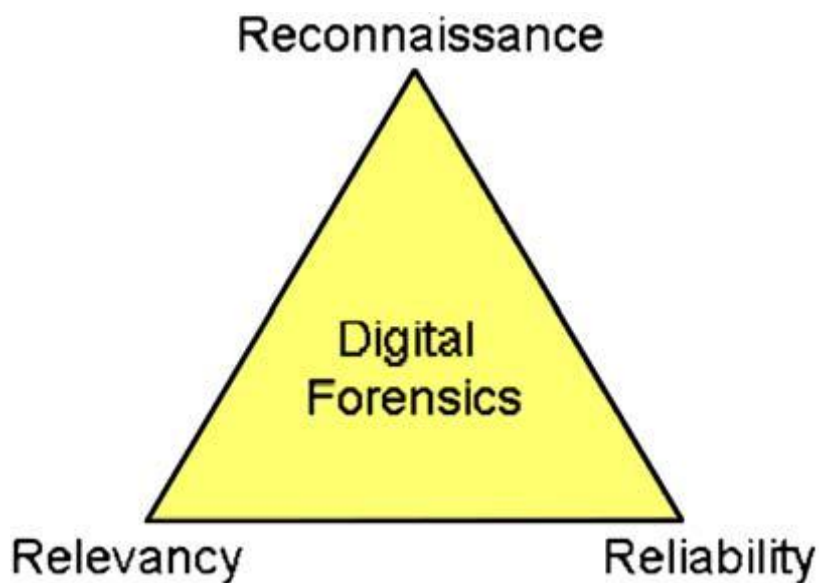


Fig. 5 – Digital Forensics Investigation Fundamentals. [73]

Digital Triage Forensics includes processes for examination of traditional crime scenes, digital crime scenes and battlefield crime scenes. [63] Digital triage collects digital evidence and processes in the first steps of investigation in order to identify what is useful and what not while being at the crime scene so as to lead the research to other possible and hidden evidence. All those evidence are meant to be used in a court of law so it is necessary to find them before they are lost or altered. So the triage tools, pro-

cesses and scripts used must obey to all forensic principles to ensure the admissibility of the collected information and evidence. [77] The Association of Chief Police Officers, ACPO, has published a guide with the basic forensic principles under the name of Good Practice Guide for Computer Based Electronic Evidence. [1]

Since criminals use information and technologies from all around the world the examiners cannot work in the opposite way. The examination team must have collaboration of numerous and different members. [6] But this makes investigation even more complex and challenging. [8] These different nationalities, cosmopolitan backgrounds, socio cultural backgrounds, legal frameworks multiply the difficulties and makes the decision-making processes highly demanding. [80]

Many organizations don't take seriously the digital forensics [81]. But they have to understand that linking an attacker to a crime isn't something easy to achieve. Organizations must be prepared for digital forensic investigations and be sure that they have taken all the measures to be ready for such examinations. [38]

Internet has developed a great need for digital investigations. Computers are now used to commit company policy violations, e-mail harassment, murder, leaks of proprietary information, embezzlement, and even terrorism. [57]

In computer forensics most professionals have been self-taught. Computer forensics investigators examine mainly hard drives, CDs, DVDs, Flash memory devices, floppies, and tapes but they must also have deep knowledge of all the computer systems. [14]

ACPO Good Practice Guide has four principles. The number is small so as to be able every investigator to understand and remember them. The Principle 1 is: no change of data must take place. [1] The Principle 2 is: the examiner to be really competent and explain his actions. [1] The Principle 3 is: that an archive of all the actions made must be recorded. [1] The Principle 4 is; that the person in charge has all the responsibility. [1] The important aspect is to show to the court that the evidence collected from the digital crime scene is exactly the same as found from the first person reaching the crime scene. If there are alterations, augmentation or decrement to the digital evidence collected it must be proven to the court what caused the variation of the analyzed data. [1] The problem with the programs used nowadays is that they often alter the data and it is crucial for the investigator to find and monitor all the alterations. Objectivity, continuity and integrity when possessing the evidence is a high priority. The tools applied must collect evidence starting from the volatile to the less volatile. [7]

In the ForensicArtifacts.com database the investigator can find potential forensic artifacts so he knows what to look for. [28] There is also the SANS resources such as the Sans Digital Forensics and Incident Response Poster-2012 or Sans forensic cheat sheets that include such information. [47] For example if we have a child pornography crime the browsing history, favorites and browsing information is the place to look for. It is important for the investigator to put some priorities before he starts collecting the data. Proper prioritization can save time, help him find evidence before they are lost, avoid mistakes and make a concrete and well monitored investigation. Data with small life time like processes, routing tables and temporary files should be a priority. [74]

In incident response there is a choice to be made. Whether to perform a complete memory acquisition or a live response. Complete memory acquisition is the analysis of the computer's physical memory with a number of tools. Some memory imaging tools are Crash Dumps, LiveKd Dumps, Hibernation Files, Firewire and Virtual Machine Imaging. [96]

Live incident response collects evidence from a computer so as to prove if an incident took place. Live incident response includes the gathering of volatile and nonvolatile data. Volatile data is data that would be lost if we cut the power off from the computer. Nonvolatile data is don't get easily lost and include useful information for digital forensic collection such as system event logs, user logons, and patch levels, among many others. Live incident response further includes the collection of information such as current network connections, running processes, and information about open files. To collect live incident response data we run commands that produce data that are send to a different storage device than the console. [72]

Memory acquisition is slow. [1] The modern hardware has big memory that includes data of past and completed processes. These data cannot be collected by the live response tools. [4]

The first responder will find live response useful if he is well prepared for the case under examination. [93] For best results both practices need to be applied.

From all the above we understand that a triage tool must comply with a number of requirements, have performance, low complexity but great adaptability.

In the following chapters we analyze, evaluate and compare four open-source triage tools used in digital forensics and incident response, assess their behavior and conclude

telling if they fulfill the purpose of the forensics analyst and remark any needed enhancements.

2.3 Testing triage tools

In the next paragraphs we test four open-source triage tools, to evaluate their behavior, their validity and conclude how useful they are for the first responder.

The four triage/incident response tools that we are going to analyze are Bulk Extractor [12], TriageIR [90], TR3Secure [89] and Kludge [46]. We tested their abilities in Microsoft Windows operating systems because based on statistics they are the most widespread operating systems for both examiners, users and criminals. [58].

To help us test TriageIR, TR3Secure and Kludge we used Windows 7 with VMware Player 8 installed including 8 different Windows OS systems. [77] Also Sandboxie 3.74, was installed. [15] [76]

We copied TriageIR v.79, Kludge-3.20110223 and TR3Secure on “E: disk” so as to use and simulate this partition as an external USB drive, as a forensic examiner would do. [77]

The tools were executed with all their options enabled in a sandboxed environment and normal environment. [77] The sandboxed environment was necessary to help us see which files are created or altered in the users system. [77]

2.4 Bulk Extractor

Bulk Extractor [12] is written in C++. It scans a file, a disk image, or a directory without parsing the file system. In addition Bulk Extractor makes histograms of features because features that are more common are more important. [10] This tool is useful for law enforcement, defense, intelligence, and cyber-investigation applications. [11]

Bulk Extractor is also useful for digital archivists. [13] Bulk Extractor has a GUI interface, the Bulk Extractor Viewer utility. Process of the results can be achieved by using the digital forensics tools in the Bit Curator environment. [13]

2.4.1 Advantages

Bulk Extractor advantages are the high speed and thoroughness. This happens because it ignores file system structure. Bulk Extractor has the ability to examine different parts of the hard disk in parallel. Bulk Extractor is thorough because it automatically detects, decompresses, and processes compressed data that have been compressed by numerous algorithms. [11] Another advantage is that Bulk Extractor can process any digital media. It can process hard drives, SSDs, optical media, camera cards, cell phones, network packet dumps, and other kinds of digital information. [11] Also it can recover more high-value forensic information than other tools. [31] Bulk Extractor is easy to use and this is why it has been used by numerous law enforcement organizations, and its acceptance spreads. [30]

2.4.2 Drawbacks

When using Bulk Extractor every media that will be examined is connected to a write-blocker of an investigator workstation and then it is processed with the Bulk Extractor tool. To start all this examination there is some time needed. Around 5 min per media. The Bulk Extractor is executed recursively without needing the attendance of the investigator, but the time needed to complete all its operations is big. [30] So Bulk Extractor works additionally to the traditional forensic procedures and its target is not to replace them. [30]

2.5 TriageIR v.0.79

When we read the manual of the TriageIR tool we see that it needs some tools added in a folder named tools which is located in the program's folder. The tools needed are: a) DumpItmemory utility [25], b) Sysinternals Suite [87], c) RegRipper [70], d) MD5deep and sha1deep [51], e) 7Zip Command Line [100]. TriageIR has 6 tabs – pages that contain all the options, see Fig. 6. In most cases the tool worked fine. [77]

The modifications that the tool makes to the computer under examination are justifiable and thus can be explained which helps to be defended in court. [77]

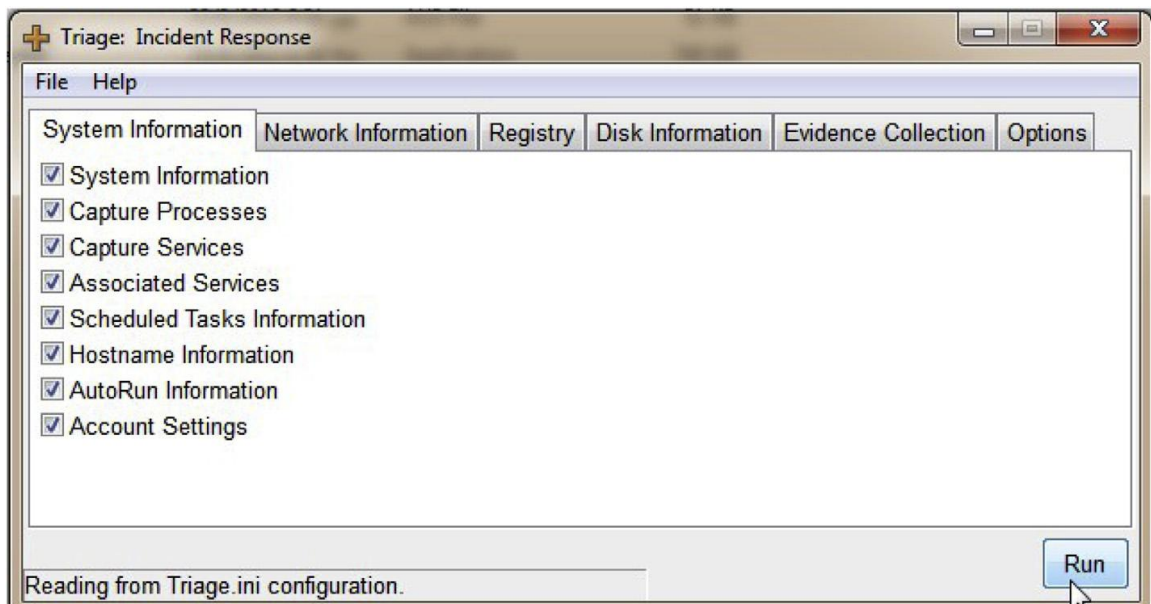


Fig. 6. TriageIR v.79 GUI.

2.5.1 Advantages

TriageIR can collect information about the startup process of the computer that is useful for malware examination. [78] Also TriageIR creates MD5 and SHA-1 hashes of evidence files. [77] This is crucial because it can prove the integrity of the evidence information.

2.5.2 Drawbacks

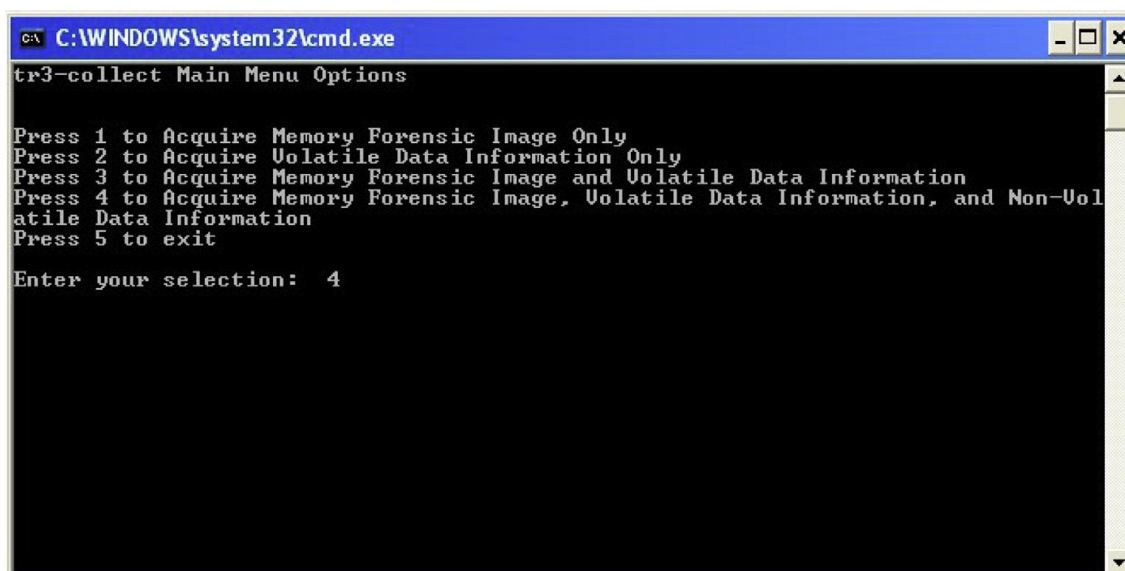
TriageIR during execution produced some errors that are due to programming faults or incompatibility of the utilities with the operating systems. TriageIR does not gather any Netbios evidence and has a problem in collecting all the event log information. [77] TriageIR has also problem collecting the security registry hive, the hard disk's directory structure and other information from the examined computer.

2.6 TR3Secure

TR3Secure data collection script uses Windows tools and tools that must be downloaded from the internet. Furthermore a .txt file with commands must be inserted in the tools folder of the program.

We chose option 4 from the tool's menu, see Fig. 7, so as to exploit all its abilities. [77] The tool did what expected in all operating systems. [77]

Like TriageIR the changes made are justifiable, they can be explained and so to be defended in the court. [77]



```
C:\WINDOWS\system32\cmd.exe
tr3-collect Main Menu Options

Press 1 to Acquire Memory Forensic Image Only
Press 2 to Acquire Volatile Data Information Only
Press 3 to Acquire Memory Forensic Image and Volatile Data Information
Press 4 to Acquire Memory Forensic Image, Volatile Data Information, and Non-Volatile Data Information
Press 5 to exit

Enter your selection: 4
```

Fig. 7. TR3Secure Main Menu.

2.6.1 Advantages

TR3Secure from a forensics view has all the functions needed including the choice to insert the case name, the examiners information, the path for storage and the execution date and time. Furthermore it records all the triage process so as to help the investigator find any errors produced.

2.6.2 Drawbacks

When we run the tool many errors occurred. The most important is that it fails on 64-bit operating systems. [77] In Windows 7 64-bit was not able to find the tools folder. [77] Also when it was executed in operating systems that used different codepage the results created needed a specific viewer to be read. [77]

2.7 Kludge 3.20110223

Kludge in the startup screen provide us 3 options (figure 8). The first is to perform simple analysis. The second is to perform a detailed analysis including timeline and registry analysis. The third is to perform the above and memory and process captures and file hashes.

Kludge was originally developed to be executed remotely by a network. This would be achieved by exploiting the administrative shares of the computer under examination. So Kludge copies all the information needed in the target computer and then it executes them. This is not a very good forensics practice because the changes to the target computer are massive. Also the administrative shares must be enabled. [77] Another option is to execute Kludge from a USB drive and save all the reports in the USB drive. [77]

When we run the Kludge by using the network the alterations to the target computer are explainable but not justifiable and so they cannot be accepted in a court. [77] But the new altered version of Kludge that was executed from a USB drive don't have this major disadvantage.

```
C:\Windows\system32\cmd.exe [#]
Running Kludge Analysis version 3.1
Please enter the following information ---
-- An Option Level (1-Simple Analysis, 2-Detailed including Timeline and Registry Analysis, 3-Includes Memory and Process captures and file Hashes)
-- An associated Ticket Number. If there is no ticket number associated, please enter 0 or none
-- The Analyst's name (e.g. john)
-- The Targeted Remote Machine Name (files will be copied to c:\Windows\Temp\analysis)
-- Your Admin Account Username (You will be prompted for your password twice)
-- A Local folder name on this machine where you want the final report to be copied to. Quote directories with spaces (e.g. "c:\Documents and Settings" or simply put a period for current directory)
-- Enter 'yes' or 'no' for GPG encryption. If 'yes', you will need your GPG Public key in a file called "pubkey.txt". The script can decrypt the Report if the Private Key is already installed on the current machine or exported to an ascii text file called "privkey.txt".
-- Enter 'yes' to have the script query a remote text file for previous incidents.
-- If Querying a share enter the share's path (e.g. \\192.168.1.2\Reports\incidents.csv)

Enter an Option Level (e.g. 2): 3
Enter the Remote Machine Name:
```

Fig. 8. Kludge script execution.

2.7.1 Advantages

Kludge is able to collect useful information that TR3Secure and TriageIR cannot. Something important for the investigator is that it collects internet history from Mozilla Firefox and Internet Explorer browsers. This is helpful in cases like grooming, bullying, spam, and other. [77] In addition it collects antivirus logs, firewall state, process dumps and memory information of running processes. [77] By creating timelines of running processes as being executed adheres to ACPO principles. [79] Furthermore the report is concentrated in a well organized html file from where the examiner can have an overview of the collected evidence. And in his way simplifying and speeding up the triage work.

2.7.2 Drawbacks

Kludge when executed produced out-of-the-box errors, programming faults and incompatibility of the utilities. It also collects only specific antivirus logs. [77] Symantec and McAfee antivirus programs share only 15% of the market. [61] It also does not collect event logs in Windows Vista, 7 and 8 operating systems. [77] Since Kludge

doesn't record an analytic log of all the processes executed it is hard to make checks and comparisons. [77] Like TriageIR does not remove added registry keys.

2.8 Synopsis

After running and testing the four tools and having found their advantages and drawbacks we come up to a synopsis about their performance.

2.8.1 Bulk Extractor

Bulk Extractor collects credit card numbers, email addresses, URLs, and other kind of information from any digital media. The tool can be executed on different data formats and collects much evidence concerning internet. Also it can detect and decompress data that have been compressed by using many different algorithms. [86] Bulk Extractor also uses GNU flex. [88] When all the evidence have been collected Bulk Extractor creates a histogram of useful features. Stop lists can also be applied. [30] In figure 9 there is an overview of the tools architecture. A graphical user interface helps the investigator to view the reports created. [30]

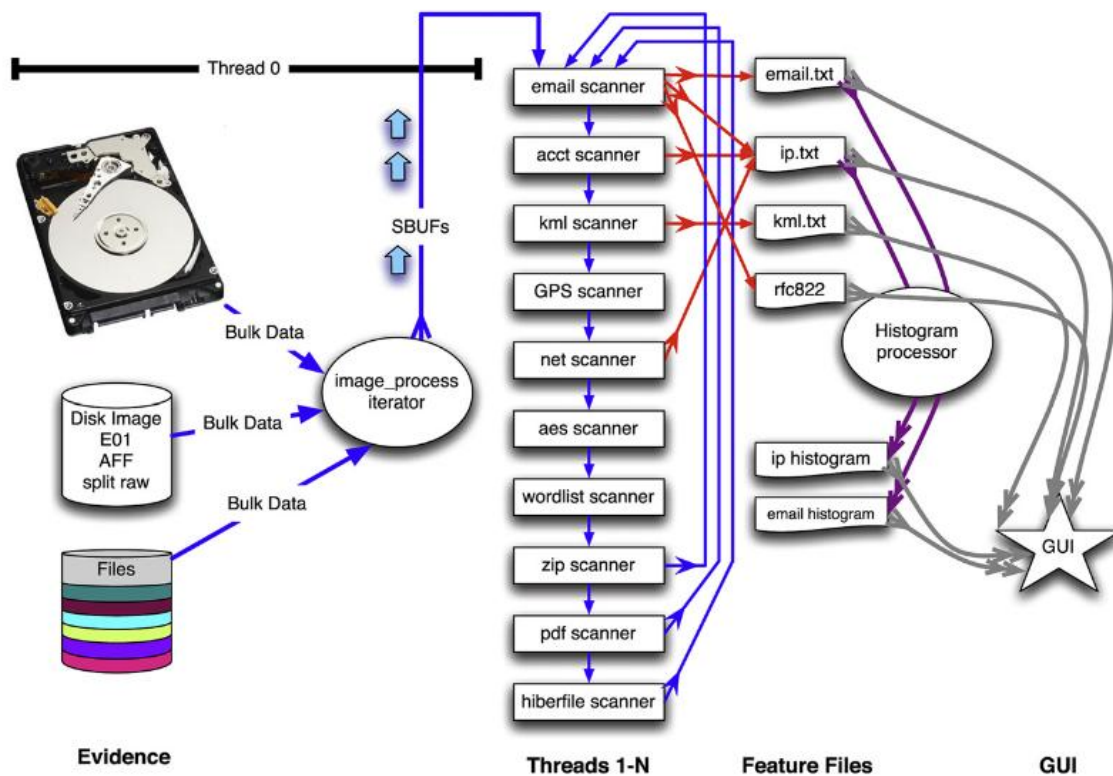


Fig. 9 Diagram showing overview of the Bulk Extractor architecture. [30]

Bulk Extractor constructs a report that have: [11]

- Credit card information.
- IP addresses found.
- Ethernet MAC addresses found.
- EXIFs from JPEGs and video segments.
- Credit card numbers.
- US and international phone numbers.
- URLs from browsers and emails.
- Email addresses.
- A histogram of terms used for internet searches.
- A list with words for password cracking.
- The wordlist in a form to be used by password-cracking programs.
- Internet domains found.
- Information about ZIP files.

- Results of specific search requests.

For all the above, two files are also created: [11]

- A stop list or a list of information not to be shown to the user.
- Histograms of features with appearance frequency.

Bulk Extractor additionally creates a report. An XML report with information concerning the source data, the compilation and execution of the tool, time used for processing and additionally includes all the evidence found. [11] Bulk Extractor is very helpful for triage. It aids to put priorities before the thorough examination begins. [30]

2.8.2 TriageIR, TR3Secure and Kludge

None of the triage tools TriageIR, TR3Secure and Kludge inform the user that has to run the tools as administrator in Windows Vista, 7 and 8 operating systems in order to work properly. [77]

In Table 1, Table 2 and Table 3 we see a picture of the incident data that the above three tools were able to collect when they were executed during the triage process. [77] The column headers of the table indicate the order of volatility scale, and the row headers show the tools tested.

Table 1 Collected forensic artifacts [77]

Tested tools – collected forensic artifacts vs. order of volatility scale.

Order of volatility (from more volatile to less volatile) ↓		TriageR 0.79	TR3Secure	Kludge 3.2
Registers and Cache	No data collected	X	X	X
Routing table, arp cache, process table, kernel statistics, memory	Network-related data → ARP cache	X	X	X
	Network-related data → Routing table		X	X
	Network-related data → DNS cache and resolution		X	X
	Network-related data → DNS Information	X		X
	Network-related data → A records			X
	Network-related data → Host file			X
	Network-related data → Netbios routing table	X		X
	Network-related data → Netbios information (sessions, connections, file transfer over netbios)	X	X	X
	Network-related data → Port to process mapping		X	
	Network-related data → TCP/UDP active connections	X	X	X
	Network-related data → TTL			X
	Network-related data → Firewall (info, status)			X
	Process data → Process File Handles	X	X	X
	Process data → Running Processes-DLLs	X	X	X
	Process data → Services			X
	Process data → Process to exe mapping		X	
	Process data → Process to user mapping		X	
	Process data → Child processes		X	
	Process data → Process dependencies		X	
	Process data → Process dumps			X
	Process data → Process memory			X
	User's activity → Active logon sessions		X	
	User's activity → Logged on users	X	X	X
	User's activity → Recent files	X		
	User's activity → Internet browsers history			X
	User's activity → Jump lists Files	X		
	User's activity → Clipboard-contents		X	X
	Registry hives → Sam	X		X
	Registry hives → Security	X		X
	Registry hives → System	X		X
	Registry hives → Software	X		X
	Registry hives → HKCU	X		X
	Registry hives → NTUSER.dat	X		X
	Registry hives → USRCLASS.dat	X		X
	Various timelines → IE Timeline			X
	Various timelines → FF Timeline			X
	Various timelines → Hard disk timeline			X
	Various timelines → Prefetch info			X
	Various timelines → Recycle Bin timeline and contents			X
	Memory image			X
	System configuration → VSS service status			X
	Prefetch files	X	X	
	Ntfs data streams		X	X
	Unsigned-executables → Uptime		X	

Table 2 Collected forensic artifacts [77]

Temporary file systems	System event logs → evt files	X		X
	System event logs → evtX files	X		
	Processed event logs → System	X		X
	Processed event logs → Security	X		X
	Processed event logs → Application event logs	X		X
	Antivirus logs			X
	No data collected		X	
Disk Remote logging and monitoring data that is relevant to the system in question	Not applicable	X	X	X
	Network-related data → Open shared files	X		
	User's activity → Remotely logged on users		X	
	User's activity → Remote users IP-addresses		X	
	No data collected			X
Physical configuration, network topology	Network-related data → Network configuration	X	X	
	Network-related data → Network Adapter info			X
	Network-related data → Routing table	X		X
	Network-related data → Host File	X		X
	Network-related data → Enabled network protocols		X	
	Network-related data → Promiscuous adapters		X	
	User's activity → Logged on users	X		
	System configuration → User accounts policy	X		
	System configuration → User groups		X	
	System configuration → Startup information	X	X	
	System configuration → Directory structure	X		

Table 3 Collected forensic artifacts [77]

Order of volatility (from more volatile to less volatile) ↓	TriageIR 0.79	TR3Secure	Kludge 3.2
System configuration → Mounted disks information	X		
System configuration → Hostname	X		
System configuration → Local shares	X		X
System configuration → Schedule tasks	X		X
System configuration → Kernel build	X		
System configuration → Register organization and owner	X		
System configuration → OS-version		X	
System configuration → Group policy listing and RSOP		X	
System configuration → Installed software		X	
System configuration → Security settings		X	
System configuration → Hardware devices		X	
System configuration → Number of processors and their type	X		
System configuration → Amount of physical memory	X		
System configuration → System's install date	X		
System configuration → System variables	X		
System configuration → System configuration			X
System configuration → Firewall configuration	X		
System configuration → Services	X		
System configuration → Type of installation	X		
System configuration → NTFS partition info	X		
Certain applications → Version and Signing info for Acrobat			X
Certain applications → Acrobat Reader			X
Certain applications → Flash			X
Certain applications → Java			X
Certain applications → Firefox			X
Certain folders structure → Program Files			X
Certain folders structure → Documents and Settings			X
Certain folders structure → Windows			X
Unsigned-Executables → Computer name			X
Unsigned-Executables → Autoruns			X
Unsigned-Executables → Startup apps			X
Unsigned-Executables → BHO's			X
Unsigned-Executables → Hotfixes and service packs			X
Unsigned-Executables → Environment Variables			X
Unsigned-Executables → Uptime			X
Unsigned-Executables → Operating System Information			X
Unsigned-Executables → Drive Information			X
Unsigned-Executables → Partition info			X
Unsigned-Executables → Users			X
Unsigned-Executables → USB device history			X
Registry files			X
Archival media	X	X	X
Not applicable			

The contents of a hard disc usually are examined in the lab so there is no use to analyze them in the triage process. [77] In Table 4 we see how effective each tool is in different operating systems. [77]

Table 4 Tool effectiveness [77]

Tool effectiveness.								
Tool	Win XP SP3 32 bit	Win XP SP2 64 bit	Win 7 32 bit	Win 7 SP1 32 bit	Win 7 64 bit	Win 7 SP1 64 bit	Win 8 32 bit	Win 8 64 bit
TriageIR 0.79	Medium effective	Medium effective	Medium effective	Medium effective	Medium effective	Medium effective	Medium effective	Medium effective
TR3Secure	Medium effective	Ineffective	Medium effective	Medium effective	Ineffective	Ineffective	Medium effective	Ineffective
Kludge 3.20110223	Medium effective	Less effective	Less effective	Less effective	Less effective	Less effective	Less effective	Less effective

In Table 5 we see the alterations caused by TriageIR, TR3Secure and Kludge on the registry and file system of every operating system. These alterations were captured with the help of Buster Sandbox Analyzer 1.87 and Sandboxie. [77] The version of the Kludge tool, that was changed to be used by a USB drive, was the most forensically correct of all the above triage tools. [77]

Table 5 Modifications produced [77]

Summary of file system and registry modifications.

OS	Tool		
	TriageIR	TR3Secure	Kludge (modified version)
Win XP SP3	FM ^a : 39 (mainly prefetch and /system32/CatRoot) RC: 33	FM: 13 (one in /system32/) RC: 21	FM: 0 RC: 4
Win 7 64 b	FM: 84 (mainly prefetch and logfiles) RC: 379	FM: 4 (mainly logfiles) RC: 71	FM: 1 (temp appdata) RC: 6
Win 7	FM: 39 (prefetch and user appdata) RC: 134	FM: 26 (mostly in prefetch, one in /system32/) RC: 131	FM: 1 (temp appdata) RC: 14
Win 8 64 b	FM: 138 (prefetch and user appdata) RC: 354	FM: 45 (mostly in /INF folder) RC: 73	FM: 0 RC: 6
Win 8	FM: 29 (prefetch and user appdata) RC: 131	FM: 19 (2 in /system32/) RC: 127	FM: 1 (temp appdata) RC: 8

^a FM: File creations/modifications – RC: Registry changes.

Summing up all the above we see that it is obvious that a triage tool have to be balanced in numerous requirements, including performance, complexity and adaptability so as to be effective and successful during the triage evaluation.

2.9 Suggestions

The four triage tools analyzed before could be adjustable in two ways: First before the execution to avoid as many errors as possible and second during the acquisition to achieve the maximum effectiveness. [77]

If we disable Prefetch on Windows the modifications created will be even less. [77] Furthermore the tools should record all the actions executed including errors and problems. [77] In addition it is advisable the tools to log and restore all registry modifications. Finally it is important that the tools enhance the collection of all history and internet activity information of all widely used browsers.

2.9.1 Bulk Extractor

Bulk Extractor is a command line tool. [31] An advanced GUI could be used for easier and faster use, even from people with little experience and a GUI for the results found. With the current version it is possible to run multiple copies of Bulk Extractor on different machines but there is no way to easily recombine the results. [31] A good direction would be to solve this problem and improve the combined functionality. Also we can extend Bulk Extractor with numerous plug-ins.

2.9.2 TriagelR 0.79

This tool must overcome problems that are encountered when it is used in different operating systems. It seems to work properly only in Windows XP and faces problems in the other releases. [97]

2.9.3 TR3Secure

This tool has compatibility problems with Windows 64bit operating systems so modifications must be made to overcome this problem. In addition it must be enhanced with capabilities such as collection of scheduled tasks, registry files, installed printers, peripherals, user logons and internet activity information. [77]

2.9.4 Kludge 3.20110223

This tool was created having in mind specific antivirus and security products. But it must broaden its use and target machines. Some other enhancements are to run from a USB stick or an external drive and save the reports there. Additionally some functions have to be altered in order to be able to execute in Windows 7 and 8 operating systems. [77]

2.10 Future work on the four tools

As it is understood from the above analysis there is no tool to cover all the purposes, be very effective and be the best for triage process. This result was obvious because there is high variety and complexity on modern media and systems. The complexity and the variety in the needs, the software and processes are increased radically so the examiner must have a wide portfolio of different tools to manage and handle every different case. [77]

In order to help the examiner choose the portfolio of tools we can put some qualitative and quantitative metrics. [77] After the examination of the four triage tools, we could apply these three metrics:

- 1) Effectiveness.
- 2) (Un)reliability.
- 3) Invariability. [77]

The use and the connection between these metrics could be a good field for future research.

Also in the future we could try to assess the tools in terms of usability, usefulness, exploitable results, validity and evaluation of reports and case-by-case analysis. [77] A future target would be to create an advanced triage tool based on the four tools mentioned above that would combine all their advantages , functions and options and none of the drawbacks.

An enhancement for Bulk Extractor would be to develop an algorithm, for distinguishing compressed data from encrypted data [33], and additionally present the percentage of encrypted data on the digital media that is under examination. [30]

The reports of Bulk Extractor can be useful when they are compared and contrasted to other data of forensic practices, for example, cross-drive analysis information. [59]

3 Problem Definition

Computer abilities and capabilities have grown radically since 1960, and they are going to develop exponentially in the future. Storage capacity has grown beyond any expectation and there is no limit for the future. [62] FBI's statistics state that the data for analysis per case have multiplied 6.65 times, from 84 GB to 559 GB, in eight years, from 2003 to 2011. [69] The expanded computing capabilities makes processing and storage cheaper. This makes the forensic examination even harder with more compute resources needed. Although this problem is known, little has been done to confront it. The blame is on both users and developers. [75]

In the triage process first we have to put priorities before starting the collection of data. Great importance must be given to the data that live for a short time like routing tables, temporary files and processes. Forensic analysts have to have a wide number of tools because no tool can apply to all the different cases. [74] So the triage tool must be flexible be able to change its functionality depending on the evidence. [77] The triage tool must collect data in very short time but this is many times overlooked. This disadvantage derives from the fact that triage tools come from traditional forensic tools which were designed to conduct the analysis at a future time. [44]

The processing speed of current digital forensic tools is not a match for the average cases because users don't have specific performance requirements and developers haven't put performance and latency to a top-level objective. [75]

SPEKTOR triage tool includes some automation, but this is done in order to be used by people who have no specific technical skills. [82] This is in violation to ACPO's second principle and this is why it is considered to be a poor practice. [77]

In the next chapters we are going to analyze the most important problems of the four tools tested before.

3.1 Bulk Extractor

Bulk data analysis has the limitation that compressed data that are fragmented into many locations are difficult to recover. However, most interesting files in forensic manner are not fragmented. The only exception is log files. [34].

Unicode and non-Latin characters create problems for bulk data analysis. One other complication is that there are many types of localized strings that are possible to be found. [30]

3.2 TriagelR v.0.79

The tool failed only in Windows 8 OS 64 bit. [77] When the tool runs its utilities create and change many files. These alterations are undesirable. [77]

By analyzing the execution and results we see that the tool violates a number of forensic principles. First of all some utilities of the tool need a hard disk partition letter as a parameter to execute properly. [77] Also when executed the tool adds registry keys but does not undo these registry alterations. Furthermore it does not record all executed commands in the created incident log file. In this way the analyst doesn't know which commands executed correctly, which failed and why. [77] Traceability is hard to achieve because the tool uses a separate command shell for each utility called. This ends after the execution and this makes the examiner not to acknowledge the errors that have been created. [77] The tool should create hashes of the reports, so as to have a complete and correct chain of custody for the data that have been either gathered or produced by the tool. Finally, when the compression ability of the tool is used, some data are not gathered. [77]

3.3 TR3Secure

Like TriageIR, the utilities used by TR3Secure alter some Windows OS files. This happens also with some temp and recent activity files. [77] In all operating systems

TR3Secure loads drivers in certain folders, alters or adds registry keys, creates or modifies folders. Finally when some utilities fail to execute the crash reports are created in specific folders. [59] [77]

3.4 Kludge 3.20110223

Kludge in all operating systems applied adds and alters registry keys, creates or modifies files, try to install drivers and changes Prefetch and the users' temp files and recent activity. [77]

Kludge uses the Hobocopy utility [43] to copy files. This utility crashed in Windows 7 and 8 OS, causing the registry files and event logs not been collected. Moreover many utilities used by Kludge in Windows 7 and 8 OS, crashed. Finally some utilities didn't execute at all in the above operating systems. [77]

4 Contribution

This paper contributes in both the theory and practice of digital forensics. The purpose is to study and analyze, the way four free triage tools work and respond during the process of installation and execution. Also highlight the advantages and drawbacks and make suggestions for improvements and better performance. Finally develop a program to help the first responder make the digital triage fast and in an easy, concrete and meaningful way.

4.1 First thoughts

The first thought was to use one or more existing tools and change the presentation of the results that they provide, when they are used for digital triage. The purpose would be to improve the results by making them more helpful and beneficial for the investigator. The target would be to ease the forensics analyst and provide him an overview with a glimpse. Help him take the right decisions fast. Demonstrate and present the contents of the hard disc in a way that helps the investigator have a more concrete image of the contents of the disc. Provide visualization, analysis and evaluation with the use of diagrams, bars, pies, percentages, images and other presentation tools.

The way to do all the above was the following: Find as many sources as possible and bibliography. Read papers, articles, published material by accredited scholars and researchers. Search the internet. Search for digital triage tools and how the presentation can be improved. Find for digital triage in forensics investigation definition, categories and characteristics, scopes, goals, objectives and achievements. Find the existing tools and what things to consider when choosing the right tool. Find regulations globally and in Greece. Find what existing tools provide, what they don't offer and why this lack is important. Why the lack of a proper and advanced presentation is so important. Make a gap analysis. Search for drawbacks and possible contribution. Improve the visualization,

analysis and evaluation with the use of diagrams, bars, pies, percentages, images and other presentation tools.

The first thought was to help the child pornography examiner. After taking an image from the hard disk, the RAM and the volatile data, then provide him with the history from all the browsers that are installed on the computer. While he is at the scene it is crucial to recover everything that is related to the examined case. When a suspect searches for child pornography in the internet or stores digital files, containing this child pornography material, many traces are left behind. The most possible scenario is that the suspect won't have great programming skills and knowledge to create a script to delete, every time, the browsing history and the evidence from all the browsers in his computer. So during the triage process, the program that I would create, would be able to collect all the evidence from the web browsers, specify attributes that stand out and present them in a useful and easy to understand way, to help the examiner.

One first approach was to use a widely available open source tool, Kludge, that collects digital evidence that the other two tools TR3Secure and TriageIR do not. It can collect internet browsers history from Mozilla Firefox and Internet Explorer. Some modifications could be made to be more focused on the latest versions of the browsers. And this because the browsers pop up often messages, to the users, to download the latest versions available. It is rare for a user to decline these prompts all the time and use a very old version of the browsers. Knowing the browsers history is crucial when the examiner searches for grooming, bullying, spam, and so on. But also Kludge can collect antivirus logs and reports of the firewall state. In addition it can collect process dumps and process-related memory for each running process. In the end Kludge produces an html file with a well organized menu which help the examiner navigate the collected digital evidence. This can simplify the work of the examiner and speed up the triage process. To enhance the usability and presentation of the tool one thought was to add statistics, diagrams, bars, pies and percentages to the reports, which could be navigated through the html file. Kludge produces .txt files for the reports. So the purpose of the new program would be to read these .txt files and produce more complicated reports with all the statistics, diagrams, bars, pies and percentages mentioned above.

However there was a major turn over in the development of the program. After reading all the sources cited in the bibliography and many more, the conclusion was that there are some tools freeware and licensed that present their results using visualization,

analysis and evaluation including the use of diagrams, bars, pies, percentages, images and other presentation tools. Such a tool is Spektor. So the new thought was to develop a tool, that would be very simple and easy to use, for the first responder, to exploit it for digital triage in the crime scene but also helpful during the analysis at the lab.

4.2 The new target

After all the above the second thought was to make a program that could search the targeted computer and find the added files, altered files, infected files or even malware constructed by the suspect.

At start, the new target, was to make a program that would search the computer under investigation for the files we want to find. To have the ability to choose the type of files we want, the domain of the computer to search in, the file the results would be saved in. The results file would show the name and the path of the files found.

The next step was to find all the MD5 hashes of the files we wanted. This could help to find which files were modified (ex .exe files for windows) in comparison to the original ones. This could be done by producing the MD5 hashes [50] of the files found and compare them with the MD5 hashes of the original files (which are unaltered).

In the next stage the above program was necessary not to be hard coded (the parameters not to be fixed in the code) but to offer the examiner the ability to insert the desired values as parameters during the execution:

1. The path where the program to search in.
2. The type of files the program to search for.
3. The name of the file with the results.
4. The path of files to be compared.
5. Where the reports with the name of the files, the path found and the MD5 hash to be saved in.

During the development many features were added, the program was upgraded many times and became more complicated, until it reached its final form presented below. The final program has the following functionality:

The user by using the command prompt can choose: a) to see the help guide of the program b) to scan a specific domain, retrieve the files chosen and create their MD5 values and c) compare the created MD5 values of the files chosen with a file containing the “clean” MD5 values of unaltered files that he wants to compare.

1. The user first inserts the option “help”, “scan” or “compare”.
2. In case of “help” the programs functionality is shown.
3. In case of “scan” without putting any parameters the default values are used.
4. In case of “scan” with parameters the inserted values are used to set a) the path for the program to search b) the extension of the files being searched and c) the path where the results file to be stored.
5. In case of “compare” the parameters are necessary. The values inserted are used a) to set the path for the file containing the MD5 of the scanned files b) the path for the file with the clean MD5 values and c) the path where the results file to be stored.

4.3 The developing process

The first attempt was to use the programming language VBScript for developing the program. [91] [92] Also there were thoughts to use the programming language Python because it is widely used for applications concerning security and digital triage. [67] [68] The final decision though was to use the programming language PHP because I had better knowledge of this language, I had use it extensively before, I knew its capabilities and the belief that I could accomplish the program till the end was strong. [65] [66]

The editor I used to write the PHP scripts was the Sublime-text editor. [84] [85] A server was used, the Wamp Server, to run the program locally and see the results, because the program doesn't have a web interface. [94] [95] So in order to run the PHP program there is a need to install PHP and a server on the examined computer. But in order to adhere to the ACPO principles there was a need to minimize the changes to the computer under investigation. To bypass this problem Bamcompile was used to convert the PHP application to standalone .exe application. [5] By transforming the *.php program into a *.exe file we give the ability to the investigator to run the program from a USB stick using only the command prompt of the computer. No need for php dlls and

installations that alter the computer files system. Bamcompile is a freeware. To use Bamcompile you have to visit the site <http://www.bambalam.se/bamcompile/> and download the win 32 zip. You extract this zip file wherever you want, for example at the desktop. In this folder you insert the scanner.php file. You open a command line prompt and write the following:

“cd C:\Users\nikos\Desktop\bamcompile1.21”. After this you write in the cmd:

“bamcompile scanner.php” and in the same folder with the scanner.php the file scanner.exe is created and is therefore ready for use.

4.4 The scanner program - How it works

The program's name is scanner.php and the standalone .exe application is scanner.exe. If the examiner wants to run the scanner.php program, first he must install a server to run it locally at the computer, like the Wamp Server, and put the scanner.php program into the www folder of the directory of the server. Then in order to run it (this is for use with Wamp Server) he must open a command line prompt and open the folder of the server where php is located, for example: cd C:\wamp\bin\php\php5.4.16\. Then depending on what he intends to achieve he can type:

- 1) .\php.exe C:\WAMP\www\scanner.php
- 2) .\php.exe C:\WAMP\www\scanner.php help
- 3) .\php.exe C:\WAMP\www\scanner.php scan
- 4) .\php.exe C:\WAMP\www\scanner.php scan --path="C:\\" --extension="exe" --log_file="C:\WindowsMD5\Cexe.txt"
- 5) .\php.exe C:\WAMP\www\scanner.php compare --log_file="C:\WindowsMD5\log.txt" --MD5_values_file="C:\WindowsMD5\clean.txt" --results_file="C:\WindowsMD5\results.txt"

The MD5_values_file must be a .txt file, with the created MD5 values from the scan option, from files that are clean and secure. The folder where the .txt file with the results will be stored must be created before the program is used.

If the analyst wants to run the program on the computer under examination by using a USB stick then he must place in the USB stick the scanner.exe program. Then in order

to run it he must open a command line prompt and type the letter of the flash drive (e.x. F:) and this will move the working directory to the root of F. Then he must open the folder where the scanner.exe program is located in. For example: cd folder\. If the scanner.exe program is located just in the F and not in a folder the following commands will do. Then depending on what he intends to achieve he can type:

- 1) scanner.exe
- 2) scanner.exe help
- 3) scanner.exe scan
- 4) scanner.exe scan --path="C:\\\" --extension="exe" --log_file="F:\Cexe.txt"
- 5) scanner.exe compare --log_file="F:\log.txt" --MD5_values_file="F:\clean.txt" --results_file="F:\results.txt"

4.4.1 Help mode

If we press as mentioned above:

- 1) .\php.exe C:\WAMP\www\scanner.php
- 2) .\php.exe C:\WAMP\www\scanner.php help
- 3) scanner.exe
- 4) scanner.exe help

Then we see the help screen of the program that shows us how it works, some instructions and what parameters to insert in each case. The help screen is appeared in the case the user doesn't know how the program works and runs the program with no parameters at all.

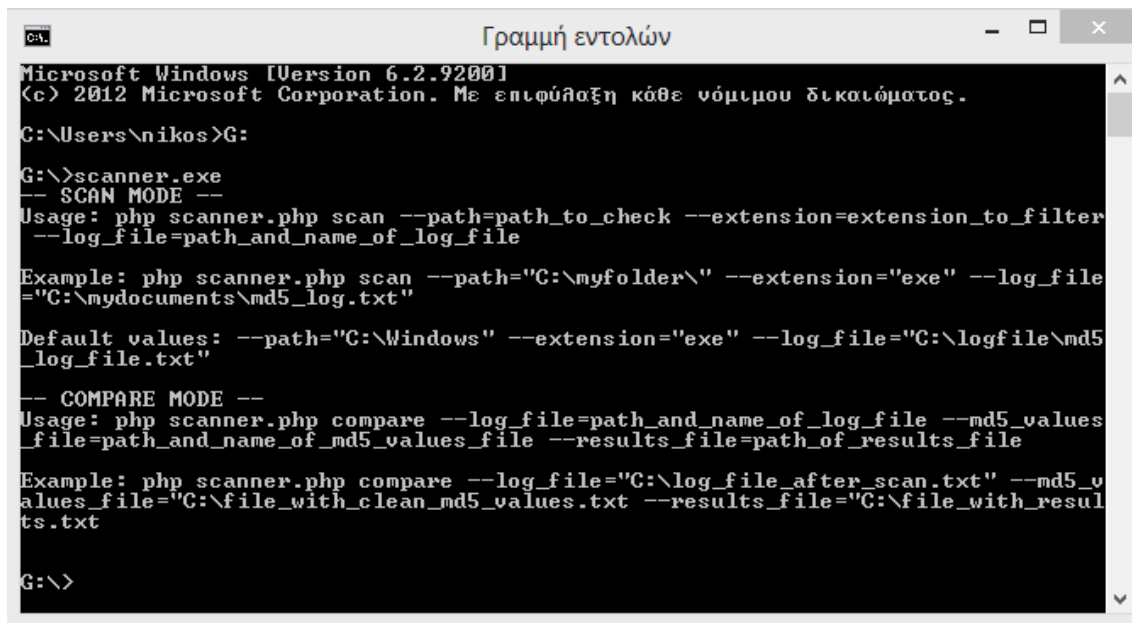
Figures 10, 11, 12 and 13 visualize this functionality of the program.

```
Γραμμή εντολών
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.
C:\Users\nikos>cd C:\wamp\bin\php\php5.4.16\
C:\wamp\bin\php\php5.4.16>.\php.exe C:\WAMP\www\scanner.php
-- SCAN MODE --
Usage: php scanner.php scan --path=path_to_check --extension=extension_to_filter
--log_file=path_and_name_of_log_file
Example: php scanner.php scan --path="C:\myfolder\" --extension="exe" --log_file
="C:\mydocuments\md5_log.txt"
Default values: --path="C:\Windows" --extension="exe" --log_file="C:\logfile\md5
_log_file.txt"
-- COMPARE MODE --
Usage: php scanner.php compare --log_file=path_and_name_of_log_file --md5_values
_file=path_and_name_of_md5_values_file --results_file=path_of_results_file
Example: php scanner.php compare --log_file="C:\log_file_after_scan.txt" --md5_v
alues_file="C:\file_with_clean_md5_values.txt --results_file="C:\file_with_resul
ts.txt
C:\wamp\bin\php\php5.4.16>
```

Fig. 10 Running the scanner.php program with no parameters.

```
Γραμμή εντολών
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.
C:\Users\nikos>cd C:\wamp\bin\php\php5.4.16\
C:\wamp\bin\php\php5.4.16>.\php.exe C:\WAMP\www\scanner.php help
-- SCAN MODE --
Usage: php scanner.php scan --path=path_to_check --extension=extension_to_filter
--log_file=path_and_name_of_log_file
Example: php scanner.php scan --path="C:\myfolder\" --extension="exe" --log_file
="C:\mydocuments\md5_log.txt"
Default values: --path="C:\Windows" --extension="exe" --log_file="C:\logfile\md5
_log_file.txt"
-- COMPARE MODE --
Usage: php scanner.php compare --log_file=path_and_name_of_log_file --md5_values
_file=path_and_name_of_md5_values_file --results_file=path_of_results_file
Example: php scanner.php compare --log_file="C:\log_file_after_scan.txt" --md5_v
alues_file="C:\file_with_clean_md5_values.txt --results_file="C:\file_with_resul
ts.txt
C:\wamp\bin\php\php5.4.16>
```

Fig. 11 Running the scanner.php program in help mode.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\nikos>G:

G:\>scanner.exe
-- SCAN MODE --
Usage: php scanner.php scan --path=path_to_check --extension=extension_to_filter
--log_file=path_and_name_of_log_file

Example: php scanner.php scan --path="C:\myfolder\" --extension="exe" --log_file
="C:\mydocuments\md5_log.txt"

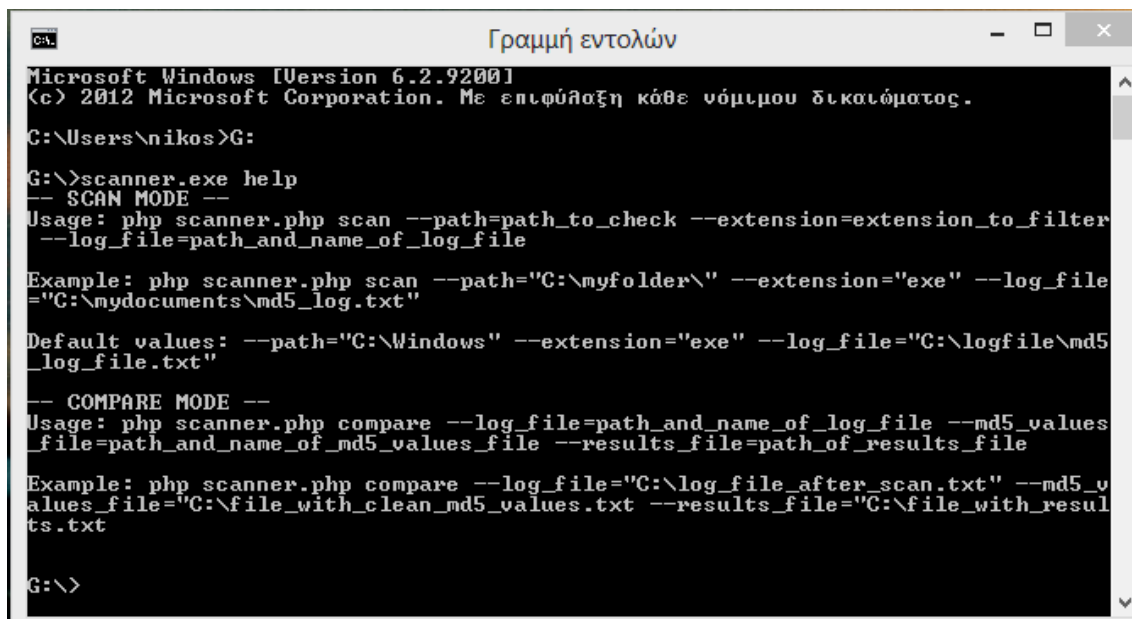
Default values: --path="C:\Windows" --extension="exe" --log_file="C:\logfile\md5
_log_file.txt"

-- COMPARE MODE --
Usage: php scanner.php compare --log_file=path_and_name_of_log_file --md5_values
_file=path_and_name_of_md5_values_file --results_file=path_of_results_file

Example: php scanner.php compare --log_file="C:\log_file_after_scan.txt" --md5_v
alues_file="C:\file_with_clean_md5_values.txt --results_file="C:\file_with_resul
ts.txt

G:\>
```

Fig. 12 Running the scanner.exe program from a USB drive with no parameters.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\nikos>G:

G:\>scanner.exe help
-- SCAN MODE --
Usage: php scanner.php scan --path=path_to_check --extension=extension_to_filter
--log_file=path_and_name_of_log_file

Example: php scanner.php scan --path="C:\myfolder\" --extension="exe" --log_file
="C:\mydocuments\md5_log.txt"

Default values: --path="C:\Windows" --extension="exe" --log_file="C:\logfile\md5
_log_file.txt"

-- COMPARE MODE --
Usage: php scanner.php compare --log_file=path_and_name_of_log_file --md5_values
_file=path_and_name_of_md5_values_file --results_file=path_of_results_file

Example: php scanner.php compare --log_file="C:\log_file_after_scan.txt" --md5_v
alues_file="C:\file_with_clean_md5_values.txt --results_file="C:\file_with_resul
ts.txt

G:\>
```

Fig. 13 Running the scanner.exe program from a USB drive in help mode.

In the above cases G drive is the USB stick.

4.4.2 Scan mode with no parameters

If we press as mentioned above:

1) `.\php.exe C:\WAMP\www\scanner.php scan`

2) `scanner.exe scan`

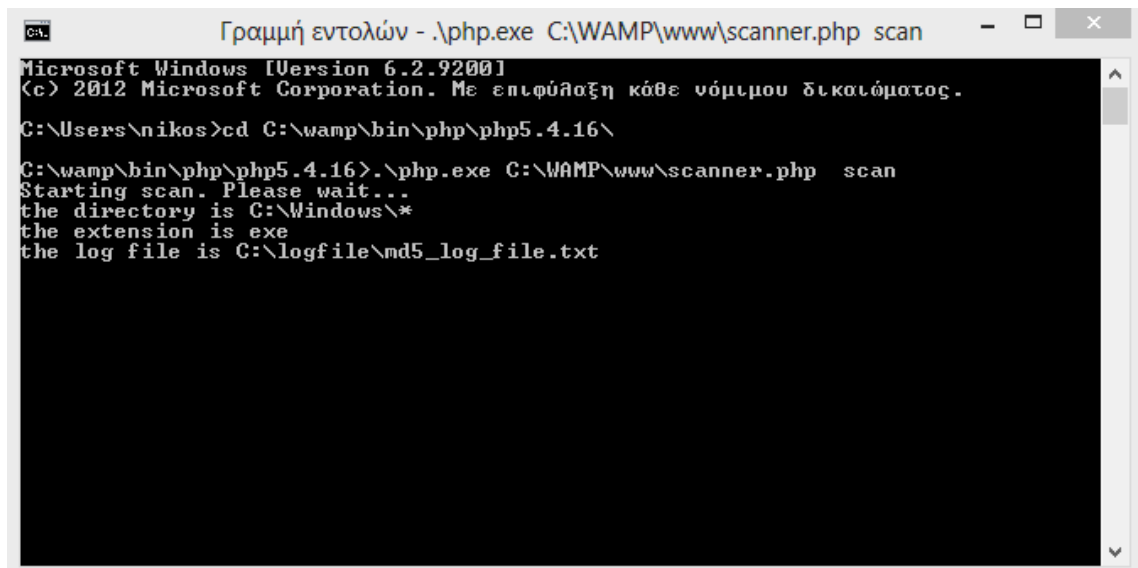
Then the program will use the scan mode with the default values (which the user can insert in the program using the Sublime-text editor) and are now set to be:

The path where the program to search in "C:\Windows".

The extension to search for "exe".

The path and the name of the .txt file to store the results "C:\logfile\md5_log_file.txt".

Figures 14, 15, 16, 17 and 18 visualize this functionality of the program.

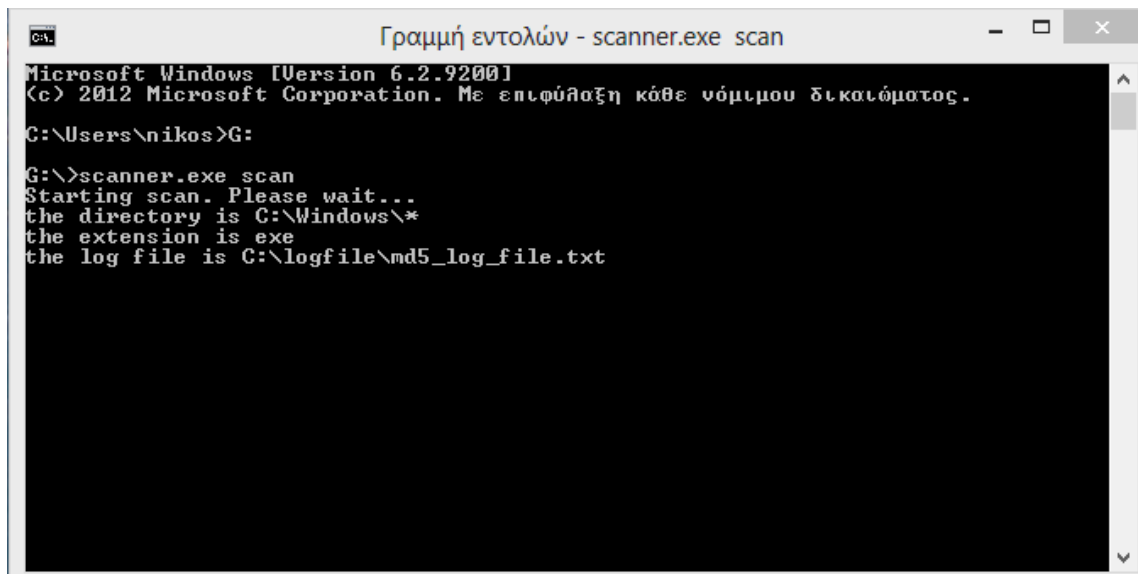


```
Γραμμή εντολών - .\php.exe C:\WAMP\www\scanner.php scan
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.
C:\Users\nikos>cd C:\wamp\bin\php\php5.4.16\
C:\wamp\bin\php\php5.4.16>.\php.exe C:\WAMP\www\scanner.php scan
Starting scan. Please wait...
the directory is C:\Windows\*
the extension is exe
the log file is C:\logfile\md5_log_file.txt
```

Fig. 14 Commands needed for running the scanner.php program in scan mode with no parameters.

```
Γραμμή εντολών
string(58) "C:\Windows\Microsoft.NET\Framework\v4.0.30319/ngentask.exe"
[101] =>
string(53) "C:\Windows\Microsoft.NET\Framework\v4.0.30319/vbc.exe"
[102] =>
string(49) "C:\Windows\Microsoft.NET\Framework\NETFXSBS10.exe"
[103] =>
string(91) "C:\Windows\Microsoft.NET\assembly\GAC_32\MSBuild\v4.0.4.0.0__b03f5f7f11d50a3a\MSBuild.exe"
[104] =>
string(103) "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\ComSvcConfig\v4.0.4.0.0__b03f5f7f11d50a3a\ComSvcConfig.exe"
[105] =>
string(111) "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\InspectUhdDialog\v4.0.6.2.0.0__31bf3856ad364e35\InspectUhdDialog.exe"
[106] =>
string(133) "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.Workflow.Compiler\v4.0.4.0.0__31bf3856ad364e35\Microsoft.Workflow.Compiler.exe"
[107] =>
string(97) "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\SMSvcHost\v4.0.4.0.0__b03f5f7f11d50a3a\SMSvcHost.exe"
[108] =>
string(99) "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\WsatConfig\v4.0.4.0.0__b03f5f7f11d50a3a\WsatConfig.exe"
[109] =>
string(89) "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\dfsvc\v4.0.4.0.0__b03f5f7f11d50a3a\dfsvc.exe"
[110] =>
string(97) "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\vmconnect\v4.0.6.2.0.0__31bf3856ad364e35\vmconnect.exe"
[111] =>
string(36) "C:\Windows\Speech\Common\sapisvr.exe"
[112] =>
string(36) "C:\Windows\System32\Boot\winload.exe"
[113] =>
string(38) "C:\Windows\System32\Boot\winresume.exe"
[114] =>
string(36) "C:\Windows\System32\Com\MigRegDB.exe"
[115] =>
string(35) "C:\Windows\System32\Com\comrepl.exe"
[116] =>
string(37) "C:\Windows\System32\Dism\DismHost.exe"
[117] =>
string(87) "C:\Windows\System32\DriverStore\FileRepository\bth.inf_x86_3033f5409fb0d31a/fsquirt.exe"
[118] =>
string(87) "C:\Windows\System32\DriverStore\FileRepository\bth.inf_x86_594bba3bef3a1195/fsquirt.exe"
[119] =>
string(87) "C:\Windows\System32\DriverStore\FileRepository\bth.inf_x86_651cbbd33256645e/fsquirt.exe"
[120] =>
string(87) "C:\Windows\System32\DriverStore\FileRepository\bth.inf_x86_6d0733bab272b88f/fsquirt.exe"
[121] =>
string(87) "C:\Windows\System32\DriverStore\FileRepository\bth.inf_x86_79a48b76ac1d2780/fsquirt.exe"
[122] =>
string(87) "C:\Windows\System32\DriverStore\FileRepository\bth.inf_x86_d5f87939da72af83/fsquirt.exe"
[123] =>
string(91) "C:\Windows\System32\DriverStore\FileRepository\cmazal.inf_x86_552d8a9147e94713/cmirmdrv.exe"
[124] =>
string(89) "C:\Windows\System32\DriverStore\FileRepository\hdart.inf_x86_aca1f15ed112685b\AERTSrv.exe"
[125] =>
string(97) "C:\Windows\System32\DriverStore\FileRepository\hdart.inf_x86_aca1f15ed112685b\DTSAudioService.exe"
[126] =>
string(95) "C:\Windows\System32\DriverStore\FileRepository\hdart.inf_x86_aca1f15ed112685b\DTSU2PAU$rv32.exe"
[127] =>
string(87) "C:\Windows\System32\DriverStore\FileRepository\hdart.inf_x86_aca1f15ed112685b\FMAPP.exe"
<more elements>...
C:\wamp\bin\php\php5.4.16>
```

Fig. 15 Results after running the scanner.php program in scan mode with no parameters.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\nikos>G:

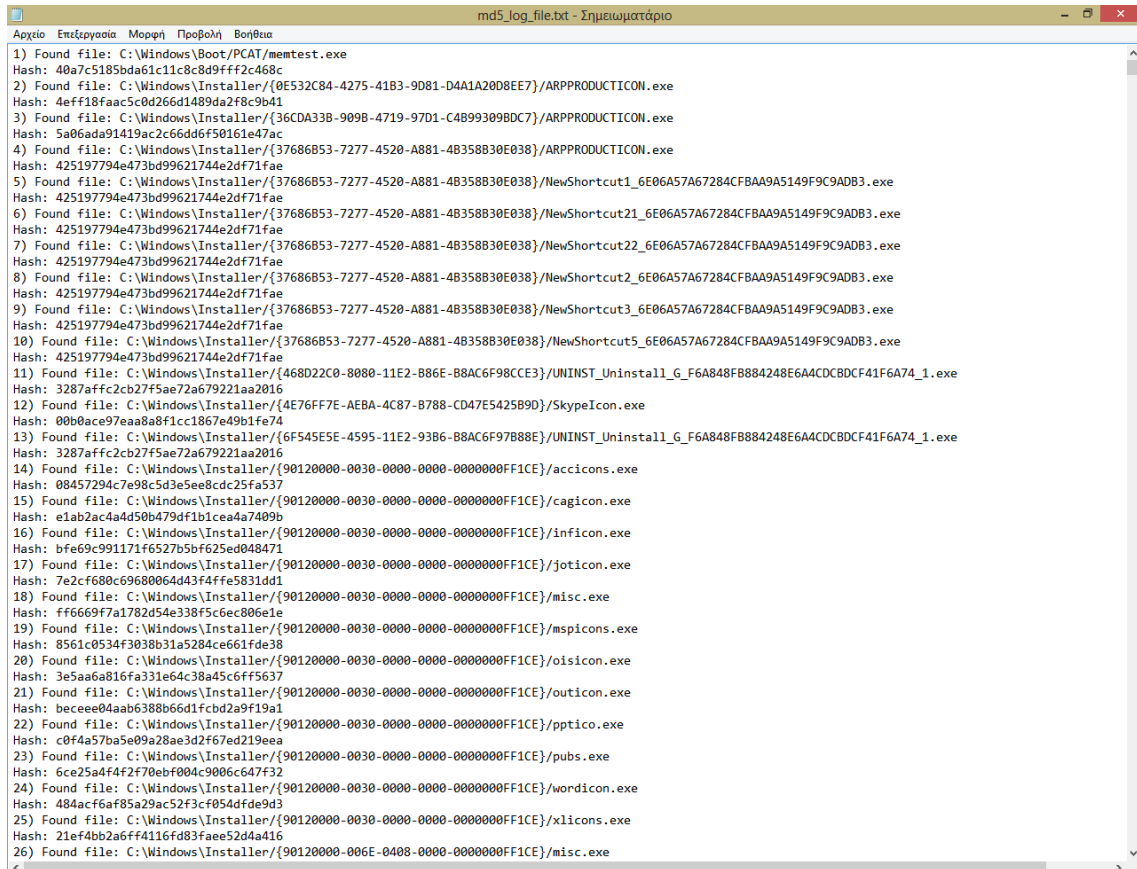
G:\>scanner.exe scan
Starting scan. Please wait...
the directory is C:\Windows\*
the extension is exe
the log file is C:\logfile\md5_log_file.txt
```

Fig. 16 Commands needed for running the scanner.exe program from a USB stick in scan mode with no parameters.

```
Γραμμή εντολών
[1740]=>
string(32) "C:\Windows\System32\vssadmin.exe"
[1741]=>
string(29) "C:\Windows\System32/w32tm.exe"
[1742]=>
string(31) "C:\Windows\System32/waitfor.exe"
[1743]=>
string(31) "C:\Windows\System32/wbadmin.exe"
[1744]=>
string(32) "C:\Windows\System32/wbengine.exe"
[1745]=>
string(31) "C:\Windows\System32/wecutil.exe"
[1746]=>
string(30) "C:\Windows\System32/wermgr.exe"
[1747]=>
string(32) "C:\Windows\System32/wevtutil.exe"
[1748]=>
string(32) "C:\Windows\System32/wextract.exe"
[1749]=>
string(29) "C:\Windows\System32/where.exe"
[1750]=>
string(30) "C:\Windows\System32/whoami.exe"
[1751]=>
string(32) "C:\Windows\System32/wiaacmgr.exe"
[1752]=>
string(31) "C:\Windows\System32/winserv.exe"
[1753]=>
string(31) "C:\Windows\System32/wininit.exe"
[1754]=>
string(31) "C:\Windows\System32/winload.exe"
[1755]=>
string(32) "C:\Windows\System32/winlogon.exe"
[1756]=>
string(33) "C:\Windows\System32/winresume.exe"
[1757]=>
string(29) "C:\Windows\System32/winrs.exe"
[1758]=>
string(33) "C:\Windows\System32/winrshost.exe"
[1759]=>
string(30) "C:\Windows\System32/winver.exe"
[1760]=>
string(34) "C:\Windows\System32/wksprbroker.exe"
[1761]=>
string(30) "C:\Windows\System32/wksprt.exe"
[1762]=>
string(31) "C:\Windows\System32/wlanext.exe"
[1763]=>
string(30) "C:\Windows\System32/wlrmrdr.exe"
[1764]=>
string(32) "C:\Windows\System32/wpnpinst.exe"
[1765]=>
string(29) "C:\Windows\System32/write.exe"
[1766]=>
string(31) "C:\Windows\System32/wscript.exe"
[1767]=>
string(35) "C:\Windows\System32/wsmprovhost.exe"
[1768]=>
string(32) "C:\Windows\System32/wsqmcons.exe"
[1769]=>
string(29) "C:\Windows\System32/wuapp.exe"
[1770]=>
string(31) "C:\Windows\System32/wuaucflt.exe"
[1771]=>
string(28) "C:\Windows\System32/wusa.exe"
[1772]=>
string(29) "C:\Windows\System32/xcopy.exe"
[1773]=>
string(32) "C:\Windows\System32/xpsrchvw.exe"
[1774]=>
string(31) "C:\Windows\System32/xwizard.exe"
[1775]=>
string(23) "C:\Windows\Temp\7za.exe"
[1776]=>
string(30) "C:\Windows\WinStore\MSHost.exe"
[1777]=>
string(41) "C:\Windows\servicing\TrustedInstaller.exe"
G:\>
```

Fig. 17 Results after running the scanner.exe program from a USB stick in scan mode with no parameters.

In the above cases G drive is the USB stick.



```
md5_log_file.txt - Σημειωματάριο
Αρχείο  Έκθεση  Μορφή  Προβολή  Βοήθεια
1) Found file: C:\Windows\Boot\PCAT/memtest.exe
Hash: 40a7c5185bda61c11c8c8d9fff2c468c
2) Found file: C:\Windows\Installer/{0E532C84-4275-41B3-9081-D4A1A20D8E7}/ARPPRODUCTICON.exe
Hash: 4eff18faac5c0d266d1489da2f8c9b41
3) Found file: C:\Windows\Installer/{36CD33B-909B-4719-97D1-C4B99309BDC7}/ARPPRODUCTICON.exe
Hash: 5a06ada91419ac2c66dd6f50161e47ac
4) Found file: C:\Windows\Installer/{37686B53-7277-4520-A881-4B358B30E838}/ARPPRODUCTICON.exe
Hash: 425197794e473bd99621744e2df71fae
5) Found file: C:\Windows\Installer/{37686B53-7277-4520-A881-4B358B30E838}/NewShortcut1_6E06A57A67284CFBAA9A5149F9C9ADB3.exe
Hash: 425197794e473bd99621744e2df71fae
6) Found file: C:\Windows\Installer/{37686B53-7277-4520-A881-4B358B30E838}/NewShortcut2_6E06A57A67284CFBAA9A5149F9C9ADB3.exe
Hash: 425197794e473bd99621744e2df71fae
7) Found file: C:\Windows\Installer/{37686B53-7277-4520-A881-4B358B30E838}/NewShortcut2_6E06A57A67284CFBAA9A5149F9C9ADB3.exe
Hash: 425197794e473bd99621744e2df71fae
8) Found file: C:\Windows\Installer/{37686B53-7277-4520-A881-4B358B30E838}/NewShortcut2_6E06A57A67284CFBAA9A5149F9C9ADB3.exe
Hash: 425197794e473bd99621744e2df71fae
9) Found file: C:\Windows\Installer/{37686B53-7277-4520-A881-4B358B30E838}/NewShortcut3_6E06A57A67284CFBAA9A5149F9C9ADB3.exe
Hash: 425197794e473bd99621744e2df71fae
10) Found file: C:\Windows\Installer/{37686B53-7277-4520-A881-4B358B30E838}/NewShortcut5_6E06A57A67284CFBAA9A5149F9C9ADB3.exe
Hash: 425197794e473bd99621744e2df71fae
11) Found file: C:\Windows\Installer/{468D22C0-8080-11E2-886E-B8AC6F98CE3}/UNINST_Uninstall_G_F6A848FB884248E6A4CDCBDC41F6A74_1.exe
Hash: 3287affc2cb27f5ae72a679221aa2016
12) Found file: C:\Windows\Installer/{4E76FF7E-AEBA-4C87-8788-CD47E5425B9D}/SkypeIcon.exe
Hash: 00b0ace97eaa8a8f1cc1867e49b1fe74
13) Found file: C:\Windows\Installer/{6F545E5E-4595-11E2-93B6-B8AC6F97888E}/UNINST_Uninstall_G_F6A848FB884248E6A4CDCBDC41F6A74_1.exe
Hash: 3287affc2cb27f5ae72a679221aa2016
14) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/accicons.exe
Hash: 08457294c7e98c5d3e5ee8cdc25fa537
15) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/cagicon.exe
Hash: e1ab2ac4a450b479df1b1cea4a7409b
16) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/inficon.exe
Hash: bfe69c91171f6527b5bf625ed048471
17) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/joticon.exe
Hash: 7e2cf680c69680064d43f4ffe5831dd1
18) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/misc.exe
Hash: ff6669f7a1782d54e338f5c6ec806e1e
19) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/mspicons.exe
Hash: 8561c0534f3038b31a5284ce661fde38
20) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/oisicon.exe
Hash: 3e5aa6a816fa331e64c38a45c6ff5637
21) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/outicon.exe
Hash: beceee04aab6388b66d1fcbd2a9f19a1
22) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/pptico.exe
Hash: c0f4a57ba5e09a28ae3d2f67ed219eea
23) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/pubs.exe
Hash: 6ce25a4f4f2f70ebf004c9006c647f32
24) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/wordicon.exe
Hash: 484ac6af85a29ac52f3cf054dfde9d3
25) Found file: C:\Windows\Installer/{90120000-0030-0000-0000-0000000FF1CE}/xlicons.exe
Hash: 21ef4bb2a6fff4116fd83faee52d4a416
26) Found file: C:\Windows\Installer/{90120000-006E-0408-0000-0000000FF1CE}/misc.exe
```

Fig. 18 The results md5_log_file.txt file after running the scanner.php or scanner.exe program in scan mode with no parameters.

4.4.3 Scan mode with parameters

We can use the scan mode with parameters if we want for example to use scanner.php to search the entire C drive, for *.docx files and store the results in the Cdocx.txt file, which will be located in the WindowsMD5 file which we will have created in advance in the C drive. In this case we have to insert in the command line the following commands:

```
1) .\php.exe C:\WAMP\www\scanner.php scan --path="C:\\" --extension="docx" --log_file="C:\WindowsMD5\Cdocx.txt"
```

Then the program will use as values (which the user can insert from the command prompt) the following:

The path where the program to search "C:\\\" (everything in C).

The extension to search for "docx".

The file to store the results "C:\WindowsMD5\Cdocx.txt".

If we want to use scanner.exe program from a USB stick to search all the D drive (partition of the hard disc) for *.txt files and store the results in the logDtxt.txt file which is located in the G drive (USB stick) we have to insert in the command line the following:

2) scanner.exe scan --path="D:\\\" --extension="txt" --log_file="G:\logDtxt.txt"

Then the program will use as values (which the user can insert from the command prompt) the following:

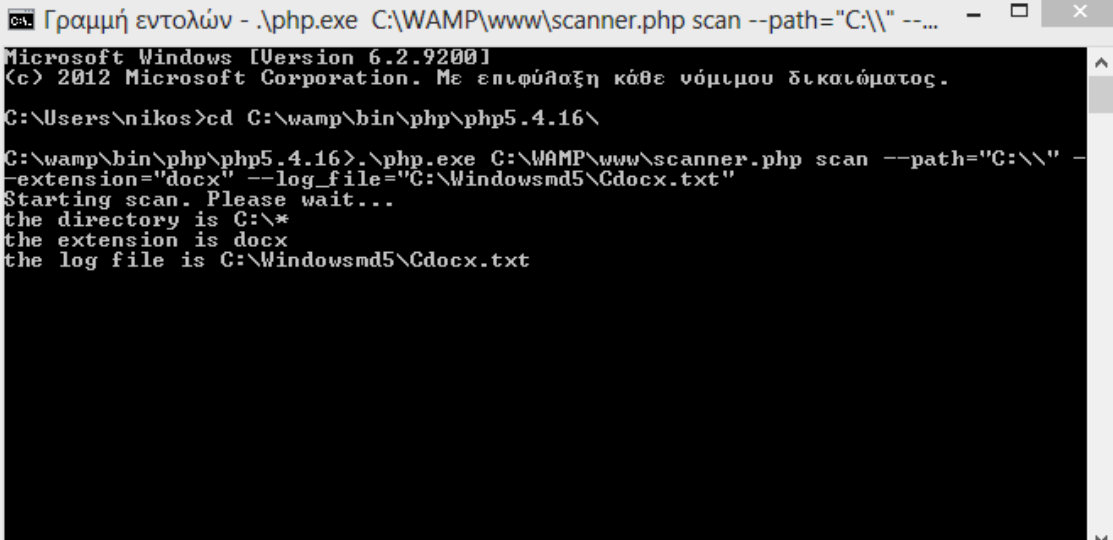
The path where the program to search "D:\\\" (everything in D).

The extension to search for "txt".

The file to store the results "G:\logDtxt.txt".

The program has the ability to search for any file extension we want .exe, .doc, .docx, .dll, .pdf, .log and so on. Also can search at any path we want, even USB sticks, and store the results wherever we want, including USB sticks, but in .txt only format.

Figures 19, 20, 21, 22, 23 and 24 visualize this functionality of the program.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\nikos>cd C:\wamp\bin\php\php5.4.16\

C:\wamp\bin\php\php5.4.16>.\php.exe C:\WAMP\www\scanner.php scan --path="C:\\\" --
extension="docx" --log_file="C:\Windowsmd5\Cdocx.txt"
Starting scan. Please wait...
the directory is C:\*
the extension is docx
the log file is C:\Windowsmd5\Cdocx.txt
```

Fig. 19 Commands needed for running the scanner.php program in scan mode with the above mentioned parameters.

```

C:\Users\nikos>cd C:\wamp\bin\php\php5.4.16\
C:\wamp\bin\php\php5.4.16>.\php.exe C:\WAMP\www\scanner.php scan --path="C:\\" --
--extension="docx" --log_file="C:\Windowsmd5\Cdocx.txt"
Starting scan. Please wait...
the directory is C:\*
the extension is docx
the log file is C:\Windowsmd5\Cdocx.txt
array(24) {
  [0] =>
  string(75) "C:\$Recycle.Bin/S-1-5-21-1123561945-1482476501-682003330-1004/$IXN
OP0H.docx"
  [1] =>
  string(66) "C:\Users\nikos\Desktop/Bike Built Buisness Plan - Contest\vpn.docx
"
  [2] =>
  string(55) "C:\Users\nikos\Desktop/bamcompile1.21/instructions.docx"
  [3] =>
  string(195) "C:\Users\nikos\Desktop/μτ||Γ||/All Chapters except from the Conc
lusions and the additional material 07-10-2013/Dissertation Nikolaos Bakirtzis/D
igital triage in forensics investigation 2013.docx"
  [4] =>
  string(89) "C:\Users\nikos\Desktop/μτ||Γ||/Dissertation Roadmap 15-07-2013/Di
ssertation Roadmap.docx"
  [5] =>
  string(109) "C:\Users\nikos\Desktop/μτ||Γ||/Interim report 22-08-2013/Digital
triage in forensics investigation 2013.docx"
  [6] =>
  string(91) "C:\Users\nikos\Desktop/μτ||Γ||/Interim report 22-08-2013/Disserta
tion Roadmap Updated.docx"
  [7] =>
  string(52) "C:\Users\nikos\Desktop/μτ||Γ||/TOOLS/Ω±Υή±óΗΗά.docx"
  [8] =>
  string(56) "C:\Users\nikos\Desktop/μτ||Γ||/guidelines/appendix.docx"
  [9] =>
  string(58) "C:\Users\nikos\Desktop/μτ||Γ||/guidelines/submission.docx"
  [10] =>
  string(74) "C:\Users\nikos\Desktop/Digital triage in forensics investigation 2
013.docx"
  [11] =>
  string(42) "C:\Users\nikos\Desktop/command prompt.docx"
  [12] =>
  string(38) "C:\Users\nikos\Desktop/references.docx"
  [13] =>
  string(38) "C:\Users\nikos\Desktop/~$ferences.docx"
  [14] =>
  string(86) "C:\Users\nikos\Desktop/~$gital triage in forensics investigation 2
013 - μ||Γ||.docx"
  [15] =>
  string(74) "C:\Users\nikos\Desktop/~$gital triage in forensics investigation 2
013.docx"
  [16] =>
  string(42) "C:\Users\nikos\Desktop/~$lk extractor.docx"
  [17] =>
  string(42) "C:\Users\nikos\Desktop/~$mmand prompt.docx"
  [18] =>
  string(36) "C:\Users\nikos\Desktop/~$óÁYIi2.docx"
  [19] =>
  string(69) "C:\Users\nikos\Desktop/~$Y ηή±ó±Y İYÿ Microsoft Office Word (2).d
ocx"
  [20] =>
  string(65) "C:\Users\nikos\Desktop/~$Y ηή±ó±Y İYÿ Microsoft Office Word.docx"
  [21] =>
  string(36) "C:\Users\nikos\Desktop/~$±ó°óΗY.docx"
  [22] =>
  string(35) "C:\Users\nikos\Desktop/±||Γ||.docx"
  [23] =>
  string(35) "C:\Users\nikos\Desktop/ίóÁYIi2.docx"
}
C:\wamp\bin\php\php5.4.16>

```

Fig. 20 Results after running the scanner.php program in scan mode with the above mentioned parameters.

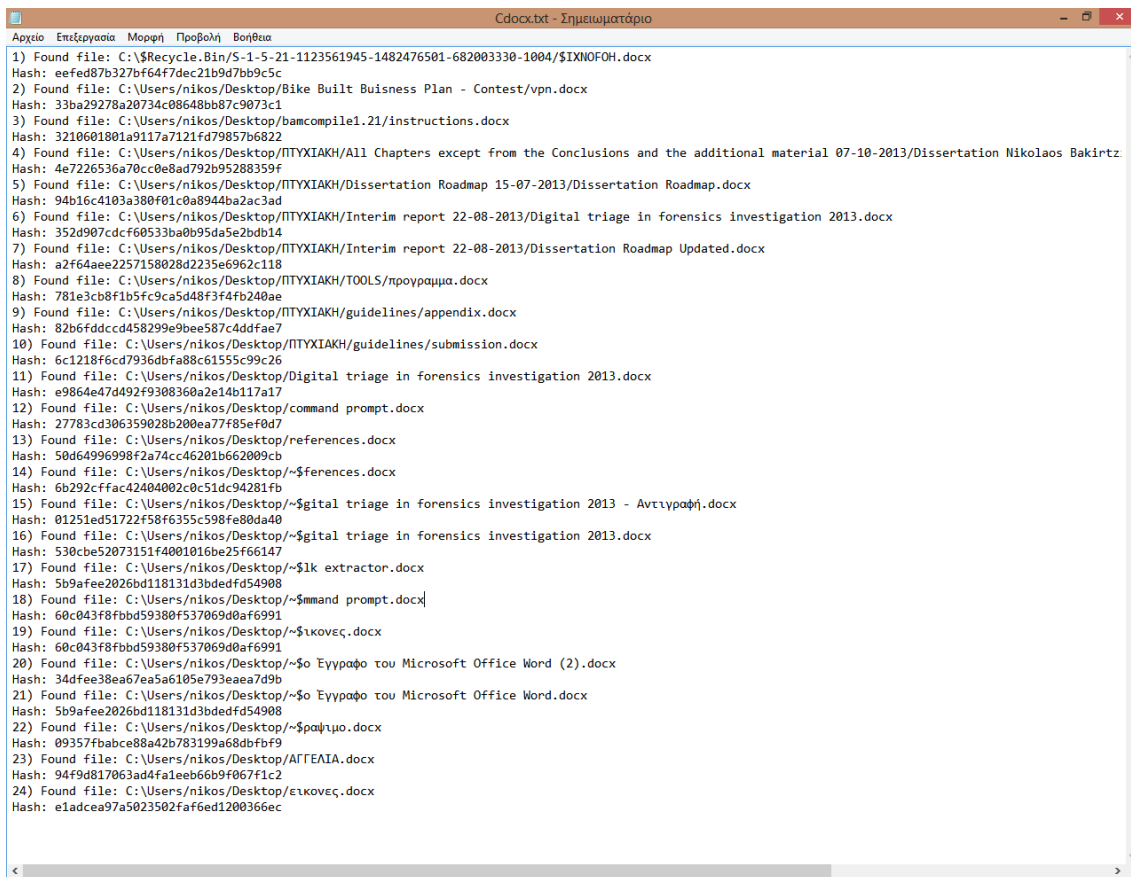


Fig. 21 The results Cdocx.txt file after running the scanner.php program in scan mode with the above mentioned parameters.

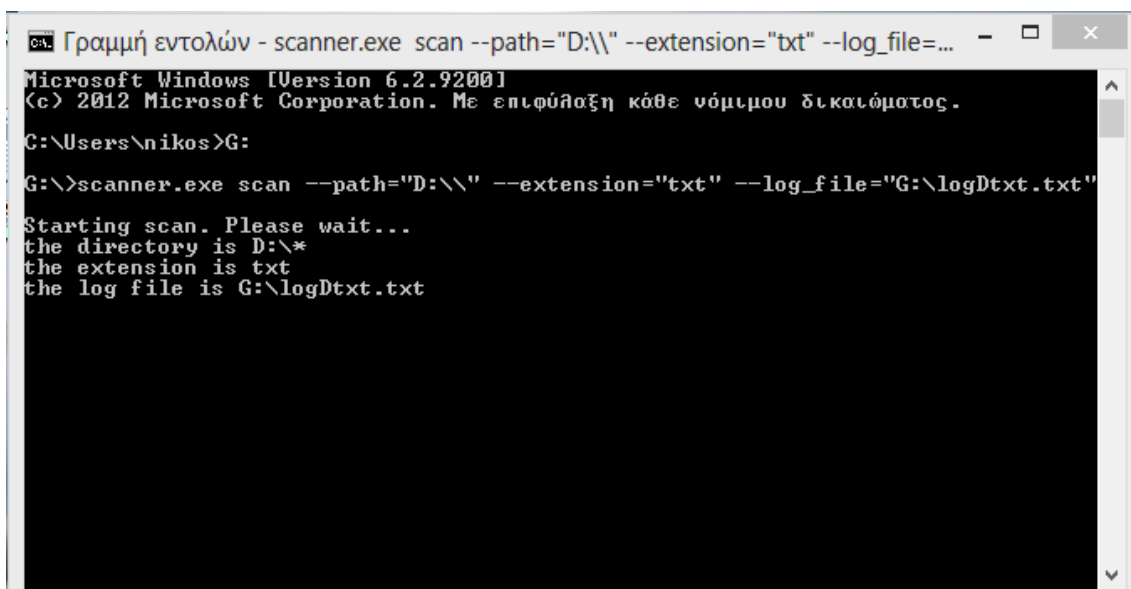


Fig. 22 Commands needed for running the scanner.exe program from a USB stick in scan mode with the above mentioned parameters.


```

Γραμμή εντολών
[1676 ]=>
string(109) "D:\...
[1677 ]=>
string(110) "D:\...
[1678 ]=>
string(116) "D:\...
[1679 ]=>
string(109) "D:\...
[1680 ]=>
string(78) "D:\...
[1681 ]=>
string(122) "D:\...
[1682 ]=>
string(77) "D:\...
[1683 ]=>
string(121) "D:\...
[1684 ]=>
string(68) "D:\...
[1685 ]=>
string(112) "D:\...
[1686 ]=>
string(108) "D:\...
[1687 ]=>
string(78) "D:\...
[1688 ]=>
string(82) "D:\...
[1689 ]=>
string(61) "D:\...
[1690 ]=>
string(61) "D:\...
[1691 ]=>
string(49) "D:\...
[1692 ]=>
string(94) "D:\...
[1693 ]=>
string(106) "D:\...
[1694 ]=>
string(99) "D:\...
[1695 ]=>
string(100) "D:\...
[1696 ]=>
string(83) "D:\...
[1697 ]=>
string(140) "D:\...
[1698 ]=>
string(137) "D:\...
[1699 ]=>
string(148) "D:\...
[1700 ]=>
string(145) "D:\...
[1701 ]=>
string(60) "D:\...
G:\>

```

Fig. 23 Results after running the scanner.exe program from a USB stick in scan mode with the above mentioned parameters. (1701 results)

A problem of the command line prompt is that it does not display words in Greek. It displays unknown symbols when it encounters Greek letters. The problem is solved in the *.txt files with the results, where every language including Greek is displayed correctly.

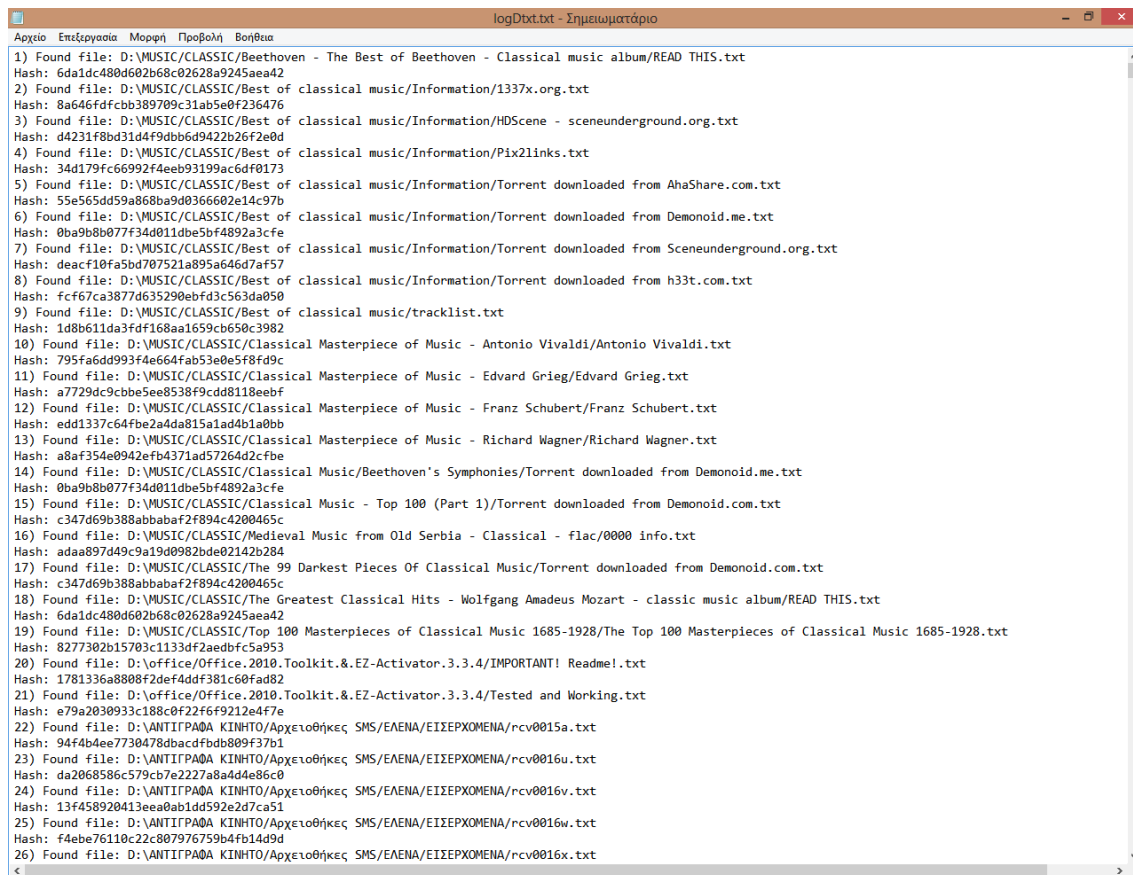


Fig. 24 The results logDbtxt.txt file after running the scanner.exe program from a USB drive in scan mode with the above mentioned parameters. (1701 results)

In all the above cases G drive is the USB stick.

4.4.4 Compare mode with parameters

The compare mode always needs parameters to run properly. If we want to use scanner.php program to compare two .txt files, the one with the MD5 values that we created by scanning the suspicious computer and the one with the clean MD5 values

that we don't want to take into account when we see the results, then we have to insert in the command line the following commands:

```
1) .\php.exe C:\WAMP\www\scanner.php compare
--log_file="C:\WindowsMD5\log.txt"
--md5_values_file="C:\WindowsMD5\clean.txt"
--results_file="C:\WindowsMD5\results.txt"
```

Then the program will use as values (which the user can insert from the command prompt) the following:

The path for the `log_file` with all the MD5 values under investigation "C:\WindowsMD5\log.txt". The `log.txt` file is the file that we created with the scan mode of the program `scanner.php` and contains all the files under investigation including their paths and their MD5 hash values.

The path for the `md5_values_file` with all the clean MD5 values that we don't want to take into consideration during the investigation is "C:\WindowsMD5\clean.txt". The `md5_values_file` is the file that we created with the scan mode of the program `scanner.php` in a previous time or in the lab, in a clean from viruses, malware and additional programs computer and that contains all the files and their MD5 values that don't oppose a threat to us. So we want to eliminate them from the investigation process.

The file to store the results of the comparison is set from the user to be "C:\WindowsMD5\results.txt". The file `WindowsMD5` is created from the user before running the program.

For the specific example we created the two files to compare as following: for the clean MD5 values file we used the scan mode of the `scanner.php` to create a file with all the *.exe files of a clean Windows 8 installation (`.\php.exe C:\WAMP\www\scanner.php scan --path="C:\Windows" --extension="exe" --log_file="C:\WindowsMD5\clean.txt"`) and for the MD5 values under investigation we used the scan mode of the `scanner.php` to create a file with all the *.exe files of the entire computer, partition C (`.\php.exe C:\WAMP\www\scanner.php scan --path="C:\\" --extension="exe" --log_file="C:\WindowsMD5\log.txt"`).

Figures 25, 26, 27, 28 and 29 visualize this functionality of the program.

```
clean.txt - Σημειωματάριο
Αρχείο  Επεξεργασία  Μορφή  Προβολή  Βοήθεια
1783) Found file: C:\Windows/assembly/NativeImages_v4.0.30319_32/vmconnect/44430f8a6396719e133b5ab441bf91d2/vmconnect.ni.exe
Hash: a7e6b4c022492d1228bca84c02263c0
1784) Found file: C:\Windows/servicing/TrustedInstaller.exe
Hash: 02d1fc0fda92fb34434166c612f95e5b
1785) Found file: C:\Windows/HelpPane.exe
Hash: d93d02787065eaafd36deec059e84afd
1786) Found file: C:\Windows/IsUninst.exe
Hash: 515e4684008e955de0c81e6a7aealc2a
1787) Found file: C:\Windows/RTHDCPL.exe
Hash: 75eb71ec21903a57e38e367cfab0c55e
1788) Found file: C:\Windows/RTLCP.exe
Hash: fb2ad52c255527fad0dcae427c707586
1789) Found file: C:\Windows/SoundMan.exe
Hash: 4ca041006428d72edc28390859668c4
1790) Found file: C:\Windows/alczwrd.exe
Hash: c9db82f8965c9fe62b9a3f03bb02143f
1791) Found file: C:\Windows/bfsvc.exe
Hash: 7d593720fc0c2c5dfb62bcc867573a21
1792) Found file: C:\Windows/dchcfcg32.exe
Hash: 9804c1eda26e70e343edee9f1807bc62
1793) Found file: C:\Windows/dciwds32.exe
Hash: 442a5ae3465cc2e532631adcf42dc523
1794) Found file: C:\Windows/dcmdev32.exe
Hash: 18a749b2bd12ed74261434656a110b5a
1795) Found file: C:\Windows/explorer.exe
Hash: eafe46b0292d2bd2467835e2acf717cc
1796) Found file: C:\Windows/hapint.exe
Hash: dfe8b187dcd083150c424024189339ec
1797) Found file: C:\Windows/hh.exe
Hash: 7837ab5539ae2bceac3ace5fb08d4ce5
1798) Found file: C:\Windows/notepad.exe
Hash: 22e1963fe26d5bceab0575eb6ff60cb5
1799) Found file: C:\Windows/regedit.exe
Hash: f71444a5935904c68e67ee3d35a4b948
1800) Found file: C:\Windows/slrundll.exe
Hash: 987fee1415e95d146804d8d008ed0162
1801) Found file: C:\Windows/sp1wow64.exe
Hash: d5abb3abe528212fc51484c92afb1b59
1802) Found file: C:\Windows/twunk_16.exe
Hash: f36a271706edd23c94956afb56981184
1803) Found file: C:\Windows/twunk_32.exe
Hash: b24577b1287a340cc76ccf2ed2c1db35
1804) Found file: C:\Windows/unin0408.exe
Hash: 2c82955895a809c5bd6d7c0978cb93fb
1805) Found file: C:\Windows/winhlp.exe
Hash: 8e6f7d51a5cb299c25621c6c1ab57e84
1806) Found file: C:\Windows/winhlp32.exe
Hash: eada08c87ad2a913563244ccf4391e5d
1807) Found file: C:\Windows/write.exe
Hash: 185b8e5d0e8ce317cd53294441fb4491
```

Fig. 25 The clean.txt file with all the *.exe files of the clean Windows 8 installation. (1807 results)

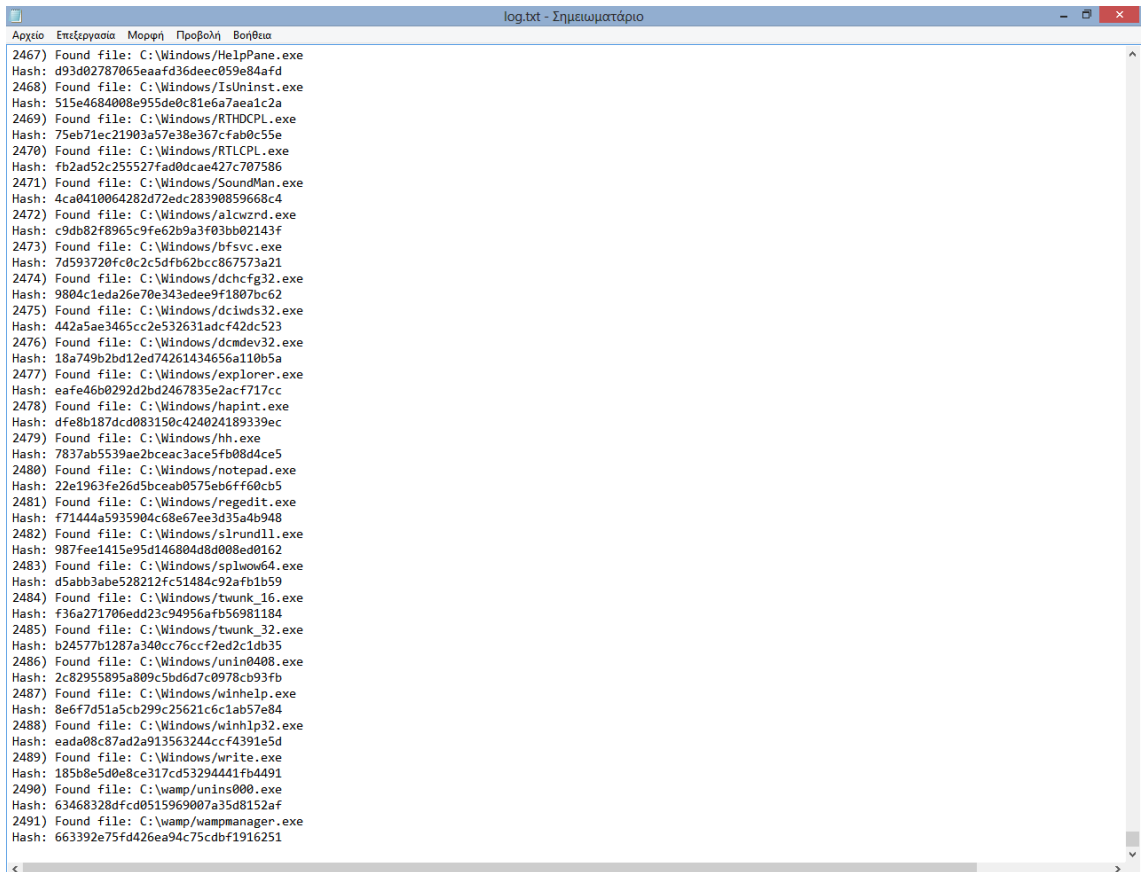


Fig. 26 The log.txt file with all the *.exe files of the entire computer, partition C.
(2491 results)

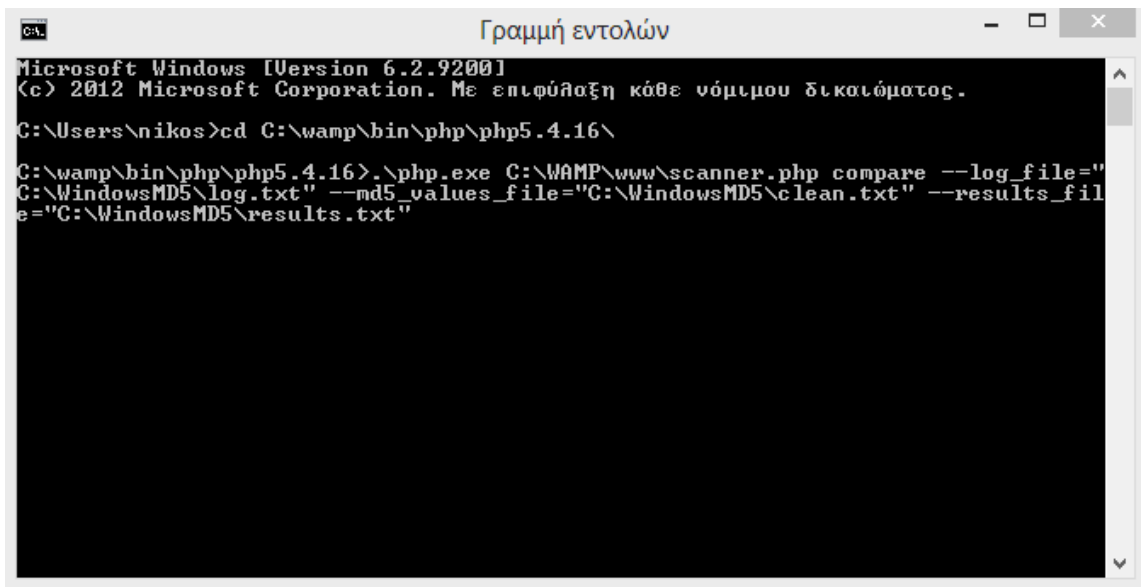


Fig. 27 Commands needed for running the scanner.php program in compare mode as mentioned above.

```
Γραμμή εντολών
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/md5d
eep-4.3/md5deep-4.3/tigerdeep64.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/md5d
eep-4.3/md5deep-4.3/whirlpooldeep.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/md5d
eep-4.3/md5deep-4.3/whirlpooldeep64.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/rifi
uti/rifiuti/rifiuti.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/srcs
/proxy.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/7z92
0.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/Soph
os Virus Removal Tool.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/catc
hme.exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/mbr.
exe
Suspicious file found -> C:\Users\nikos\Desktop\μτ||μ||\TOOLS/super kludge/mdd_
1.3.exe
Suspicious file found -> C:\Users\nikos\Desktop\scanner.exe
Suspicious file found -> C:\Users\nikos\Desktop\scanner_version_2.exe
Suspicious file found -> C:\Users\nikos\Desktop\scanner_version_3.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\ApacheMonitor.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\ab.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\abs.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\htcacheclean.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\htdbm.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\htdigest.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\htpasswd.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\httpd.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\htt2dbm.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\logresolve.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\openssl.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin\rotatelogs.exe
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\bin>wintty.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\echo.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\innochecksum.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\my_print_defaults.exe

Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\myisam_ftdump.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\myisamchk.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\myisamlog.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\myisampack.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_client_test.exe

Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_client_test_emb
bedded.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_config_editor.e
xe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_embedded.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_plugin.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_tzinfo_to_sql.e
xe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_upgrade.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqldadmin.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqldbinlog.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqlcheck.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqld--debug.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqld.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqldump.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqlimport.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqlshow.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqslap.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqltest.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqltest_embedded.ex
e
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\perror.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\replace.exe
Suspicious file found -> C:\wamp\bin\mysql\mysql5.6.12\bin\resolveip.exe
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php-cgi.exe
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php-win.exe
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php.exe
Suspicious file found -> C:\wamp\tools\xdc\xdc.exe
Suspicious file found -> C:\wamp\unins000.exe
Suspicious file found -> C:\wamp\wampmanager.exe

C:\wamp\bin\php\php5.4.16>
```

Fig. 28 The results after running the scanner.php program in compare mode.

```

results.txt - Σημειωματάριο
Αρχείο  Επεξεργασία  Μορφή  Προβολή  Βοήθεια
583) Suspicious file Found -> C:\Users\nikos\Desktop/scanner_version_2.exe
584) Suspicious file Found -> C:\Users\nikos\Desktop/scanner_version_3.exe
585) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\ApacheMonitor.exe
586) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\ab.exe
587) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\abs.exe
588) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\htcacheclean.exe
589) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\htdbm.exe
590) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\htdigest.exe
591) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\htpasswd.exe
592) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\httpd.exe
593) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\httpd2dm.exe
594) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\logresolve.exe
595) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\openssl.exe
596) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\rotatelogs.exe
597) Suspicious file Found -> C:\wamp\bin\apache\Apache2.4.4\bin\wintty.exe
598) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\echo.exe
599) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\innochecksum.exe
600) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\my_print_defaults.exe
601) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysam_ftdump.exe
602) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysamchk.exe
603) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysamlog.exe
604) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysampack.exe
605) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe
606) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_client_test.exe
607) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_client_test_embedded.exe
608) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_config_editor.exe
609) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_embedded.exe
610) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_plugin.exe
611) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_tzinfo_to_sql.exe
612) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysql_upgrade.exe
613) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqladmin.exe
614) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqlbinlog.exe
615) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqlcheck.exe
616) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqld-debug.exe
617) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqld.exe
618) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqldump.exe
619) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqlimport.exe
620) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqlshow.exe
621) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqslap.exe
622) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqctest.exe
623) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\mysqctest_embedded.exe
624) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\perror.exe
625) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\replace.exe
626) Suspicious file Found -> C:\wamp\bin\mysql\mysql5.6.12\bin\resolveip.exe
627) Suspicious file Found -> C:\wamp\bin\php\php5.4.16\php-cgi.exe
628) Suspicious file Found -> C:\wamp\bin\php\php5.4.16\php-win.exe
629) Suspicious file Found -> C:\wamp\bin\php\php5.4.16\php.exe
630) Suspicious file Found -> C:\wamp\tools\xdc\xdc.exe
631) Suspicious file Found -> C:\wamp\unins000.exe
632) Suspicious file Found -> C:\wamp\wampmanager.exe

```

Fig. 29 The results.txt file with the results after running the scanner.php program in compare mode with the above mentioned parameters. (632 results)

If we want to use scanner.exe program, which is located in the G drive (USB stick), to compare two .txt files, the one with the MD5 values that we created by scanning the suspicious computer and the one with the clean MD5 values that we don't want to take into account when we see the results, then we have to insert in the command line the following commands (after making the working directory of the CMD the G:):

```
2) scanner.exe compare --log_file="G:\log.txt" --md5_values_file="G:\clean.txt"
--results_file="G:\results.txt"
```

Then the program will use as values (which the user can insert from the command prompt) the following:

The path for the log_file with all the MD5 values under investigation "G:\log.txt". G drive is the USB stick. The log.txt is the file that we created with the scan mode of the program scanner.exe and contains all the files under investigation, including their paths and their MD5 values.

The path for the md5_values_file with all the clean MD5 values that we don't want to take into consideration during the investigation "G:\clean.txt". The md5_values_file is the file that we created with the scan mode of the program scanner.exe, in a previous time or in the lab, in a clean from viruses, malware and additional programs computer and contains all the files and their MD5 values that don't oppose a threat to us. So we want to eliminate them from the investigation process.

The file to store the results from the comparison is set from the user to be "G:\results.txt". As mentioned above G drive is the USB stick.

For the specific example we created the two files to make the comparison as following: for the clean MD5 values file we used the scan mode of the scanner.exe to create a file with all the *.dll files of a clean Windows 8 installation (scanner.exe scan --path="C:\Windows" --extension="dll" --log_file="G:\clean.txt") and for the MD5 values under investigation we used the scan mode of the scanner.exe to create a file with all the *.dll files of the entire computer, partition C (scanner.exe scan --path="C:\\ " --extension="dll" --log_file="G:\log.txt").

The program has the ability to search for any file extension we want .exe, .doc, .docx, .dll, .pdf, .log and so on. It can also search at any path we want, even USB sticks, and store the results wherever we want, including USB sticks, but in .txt only format.

In some cases, depending on the Operating System (Windows 8, 7, Vista etc), the CLI mode must be executed as administrator. This must be done in order for the compare mode of the scanner.exe program to work properly. In these cases there are not enough privileges, for the program, to execute all its functions, concerning the files and wherever these might have been placed.

Figures 30, 31, 32, 33 and 34 visualize this functionality of the program.


```
clean.txt - Σημειωματάριο
Αρχείο  Επεξεργασία  Μορφή  Προβολή  Βοήθεια
12674) Found file: C:\Windows\diagnostics\system/DeviceCenter/DiagPackage.dll
Hash: 48025f5096666c9ccb03d2fc775ed7f
12675) Found file: C:\Windows\diagnostics\system/HomeGroup/DiagPackage.dll
Hash: 7dec7fea0f9612eba5125015bd109f5
12676) Found file: C:\Windows\diagnostics\system/HomeGroup/Microsoft-Windows-HomeGroupDiagnostic.Interop.dll
Hash: 094829f4bd83035e388dd234e45f86e
12677) Found file: C:\Windows\diagnostics\system/HomeGroup/Microsoft-Windows-HomeGroupDiagnostic.NetListMgr.Interop.dll
Hash: 4dad5b3263a2ac2baeafc450ded119c0
12678) Found file: C:\Windows\diagnostics\system/IEBrowseWeb/DiagPackage.dll
Hash: ff9f67ab23bc66ff5705d56cc2963917
12679) Found file: C:\Windows\diagnostics\system/IESecurity/DiagPackage.dll
Hash: 9b1ad6f95b0bccdb83e053557bbd9f2d
12680) Found file: C:\Windows\diagnostics\system/Networking/DiagPackage.dll
Hash: 5a72e3b2d6e9e649e6d313ee57b89f3
12681) Found file: C:\Windows\diagnostics\system/Networking/NetworkDiagnosticSnapIn.dll
Hash: 2fa53b758a8c7f3fa66836c89339afe7
12682) Found file: C:\Windows\diagnostics\system/PCW/DiagPackage.dll
Hash: 7b4ff08e778f07ec46298fd22ab797d7
12683) Found file: C:\Windows\diagnostics\system/Performance/DiagPackage.dll
Hash: a00be81c36cf2651c7ce1e2eedbdfffd
12684) Found file: C:\Windows\diagnostics\system/Power/DiagPackage.dll
Hash: 876495f00e6cb496ecb72d8f5bdb86bd
12685) Found file: C:\Windows\diagnostics\system/Printer/DiagPackage.dll
Hash: 276269e48e0acf045eac10fb10f797e5
12686) Found file: C:\Windows\diagnostics\system/Printer/UpdatePrinterDriver.dll
Hash: 83d89e1657e440721058f682aec0ba60
12687) Found file: C:\Windows\diagnostics\system/Search/DiagPackage.dll
Hash: bdd23008f07a0611090c910b545ba2cb
12688) Found file: C:\Windows\diagnostics\system/UsbCore/DiagPackage.dll
Hash: 5115de580fbde0f9f6304600489eb250
12689) Found file: C:\Windows\diagnostics\system/WindowsMediaPlayerConfiguration/DiagPackage.dll
Hash: 8a63eb8a9402cefa0d8ea06e6962b3cb
12690) Found file: C:\Windows\diagnostics\system/WindowsMediaPlayerMediaLibrary/DiagPackage.dll
Hash: a6a81efb930c226b06e92ffbec3fcb5a
12691) Found file: C:\Windows\diagnostics\system/WindowsMediaPlayerPlayDVD/DiagPackage.dll
Hash: 3040be46af9f50a6c600dfd19a4c1220
12692) Found file: C:\Windows\diagnostics\system/WindowsUpdate/DiagPackage.dll
Hash: fadb557dd4aa5f2d69b2c07c22638845
12693) Found file: C:\Windows\servicing/CbsApi.dll
Hash: f1720df60212ba9baa98a542119d69a4
12694) Found file: C:\Windows\servicing/CbsMsg.dll
Hash: 4b805e20ee9dc7897ad390459c006d34
12695) Found file: C:\Windows\servicing/wrprintapi.dll
Hash: 9585f51ff09bdfa336f1475520583ca7
12696) Found file: C:\Windows\RtlExUpd.dll
Hash: 2a7b78f4cfa0f1a5655891ddaacefad9
12697) Found file: C:\Windows\twain.dll
Hash: 0bea3f79a36b1f67b2ce0f595524c77c
12698) Found file: C:\Windows\twain_32.dll
Hash: da7eb5d3652fe2b1676aaa9e6e241e68
```

Fig. 30 The clean.txt file with all the *.dll files of the clean Windows 8 installation.
(12698 results)

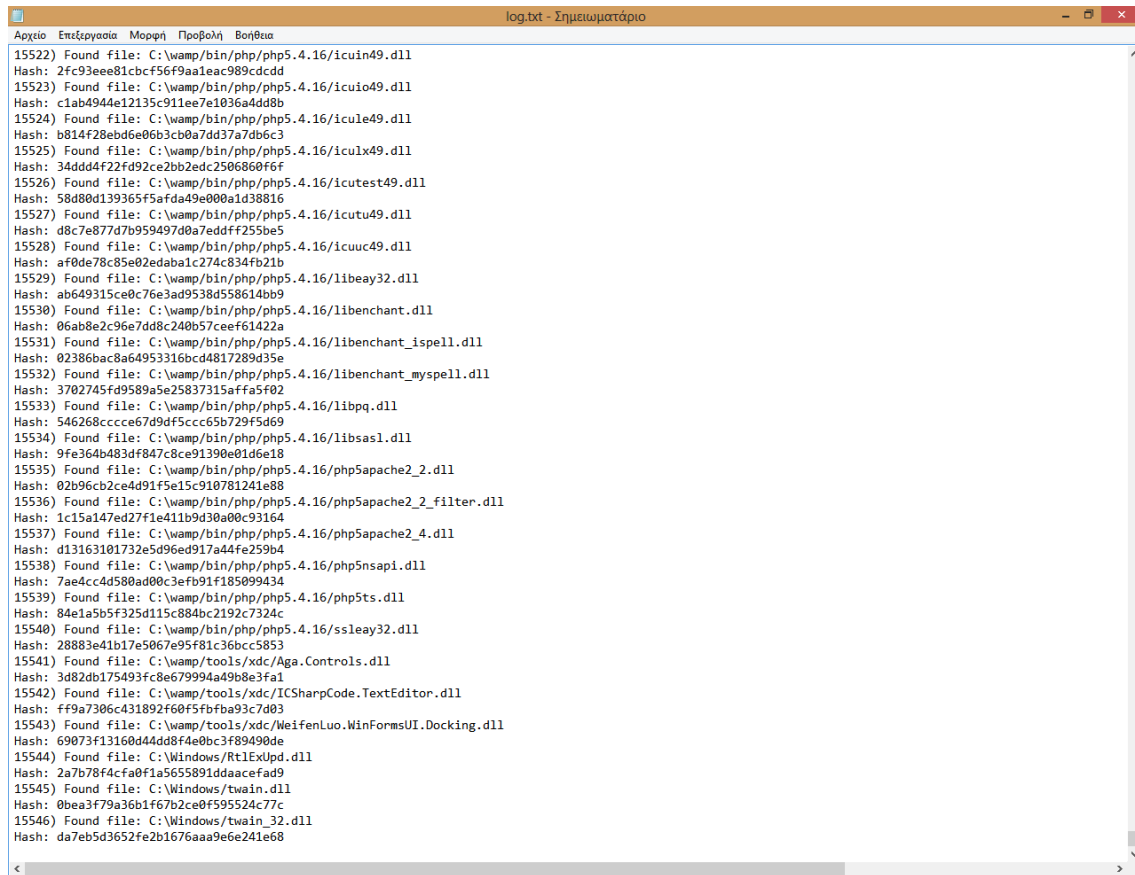


Fig. 31 The log.txt file with all the *.dll files of the whole computer, partition C.
(15546 results)

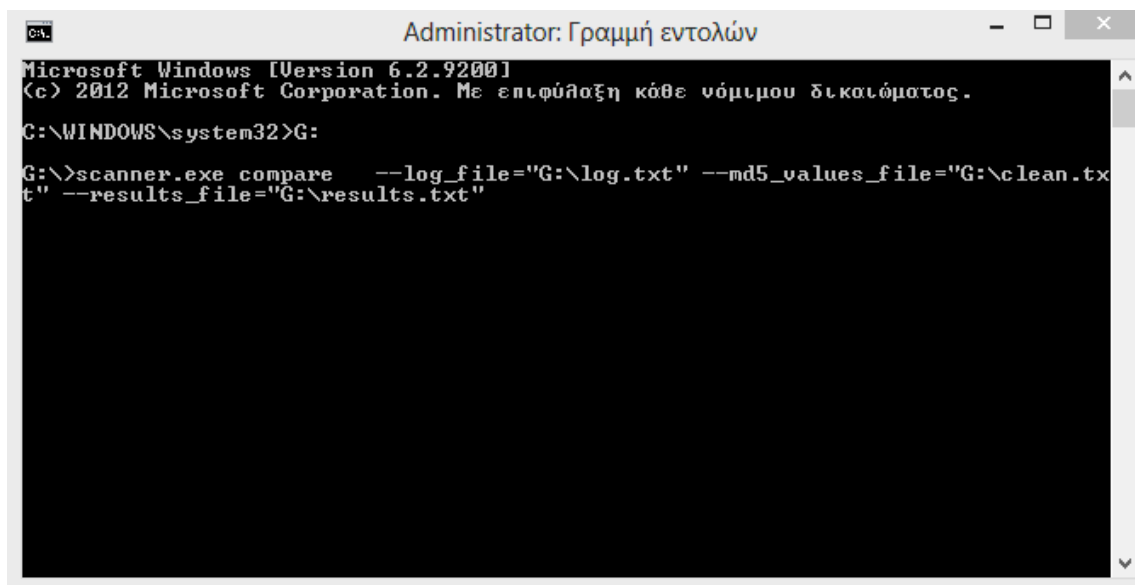


Fig. 32 Commands needed for running the scanner.exe program in compare mode as mentioned above. (run as administrator)

```
Administrator: Γραμμή εντολών
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_pgsql.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_sqlite.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pgsql.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_shmop.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_snmp.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_soap.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_sockets.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_sqlite3.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_sybase_ct.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_tidy.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_xmlrpc.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_xsl.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\zend_ext\php_xdebug-2.2.3-5.4-uc9.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\glib-2.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\gmodule-2.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\icudt49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\icuin49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\icuiio49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\icule49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\iculx49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\icutest49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\icutu49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\icuuc49.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\libeay32.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\libenchant.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\libenchant_ispell.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\libenchant_myspell.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\libpq.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\libsasl.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5apache2_2.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5apache2_2_filter.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5apache2_4.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5nsapi.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5ts.dll
Suspicious file found -> C:\wamp\bin\php\php5.4.16\ssleay32.dll
Suspicious file found -> C:\wamp\tools\xdc\Aga.Controls.dll
Suspicious file found -> C:\wamp\tools\xdc\ICSharpCode.TextEditor.dll
Suspicious file found -> C:\wamp\tools\xdc>WeifenLuo.WinFormsUI.Docking.dll
G:\>
```

Fig. 33 The results after running the scanner.exe program in compare mode as mentioned above.

```
results.txt - Σημειωματάριο
Αρχείο  Επεξεργασία  Μορφή  Προβολή  Βοήθεια
2437 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_intl.dll
2438 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_ldap.dll
2439 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_mbstring.dll
2440 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_mysql.dll
2441 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_mysqli.dll
2442 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_oc18.dll
2443 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_oc18_11g.dll
2444 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_openssl.dll
2445 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_firebird.dll
2446 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_mysql.dll
2447 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_oci.dll
2448 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_odbc.dll
2449 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_pgsql.dll
2450 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pdo_sqlite.dll
2451 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_pgsql.dll
2452 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_shmop.dll
2453 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_snmp.dll
2454 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_soap.dll
2455 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_sockets.dll
2456 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_sqlite3.dll
2457 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_sybase_ct.dll
2458 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_tidy.dll
2459 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_xmlrpc.dll
2460 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ext\php_xsl.dll
2461 Suspicious file found -> C:\wamp\bin\php\php5.4.16\zend_ext\php_xdebug-2.2.3-5.4-vc9.dll
2462 Suspicious file found -> C:\wamp\bin\php\php5.4.16\glib-2.dll
2463 Suspicious file found -> C:\wamp\bin\php\php5.4.16\gmodule-2.dll
2464 Suspicious file found -> C:\wamp\bin\php\php5.4.16\icudt49.dll
2465 Suspicious file found -> C:\wamp\bin\php\php5.4.16\icuio49.dll
2466 Suspicious file found -> C:\wamp\bin\php\php5.4.16\icuio49.dll
2467 Suspicious file found -> C:\wamp\bin\php\php5.4.16\icule49.dll
2468 Suspicious file found -> C:\wamp\bin\php\php5.4.16\iculx49.dll
2469 Suspicious file found -> C:\wamp\bin\php\php5.4.16\icutest49.dll
2470 Suspicious file found -> C:\wamp\bin\php\php5.4.16\icutu49.dll
2471 Suspicious file found -> C:\wamp\bin\php\php5.4.16\icuuc49.dll
2472 Suspicious file found -> C:\wamp\bin\php\php5.4.16\libeay32.dll
2473 Suspicious file found -> C:\wamp\bin\php\php5.4.16\libeay32.dll
2474 Suspicious file found -> C:\wamp\bin\php\php5.4.16\libeay32.dll
2475 Suspicious file found -> C:\wamp\bin\php\php5.4.16\libeay32.dll
2476 Suspicious file found -> C:\wamp\bin\php\php5.4.16\libpq.dll
2477 Suspicious file found -> C:\wamp\bin\php\php5.4.16\libsasl.dll
2478 Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5apache2_2.dll
2479 Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5apache2_2_filter.dll
2480 Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5apache2_4.dll
2481 Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5sapi.dll
2482 Suspicious file found -> C:\wamp\bin\php\php5.4.16\php5sapi.dll
2483 Suspicious file found -> C:\wamp\bin\php\php5.4.16\ssleay32.dll
2484 Suspicious file found -> C:\wamp\tools\xdc\Aga.Controls.dll
2485 Suspicious file found -> C:\wamp\tools\xdc\ICSharpCode.TextEditor.dll
2486 Suspicious file found -> C:\wamp\tools\xdc\MeifenLuo.WinFormsUI.Docking.dll
```

Fig. 34 The results.txt file with the results after running the scanner.exe program in compare mode with the above mentioned parameters. (2486 results)

As we can verify the results.txt file contains all the *.dll files of the whole partition C except the *.dll files that are included in the Windows 8 installation file.

In the next chapters, two versions of the scanner.php and scanner.exe programs are presented, which implement some differences and alterations that could be useful to the investigator.

4.5 The scanner_version_2 program

This version of scanner.php and scanner.exe has this difference: for the comparison in the compare mode, uses as the md5_values_file with all the clean MD5 values, a *.txt file that contains only MD5 values and nothing else. This *.txt file consists of one MD5 value in each separate line and nothing else.

So when the user inserts from the command prompt, the path of the md5_values_file with all the clean MD5 values (that we don't want to take into consideration during the investigation), this file consists of one MD5 value in each separate line and nothing else.

All the other functionality of the program remains unchanged.

This differentiation was necessary in order, for the program, to have the ability to utilize and exploit possible dump files from other programs or other created files that contain only MD5 values. This makes it more valuable for commercial use. The first program has the restriction that works only with files that were created with the scan mode of the same program.

4.6 The scanner_version_3 program

This version of scanner.php and scanner.exe has this difference from the first program: for the comparison, in the compare mode, uses as the md5_values_file, with all the clean MD5 values, a *.txt file that contains only MD5 values and nothing else, but also has the ability to create these *.txt files with only MD5 values inside by using the scan mode of this version of the program.

So the user has the ability to create a *.txt file with only MD5 values inside but also afterwards compare this file with the log file (log_file) created from the first or the second program.

All the other functionality of the program, and the way it is used, remains unchanged.

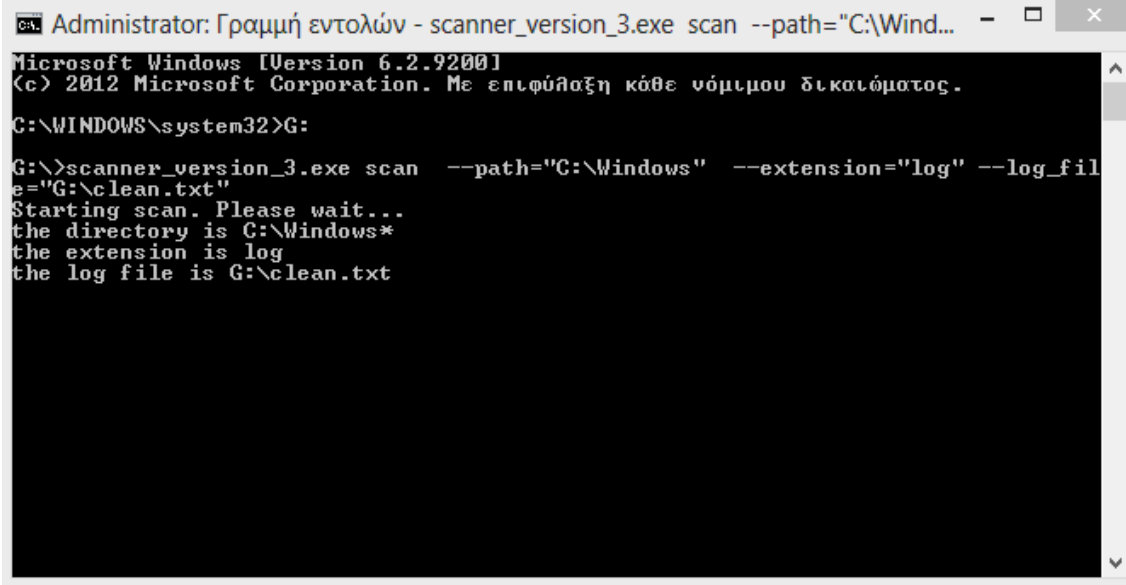
This differentiation was necessary, in order for the user to have the ability to create himself the *.txt files with only MD5 values inside, in order to use them as input for other programs or for investigation reasons. This makes it more valuable for trial and investigation use.

In the below example we used the scanner_version_2.exe and scanner_version_3.exe programs from a USB stick (G drive) to demonstrate their use.

For the specific example we created the two files to make the comparison as following: for the clean MD5 values file, with only MD5 values inside, we used the scan mode

of the scanner_version_3.exe to create a file with all the *.log files of a clean Windows 8 installation (scanner_version_3.exe scan --path="C:\Windows" --extension="log" --log_file="G:\clean.txt") and for the MD5 values under investigation (.txt file that includes the path and the hash code of each file) we used the scan mode of the scanner_version_2.exe to create a file with all the *.log files of the entire computer, partition C (scanner_version_2.exe scan --path="C:\\ " --extension="log" --log_file="G:\log.txt").

Figures 35, 36, 37, 38, 39, 40, 41, 42 and 43 visualize this functionalities of the programs.



```
Administrator: Γραμμή εντολών - scanner_version_3.exe scan --path="C:\Wind... - □ ×
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.
C:\WINDOWS\system32>G:
G:\>scanner_version_3.exe scan --path="C:\Windows" --extension="log" --log_file="G:\clean.txt"
Starting scan. Please wait...
the directory is C:\Windows*
the extension is log
the log file is G:\clean.txt
```

Fig. 35 Creating the clean.txt file with only MD5 values inside by using the scanner_version_3.exe program in scan mode.

```
Administrator: Γραμμή εντολών
CLR_v4.0_32/UsageLogs/NGenTask.exe.log"
[63]=>
string(103) "C:\Windows\System32/config/systemprofile/AppData/Local/Microsoft/
CLR_v4.0_32/UsageLogs/taskhost.exe.log"
[64]=>
string(105) "C:\Windows\System32/config/systemprofile/AppData/Local/Microsoft/
CLR_v4.0_32/UsageLogs/taskhost.exe.log"
[65]=>
string(31) "C:\Windows\System32/sru/SRU.log"
[66]=>
string(36) "C:\Windows\System32/sru/SRU00502.log"
[67]=>
string(36) "C:\Windows\System32/sru/SRU00503.log"
[68]=>
string(36) "C:\Windows\System32/sru/SRU00504.log"
[69]=>
string(59) "C:\Windows/Temp/vmware-SYSTEM/vmware-usbarb-SYSTEM-3068.log"
[70]=>
string(59) "C:\Windows/Temp/vmware-SYSTEM/vmware-usbarb-SYSTEM-5724.log"
[71]=>
string(37) "C:\Windows/Temp/ASPNETSetup_000000.log"
[72]=>
string(28) "C:\Windows/Temp/MpCmdRun.log"
[73]=>
string(32) "C:\Windows/Temp/Silverlight0.log"
[74]=>
string(34) "C:\Windows/Temp/SilverlightMSI.log"
[75]=>
string(36) "C:\Windows/Temp/chrome_installer.log"
[76]=>
string(26) "C:\Windows/Temp/vminst.log"
[77]=>
string(28) "C:\Windows/Temp/winstore.log"
[78]=>
string(29) "C:\Windows/WinSxS/pogexec.log"
[79]=>
string(33) "C:\Windows/debug/WIA/wiatracer.log"
[80]=>
string(24) "C:\Windows/debug/mrt.log"
[81]=>
string(27) "C:\Windows/debug/mrteng.log"
[82]=>
string(29) "C:\Windows/debug/netlogon.log"
[83]=>
string(27) "C:\Windows/debug/sammui.log"
[84]=>
string(31) "C:\Windows/inf/setupapi.app.log"
[85]=>
string(47) "C:\Windows/inf/setupapi.dev.20130730_200027.log"
[86]=>
string(31) "C:\Windows/inf/setupapi.dev.log"
[87]=>
string(33) "C:\Windows/inf/setupapi.setup.log"
[88]=>
string(37) "C:\Windows/security/logs/scesetup.log"
[89]=>
string(27) "C:\Windows/security/edb.log"
[90]=>
string(25) "C:\Windows/DtcInstall.log"
[91]=>
string(23) "C:\Windows/KB835221.log"
[92]=>
string(29) "C:\Windows/MSI30-KB884016.log"
[93]=>
string(19) "C:\Windows/PFR0.log"
[94]=>
string(28) "C:\Windows/WindowsUpdate.log"
[95]=>
string(22) "C:\Windows/chipset.log"
[96]=>
string(18) "C:\Windows/iis.log"
[97]=>
string(27) "C:\Windows/vmgcoinstall.log"
>
Warning: md5_file(res:///PHP/C:\Windows\System32\catroot2\edb.log): failed to open stream: Permission denied in scanner_version_3.php on line 1
G:>
```

Fig. 36 The results after running the scanner_version_3.exe program in scan mode.

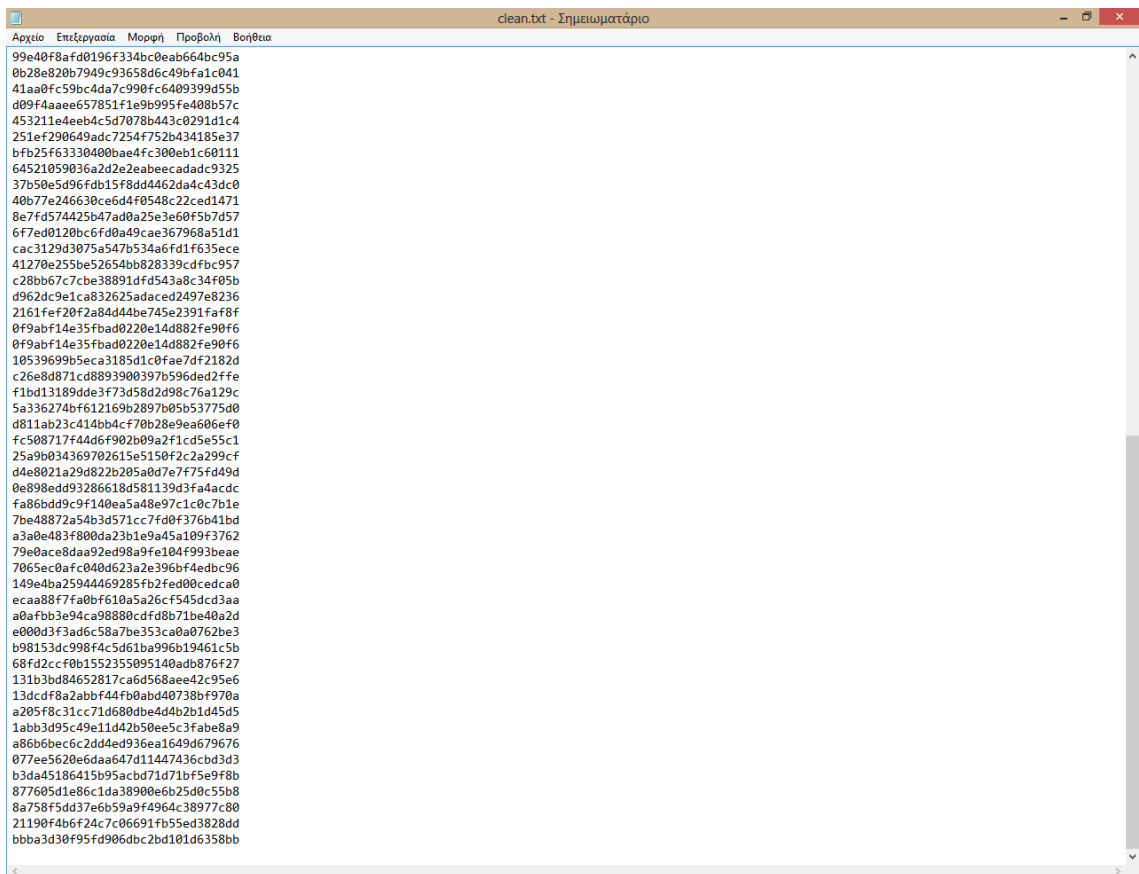


Fig. 37 The clean.txt file after running the scanner_version_3.exe program in scan mode.

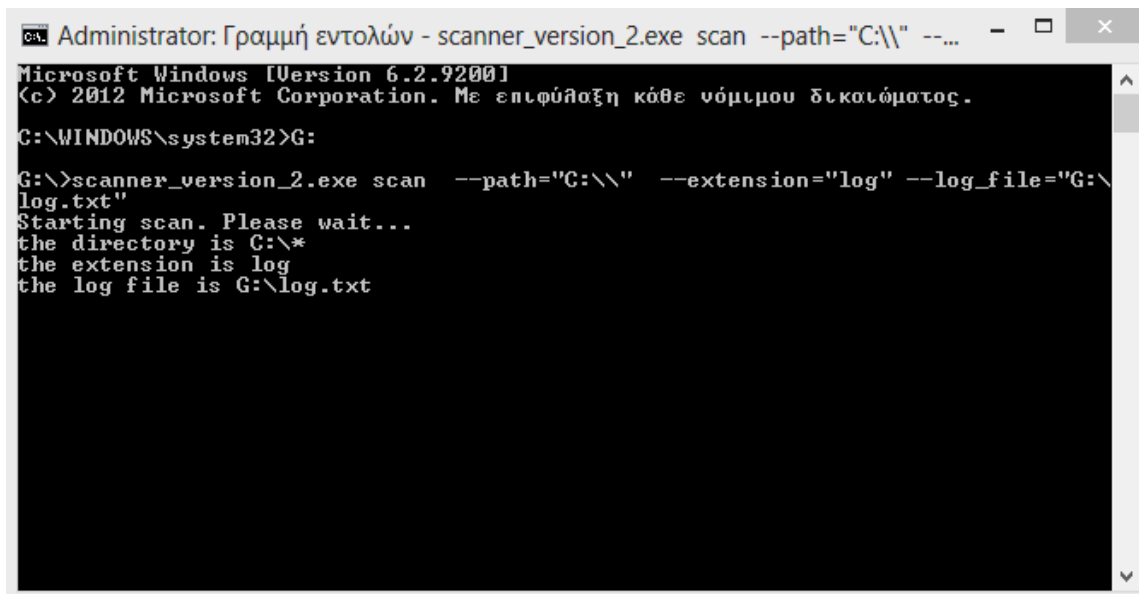


Fig. 38 Creating the log.txt file by using the scanner_version_2.exe program in scan mode.


```
Administrator: Γραμμή εντολών
string(28) "C:\Windows/Temp/MpCmdRun.log"
[218]=>
string(32) "C:\Windows/Temp/Silverlight0.log"
[219]=>
string(34) "C:\Windows/Temp/SilverlightMSI.log"
[220]=>
string(36) "C:\Windows/Temp/chrome_installer.log"
[221]=>
string(26) "C:\Windows/Temp/vminst.log"
[222]=>
string(28) "C:\Windows/Temp/winstore.log"
[223]=>
string(29) "C:\Windows/WinSxS/poqexec.log"
[224]=>
string(33) "C:\Windows/debug/WIA/wiatraces.log"
[225]=>
string(24) "C:\Windows/debug/mrt.log"
[226]=>
string(27) "C:\Windows/debug/mrteng.log"
[227]=>
string(29) "C:\Windows/debug/netlogon.log"
[228]=>
string(27) "C:\Windows/debug/sammui.log"
[229]=>
string(31) "C:\Windows/inf/setupapi.app.log"
[230]=>
string(47) "C:\Windows/inf/setupapi.dev.20130730_200027.log"
[231]=>
string(31) "C:\Windows/inf/setupapi.dev.log"
[232]=>
string(33) "C:\Windows/inf/setupapi.setup.log"
[233]=>
string(37) "C:\Windows/security/logs/scesetup.log"
[234]=>
string(27) "C:\Windows/security/edb.log"
[235]=>
string(46) "C:\inetpub/logs/LogFiles/W3SVC1/u_ex130221.log"
[236]=>
string(46) "C:\inetpub/logs/LogFiles/W3SVC1/u_ex130418.log"
[237]=>
string(46) "C:\inetpub/logs/LogFiles/W3SVC1/u_ex130422.log"
[238]=>
string(46) "C:\inetpub/logs/LogFiles/W3SVC1/u_ex130423.log"
[239]=>
string(46) "C:\inetpub/logs/LogFiles/W3SVC1/u_ex130424.log"
[240]=>
string(46) "C:\inetpub/logs/LogFiles/W3SVC1/u_ex130425.log"
[241]=>
string(47) "C:\wamp/bin/apache/Apache2.4.4/logs/install.log"
[242]=>
string(22) "C:\wamp/logs/mysql.log"
[243]=>
string(41) "C:\System Volume Information/tracking.log"
[244]=>
string(25) "C:\Windows/DtcInstall.log"
[245]=>
string(23) "C:\Windows/KB835221.log"
[246]=>
string(29) "C:\Windows/MSI30-KB884016.log"
[247]=>
string(19) "C:\Windows/PFRO.log"
[248]=>
string(28) "C:\Windows/WindowsUpdate.log"
[249]=>
string(22) "C:\Windows/chipset.log"
[250]=>
string(18) "C:\Windows/iis.log"
[251]=>
string(27) "C:\Windows/vmgcoinstall.log"
}
Warning: md5_file(res:///PHP/C:\Users\nikos\AppData\Local\Microsoft\Windows\WebCache\U01.log): failed to open stream: Permission denied in scanner_version_2.php on line 1
Warning: md5_file(res:///PHP/C:\Windows\System32\catroot2\edb.log): failed to open stream: Permission denied in scanner_version_2.php on line 1
G:\>
```

Fig. 39 The results after running the scanner_version_2.exe program in scan mode.

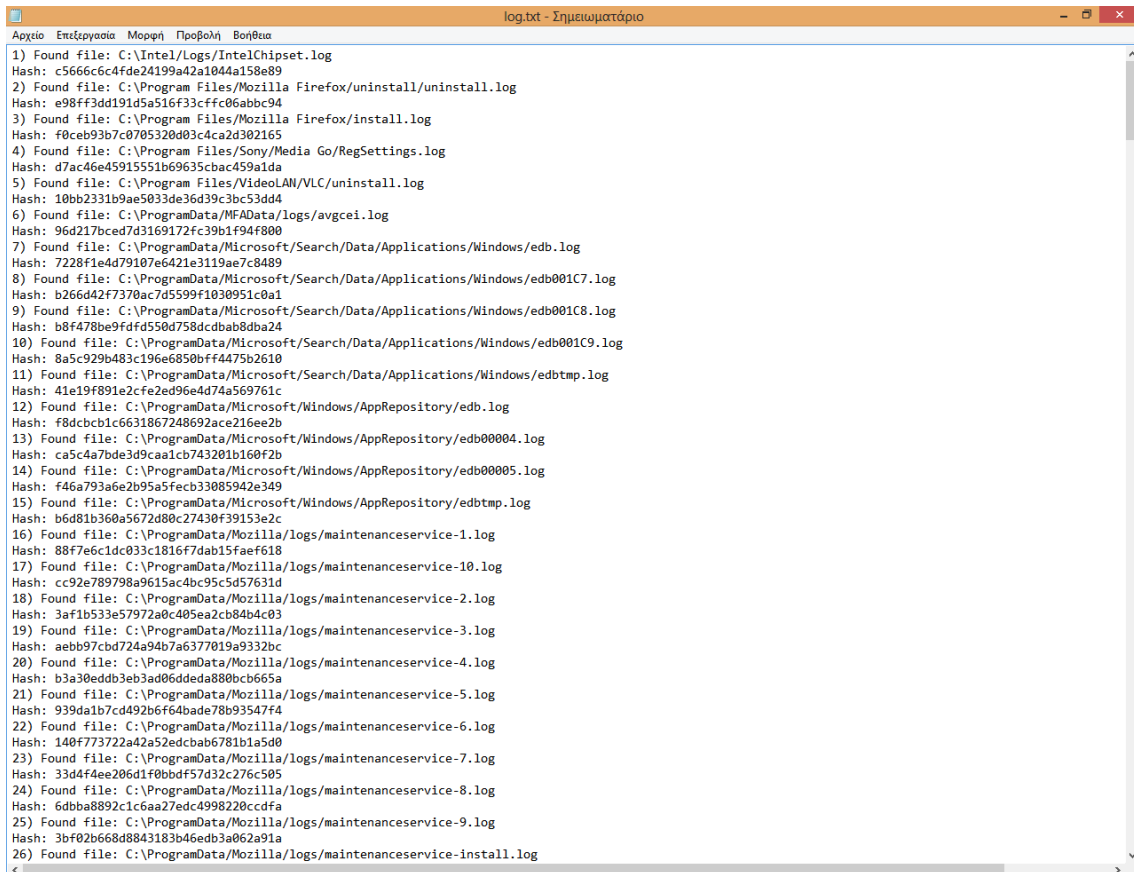


Fig. 40 The log.txt file after running the scanner_version_2.exe program in scan mode.

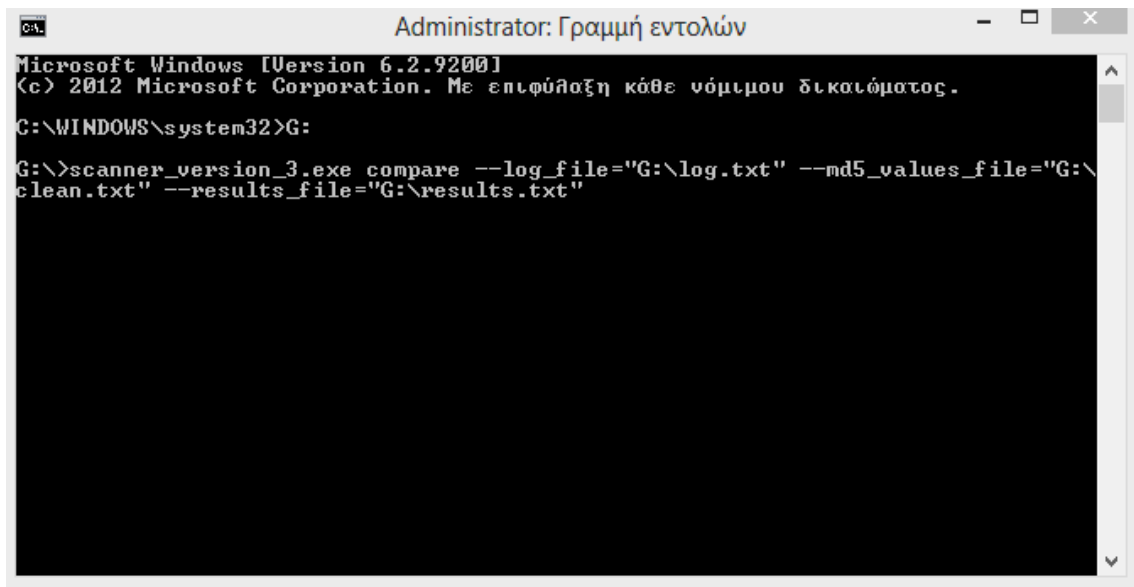


Fig. 41 Creating the results.txt file with the results by using the scanner_version_3.exe program in compare mode.

```

Administrator: Γραμμή εντολών
Suspicious file found -> C:\Users\nikos\AppData\Roaming\Sony\Media Go Installer\msi.log
Suspicious file found -> C:\Users\nikos\Desktop\Μεταφραστής\TOOLS\super kludge\Wangdera.Controls-0.9.1.0-src/docs/Documentation.log
Suspicious file found -> C:\Users\nikos\Documents\Virtual Machines\Windows XP Professional\vmware-0.log
Suspicious file found -> C:\Users\nikos\Documents\Virtual Machines\Windows XP Professional\vmware.log
Suspicious file found -> C:\Users\nikos\Documents\Virtual Machines\Windows XP Professional\yprintproxy.log
Suspicious file found -> C:\Windows\Logs\CBS\CBS.log
Suspicious file found -> C:\Windows\ServiceProfiles\LocalService\AppData\Local\Temp\MpCmdRun.log
Suspicious file found -> C:\Windows\SoftwareDistribution\DataStore\Logs\edb.log
Suspicious file found -> C:\Windows\System32\catroot2\edb.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgadvisor.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgcfg.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgchjw.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgcore.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgcsl.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgdiagex.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgdiskdrv.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgemc.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgidpagent.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgidpdrv.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgidpeh.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgldr.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgmf.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgmsgdisp.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgns.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgrs.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgsched.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgsecapi.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgshred.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgtdi.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgwd.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\avgwdsvc.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\commonpriv.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\fixcfg.log
Suspicious file found -> C:\Windows\System32\config\systemprofile\AppData\Local\Aug2014\log\lng.log
Suspicious file found -> C:\Windows\System32\sru\SRU.log
Suspicious file found -> C:\Windows\System32\sru\SRU00506.log
Suspicious file found -> C:\inetpub\logs\LogFiles\W3SUC1\u_ex130221.log
Suspicious file found -> C:\inetpub\logs\LogFiles\W3SUC1\u_ex130418.log
Suspicious file found -> C:\inetpub\logs\LogFiles\W3SUC1\u_ex130422.log
Suspicious file found -> C:\inetpub\logs\LogFiles\W3SUC1\u_ex130423.log
Suspicious file found -> C:\inetpub\logs\LogFiles\W3SUC1\u_ex130424.log
Suspicious file found -> C:\inetpub\logs\LogFiles\W3SUC1\u_ex130425.log
Suspicious file found -> C:\wamp\bin\apache\Apache2.4.4\logs\install.log
Suspicious file found -> C:\wamp\logs\mysql.log
Suspicious file found -> C:\System Volume Information\tracking.log
Suspicious file found -> C:\Windows\WindowsUpdate.log

```

Fig. 42 The results after running the scanner_version_3.exe program in compare mode.

Fig. 43 The results.txt file with the results after using the scanner_version_3.exe program in compare mode (with the process mentioned above).

As we can confirm, the results.txt file contains all the *.log files of the whole partition C, except the *.log files that are included in the Windows 8 installation file.

4.7 Advantages

The scanner.php program is a simple and functional tool. It is written in a well known language (PHP) with million users around the world, with numerous applications and a wide variety of implementations. There are many plug ins that can be attached to this program. In addition this program can be easily incorporated into other programs. It is easy to carry, because of its minimum size. Also it is user friendly because it is easy to run even from someone with little knowledge and little expertise. The parameters are no longer hard coded and they are easy to be inserted or changed from

the command line prompt. It is very fast, reliable and trustworthy because of its simplicity and the small number of code lines.

4.8 Drawbacks

The program is very simple. So it has only a small number of functions, options, commands and parameters. It can be used only by using the command line prompt. A web interface could make it more user friendly and more attractive for the simple user.

4.9 Evaluation

The program was tested in all possible scenarios that could be encountered by a digital forensics investigator or by a simple user. In all conditions the program worked as expected and the results were correct.

4.10 Suggestions

There are some things that must be taken into consideration before using the program. First of all the investigator must have knowledge, of what he is searching for, inside the computer under investigation. If the program is used for digital triage, the time is limited. If the program is used in the examination laboratory, there is plenty of time to do many tests and search for a wide variety of traces, depending always on the case under investigation.

Additionally the user of the program must create a wide range of correct clean MD5 files. In this way he will be able to use the correct files in every specific and different case. He must create files with clean MD5 values of all the windows versions and releases installations, Linux, Ubuntu and other OS installations, clean installations of programs that are widely used, a wide number of software installations, browser installations and many more. So in each case he will be able to make the comparison using

the correct pre constructed clean MD5 files, and therefore produce solid and helpful findings. In addition the user must always have updated clean MD5 files from all the new releases and updated programs. Nowadays the Operating Systems whether it is Windows or any other, they update very frequently and automatically. Also all the programs in the modern computers from the antivirus programs to media players update all the time, so there is a need to create the clean MD5 files from all the up to date versions of these programs.

Furthermore this program has a certain functionality and the capabilities are constrained and predefined. The investigator and user of this program in a real scenario cannot depend only on this tool. Digital triage and digital investigation are very wide and complex fields. In order to achieve the necessary findings many tools must be applied. As mentioned in the chapters above there is a great number of tools with different potentials, different targets, different use, with case oriented applications producing different results and findings. The investigator must be very well informed and prepared in the digital triage and digital investigation field in order to achieve the required results and eventually success. A wide range portfolio of programs is unquestionable. A true necessity in every investigation.

4.11 Future Work – Enhancements

In the future many things can be done to improve the program and its functionality. First of all a web interface could be developed to make it more attractive and user friendly.

Functionalities can be added, including the ability to search for more difficult files and do a wide variety of comparisons. An upgrade would be to have libraries with clean MD5 values, from various program installations, and all the possible Windows installations, and be able to choose each time the files we want to compare. Furthermore, when running, the program could find automatically the versions of installations and make the comparisons without any help or any intervention from the user.

The results could be presented in a web environment using statistics, diagrams, bars, pies and percentages for the reports. This could help the user take fast decisions, come quickly with the correct conclusions and decide in which way to proceed.

The program as mentioned above can be incorporated in other digital triage and digital investigation tools to enhance their functionality and usability. Also other tools and programs could be added in this program to make it more complex and become useful in more difficult cases. The simplicity of the program and the PHP language make all the above an easy task for the programmer. In addition it is easy to transform this program in other languages like C, C++, Java, VBscript, Python and many more.

This program could also be used for Data Reduction. It could be implemented like the FTK (Forensic Toolkit). By using the Known File Filter, or KFF, we can eliminate or highlight known files, by using MD5 hashes that are generated by the user or from NIST or Hashkeeper. [29] [45] [56] [42] These KFF lists are MD5 values from files that we don't care to examine and we want to avoid analyze them. Also they are files from similar, previews, or other suspicious real cases.

Further more, plug-ins and extensions could be found to enhance the functionality of the program. Many tools like this could be integrated to create a bigger and better tool. The new tool, after the integration, would have none of the drawbacks but all the advantages of each individual tool.

4.12 Conclusion

Digital Forensics is a very demanding and exciting sector but it has many difficulties. There are many types of data to be analyzed, the data size is enormous and there is lack of skilled analysts. The above problems will get worse in the future, and the only way to keep up is to develop new techniques that will enhance our ability to collect, maintain and analyze big data sets with information. [32]

5 Personal Reflection

From my personal experience, while experimenting with the four forensics tools Bulk Extractor, TriageIR, TR3Secure and Kludge I encountered all their problems and disadvantages. First of all I installed these programs in 6 different home desktop computers with different versions of Windows operating systems (Windows XP, Vista, 7, 8, 32 and 64 bit). The installation of the tools may look simple to the medium user but when it comes to start using them, then the problems start. Most of the times they don't even work and they don't create any reports. When running they display error messages and when they work they don't produce all the reports they claim to be able to produce. Some reports are created and some other not, depending each time on the operating system. The language of the OS affected their compatibility and implementation. So it was obvious that they were OS sensitive. In Windows 8, the most recent of the OS, the problems were even worse. Probably new releases of the tools should be developed. In the internet there is not enough information by the developers or the users to bypass all the problems encountered, that could help and assist in making them work. The collection of browsing history, which was the initial target, by using Kludge, was never produced in any computer tested. I tried many solutions, personal and suggested by the internet, but with no luck. My personal opinion is that these programs address to the expert user that knows exactly what he wants and how to achieve it. They are not very useful for the beginner and the medium user. Anyone choosing them would have problems in using these tools. And in a real case scenario these problems and malfunctions would be at least catastrophic for triage and analysis. My research showed that the most digital triage tools have numerous problems and drawbacks and this is expected when considering the nature of the investigation. Also most of them are written in simple DOS batch scripting, a very poor programming practice. In addition most of them need improvements and enhancements.

All the above, lead me first to think, the development of a program that would collect all the browsing history from all the browsers installed in a computer. After reading

many articles and conducted extensive research I understood that all the new browsers (latest releases) were very complicated and needed special tools, specifically designed for each and every one of them, to be able to extract the browsing history and other useful information stored in the browsers. For example Mozilla Firefox needs the SQLite Manager tool. [83] Developing a tool that could be able to process all the available browsers in all their possible and available releases would be something really demanding, complicated and advanced. In a real case scenario having a tool to fetch the browsing history of a specific only browser would be of no use. The suspect has the ability to install many browsers in his personal computer and use any of them at will.

These findings lead me to the decision to develop my own program to help the digital forensics investigator in my way. To develop a program that could help him during the digital triage but also to be useful it in the lab as well. So the idea before starting developing my own program was to create a script that could find all the .exe files in a computer, to reject the default .exe files from Windows and present only the .exe files installed by the user. This could be achieved by finding all the .exe files installed in the computer, calculating their MD5 and then compare these MD5 values with an archive file / database with the produced MD5 values of a Windows installation only. For example notepad.exe in Windows 7 always gives the same MD5. So comparing this MD5 with a database of MD5 can be distinguished and be ignored during the presentation of the installed applications to the investigator. The general solution is easy, but there are many differentiations in each case that could make the whole process very complicated.

So a tool was developed that has the ability to search a whole computer or any partition or file chosen, for files and programs installed or created by the user and that may have any possible extension. This was possible by comparing the MD5 hashes of the files under investigation. In this way the examiner can search in a very short time the computer under examination for all the installed and created files or programs, but also for altered programs, possible malware and harmful programs.

This program will gain even greater usability if it is incorporated into other digital triage programs or if it is enhanced with more advanced functionality.

I hope that this paper will help others to learn not only from my paper research but also from the program I developed, the presented personal experiences, and any mistakes that were made along the way. Writing tools that will be used for digital forensics is very difficult and very different from other programming attempts because of the

multiple data types that need to be taken into consideration, the desire for high performance, the advanced skills needed from users, and the requirement the software to run without ever crashing. Since the digital forensics field grows continuously, many people are engaged in the practice of writing software for digital forensics. Only a small number of today's forensic tool developers have adequate knowledge and training in developing and design software. A number of them do not even think themselves of being programmers. [32] And this belief must change.

6 Conclusions

Appreciating the change of technology, and understanding the nature of the threat, the evolving discipline of anti-forensics and increasing application of cryptography, we understand that the domain of forensics has an extremely challenging and exciting future ahead of it. However, the need for organizations to equip themselves with a forensic capability is becoming essential in order to combat and manage incidents effectively. [55]

To appreciate the necessity of computer forensics in the organization, first we have to understand the scale and nature of the threats. Unfortunately, truly understanding the scale of the threat is difficult as the reporting of cybercrime is relatively patchy. Many organizations see such reporting as something that will affect their brand image and reputation. Whilst discussions are being held in some countries about implementing laws to force organizations into reporting incidents, at this stage the industry relies upon survey statistics to appreciate the threat. [55]

It is no longer enough to pull the plug and take the computer to the lab when making a digital examination. Technology changes continuously and digital forensic analysts have to learn new methods and develop tools to succeed. This is necessary in a live response case. [93]

Forensic analysis help to identify privacy issues, detect forgery and manipulation, establish a chain of custody for sources and employ write protection for capture or transfer. It can find content and metadata, help indexing and searching by examiners, and enable audit control. [48]

6.1 Digital forensics is different

Digital Forensics software development makes it really different from other branches. These differences are: data diversity, data scale, temporal diversity, human capital, and the so-called “CSI effect.” [32]

6.1.1 The challenge of data diversity

Digital Forensics and other kinds of software have a big difference in the range of data that have to be examined. Most software development is specialized in a specific problem. Digital Forensics has as target all the data that can be stored or transmitted using computer and digital media. In addition most software tools work only with proper input and crash otherwise. Digital Forensics tools don't have this luxury. They must run at any circumstances and produce any results possible. [32]

6.1.2 Data scale

A second problem in the development of digital forensics tools is the huge bulk of information that has to be analyzed. Furthermore there is the problem of the difference between the storage and performance bottlenecks.

Examiners have to analyze new and up to date computer systems. So they are using high end computers to examine high end computers and they also have to examine information in hours that the suspect needed weeks, months, or even years to create. We will never overcome the performance lack. When the analysis will move to the cloud we will have to examine multi-terabyte data caused by the cloud-based crime. [32]

6.1.3 Temporal diversity: the never-ending upgrade cycle

Most organizations believe that upgrades in software are a difficult process that can cause problems and incompatibilities. So they use out-of-date operating systems and update only when they buy the new hardware.

Digital forensics examiners do not have this flexibility. They have to update the software all the time because the target is not only the obsolete but also the newest software and hardware possible. Nowhere else it is so important to upgrade the software when the new software is released. In upgrades, two things have to be taken into account. The examiners tool's version and the target's version [32]

6.1.4 Human capital demands and limitations

Digital forensics tools users are from law enforcement, with little knowledge in computer science. They have strict deadlines and are exhausted. Certifications and degrees certainly help this situation but cannot solve the problem. Analysis may have as target any possible kind of information. So many organizations train their own developers to create the needed software. [32]

6.1.5 The CSI effect

CSI Effect is the belief, that when television shows crime scenes, investigations, forensics, courts, juries, judges and prosecutors to have exceptionally high demanding concerning what forensic analysis can actually achieve, this is also true in reality. On television every digital forensics examiner knows every tool, correlation is easy and instantaneous, there are never false positives, overwritten data can easily be recovered, encryption can be cracked, it is impossible to delete anything and the tools never crash. In reality things are not so easy. Overwritten data cannot be recovered and modern encryption algorithms can be decrypted only by using password cracking. [32]

6.1.6 The cost of development and the role of government

Digital forensics tool development is exceptionally expensive and the software produced has small number of users. The more sophisticated the analysis, the smaller the market. Few digital forensics companies have been commercially successful. It is not that digital forensics is an immature market with customers only from the government; it is that digital forensics is a mature market with high and increasing development costs. These high development costs in addition to customers being federal, state and local governments makes it difficult for the traditional commercial software development model to be successfully applicable. [32]

6.2 Lessons learned developing digital forensics tools

In this chapter we discuss about software engineering and design issues that have been confronted when developing the above mentioned digital forensics tools. [32]

6.2.1 Platform and language

Windows is the most widespread operating system used by computer forensics developers and examiners. But also Linux and MacOS seem to consolidate their places. An easy way to write multi-platform forensics tools is to use C, C++, C#, Java or Python because tools in these languages can be easily transferred between these three platforms. C was historically the preferred developer's language but nowadays many have shifted to C++. Many believed that Java was running much slower than C/C++. But the testing so far shows that this belief is only partially true.

Writing programs in Python is quite easy but the experience so far shows that these programs are slow and intensive for memory. [32]

6.2.2 Parallelism and high performance computing

The data scale problem led many researchers to spend lots of effort and time on issues such as multithreading and high performance computing in an effort to gain more performance. But until now the efforts made are mixed and the results produced not so clear. [32]

6.2.3 All-in-one tools vs. single-use tools

There are many kinds of forensic investigations and the same tool many times has to be used at the same data but for different reasons. This difference in the use cases com-

plicates the programs development, documentation, and training. Some say it is better to have a single tool than many because:

- If there are many tools, the investigators will try to have them all.
- Whatever a digital forensics program does: decoding and enumerating data, data ingest; preparing a report is necessary whatever the expected results may be.
- There is a standard cost to packaging, distributing, and promoting a program. When a tool has many functions the cost is distributed to a broader base.

An opinion to solve the problem of different use cases is the programs to organize the results into different partitions or files. For example using one section storing information needed for usual cases, and another saving all the extracted data. [32]

6.2.4 Evidence container file formats

The diversity of programs and the lack of proper user training make it necessary for forensic software to be able to process inputs at any format. In real cases a single input layer should enable programs to transparently manage disk images in raw, split-raw, EnCase or AFF formats. [32]

6.3 Conclusion

This paper makes contributions not only to the theory but also the practice of digital forensics. First, it shows that all the existing freeware and licensed forensics tools have many drawbacks and incompatibilities. This paper presents Bulk Extractor, TriageIR, TR3Secure and Kludge, powerful freeware tools to perform bulk data analysis. It highlights the experience of installing and using these tools noticing all the advantages, drawbacks, enhancements, suggestions and future work. It presents a detailed evaluation comparing Bulk Extractor, TriageIR, TR3Secure and Kludge. It shows that each tool has a strong and a weak point and there is no solution for every case that an examiner may encounter.

Finally, this paper presents a tool, developed by the author, which can search a whole computer or any partition or file chosen, for files and programs installed or creat-

ed by the user and that may have any possible extension. This was possible by comparing the MD5 hashes of the files. In this way the examiner can search in a very short time the computer under examination for all the installed and created files or programs, altered programs, but also for possible malware and harmful programs.

This program will have even greater usability if it is incorporated into other digital triage programs or if it is enhanced with more advanced functionality.

References

- [1] ACPO. (2008) *Good practice guide for computer-based electronic evidence*. ACPO, United Kingdom.
- [2] Adams, R. (2012) The Advanced Data Acquisition Model (ADAM): A process model for digital forensic practice. *National Library of Australia*, Murdoch University, Australia.
- [3] Adelstein, F., and Golden, G. (2008) Live Forensics Tutorial, Part 1 Traditional Forensics. *Usenix Annual Technical Conference 2008*, Boston, Massachusetts, USA.
- [4] Aljaedi, A., Lindskog, D., Zavorsky, P., Ruhl, R., and Almari, F. (2011) Comparative analysis of volatile memory forensics: live response vs. memory imaging. *Proceedings of the 2011 IEEE third international conference on Social Computing (SocialCom)*, Boston, MA, USA, pp.1253–1258.
- [5] Bamcompile (2013) <http://www.bambalam.se/bamcompile/>.
- [6] Bednar, P., Katos, V., and Hennell, C. (2008) Cyber-Crime Investigations: Complex Collaborative Decision Making. *Third International Annual Workshop on Digital Forensics and Incident Analysis, 2008 IEEE*.
- [7] Brezinski, D., and Killalea, T. (2002) Guidelines for evidence collection and archiving. *The Internet Society*.
- [8] Broadhurst, R. (2006) Developments in the Global Law Enforcement of Cyber-Crime. *Policing: An International Journal of Police Strategies & Management*, vol. 29, no.3, *Emerald*, pp. 408- 433.
- [9] Brownlee, N., Guttman, E., (1998) Expectations for computer security incident response. *The Internet Society*.
- [10] Bulk Extractor (2013) http://www.securitytube-tools.net/index.php@title=Bulk_Extractor.html.
- [11] Bulk Extractor (2013) http://www.forensicswiki.org/wiki/Bulk_extractor.
- [12] Bulk Extractor (2013) http://digitalcorpora.org/downloads/bulk_extractor/.

- [13] Bulk Extractor (2013) http://wiki.bitcurator.net/index.php?title=Using_bulk_extractor_to_Find_Personal_Identifiable_Information_%28PII%29_on_a_Disk_Image.
- [14] Bunting, S. (2008) *EnCase Computer Forensics, The Official EnCE : EnCase Certified Examiner, Study Guide, Second Edition*. Wiley Publishing, Inc, Indianapolis, Indiana, USA.
- [15] Buster Sandbox Analyzer (BSA) (2013) <http://bsa.isoftware.nl/>.
- [16] Cantrell, G., Dampier, D., Yoginder, S., Dandass, N., and Bogen, C. (2012) Research toward a Partially-Automated, and Crime Specific Digital Triage Process Model. *Computer and Information Science*, Vol. 5, No. 2, March.
- [17] Carrier, B. (2006) Basic Digital Forensic Investigation Concepts. <http://www.digital-evidence.org/>.
- [18] Carrier, B. (2001) Defining digital forensic examination and analysis tools. *International Journal of Digital Evidence, Digital Research Workshop II*.
- [19] Carvey, H. (2012) *Windows Forensic Analysis Toolkit, Advanced Analysis Techniques for Windows 7*. Syngress, Elsevier, USA.
- [20] Casey, E. (2009) *Handbook of Digital Forensics and Investigation*. Academic Press.
- [21] Computer forensics (2013) https://en.wikipedia.org/wiki/Computer_forensics.
- [22] Computer forensics (2013) <http://computer-forensics.safemode.org/>.
- [23] Dickson, M. (2010) *Digital Triage*. Institute for Advanced Studies, SCDEA (Scottish Crime and Drug Enforcement Agency).
- [24] Digital forensics (2013) http://en.wikipedia.org/wiki/Digital_forensics.
- [25] DumpIt memory utility (2013) <http://www.moonsofs.com/windows-memorytoolkit/>.
- [26] Dunbar, B. (2001) A detailed look at Steganographic: Techniques and their use in an Open-Systems Environment. *SANS Institute InfoSec Reading Room*.
- [27] Eoghan, C. (2004) *Digital Evidence and Computer Crime, Second Edition*. Academic Press, Elsevier.
- [28] Forensic Artifacts (2013) <http://forensicartifacts.com/>.
- [29] Forensic Toolkit (2013) http://www.forensicswiki.org/wiki/Forensic_Toolkit.
- [30] Garfinkel, S. (2013) *Digital media triage with bulk data analysis and bulk extractor*. *Computers and Security* 32, pp. 56-72.

- [31] Garfinkel, S. (2012) *Using bulk extractor for digital forensics triage and cross-drive analysis*. Navy's Research University, Monterey, California, USA.
- [32] Garfinkel, S. (2012) Lessons learned writing digital forensics tools and managing a 30TB digital evidence corpus. *Digital Investigation 9* S80–S89, Elsevier.
- [33] Garfinkel, S., Nelson, A., White, D., and Rousev, V. (2010) Using purpose-built functions and block hashes to enable small block and sub-file forensics. *Proceedings of the tenth annual DFRWS conference*, Elsevier, Portland.
- [34] Garfinkel, S. (2007) Carving contiguous and fragmented files with fast object validation. *Proceedings of the 7th annual Digital Forensic Research Workshop (DFRWS)*, vol. 4, Elsevier, Pittsburgh, pp. 2 – 12.
- [35] Garfinkel, S. (2006) Forensic Feature Extraction and Cross-Drive Analysis. *Digital Investigation 3S* S71–S81, Harvard University, Cambridge, USA.
- [36] Geiger, M. (2005) Evaluating Commercial Counter-Forensic Tools. *2005 Digital Forensic Research Workshop (DFRWS)*, New Orleans, USA.
- [37] Gomez, L.S.M. (2012) *Triage in-Lab: case backlog reduction with forensic digital profiling*. Simposio Argentino de Informatica y Derecho, JAIIO - SID 2012, Argentina.
- [38] Grobler, C., Louwrens, C., and Solms, S. (2010) *A framework to guide the implementation of Proactive Digital Forensics in organizations*. 2010 International Conference on Availability, Reliability and Security, Academy for Information Technology, University of Johannesburg, Johannesburg, South Africa.
- [39] Gunsch, G. (2002) An Examination of Digital Forensic Models. *International Journal of Digital Evidence*, Fall, Volume 1, Issue 3.
- [40] Haggerty, J., and Taylor, M. (2006) *Managing corporate computer forensics*. Computer Fraud & Security Magazine, June 2006, School of Computing and Mathematical Sciences, Liverpool, John Moores University, United Kingdom.
- [41] Halderman, A., Schoen, S., Heninger, N., Clarkson, W., Paul, W., Calandrino, J., Feldman, A., Appelbaum, J., and Felten, E. (2008) Lest We Remember: Cold Boot Attacks on Encryption Keys. *17th USENIX Security Symposium*, Princeton University, San Jose, California, USA.
- [42] Hashkeeper (2013) <http://www.forensicswiki.org/wiki/Hashkeeper>.
- [43] Hobocopy Utility (2013) <https://github.com/candera/hobocopy/downloads>.
- [44] Horsman, G., Laing, C., and Vickers, P. (2011) *A case based reasoning system for automated forensic examinations*. PGNET 2011, the 12th annual postgraduate symposi-

um on the convergence of telecommunications, networking and broadcasting, 27–28 June, 2011, Liverpool.

[45] KFF (2013) <http://www.forensicfocus.com/Forums/viewtopic/t=7307/>.

[46] Kludge-3.20110223 (2013) <http://theinterw3bs.com/?pO503>.

[47] Lee, R., (2012) Sans-Digital-Forensics-and-Incident-Response-Poster 2012. *Sans DFIR Faculty*.

[48] Leighton, J. (2012) *Digital Forensics and Preservation*. DPC Technology Watch Report, Digital Preservation Coalition, Great Britain.

[49] Leigland, R. (2004) A Formalization of Digital Forensics. *International Journal of Digital Evidence*, Fall, Volume 3, Issue 2, University of Idaho, USA.

[50] MD5 (2013) <http://www.forensicswiki.org/wiki/MD5>.

[51] Md5deep and Sha1deep utilities (2013) <http://md5deep.sourceforge.net/>.

[52] Merriam-Webster dictionary Free (2013) <http://www.merriam-webster.com/dictionary/>.

[53] Mislán, R., Casey, E., and Kessler, G. (2010) The growing need for on-scene triage of mobile devices. *Digital Investigation*, pp. 112–124.

[54] Mobile device forensics (2013) http://en.wikipedia.org/wiki/Mobile_device_forensics.

[55] Nathan, C. (2010) *Computer Forensics A Pocket Guide*. IT Governance Publishing, United Kingdom.

[56] National Software Reference Library (2013) http://www.forensicswiki.org/wiki/National_Software_Reference_Library.

[57] Nelson, B., Phillips, A., and Steuart, C. (2010) *Guide To Computer Forensics And Investigations, Third Edition*. Course technology, Cengage Learning.

[58] Netmarketshare. (2013) Market share statistics for internet technologies. Netmarketshare.

[59] Nirsoft web browsers tools package (2013) http://www.nirsoft.net/web_browser_tools.html.

[60] Noblett, M., Pollitt, M., and Presley, L. (2000) Recovering and examining computer forensic evidence. *Forensic Science Communications*, Volume 2, Number 4, FBI, USA.

[61] OPSWAT. (2012) Antivirus market analysis. San Francisco, USA.

[62] Patterson, D. (2004) Latency lags bandwidth. *Communications of the ACM*. Volume 47, Issue 10, October 2004, pp. 71-75, New York, USA.

- [63] Pearson, S., and Watson, R. (2010) *Digital Triage Forensics: processing the digital crime scene*. Syngress.
- [64] Phillip, A., Cowen, D., and Davis, C. (2009) *Hacking Exposed: Computer Forensics*. McGraw Hill Professional.
- [65] PHP (2013) <http://el.wikipedia.org/wiki/PHP>.
- [66] PHP (2013) <http://php.net/>.
- [67] Python (2013) <http://el.wikipedia.org/wiki/Python>.
- [68] Python (2013) <http://www.python.org/>.
- [69] RCFL (2013) RCFL annual reports FY2003 – FY2012, USA.
- [70] RegRipper (2013) <http://code.google.com/p/winforensicaanalysis/downloads/>list.
- [71] Reith, M., Carr, C., and Gunsch, G. (2002) An examination of digital forensic models. *International Journal of Digital Evidence*.
- [72] Resendez, I., Martinez, P., and Abraham, J. (2008) An Introduction to Digital Forensics, University of Texas Pan American.
- [73] Ricci, S.C. (2006) FORZA – Digital forensics investigation framework that incorporate legal issues. *Digital investigation* 3S S29 – S36.
- [74] Rogers, M.K., Goldman, J., Mislán, R., Wedge, T., and Debrotá, S. (2006) Computer forensics field triage process model. *Proceedings of the conference on Digital Forensics, Security and Law*, Las Vegas, Nevada, USA, pp. 27–40.
- [75] Roussev, V., Quates, C., and Martell, R. (2013) Real-time digital forensics and triage. *Digital Investigation*, University of New Orleans, USA.
- [76] Sandboxie (2013) <http://www.sandboxie.com>, <http://www.sandboxie.com/index.php?AllPages>.
- [77] Shiaeles, S., et al. (2013) On-scene triage open source forensic tool chests: Are they effective?. *Digital Investigation*.
- [78] Skoudis, E. (2006) *Windows Command-Line Kung Fu with WMIC*. SANS Technology Institute.
- [79] Sleuthkit (2013) <http://wiki.sleuthkit.org>.
- [80] Sliter, J. (2006) Organized Crime in Business. *Journal of Financial Crime*, vol. 13, no.4, *Emerald*, pp. 383-386.
- [81] Sommer, P. (2005) Directors and Corporate Advisors' Guide to Digital Investigations and Evidence. *Information Assurance Advisory Council*, United Kingdom.
- [82] SPEKTOR triage tool (2013) <http://www.evidencetalks.com/index.php?>

- optionOcom_content&viewOcategory&layoutOblog&idO83&ItemidO513.
- [83] SQLite Manager (2013) <https://addons.mozilla.org/en/firefox/addon/sqlite-manager/>.
- [84] Sublime Text Editor (2013) http://en.wikipedia.org/wiki/Sublime_Text.
- [85] Sublime Text Editor (2013) <http://www.sublimetext.com/>.
- [86] Suiche, M. (2008) Windows hibernation file for fun 'n' profit. *Black hat 2008*.
- [87] Sysinternals Suite (2013) <http://technet.microsoft.com/en-us/sysinternals/bb84206>.
- [88] The Flex Project (2013) <http://flex.sourceforge.net/>.
- [89] TR3Secure (2013) http://code.google.com/p/jiir-resources/downloads/detail?nameOtr3secure_data-collection-script.zip&canO2&qO.
- [90] TriageIR v.79 (2013) <http://code.google.com/p/triage-ir/downloads/list>.
- [91] VBScript (2013) <http://en.wikipedia.org/wiki/VBScript>.
- [92] VBScript (2013) <http://msdn.microsoft.com/en-us/library/t0aew7h6%28v=vs.84%29.aspx>.
- [93] Waits, C., Akinyele, J.A., Nolan, R., and Rogers, L. (2008) *Computer forensics: results of live response inquiry vs. memory image analysis*. CERT Digital Intelligence and Investigation Directorate (DIID), Software Engineering Institute, Carnegie Mellon University.
- [94] WAMP Server (2013) <http://en.wikipedia.org/wiki/WAMP>.
- [95] WAMP Server (2013) <http://www.wampserver.com/en/>.
- [96] Wikipedia - Memory Imaging (2013) http://www.forensicswiki.org/wiki/Tools:Memory_Imaging.
- [97] Windows XP Embedded (WinXPe) OS (2013) <http://www.microsoft.com/windowseembedded/en-us/develop/windows-xp-embedded-fordevelopers.aspx>.
- [98] Wolfe, P. (2007) EXP-SA: Prediction and Detection of Network Membership through Automated Hard Drive Analysis. *National Science Foundation*, Harvard University, Cambridge, USA.
- [99] Yasinsac, A., Erbacher, R., Marks, D., and Pollitt, M. (2003) Computer forensics education. *IEEE Security & Privacy*.
- [100] 7Zip Command Line (2013) <http://www.7-zip.org/>.

Appendix

1) The program scanner.php in the PHP programming language:

```
<?php

error_reporting(E_ERROR | E_WARNING | E_PARSE);

//solution to return *.* files

$array = array();
function recursiveGlob($dir, $ext) {
    global $array;
    $globFiles = glob("$dir/*.${ext}");
    $globDirs = glob("$dir/*", GLOB_ONLYDIR);

    foreach ($globDirs as $dir) {
        recursiveGlob($dir, $ext);
    }

    foreach ($globFiles as $file) {
        if(!in_array($file,$array)) {
            array_push($array,$file);
        }
    }
    return $array;
}
```

//only use in CLI mode because of extra large buffer and execution time

```
if ($argv[1] == "help" || $argv[1] == null) {
    echo "-- SCAN MODE --\r\n";
    echo "Usage: php scanner.php scan --path=path_to_check --
extension=extension_to_filter --log_file=path_and_name_of_log_file \r\n";
    echo "\r\n";
    echo "Example: php scanner.php scan --path=\"C:\myfolder\\" --extension=\"exe\" --
log_file=\"C:\mydocuments\md5_log.txt\" \r\n";
    echo "\r\n";
    echo "Default values: --path=\"C:\Windows\" --extension=\"exe\" --
log_file=\"C:\\logfile\md5_log_file.txt\" \r\n";
    echo "\r\n";
    echo "-- COMPARE MODE --\r\n";
    echo "Usage: php scanner.php compare --log_file=path_and_name_of_log_file --
md5_values_file=path_and_name_of_md5_values_file --
results_file=path_of_results_file \r\n";
    echo "\r\n";
    echo "Example: php scanner.php compare --log_file=\"C:\\log_file_after_scan.txt\" --
md5_values_file=\"C:\\file_with_clean_md5_values.txt --
results_file=\"C:\\file_with_results.txt \r\n";
    echo "\r\n";
    die();
}

if ($argv[1] == "debug") {

    for ($i=2; $i < $argc; $i++) {
        $tempvar = explode("=", $argv[$i]);
```

```

        echo $tempvar[1]."\r\n";
    }
}

if ($argv[1] == "scan") {
    echo "Starting scan. Please wait...\r\n";
    for ($i=2; $i < $argc ; $i++) {
        $tempvar = explode("=", $argv[$i]);
        switch ($tempvar[0]) {
            case '--path':
                $directory = $tempvar[1]."*";
                break;
            case '--extension':
                $extension = $tempvar[1];
                break;
            case '--log_file':
                $log_file = $tempvar[1];
                break;
        }
    }
    if ($argc < 5) {
        if (!$directory) {
            $directory = "C:\Windows\\".*";
        }
        if (!$extension) {
            $extension = "exe";
        }
        if (!$log_file) {
            $log_file = "C:\\logfile\md5_log_file.txt";
        }
    }
}

```

```

    }
}

echo "the directory is ".$directory."\r\n";
echo "the extension is ".$extension."\r\n";
echo "the log file is ".$log_file."\r\n";

$files = recursiveGlob($directory,$extension);
$fh = fopen($log_file, "w") or die("can't open file \r\n");
var_dump($files);
$i = 1;
foreach ($files as $file) {
    $stringData = $i.") Found file: ".$file."\r\n";
$stringData .= "Hash: ".md5_file($file).\r\n";
    $i++;
fwrite($fh, $stringData);
}

fclose($fh);

}

if ($argv[1] == "compare") {
    echo "Starting compare. Please wait...\r\n";
    for ($i=2; $i < $argc ; $i++) {
        $tempvar = explode("=", $argv[$i]);
        switch ($tempvar[0]) {
            case '--md5_values_file':
                $md5_values_file = $tempvar[1];

```

```

        break;

        case '--log_file':

            $log_file = $tempvar[1];

            break;

        case '--results_file':

            $results_file = $tempvar[1];

            break;

        default:

            echo "error: loop defaulted\r\n";

            break;

    }

}

if ($argc < 4) {

    echo "Wrong count of parameters. Please type php scanner.php help for more in-
fo";

    die();

}

$scanned      =      file($log_file,      FILE_IGNORE_NEW_LINES      |
FILE_SKIP_EMPTY_LINES);

$clean_md5     =     file($md5_values_file,  FILE_IGNORE_NEW_LINES     |
FILE_SKIP_EMPTY_LINES);

$clean_md5_size = count($clean_md5);

$i = 1;

$fh = fopen($results_file, "w") or die("can't open file \r\n");

foreach ($scanned as $line_num => $line) {

    $check = explode(" ", $line);

    if ($check[0] == "Hash:") {

```

```

foreach ($clean_md5 as $clean_line_num => $clean_line) {
    $md5_check = explode(" ", $clean_line);
    if ($md5_check[0] == "Hash:") {
        if ($check[1] == $md5_check[1]) {
            break;
        }
        if (($clean_line_num + 1) == $clean_md5_size) {
            echo "Suspicious file found -> ".$pathname[1]."\r\n";
            $stringData = $i.") Suspicious file found -> ".$pathname[1]."\r\n";
            $i++;
            fwrite($fh, $stringData);
        }
    }
}
} else {
    $pathname = explode(":", $line);
}
}
fclose($fh);
}

?>

```

2) The program scanner_version_2.php in the PHP programming language:

```
<?php
```

```
error_reporting(E_ERROR | E_WARNING | E_PARSE);
```

```
//solution to return *.* files
```

```
$array = array();  
function recursiveGlob($dir, $ext) {  
    global $array;  
    $globFiles = glob("$dir/*.${ext}");  
    $globDirs = glob("$dir/*", GLOB_ONLYDIR);  
  
    foreach ($globDirs as $dir) {  
        recursiveGlob($dir, $ext);  
    }  
  
    foreach ($globFiles as $file) {  
        if(!in_array($file,$array)) {  
            array_push($array,$file);  
        }  
    }  
    return $array;  
}
```

```
//only use in CLI mode because of extra large buffer and execution time
```

```
if ($argv[1] == "help" || $argv[1] == null) {  
    echo "-- SCAN MODE --\r\n";  
    echo "Usage: php scanner.php scan --path=path_to_check --  
extension=extension_to_filter --log_file=path_and_name_of_log_file \r\n";  
    echo "\r\n";  
    echo "Example: php scanner.php scan --path=\"C:\myfolder\\" --extension=\"exe\" --  
log_file=\"C:\mydocuments\md5_log.txt\" \r\n";  
}
```

```

    echo "\r\n";

    echo "Default values: --path=\"C:\Windows\" --extension=\"exe\" --
log_file=\"C:\\logfile\md5_log_file.txt\" \r\n";

    echo "\r\n";

    echo "-- COMPARE MODE --\r\n";

    echo "Usage: php scanner.php compare --log_file=path_and_name_of_log_file --
md5_values_file=path_and_name_of_md5_values_file --
results_file=path_of_results_file \r\n";

    echo "\r\n";

    echo "Example: php scanner.php compare --log_file=\"C:\\log_file_after_scan.txt\" --
md5_values_file=\"C:\\file_with_clean_md5_values.txt --
results_file=\"C:\\file_with_results.txt \r\n";

    echo "\r\n";

    die();
}

if ($argv[1] == "debug") {

    for ($i=2; $i < $argc; $i++) {
        $tempvar = explode("=", $argv[$i]);
        echo $tempvar[1]."\r\n";
    }
}

if ($argv[1] == "scan") {
    echo "Starting scan. Please wait...\r\n";
    for ($i=2; $i < $argc ; $i++) {
        $tempvar = explode("=", $argv[$i]);
        switch ($tempvar[0]) {
            case '--path':

```



```

    $directory = $tempvar[1]."*";
    break;
    case '--extension':
    $extension = $tempvar[1];
    break;
    case '--log_file':
    $log_file = $tempvar[1];
    break;
}
}
if ($argc < 5) {
    if (!$directory) {
        $directory = "C:\Windows\*";
    }
    if (!$extension) {
        $extension = "exe";
    }
    if (!$log_file) {
        $log_file = "C:\\logfile\md5_log_file.txt";
    }
}

echo "the directory is ".$directory."\r\n";
echo "the extension is ".$extension."\r\n";
echo "the log file is ".$log_file."\r\n";

$files = recursiveGlob($directory,$extension);
$fh = fopen($log_file, "w") or die("can't open file \r\n");
var_dump($files);

```

```

$i = 1;
foreach ($files as $file) {
    $stringData = $i.") Found file: ".$file."\r\n";
    $stringData .= "Hash: ".md5_file($file).\r\n";
    $i++;
    fwrite($fh, $stringData);
}

fclose($fh);

}

if ($argv[1] == "compare") {
    echo "Starting compare. Please wait...\r\n";
    for ($i=2; $i < $argc ; $i++) {
        $tempvar = explode("=", $argv[$i]);
        switch ($tempvar[0]) {
            case '--md5_values_file':
                $md5_values_file = $tempvar[1];
                break;
            case '--log_file':
                $log_file = $tempvar[1];
                break;
            case '--results_file':
                $results_file = $tempvar[1];
                break;
            default:
                echo "error: loop defaulted\r\n";
                break;
        }
    }
}

```

```

    }
}

if ($argc < 4) {
    echo "Wrong count of parameters. Please type php run.php help for more info";
    die();
}

$scanned      =      file($log_file,      FILE_IGNORE_NEW_LINES      |
FILE_SKIP_EMPTY_LINES);
$clean_md5     =     file($md5_values_file,  FILE_IGNORE_NEW_LINES  |
FILE_SKIP_EMPTY_LINES);
$clean_md5_size = count($clean_md5);
$i = 1;

$fh = fopen($results_file, "w") or die("can't open file \r\n");
foreach ($scanned as $line_num => $line) {
    $check = explode(" ", $line);
    if ($check[0] == "Hash:") {
        foreach ($clean_md5 as $clean_line_num => $clean_line) {
            if ($check[1] == $clean_line) {
                break;
            }
        }
        if (($clean_line_num + 1) == $clean_md5_size) {
            echo "Suspicious file found -> ".$pathname[1]."\r\n";
            $stringData = $i.") Suspicious file found -> ".$pathname[1]."\r\n";
            $i++;
            fwrite($fh, $stringData);
        }
    }
}
}

```

```

    }
    else {
        $pathname = explode(" ", $line);
    }
}
fclose($fh);
}

?>

```

3) The program scanner_version_3.php in the PHP programming language:

```

<?php

error_reporting(E_ERROR | E_WARNING | E_PARSE);

//solution to return *.* files

$array = array();
function recursiveGlob($dir, $ext) {
    global $array;
    $globFiles = glob("$dir/*.$ext");
    $globDirs = glob("$dir/*", GLOB_ONLYDIR);

    foreach ($globDirs as $dir) {
        recursiveGlob($dir, $ext);
    }

    foreach ($globFiles as $file) {

```

```

        if(!in_array($file,$array)) {
            array_push($array,$file);
        }
    }
    return $array;
}

```

//only use in CLI mode because of extra large buffer and execution time

```

if ($argv[1] == "help" || $argv[1] == null) {
    echo "-- SCAN MODE --\r\n";
    echo "Usage: php scanner.php scan --path=path_to_check --
extension=extension_to_filter --log_file=path_and_name_of_log_file \r\n";
    echo "\r\n";
    echo "Example: php scanner.php scan --path=\"C:\myfolder\\" --extension=\"exe\" --
log_file=\"C:\mydocuments\md5_log.txt\" \r\n";
    echo "\r\n";
    echo "Default values: --path=\"C:\Windows\" --extension=\"exe\" --
log_file=\"C:\\logfile\md5_log_file.txt\" \r\n";
    echo "\r\n";
    echo "-- COMPARE MODE --\r\n";
    echo "Usage: php scanner.php compare --log_file=path_and_name_of_log_file --
md5_values_file=path_and_name_of_md5_values_file --
results_file=path_of_results_file \r\n";
    echo "\r\n";
    echo "Example: php scanner.php compare --log_file=\"C:\\log_file_after_scan.txt\" --
md5_values_file=\"C:\\file_with_clean_md5_values.txt --
results_file=\"C:\\file_with_results.txt \r\n";
    echo "\r\n";

```

```

    die();
}

if ($argv[1] == "debug") {

    for ($i=2; $i < $argc; $i++) {
        $tempvar = explode("=", $argv[$i]);
        echo $tempvar[1]."\r\n";
    }
}

if ($argv[1] == "scan") {
    echo "Starting scan. Please wait...\r\n";
    for ($i=2; $i < $argc ; $i++) {
        $tempvar = explode("=", $argv[$i]);
        switch ($tempvar[0]) {
            case '--path':
                $directory = $tempvar[1]."*";
                break;
            case '--extension':
                $extension = $tempvar[1];
                break;
            case '--log_file':
                $log_file = $tempvar[1];
                break;
        }
    }
}

if ($argc < 5) {
    if (!$directory) {

```

```

    $directory = "C:\Windows\*";
}
if (!$extension) {
    $extension = "exe";
}
if (!$log_file) {
    $log_file = "C:\\logfile\md5_log_file.txt";
}
}

echo "the directory is ".$directory."\r\n";
echo "the extension is ".$extension."\r\n";
echo "the log file is ".$log_file."\r\n";

$files = recursiveGlob($directory,$extension);
$fh = fopen($log_file, "w") or die("can't open file \r\n");
var_dump($files);

foreach ($files as $file) {

    $stringData = md5_file($file).\r\n";

    fwrite($fh, $stringData);
}

fclose($fh);
}

```

```

if ($argv[1] == "compare") {
    echo "Starting compare. Please wait...\r\n";
    for ($i=2; $i < $argc ; $i++) {
        $tempvar = explode("=", $argv[$i]);
        switch ($tempvar[0]) {
            case '--md5_values_file':
                $md5_values_file = $tempvar[1];
                break;
            case '--log_file':
                $log_file = $tempvar[1];
                break;
            case '--results_file':
                $results_file = $tempvar[1];
                break;
            default:
                echo "error: loop defaulted\r\n";
                break;
        }
    }
}

if ($argc < 4) {
    echo "Wrong count of parameters. Please type php run.php help for more info";
    die();
}

$scanned      =      file($log_file,      FILE_IGNORE_NEW_LINES      |
FILE_SKIP_EMPTY_LINES);

$clean_md5    =      file($md5_values_file, FILE_IGNORE_NEW_LINES    |
FILE_SKIP_EMPTY_LINES);

$clean_md5_size = count($clean_md5);

```



```

$i = 1;

$fh = fopen($results_file, "w") or die("can't open file \r\n");
foreach ($scanned as $line_num => $line) {
    $check = explode(" ", $line);
    if ($check[0] == "Hash:") {
        foreach ($clean_md5 as $clean_line_num => $clean_line) {
            if ($check[1] == $clean_line) {
                break;
            }
            if (($clean_line_num + 1) == $clean_md5_size) {
                echo "Suspicious file found -> ".$pathname[1]."\r\n";
                $stringData = $i.") Suspicious file found -> ".$pathname[1]."\r\n";
                $i++;
                fwrite($fh, $stringData);
            }
        }
    }
    else {
        $pathname = explode(":", $line);
    }
}
fclose($fh);
}

?>

```