Comparative Analysis of Network Forensic Tools on Different Operating Systems

Damir Delija, Ivan Mohenski, Goran Sirovatka
Tehničko Veleučilište u Zagrebu, Zagreb, Hrvatska
ddelija@tvz.hr, ivan.mohenski@tvz.hr,gsirovatka@tvz.hr

***Abstract* - This paper deals with the theoretical and practical elaboration of the mentioned topics, which gives an insight into digital forensics, network forensics, collection and analysis of digital evidence, and the tools with which the above is done. An insight into the comparison of tools is provided, which begins with the selection of tools according to the stated criteria and the description of the tools with the operating systems on which they run. The comparison itself begins with defining tasks to test functionality. Upon completion of the above tasks, the obtained results are recorded and later used to analyze the performance of the tool. After the tool comparison, the obtained results are scored and ranked, and then a conclusion is given about the acquired knowledge and experience during the preparation of this paper.**

Keywords: forensics, digital, network, tools, analysis

I. INTRODUCTION

With the development of the first network in the 1960s, numerous opportunities and possibilities emerged that progressed and grew over the years. With the development of networks and networked devices, their misuse also appeared, and for that reason, a branch of network forensics developed from digital forensics. Digital forensics is a science that aims to collect, store, retrieve, analyze and document digital evidence, or data that is stored, processed, or transmitted in digital form. Network forensics is a branch of digital forensics that deals with the collection, analysis, processing, and presentation of digital evidence obtained from network traffic. Nowadays, it is a very important branch of digital forensics, since almost every computer or device communicates with each other via a network. Precisely for this reason, the network is a very common vector of attacking or carrying out unauthorized actions. Every attack permitted or unauthorized action, travels across the network in the form of network traffic and leaves a certain trace, and it is this trace that often serves as digital evidence. Digital evidence is considered to be any data stored in digital form that contains information related to the commission of an illegal act [1, p. 4].

With the increasing complexity of networks, network devices, and thus network forensics procedures, there is a need for stable, high-quality, and reliable network forensics tools. The topic of this paper is exactly on the trace of the above, and it is the elaboration and implementation of comparative analysis of network forensics tools on different operating systems.

The topic of this paper is focused on one of the software solutions, and the existing software solutions are: packet capturing tools, pattern matching engines), data flow and analysis tools, and full-scale analysis tools. In essence to compare and rank tools depending on succeeding in solving tasks.

Network traffic, if collected, processed, and presented by the prescribed rules and procedures, provides the investigator with an accurate insight into events that took place on the network. Depending on the complexity of the task and the skill of the investigator, it is possible to single out the parts from which attempts to carry out or actual conduction of illegal actions on the network can be seen.

In preparation, an in-depth analysis of available sources and titles was performed to establish a tool evaluation system. The idea was to understand how the problem was approached earlier and how it was thought. In this step, the most useful were the titles: “Cyber Forensic Tools: A Review” [10], "A study on digital forensic tools" [11], "Forensic investigation of cross-platform massively multiplayer online games: Minecraft as a case study" [12] and “Big Data Forensics: Hadoop Distributed File Systems as a Case Study BT - Handbook of Big Data and IoT Security” [13]. During preparation a new metrics for network forensic tools evaluation was introduced and tested.

Network traffic collection can be done in several ways, and the basic division is done into the passive and active collection. The passive collection is the collection of network traffic that does not affect the state of the network and requires minimal interaction of investigators with the network. It is done by collecting network traffic that travels by wire, wirelessly, or through specific network devices such as a switch. If network traffic cannot be collected without minimal interaction with the network, but the investigator is forced to generate some traffic on the network to achieve the above, such collection is called an active collection. Disadvantages of active network traffic collection are the possible destruction of existing evidence and the potential activation of network security system alarms [2, p. 17].

Network traffic collection tools may vary depending on how they work. There are hardware tools for collecting network traffic and software solutions to achieve the same.

II. COMPARATIVE ANALYSIS OF TOOLS

The focus of this paper is on the comparative analysis of the aforementioned full-scale analysis tools. For that, it is necessary to get acquainted with each of the tools being analyzed, with the operating systems on which the analysis is performed on and to define the comparison criteria.

A. Methodology

The chosen methodology is based on qualitative and quantitative analysis of network forensics tools. It was carried out through four phases:

- Qualitative phases, which consisted of:

  1) defining the method of tool selection, defining evaluation tasks, and selecting the scoring algorithm

  2) selection of tools for network forensics on various operational systems

- Quantitative phases, which consist of:

  3) measuring tool behavior on chosen tasks

  4) scoring and ranking of tools according to the achieved results.

B. Wireshark

Wireshark is a free network traffic monitoring tool. It is used by network administrators to troubleshoot problems or find security vulnerabilities. It is also used by development engineers as a tool to test the implementation of new protocols. The area in which the Wireshark tool relates to this work is network forensics and its purpose to find evidence in traffic traveling across the network. Some of the important features of Wireshark tools are compatible with Windows and Unix operating systems, the ability to collect live traffic and save it, the ability to open files containing traffic captured by another tool, import text files containing a hexadecimal record of downloaded network traffic, export certain packets to others file formats, packet filtering, conditional packet search and display of traffic statistics. Wireshark is exclusively a tool for reviewing and analyzing network traffic [3].

C. Alternative tools selection

One of the criteria, also the main one, is that the tool must be an adequate replacement for the Wireshark tool for a given operating system. Other criteria to narrow the choice of tools are availability (must be free), ability to work with a graphical interface if possible, and popularity. Information on the popularity of each tool is collected through the *AlternativeTo.net* website. Given that some of the tools are not found on the said website, it is understandable that information on their popularity is not available so this criterion is not an exclusive criterion.

Short research reveals an article in which, according to Stephen Cooper, the five best alternatives to the Wireshark tool are Savvius Omnipeek, Ettercap, Kismet, SmartSniff, and EtherApe [4]. Given that some of these tools do not meet the given criteria and that the desire is to achieve an objective analysis that includes multiple tools from multiple different sources, there is a need for expanding the source list of alternative tools. These sources for an extended list of tools are the websites *AlternativeTo.ne*t, *guru99.com*, *educba.com*, and *techwiser.com*.

The list of tools gathered from these sources can be seen in Table I below.

TABLE I LIST OF TOOLS AND CRITERIA

	Win 10	**Mac OS**	**Kali**	**Available**	**GUI**	**Popularity**
Wireshark	YES	YES	YES	YES	YES	
Ettercap	/	YES	YES	YES	YES	15
Etherape	/	/	YES	YES	YES	/
Kismet	YES	YES	YES	YES	YES	/
Network Miner	YES	/	/	YES	YES	13
TcpDump	/	YES	YES	YES	/	50
WinDump	YES	/	/	YES	/	/
Cloud Shark	/	/	YES	NO	YES	9
Sysdig	/	/	YES	NO	YES	8
Colasoft Capsa	YES	/	/	NO	YES	4
Debokee	/	YES	/	YES	YES	2
Etheral	Part of PRTG Monitor tools					5
Intercepter NG	Contains malicious code					14
Nethogs	/	/	YES	YES	YES	19
MNM	YES	/	/	YES	YES	17
SmartSniff	YES	/	/	YES	YES	12
PacketSled	Part of MixMod tools					3
Scapy	/	/	YES	YES	/	7
Cain and Abel	YES	/	/	YES	YES	/
Savvius Omnipeek	YES	/	/	NO	YES	/
Packet Peeper	/	YES	/	YES	YES	2
CPA	/	YES	/	YES	YES	2
KisMac	/	YES	/	YES	YES	/

Selected tools are: for Windows NetworkMiner, for MacOS PacketPeeeper, and for Kali Linux Ettercap.

The reason for choosing the NetworkMiner tool is its high popularity compared to other tools NetworkMiner is an open-source tool designed for the Windows operating system, and its primary purpose is to analyze network traffic. It is used in the passive collection of network traffic and provides the possibility of analyzing the same to detect operating system from which or to which network traffic travels, also to detect information on sessions, open ports, and similar items. Manufactured in 2007, it gained great popularity relatively quickly due to the ability to perform network traffic analysis in a relatively simple way.

The Packet Peeper tool, which is used as an alternative tool on MacOS in this paper. Packet Peeper is a free tool designed to monitor and collect network traffic, made exclusively for MacOS operating systems. It provides several features such as composing TCP streams, packet filtering, support for pcap/tcpdump files, and much more [7].

The Etterercap tool was chosen because it is listed as one of the best alternatives to the Wireshark tool according to Stephen Cooper [8]. Ettercap is a free tool designed to help perform "Man in the middle" attacks. That is why it is equipped with the ability to monitor traffic, filter it, analyze the network and devices on it. The Ettercap tool can be used in three modes: graphical interface mode, command-line mode, and ncurses mode. Primarily designed for Unix operating systems with the ability to run the same on Mac operating systems. The version for the Windows operating system existed until 2011 when the manufacturer stopped providing upgrades and support. The latest version for Unix operating systems was released on 01.08.2020. [9].

D. Functionality testing tasks

Testing the functionality of the tool is carried out through three tasks. The first task is to collect traffic with selected tools and compares their capabilities in this mode. Each subsequent task, i.e. tasks 2 and 3, uses the already collected traffic in the form of a .pcap file.

In the first task, it is necessary to begin collecting traffic with the selected tool and then generate specific traffic. Traffic is generated by „pinging" via the command line web address *www.tvz.hr*. After the ping, it is necessary to visit the specified web address. Finally, it is necessary to download the image that contains the logo of the Polytechnic of Zagreb. Upon completion of the generated traffic, it is necessary to save it to the local computer from which the collection was performed. When saving, it is necessary to record the number of formats in which the collected traffic file can be saved and the size of the saved file.

The second task is all about gathering information from predefined traffic through series of subtasks. In those subtasks following information can be gathered: traffic recording time, the number of captured packets, used protocols, most used protocols, traffic filtering, HTTP status code, hostnames and visited web pages, used web browser, characteristic actions for TCP protocol, MAC and IP addresses of devices, used OS, opened ports and BSSID and SSID of Access Point.

The third task is testing the exclusion of files from .pcap files. The same is done through series of subtasks where it is needed to find out and exclude e-mail correspondence, contents of e-mail correspondence, attachments from e-mail correspondence, IM correspondence, contents of IM correspondence, attachments from IM correspondence, and .png files.

E. Comparison of obtained results

Upon completion of solving the tasks that examine the functionality of the tools, the obtained results are compared. To make the comparison clearer, a table is created in which the results are listed in the order of solving the tasks. The obtained results are recorded in two forms. The first form is the value of the result obtained, such as the number of packages contained in the file, while the second form is "yes/no / partial" which provides insight into the information whether the specified action can be done using a particular tool or not. If the stated result is "partial", it indicates that there is a partial record of the requested information from which the answer can be deduced. These results are visible in table II below.

TABLE II RESULTS OF SOLVING GIVEN TASKS

Task		**Windows**		**MacOS**		**Kali Linux**	
		Wireshark	NetworkMine	Wireshark	Packet Peeper	Wireshark	Ettercap
1	Number of saving formats	21	1	21	1	21	no
	Saved file size	553 KB	112 KB	1,5 MB	889 KB	639 KB	689 KB
2	Capturing time	47 sec	no	47 sec	47 sec	47 sec	no
	Captured packets	548	no	548	548	548	548
	Protocols used	TCP / UDP	no	TCP / UDP	TCP / UDP	TCP / UDP	TCP / UDP
	Most common protocol	TCP	no	TCP	TCP	TCP	TCP
	Traffic filtering	yes	partial	yes	yes	yes	yes
	HTTP status code	200 OK	200 OK	200 OK	200 OK	200 OK	200 OK
	Name discovery	yes	yes	yes	no	yes	yes
	Visited web sites	yes	yes	yes	yes	yes	yes
	Web browser being used	yes	yes	yes	yes	yes	yes
	TCP characteristics	yes	no	yes	yes	yes	no
	MAC device addresses	yes	yes	yes	yes	yes	yes
	IP device addresses	yes	yes	yes	yes	yes	yes
	OS discovery	partial	yes	partial	partial	partial	partial
	Opened ports	partial	yes	partial	partial	partial	yes
	AP BSSID-a discovery	yes	no	yes	no	yes	no
	AP SSID-a discovery	yes	no	yes	no	yes	no
3	E-mail discovery	yes	yes	yes	yes	yes	yes
	E-mail contents	yes	yes	yes	yes	yes	yes
	Attachment exemption	yes	yes	yes	yes	yes	no
	IM discovery	yes	yes	yes	yes	yes	yes
	IM contents	yes	yes	yes	yes	yes	yes
	IM attachment exemption	yes	yes	yes	yes	yes	no
	PNG exemption	yes	yes	yes	yes	yes	yes

F. Wireshark comparison on different OS

From the achieved results presented in Table II, it can be seen that the results of the Wireshark tool, solving the second and third tasks, are identical on all the previously mentioned operating systems. The difference in the achieved results is visible in the first task of collecting traffic. This difference is not due to differences in tool functionality on different operating systems, but because of the operating system itself, services, and background processes running. Although the collection of traffic was done on computers that had freshly installed operating systems with a fairly identical set of installed programs, the differences are visible. Also, it should be noted that during traffic collection, the only computer that generated traffic on the network was the computer from which the collection was performed. From the above facts, it can be concluded that the Windows operating system generated the least network traffic with network traffic generated for tool comparison.

G. Tool comparison on Windows OS

Solving the first task using Wireshark and NetworkMiner tools is equally successful. The difference observed in the results is in the number of formats in which the captured traffic is stored and in the size of the captured traffic file itself, which indicates the fact that the Wireshark tool was more successful in capturing, and that it captured more useful information from generated network traffic. Solving the second task in which information was excluded from traffic according to the presented results indicates the fact that Wireshark was more successful in solving this task as well. Solving subtasks related to statistics, protocols, traffic capturing, and access point information discovery was not successful with the NetworkMiner tool while solving them with the Wireshark tool was fairly straightforward. The determination of operating systems and open ports with the Wireshark tool was done partially, i.e. the above information could be deduced from the TTL package parameters and the amount of traffic per port, while the NetworkMiner tool provided a simple insight into that information. Solving other subtasks was equally successful and complex. Solving the third task, according to the above results, is equally successful with both tools. The difference in solving the third task is in the simplicity, ie the process of extracting files with the NetworkMiner tool is much simpler considering that all files in traffic are recognized and excluded by the tool when loading traffic file. Extracting files with the Wireshark tool was a challenge, especially when it came to a .docx file. The advantage of the Wireshark tool in extracting files from traffic is the ability to extract them through a hexadecimal record in traffic, and save files using tools for processing such records. This is not possible with the NetworkMiner tool because it does not support working directly with packets and their hexadecimal records, but only with interpreted data, which provides the possibility of tool errors in the form of not recognizing files or records that are important.

H. Tool comparison on Mac OS

Solving the first task with Wireshark and PacketPeeper tools differed in the amount of possible file storage formats of the collected traffic and the size of the saved files. The file collected by the Wireshark tool is significantly larger than the file collected by the Packet Peeper tool, which also indicates a larger amount of data collected from online traffic. The results of solving the second task differ in the success of solving the task related to discovering the device name and collecting information about access points. Device name detection is not possible with the Packet Peeper tool because there is no name resolution option, and access point information collection was not possible due to the inability to open the required file containing traffic related to that task. Both of these tools are equally successful in determining operating systems and open ports, i.e. both tools provide information on the TTL parameter of the packet and the amount of traffic per port from which the required information can be deduced, but not unequivocally determined. Solving the third task is successful with both tools. The difference between these tools in extracting files from traffic is in the fact that the Wireshark tool provides method for extracting files through the "Export objects" built-in function and through the hexadecimal package record. The Packet Peeper tool provides the ability to exclude traffic exclusively through the hexadecimal record of the contents of individual packages, which makes it challenging.

I. Tool comparison on Kali Linux

The tools used on the Kali Linux operating system are Wireshark and Ettercap. The results achieved on the first task differ in the ability to save the collected traffic. The Ettercap tool, unlike the Wireshark tool, cannot save the collected traffic within the graphical mode, but it is possible to save it only using the command line. The sizes of the captured traffic do not differ drastically, but in a few KB, which indicates almost the same amount of data collected by both tools. The success of solving the second task is the same, and the differences are visible in solving subtasks that require direct interaction with packages which the Ettercap tool does not provide. Unlike the Wireshark tool, the Ettercap tool provides only the ability to work with interpreted data, but not at the level as it was on the Windows operating system using the Network miner tool, but at the level below. It provides the possibility of working with interpreted data, but also provides insight into the hexadecimal record of the same through the connections. The inability to solve the subtask related to collecting access point information is also present with the Ettercap tool, while the same Wireshark performed flawlessly. Solving the third task presents the problem of simply excluding files. Unlike the Wireshark tool, which offers the ability to easily exclude files through built-in functions and complexly exclude from a hexadecimal record, Ettercap provides only the ability to exclude files from a hexadecimal connection content record. Although it is possible to exclude files from hexadecimal records, the Ettercap tool did not successfully exclude attachments sent over the network.

J. Tool ranking

The ranking of tools is done by scoring the tools, which is done depending on the success of the tool in solving tasks.

TABLE III SCORING CRITERIA

Task	Number of points	Description
1	3 points for task solving	1 point for traffic recording 1 point for saving recorded traffic 1 point for more than one save format
2	1 point	per item listed in table II
3	1 point	per item listed in table II

According to the scoring system in Table III, the Wireshark tool achieves 25/26 points on all operating systems, NetworkMiner 17.5 / 26 points, Packet Peeper 20.5 / 26 points, and Ettercap tool 17.5 / 26 points. The first place belongs to the Wireshark tool due to the most successfully solved tasks and the ability to work on multiple operating systems. Second place, and also the title of the best alternative tool for Wireshark tool, belongs to the Packet Peeper due to the success of solving these tasks. The third-place belongs to the NetworkMiner tool due to the achieved results and ease of use. In the last place is the Ettercap tool due to its complexity of performing basic network traffic monitoring procedures and the achieved results of solving tasks.

The ranking of these tools according to the acquired knowledge and experience about them during the preparation of this paper is shown below.

1. Wireshark
2. PacketPeeper
3. NetworkMiner
4. Ettercap

The results show that the Wireshark tool was more successful in solving tasks on all operating systems. The best alternative tool is Packet Peeper according to task resolution performance, while the remaining two tools NetworkMiner and Ettercap are equally successful. The creation of the ranking list and the ranking of the tool also includes the subjective impression of the examiner, which is greatly influenced by the factor of the difficulty of solving the stated tasks with the selected tool.

In further research, one should also compare ease of use and time to master the basic functionality of the tools, since the popularity ranking presented in Table I does not address these issues correctly.

III. CONCLUSION

The results achieved show that the results achieved by the Wireshark tool on all operating systems are identical and better than the results of all other tools. The best alternative to the Wireshark tool was found on the MacOS operating system, and that is Packet Peeper, which is the most similar tool to the Wireshark tool in its capabilities. In terms of ease of use and getting results quick and easy, NetworkMiner proved to be the best tool. The Ettercap tool, while not an easy tool to master, has proven useful in analyzing network traffic while providing additional options for modifying network traffic, which none of the aforementioned tools can. Given the stated advantages and disadvantages of the tools used in the comparative analysis, it can be seen that each of the tools is better than the others in at least one segment, be it functionality, simplicity, or additional features. What all tools have in common is the availability and success of solving these tasks to some degree. The good practice gained from experience is to use multiple tools to solve a particular problem if possible. As for the Wireshark tool and its alternatives, the description is correct, all the listed alternative tools represent just that, adequate replacements.

REFERENCES

[1] A. Arnes, (2018), *Digital Forensics,* India, Indianapolis, Indiana, Wiley
[2] Ensia, (2019), *Introduction to Network Forensics, Greece*, Ensia
[3] Wireshark User's Guide, https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs, (visited 07.06.2020.)
[4] 5 Best Wireshark alternative packet sniffers, https://www.comparitech.com/net-admin/best-wireshark-alternatives/, (visited 07.06.2020.)
[5] Microsoft Network Monitor, https://en.wikipedia.org/wiki/Microsoft_Network_Monitor, (visited 08.08.2020.)
[6] NetworkMiner, https://www.netresec.com/?page=Networkminer, (visited 10.08.2020.)
[7] PacketPeeper, https://packetpeeper.org, (visited 10.08.2020.)
[8] 5 Best Wireshark alternative packet sniffers, https://www.comparitech.com/net-admin/best-wireshark-alternatives/, (visited 07.06.2020.)
[9] About the ettercap project, https://www.ettercap-project.org/about.html, (visited 07.06.2020.)
[10] B. V Prasanthi, "Cyber Forensic Tools: A Review," Int. J. Eng. Trends Technol., vol. 41, no. 5, pp. 266–271, 2016, doi:10.14445/22315381/ijett-v41p249.
[11] K. Ghazinour, D. M. Vakharia, K. C. Kannaji, and R. Satyakumar, "A study on digital forensic tools," IEEE Int. Conf. Power, Control. Signals Instrum. Eng. ICPCSI 2017, pp. 3136–3142, 2018, doi: 10.1109/ICPCSI.2017.8392304.
[12] D. C. P. J. Taylor et al., "Forensic investigation of cross platform massively multiplayer online games: Minecraft as a case study," Sci. Justice, vol. 59, no. 3, pp. 337–348, 2019, doi: https://doi.org/10.1016/j.scijus.2019.01.005.
[13] M. Asim, D. R. McKinnel, A. Dehghantanha, R. M. Parizi, M. Hammoudeh, and G. Epiphaniou, "Big Data Forensics: Hadoop Distributed File Systems as a Case Study BT - Handbook of Big Data and IoT Security," A. Dehghantanha and K.-K. R. Choo, Eds. Cham: Springer International Publishing, 2019, pp. 179–210.