Software System For Automatic Reaction To Network Anomalies And In Real Time Data Capturing Necessary For Investigation Of Digital Forensics

Mladen Vukašinović
Faculty of Information Technology
Mediterranean University
Podgorica, Montenegro
mladen@ac.me

***Abstract*—Digital forensics has a technical component, and tools in the form of appropriate software and hardware, but also a legal component aimed at respecting certain principles, rules and methodologies. There is a number of commercial tools in the market and the selection of appropriate tools depends on their usability in specific cases. This paper proposes a software tool that could be installed on the computer system. It would automatically respond to anomalies in the system and collect digital data. Such data would be stored and later used as digital forensics evidence. This paper presents the experimental results of a simulation of the network intrusion using Back Track to host and disclosure of that host using Wireshark and Netflow Analyzer.**

Keywords*—*digital forensics; digital data; the protection of computer networks; Wireshark; Netflow Analyzer

I. Introduction

There are many issues digital forensic investigation is facing.

Digital origin describing the previous or the history of a digital object is a key feature for forensic investigation.

Digital evidence is a necessary but not a sufficient condition for building a case. Such evidence is fragile and can easily be modified, destroyed and lost. [3]

When it comes to the preservation of digital evidence, investigators face many difficulties when trying to obtain information from the disk of the launched system using just a software approach. The software must communicate with the operating system all the time while the system is running. Investigators have suggested a variety of hardware and software solutions. For them to be fully effective they must be installed before the incident occurs.

This paper proposes a software solution that would be installed on the computer system which would automatically respond to anomalies in the system and automatically collect digital data. All data would be stored and later could reviewed in the original composition and check whether they can serve as evidence for digital forensics.

II. Digital Forensics

Digital forensics is the science of identification, collection, preservation and presentation of data that are electronically processed and placed on the computer media while preserving the integrity of the original evidence.

Digital forensics is a relatively new scientific discipline that has the potential to significantly affect specific types of investigations and prosecutions. It differs significantly from the traditional forensic disciplines. Tools and techniques that this discipline requires are easily accessible to everyone. Digital forensic experts analyze and examine evidence at each location, not only in a controlled environment.

Digital forensic investigation is a process which develops and tests various theories through hypothesis by analyzing digital devices, media which present relevant evidence in the court proceedings by using scientific methods and technologies. The main objective of the investigation is to establish the truth on unlawful activity and the manner of execution of criminal offense. Digital evidence in this case is a digital object that contains reliable information which supports or refutes the hypothesis [5].

Digital forensic investigation is based on the principles of digital forensic science.

When a digital criminal activity occurs, digital forensic investigators conduct investigation, go to the crime scene, make photos of the real situation they find and write reports with relevant information pertaining to this criminal activity.

III. Using Investigative Methodology

The investigative methodology includes forensic analysis of all types of digital investigations of criminal offenses.

The objective of defining the model of digital investigations is to formulate and standardize the process of digital investigations [5]. They should be applicable to all types of current digital crimes as well as those that will occur in the future. Also they need to overcome the limitations of the existing models and provide a standardized framework that supports all phases of investigation. The fact that digital evidence are not only on computers but also on various other media (mobile phones, e-mail, web pages, and social networks) should also be taken into account [6].
There are several types of investigative methods (model): the DFRWS model, the Abstract Digital Forensic Model, the Ciardhuain model, The Beebe and Clark model, the Kruse and Heiser model, DOJ model, the Carrier and Spafford model, Model “Incident Response” and the Eoghan Casey and model. There are several things they have in common: identification/handling, forensic acquisition, forensic analysis and presentation.

These models should help digital investigators apply some of these models to a specific investigation, depending on the case.

In this paper we have used the following models: the “Incident Response” model and the Eoghan Casey model.

IV. Using Software System For Automatic Reaction At Fault In The Network

Experiments were carried out on Pentiumu® Dual-Core CPU E5200 @ 2.50GHz, 2.50GHz, 2.00GBRAM 32-bit Operating System, x64-based processor, and Microsoft Windows 7. The Whireshark Version of 1.12.2 (v1.12.2-0 - g898fa22 from master-1.12) was installed on it and NetFlow Analyzer 9.8.5 Licence free type was used.

NetFlow Analyzer was installed on the HP Server Blade. The information flow was collected from the central router located on the border between the Internet and intranet computer network, as well as from the device presented as firewall. Hostnames were entered in the DNS software. Microsoft Windows XP Professional (SP2) 2002 Version was installed on Intel® Pentiumu® Duo CPU @ 2.80GHz, 2.79GHz, 0.768GB RAM via Live CD Back Treck 3 7.1.12 version.

The intrusion was conducted via a randomly selected computer from network. The Live CD BackTrack 3 was started on the computer. The computer got a free address via DHCP. The IP address that was not registered in DNS was detected via the NetFlow Analyzer software. Then the Whireshark software was started which “intercepted data in real time” from that IP address.

Figure 1 shows data captured by the NetFlow Analyzer. The graphs show the IP address that was not entered in DNS, 89.188.47.71 from which the attack on the network was conducted. The graphs also show the used destination IP addresses, their numerical and literal depictions, the used applications, ports, protocols, data size, the time when it happened, and so on.

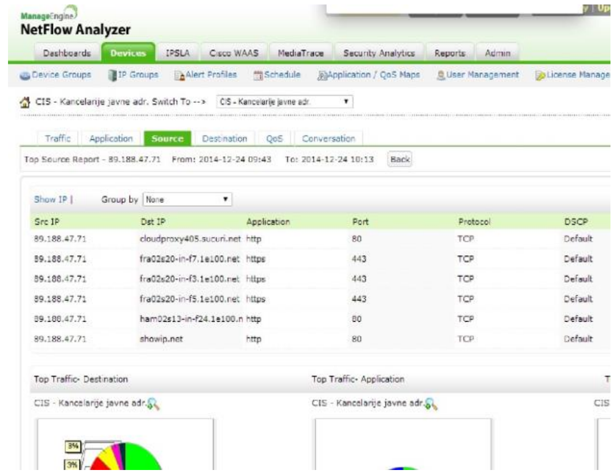

Figure 1. An overview of the connections executed by a specific host with a depicted DNS

Based on Figure 1 it can be concluded that the host communicates with various hosts out of the computer network at ports 80 used by the http protocol and ports 443 used by the https protocol. TCP was the protocol used for communication. The generated traffic to each address is also shown. The following Figure shows an overview of literal addresses that communicate, which was withdrawn from the DNS server.

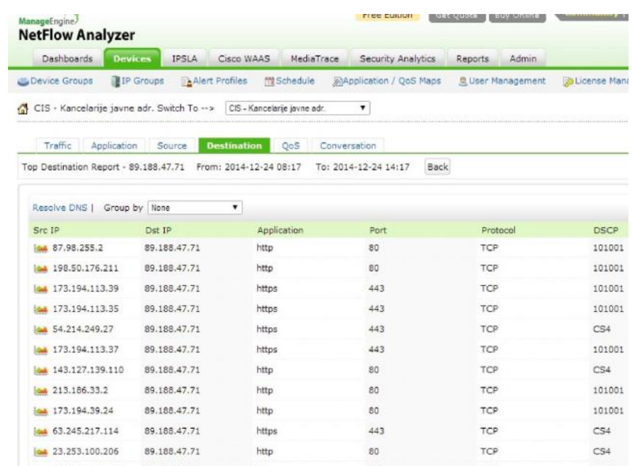

Figure 2. An overview of the connections executed by a specific host without a depicted DNS

Based on Figure 2, it can be concluded that different hosts outside the computer network communicate with the monitored host at ports 80 used by the http application and ports 443 used by the https application. The protocol used for communication was TCP, as well as the following special processes: DSCP, 101001 and CS4. The generated traffic to any address in MB (KB) and the percentage traffic to individual addresses can also be seen.

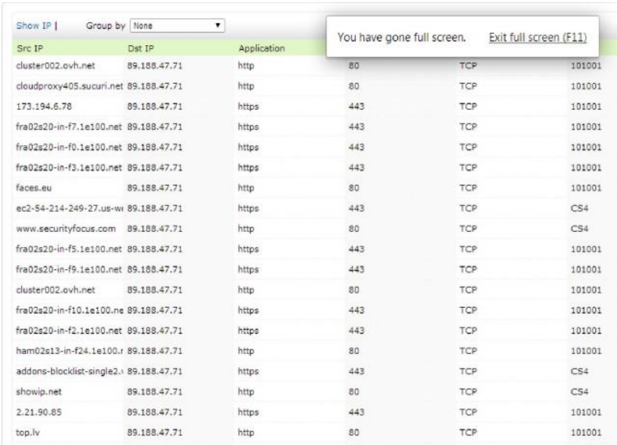

Figure 3. An overview of the connections executed by a specific host with a depicted DNS

Based on Figure 3, it can be concluded that different hosts outside the computer network communicate with the monitored host. The literal displays of those hosts are also shown since the DNS server data were used. The communication was conducted at ports 80 used by the http application and ports 443 used by the https application. The protocol used for communication was TCP, as well as the following special processes: DSCP, 101001 and CS4. The generated traffic to any address in MB (KB) and the percentage traffic to individual addresses can also be seen.

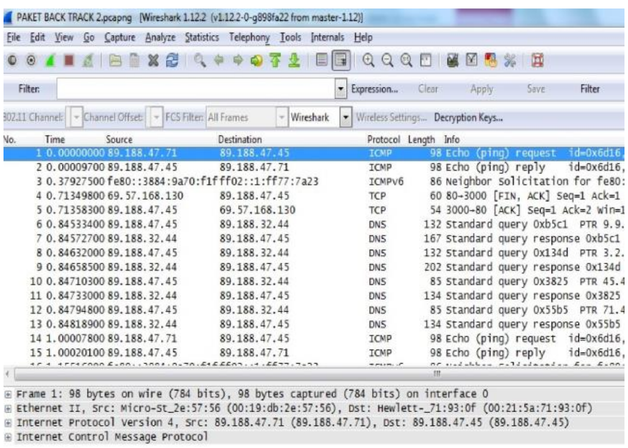

Figure 4. Intercepted packets from all IP addresses in the network, without the use of filters

Figure 4 shows all addresses which communicate with our host, the protocols used in communication, the length of the sent packages and description of individual communication.

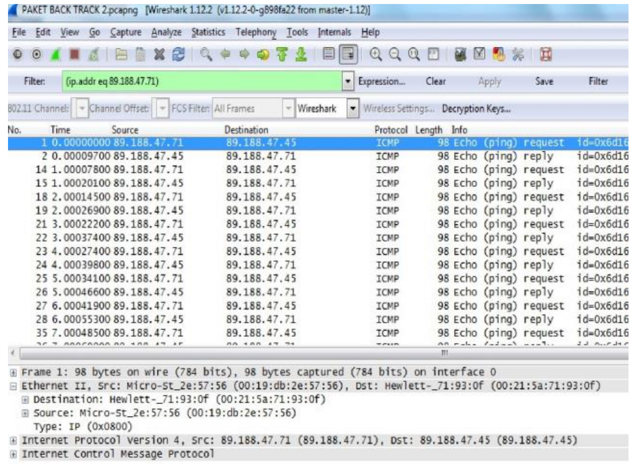

Figure 5. A list of intercepted packets from the monitored host of the ICMP protocol

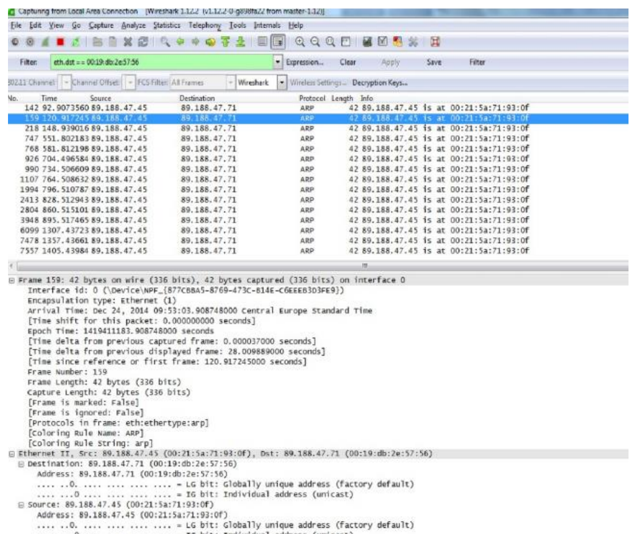

Figure 6. A list of captured packages from the monitored host

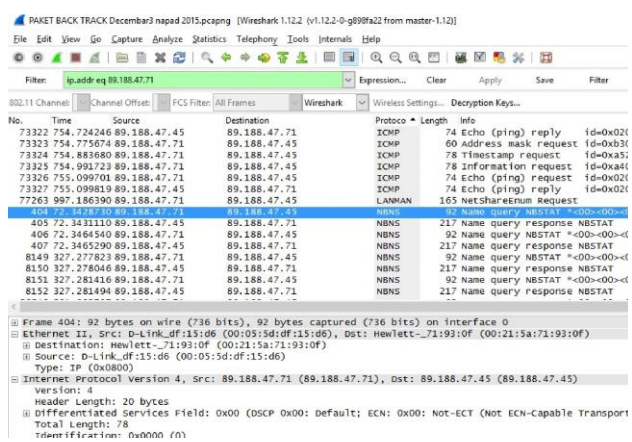

Figure 7. A list of captured packages from the monitored host

Based on Figure 7 it can be concluded that the monitored host ARP protocol was used for communication with our host with the repeated request. The analysis of the contents of the package ahs shown that the monitored host IP address 89.188.47.71 has MAK (physical address) address of 00:19:db:2e:57:56.

Based on Figure 8 it can be concluded that the monitored host used ICMP, LANMAN and NBNS protocols. The LANMAN protocol was used to scan the ports on our hosts. NBSTAT was used to watch TCP-IP information and other details of our computer.

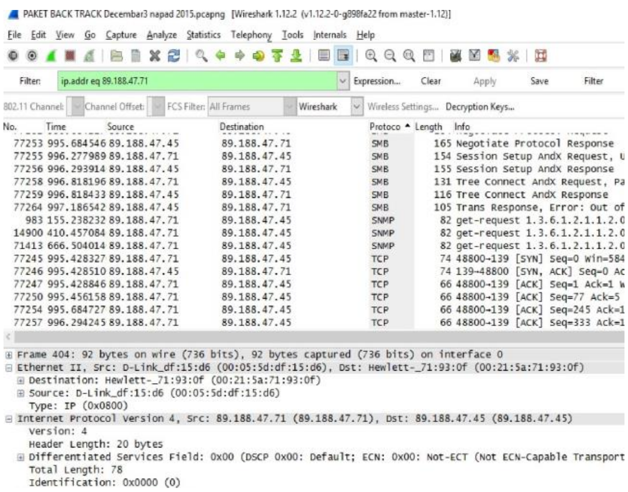

Figure 8. A list of captured packages from the monitored host, ARP protocol

The following commands were used in Whireshark to filter data: (eq ip.addr 89.188.47.71) (ip.addr eq eq ip.addr 89.188.47.45 and 89.188.47.71).

In Figure 8 the original IP address can be seen as well as the destination IP address, used protocols, the information about the length of the package, time, MAC (physical address) of the computer (Figure 6), and so on.

Based on the obtained live captured data, we can conclude that in order to discover password on the host, the monitored host was used to scan the computer network and the frequently sending ping packets. For all these actions the following protocols were used: TCP, UDP, ARP, NBNS, SMB, SNMP, and LANMAN.

Since we have the information about the IP address and MAC address of the host, the physical location of the computer from which the intrusion into the network was conducted, can be detected.

V. Comparative Analysis Of Capturing Packets Using Wireshark Netflow Analyzer

Wireshark and analysis of malicious software:

Wireshark allows the detection of the presence of malicious programs. If it does not do that in a direct way if there is a network communication, then almost certainly it is a malicious program that communicates with the network. When analyzing the malicious programs Wireshark is used to detect IP addresses from which malicious program wants to communicate this data to identify a sent malicious program. Detection of IP addresses can be extremely helpful in identifying the authors of malicious programs, and data identification is of great importance in determining the effects of malicious programs. Data identification is carried out by monitoring the suspicious TCP / UDP / ARP / SMB / SNMP / NBNS / LANMAN flows. In the analysis it is useful to collect large amounts of data, then remove potentially compromised computer from the network and perform analysis.

The analysis process itself largely depends on the actual analysis, and it is generally difficult to describe it. Once you establish the existence of a malicious use of network, the next step is to identify the program that is causing it. For this purpose it is better to use a program such as Network Monitor, which offers the option of filtering processes that communicate with the network.

Netflow Analyzer and analysis of malicious software:

NetFlow Analyzer can be used as a collector to get the information on the amount of generated network traffic, applications, sources and destination resorts participating in conversations, as well as the details of the protocols used, ports and other features of network traffic in the selected time period.

In addition to this information it is possible to get an overview of a particular host connection, traffic limitations of certain IP groups, interface, the percentages of the most common protocols, application and identification of top conversation hosts in the network. Besides the statistical data it is possible to create various types of alerts in the exercise of given parameters.

It is possible to find out the IP address of the host, application type, port, protocol, and traffic realized in a particular time interval. Flow technologies enable obtaining all the information of interest quickly and easily in the form of reports in customizable formats used to monitor the network in the real time or to deal with the analysis of utilization of network resources at different times of the day, week or month.

While Wireshark captures data in the national transport network, Netflow Analyzer captures data that is generated out of the monitored host network.

The combination of these two software tools allows capturing data used in a communication between the hosts on the external network. The disadvantages of this system are the following: Wireshark must be run by an administrator after detecting threats, and Netflow Analyzer does not capture data in domestic service.

VI. Automatic Alert Notification And Automatic Wireshark Launch

Based on the experimental results, the programme code in the visual basic was proposed which can start Wireshark in detecting threats as well as programme codes which can block the protocols, ports and applications which pose a threat to a computer system. Also connecting interface switches in a computer network with Netflow Analyzer was also given in order to detect traffic within a computer network as well, not only from the outside. Also a proposal for creating an alarm that will continue to react to the emergence of protocols, ports, applications and DSCP processes which pose a threat to a computer system in the attack from the network was also given.

NetFlow Analyzer has the ability to send not only alerts to the administrator's address but also the SNMP trap messages. In this case one of the parts of the SNMP protocol is used - Trap. Trap is a one-direction message from network devices such as routers, switches or servers (NetFlow Analyzer server / collector) which is sent to SNMP trap collectors (Application Manager). Trap collector represents a combination of hardware and software used to collect messages. SNMP trap messages is transmitted via the UDP protocol, which means that there is no guarantee that the message will be registered by the trap collector. The best-known SNMP trap messages are link up, link down, and agent reset (when the device is switched on again). In addition to the usual messages, there are special trap messages that are created by different hardware and software manufacturers which give them diverse meanings. NetFlow Analyzer collector has a set of SNMP trap messages that are created on the basis of given parameters, when an alert occurs which was previously created. Alerts that are created by NetFlow Analyzer server can be forwarded via trap messages to the Trap collector server. The Trap collector server has the ability to save and storage alerts in one place.

Preparations for the forwarding of SNMP trap messages consist of the following steps:

• UDP port configuration on the side of the trap collector on which listening will be carried out (UDP port number is 162)

• Selection of the type of SNMP Trap alerts on the side of the NeFlow Analyzer server and parameters so that the messages can be forwarded to the Trap collector server. Parameters including inserting the following data:

• <Server Name> - The name or IP address of the server that is running trap collector.

• <Port Number> - number of the port where the collector listens trap trap messages.

• <Community String> - String of characters set to the trap collector.

Using SNMP trap messages and Trap collectors with defined filters and adequate actions, responses to the received alerts can be automated.

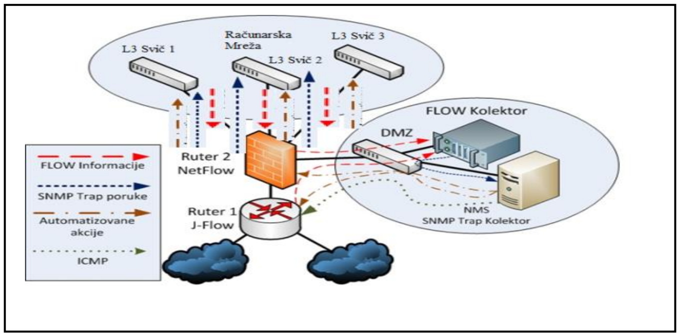

Fig. 9. Elements of control systems - Connecting switches, Flow collector

The flow collector (agent) is based on Ubuntu (Linux) server operating system, and SNMP Trap collector (manager) on the Windows operating system, while programs that perform modification of device configurations are written in Visual Basic script programming language with CLI device commands. An example of the automatic control system is implemented on heterogeneous architectures.

The network devices from Figures 13, L3 switch 1, 2 L3 switch, L3 switch 3 exported information to the flow collector. These devices need to be configured previously in order to be able to export the information flow properly. The flow collector collects flow information obtained by the exporter from the device and performs their analysis and display. Filters are set on the flow collector which in case of exceeding threshold or realization of given parameters create alert messages. Warning messages with the information sent as an email message to the administrator or as a SNMP Trap message to the NMS (network management system) servers, and SNMP Trap collectors.

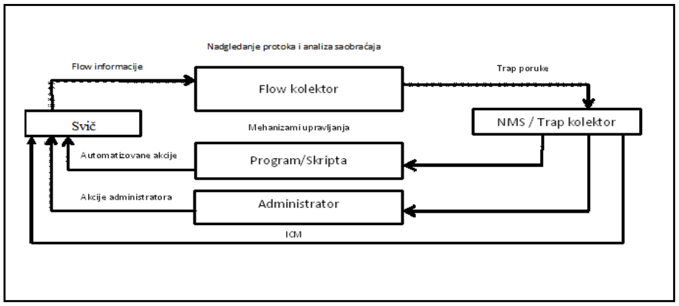

Fig. 10. Schematic representation of management process

Visual Basic Script (Program code 1), which will launch trap collector is aimed at starting the Wireshark program:

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run "C:\putty.exe 89.188.47.45 -l user -pw password"
 WshShell.Run "C:\Program Files\Wireshark>Wireshark.exe"
WshShell.AppActivate "89.188.47.45 - Wireshark"
WshShell.SendKeys "ctrl+e"
```

Visual Basic Script (Program code 2), which will launch a trap collector is aimed to block the SMB protocol on the host:

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run"C:\putty.exe 192.168.1.1 -l user -pw password"
WScript.Sleep 5000
WshShell.AppActivate "192.168.1.1 - PuTTY"
WshShell.SendKeys "enable{ENTER}"
WshShell.SendKeys "password{ENTER}"
WshShell.SendKeys "configure terminal{ENTER}"
WshShell.SendKeys "int GigabitEthernet 0/3{ENTER}"
WshShell.SendKeys "access-list host_access_in extended deny SMB host 89.188.47.71 any eg 3020{ENTER}"
WshShell.SendKeys "exit{ENTER}"
WshShell.SendKeys "exit{ENTER}"
```

Visual Basic Script (Programming code 3), which will launch trap collector is aimed at blocking the NBNS protocol on the host:

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run"C:\putty.exe 192.168.1.1 -l user -pw password"
WScript.Sleep 5000
WshShell.AppActivate "192.168.1.1 - PuTTY"
WshShell.SendKeys "enable{ENTER}"
WshShell.SendKeys "password{ENTER}"
WshShell.SendKeys "configure terminal{ENTER}"
WshShell.SendKeys "int GigabitEthernet 0/3{ENTER}"
WshShell.SendKeys "access-list  host_access_in extended
deny NBNS host 89.188.47.71 any eg  any{ENTER}"
WshShell.SendKeys "exit{ENTER}"
WshShell.SendKeys "exit{ENTER}"
```

Visual Basic Script (Program code 4), which will launch trap collector is aimed at blocking the IP protocol on the host:

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run C:\putty.exe 192.168.1.1 -l user -pw password"
WScript.Sleep 5000
WshShell.AppActivate "192.168.1.1 - PuTTY"
WshShell.SendKeys "enable{ENTER}"
WshShell.SendKeys "password{ENTER}"
WshShell.SendKeys "configure terminal{ENTER}"
WshShell.SendKeys "int GigabitEthernet 0/3{ENTER}"
WshShell.SendKeys "access-list  host_access_in extended
deny LANMAN host 89.188.47.71 any eg  any{ENTER}"
WshShell.SendKeys "exit{ENTER}"
WshShell.SendKeys "exit{ENTER}"
```

Visual Basic Script (program code 5), which will launch trap collector is aimed at blocking the ARP protocol on the host:

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run"C:\putty.exe 192.168.1.1 -l user -pw password"
WScript.Sleep 5000
WshShell.AppActivate "192.168.1.1 - PuTTY"
WshShell.SendKeys "enable{ENTER}"
WshShell.SendKeys "password{ENTER}"
WshShell.SendKeys "configure terminal{ENTER}"
WshShell.SendKeys "int GigabitEthernet 0/3{ENTER}"
WshShell.SendKeys "access-list  host_access_in extended
deny ARP host 89.188.47.71 any eg  any{ENTER}"
WshShell.SendKeys "exit{ENTER}"
WshShell.SendKeys "exit{ENTER}"
```

Visual Basic Script (Program code 6), which will launch trap collector is aimed at blocking the TCP protocol on the host:

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run"C:\putty.exe 192.168.1.1 -l user -pw password"
WScript.Sleep 5000
WshShell.AppActivate "192.168.1.1 - PuTTY"
WshShell.SendKeys "enable{ENTER}"
WshShell.SendKeys "password{ENTER}"
WshShell.SendKeys "configure terminal{ENTER}"
WshShell.SendKeys "int GigabitEthernet 0/3{ENTER}"
WshShell.SendKeys "access-list  host_access_in extended
deny TCP host 89.188.47.71 any eg  any{ENTER}"
WshShell.SendKeys "exit{ENTER}"
WshShell.SendKeys "exit{ENTER}"
```

Visual Basic Script (program code 7) which will launch trap collector is aimed at blocking the IP protocol on the host:

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run"C:\putty.exe 192.168.1.1 -l user -pw password"
WScript.Sleep 5000
WshShell.AppActivate "192.168.1.1 - PuTTY"
WshShell.SendKeys "enable{ENTER}"
WshShell.SendKeys "password{ENTER}"
WshShell.SendKeys "configure terminal{ENTER}"
WshShell.SendKeys "int GigabitEthernet 0/3{ENTER}"
WshShell.SendKeys "access-list  host_access_in extended
deny IP host 89.188.47.71 any eg  any{ENTER}"
WshShell.SendKeys "exit{ENTER}"
WshShell.SendKeys "exit{ENTER}"
```

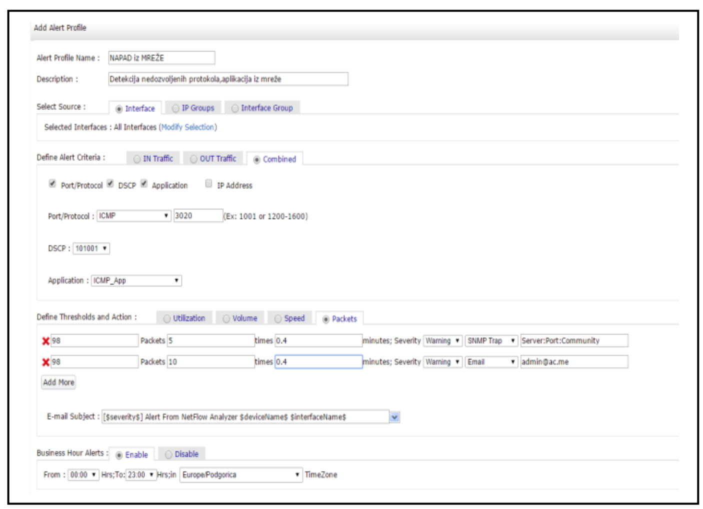

Fig. 11. Sending SNMP Trap alert messages for the ICMP protocol, DSCP-101001 and ICMP application

Figure 11 shows an example of creating an alert and forwarding it via SNMP Trap messages and via email messages. Messages will be created when you fill the parameters to define. In this regarding the cases messages will be sent if the application is sent ICMP packet length of 98 five times within 40 seconds.

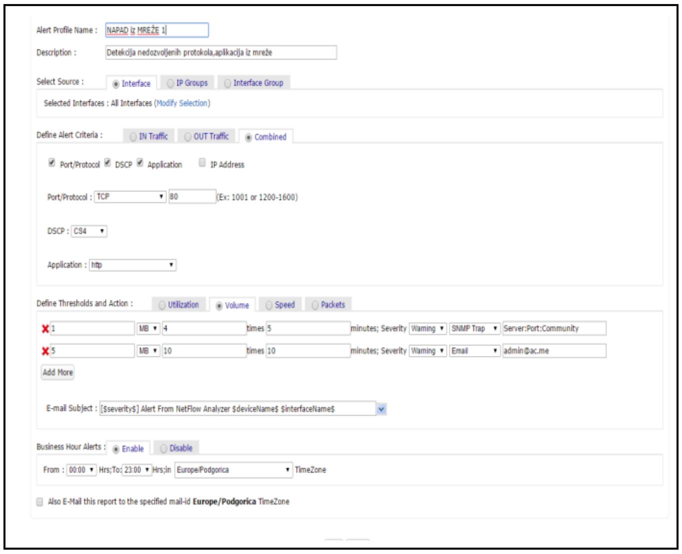

Fig. 12. Sending SNMP Trap alert messages for TCP protocol, DSCP-CS4 and HTTP application

In Figure 12 shows an example of creating an alert and forwarding via SNMP Trap messages as well as via emil messages. In this case, the message will be sent if the application is http traffic rules than 1 MB, and if it is repeated four to five times within 5 minutes will be sent Trap message, and if traffic is 5 MB, and if it is repeated 10 times within 10 minutes Email will be sent to the administrator.

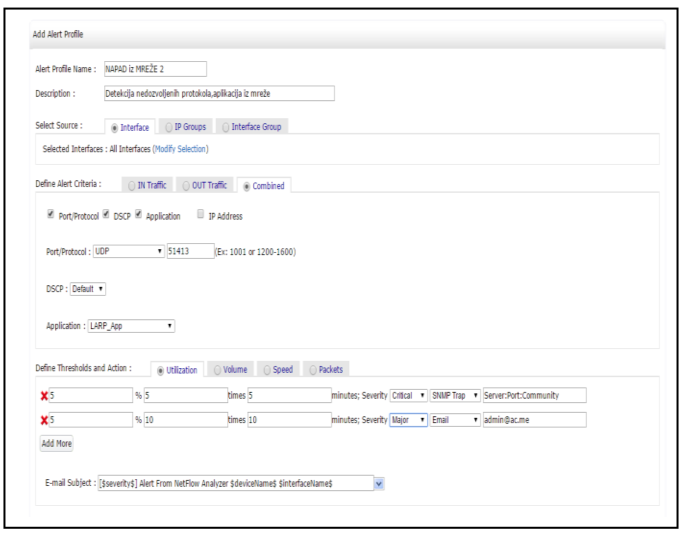

Fig. 13. Sending SNMP Trap alert messages for UDP protocol, LARP application

In Figure 13 shows an example of creating an alert and forwarding via SNMP Trap messages as well as via emil messages. In this case, the message will be sent if the interface utilization of 5% and if this is repeated 5 five times within 5 minutes Trap message will be sent, and if use of the interface between 5% and if this is repeated 10 times within 10 minutes Email will be sent to the administrator.

Actions incoming trap messages can be different only need to determine what action will be made to the appropriate type of warning.

Using this type of software management would allow the automatic response to the anomalies in the network and capture live data to assist in the investigation of digital forensics.

VII. Conclusion

Digital forensic investigation is a process which develops and tests various theories through hypothesis by analyzing digital devices and media which represent relevant evidence in the court proceedings all by using scientific methods and technologies. The main objective of investigation is to establish the truth about the unlawful activity and the manner of committing criminal offense. Digital evidence in this sense is a digital object that contains reliable information to support or refuse the hypothesis.

Each process has its own initial step which may be signalled by an alert of a protection system - a system for detection of an attack or a system for detection of malicious activities, sensors of protection at the network, or the administrator’s system after reviewing log files. It can be initiated in the traditional way, in case the user reports any criminal activity, the consequence of which is sending the investigation team to the crime scene.

Experimenting means testing new and untried techniques and methods that are based on a scientific basis with rigorous documentation for testing purposes. The result of the experiment could be either rejected or generally accepted. This paper presents a proposal for the automatic response of the system software to the perceived anomalies in the computer system. The system would automatically collect live data, which would be stored and would react with counter-measures to a specific incident.

References

[1] Reith M., Carr C., Gunsch G., An examination of digital forensics, International Journal of Digital Evidence, 1(3), 2002, 12-374267-4.

[2] Prosise C. Mandia K, Incident response and computer forensics, second edition 2003.

[3] Casey E., Digital Evidence and Computer crime, Academic press, San Diego 2000.

[4] CSCare, Inc., “Trap Console - User’s Guide“, San Jose, CA 95134, September 2006.

[5] Eoghan Casey, Handbook of Digital Forensics and Investigation, Elsevier Academic Press 2010, ISBN 978-0.