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.

NetFlow Analyz	zer		_		γ Up
Dashboards 0	Devices IPSLA	Cisco WAAS Mer	diaTrace Security Analytic	s Reports Admin	
Device Groups	IP Groups DAle	rt Profiles 🛛 📆 Schedi	Ile Application / QoS Ma	aps 🧕 User Management	Dicense Manage
CIS - Kancelarije ja	ivne adr. Switch To -	-> CIS - Kancelarije javn	e adr. 🔻		-
Traffic Applic	ation Source	Destination QoS	Conversation		
Top Source Report - 8	9.188.47.71 From:	2014-12-24 09:43 T	o: 2014-12-24 10:13 Back)	
Show IP Gro	oup by None				
Sinc 1P	Dat IP	Application	Port	Protocol	DSCP
89.188.47.71	cloudproxy405	sucuri.net http	80	TCP	Default
89.188.47.71	fra02s20-in-f7	1e100.net https	443	TCP	Default
89.188.47.71	fra02s20-in-f3	1e100.net https	443	TCP	Default
89.188.47.71	fra02s20-in-f5	1e100.net https	443	TCP	Default
89.100.47.71	ham02s13-in-f	24.1e100.n http	80	TCP	Default
89.188.47.71	showip.net	http	80	TCP	Default
Top Traffic- Destina	tion		Top Traffic- Application		т
CIS - Kancelarije ja	vne odr.🕵		CIS - Kancelarije javne	adr. 💦	CIS
355					
3%					

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.

etFlow Analyze	_				
Dashboards Dav	ices IPSLA	Cisco WAAS MediaTrac	e Security Analytics	Reports Admin	
Device Groups	Groups 🔂 Alert	Profiles 🛗 Schedule	Application / QoS Maps	🚊 User Management	Dicense Ma
CIS - Kancelarije javne	adr. Switch To>	CIS - Kancelarije javne adr.	•		
Traffic Applicatio	n Source De	estination QoS Co	nversation		
op Destination Report - 8	9.188.47.71 From	: 2014-12-24 08:17 To:	2014-12-24 14:17 Back	¢	
Resolve DNS Group	by None	*			
Src IP	Dst IP	Application	Port	Protocol	DSCP
87.98.255.2	89.188.47.71	http	80	TCP	10100
198.50.176.211	89.188.47.71	http	80	TCP	10100
173.194.113.39	89.188.47.71	https	443	TCP	10100
173.194.113.35	89.188.47.71	https	443	TCP	10100
54.214.249.27	89.188.47.71	https	443	TCP	CS4
173.194.113.37	89.188.47.71	https	443	TCP	10100
i43.127.139.110	89.188.47.71	http	80	TCP	CS4
213.186.33.2	89.188.47.71	http	80	TCP	10100
173.194.39.24	89.188.47.71	http	80	TCP	10100
63.245.217.114	89.188.47.71	https	443	TCP	CS4
23.253.100.206	89.188.47.71	http	80	TCP	C54

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.

Show IP Group by	/ None		You have gone	full screen	Exit full screen (F11)
Src IP	Dst IP	Application	fou have gone	Tull screen.	LAIL ION SCIECT (1 11/
cluster002.ovh.net	89.188.47.71	http	80	TCP	10100
cloudproxy405.sucuri.net	89.188.47.71	http	80	TCP	10100
173.194.6.78	89.188.47.71	https	443	TCP	10100
fra02s20-in-f7.1e100.net	89.188.47.71	https	443	TCP	10100
fra02s20-in-f0.1e100.net	89.188.47.71	https	443	TCP	10100
fra02s20-in-f3.1e100.net	89.188.47.71	https	443	TCP	10100
faces.eu	89.188.47.71	http	80	TCP	10100
ec2-54-214-249-27.us-w	89.188.47.71	https	443	TCP	CS4
www.securityfocus.com	89.188.47.71	http	80	TCP	CS4
fra02s20-in-f5.1e100.net	89.188.47.71	https	443	TCP	10100
fra02s20-in-f9.1e100.net	89.188.47.71	https	443	TCP	10100
cluster002.ovh.net	89.188.47.71	http	80	TCP	10100
fra02s20-in-f10.1e100.ne	89.188.47.71	https	443	TCP	10100
fra02s20-in-f2.1e100.net	89.188.47.71	https	443	TCP	10100
ham02s13-in-f24.1e100.r	89.188.47.71	http	80	TCP	10100
addons-blocklist-single2.	89.188.47.71	https	443	TCP	CS4
showip.net	89.188.47.71	http	80	TCP	C54
2.21.90.85	89.188.47.71	https	443	TCP	10100
top.lv	89.188.47.71	http	80	TCP	10100

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.

Eile	Edit View	Go Capture Analyze Sta	tistics Telephony Iools Inte	rnals <u>H</u> elp	
0	0 🥻 🔳	1 B B X 2 1	q 🔹 🗢 😜 🐺 🛓 i E	Q Q Q	L 🖸 🖌 🖼 🔨 🥵 % 🛛 😫
F	ilter:			Expression	Clear Apply Save Filter
02.1	I Channel:	Channel Offset	All Frames 👻 Wireshar	Wireless Settir	igs Decryption Keys
lo.	Time	Source	Destination	Protocol I	Length Info
	1 0.0000	00000 89.188.47.71	89.188.47.45	ICMP	98 Echo (ping) request id=0x6d10
	2 0.0000	9700 89.188.47.45	89.188.47.71	ICMP	98 Echo (ping) reply id=0x6d16
	3 0. 3792	27500 fe80::3884:9a70	:f1fff02::1:ff77:7a23	ICMPV6	86 Neighbor Solicitation for fe80
	4 0.7134	9800 69. 57. 168. 130	89.188.47.45	TCP	60 80-3000 [FIN, ACK] Seg=1 Ack=1
	5 0.7135	58300 89.188.47.45	69.57.168.130	TCP	54 3000-80 [ACK] Seg=1 Ack=2 Win=
	6 0.8453	3400 89.188.47.45	89.188.32.44	DNS	132 Standard guery 0xb5c1 PTR 9.9
	7 0.8457	2700 89.188.32.44	89.188.47.45	DNS	167 Standard query response 0xb5c1
	8 0.8463	82000 89.188.47.45	89.188.32.44	DNS	132 Standard query 0x134d PTR 3.2
	9 0.8465	58500 89.188.32.44	89.188.47.45	DNS	202 Standard query response 0x1340
	10 0.8471	0300 89.188.47.45	89.188.32.44	DNS	85 Standard query 0x3825 PTR 45.
	11 0.8473	3000 89.188.32.44	89.188.47.45	DNS	134 Standard query response 0x3825
	12 0.8479	4800 89.188.47.45	89.188.32.44	DNS	85 Standard guery 0x55b5 PTR 71.
	13 0.8481	8900 89.188.32.44	89.188.47.45	DNS	134 Standard query response 0x55b5
	14 1.0000	7800 89.188.47.71	89.188.47.45	ICMP	98 Echo (ping) request id=0x6d16
	15 1.0002	20100 89.188.47.45	89.188.47.71	ICMP	98 Echo (ping) reply id=0x6d16
1	40.4.4004	COOD &	. 24 2 2 2 0 1		or within Fililians and Eror
1					

■ Enternet 11, SrC: WICro-St_2er57:50 (00:13:00:2er57:50), US1: NewTett=_71:95:07 (00:21:3a71:95: @ Internet Protocol version 4, Src: 89.188.47.71 (89.188.47.71), DS1: 89.188.47.45 (89.188.47.45) @ Internet Control Message Protocol

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.

File	Edit View	Go Capture Analyze	Statistics Telephony Tools In	temals <u>H</u> elp				
0	•	d 🖻 🗎 🗙 🔁	् 🗢 🗢 😜 🕃 🛓 [0 👧 🕅	6	
1	Filter: (ip.ad	idr eq 89.188.47.71)		Expression	Clear	Apply	Save	Filter
02.1	11 Channel:	Channel Offset: 🔫 FCS	Filter All Frames Wireshi	ark 💽 Wireless Settin	gs Decryptio	n Keys		
lo.	Time	Source	Destination	Protocol L	ength Info			
	1 0.0000	0000 89.188.47.71	89.188.47.45	ICMP	98 Echo	(ping)	request	id=0x6d1
	2 0.0000	9700 89.188.47.45	89.188.47.71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	14 1.0000	7800 89.188.47.71	89.188.47.45	ICMP	98 Echo	(ping)	request	id=0x6d1
	15 1.0002	0100 89.188.47.45	89.188.47.71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	18 2.0001	4500 89.188.47.71	89.188.47.45	ICMP	98 Echo	(ping)	request	id=0x6d1
	19 2.0002	6900 89.188.47.45	89.188.47.71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	21 3.0002	2200 89.188.47.71	89.188.47.45	ICMP	98 Echo	(ping)	request	id=0x6d1
	22 3,0003	7400 89.188.47.45	89,188,47,71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	23 4.0002	7400 89.188.47.71	89,188,47,45	ICMP			request	id=0x6d1
	24 4,0003	9800 89.188.47.45	89,188,47,71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	25 5,0003	4100 89, 188, 47, 71	89, 188, 47, 45	ICMP			request	id=0x6d1
	26 5,0004	6600 89, 188, 47, 45	89,188,47,71	ICMP	98 Echo			id=0x6d1
	27 6,0004	1900 89.188.47.71	89,188,47,45	ICMP			request	id=0x6d1
		5300 89, 188, 47, 45	89, 188, 47, 71	TCMP	98 Echo			id=0x6d1
		8500 89, 188, 47, 71	89, 188, 47, 45	ICMP			request	id=0x6d1
			00 100 17 71	7000	00			1
					m			

B Destination: Hewlett-_71:93:0f (00:21:5a:71:93: Source: Micro-St_2e:57:56 (00:19:db:2e:57:56) Type: IP (0x0800)

Type: IP (0X0800) ⊕ Internet Protocol version 4, Src: 89.188.47.71 (89.188.47.71), Dst: 89.188.47.45 (89.188.47.45) ⊕ Internet Control Message Protocol

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

ie gun ziew go Lapture Bualyze Sta	stistics Telephony Iools Int	emals Help				
	0. 4 4 4 7 2 1		Q. 🖸 📓 🖬 🕵	* 8		
Filter: eth.dst == 00:19.db.2e57.56		· Expression	Clear Apply	Save	Filter	
	2 51	and the second s		Jare	100	
211 Channel - Channel Offset - FCS Filte		1112	ings Decryption Keys			
Time Source	Destination		Length Info			
142 92.9073560 89.188.47.45	89.188.47.71	ARP			: 00:21:5a:71:93:0f	
159 120, 917245 89, 188, 47, 45 218 148, 939016 89, 188, 47, 45	89, 188, 47, 71 89, 188, 47, 71	ARP			00:21:5a:71:93:0f 00:21:5a:71:93:0f	
218 148. 939016 89. 188. 47. 45 747 551, 802183 89, 188. 47. 45	89.188.47.71	ARP			00:21:5a:71:93:0F	
768 581, 812195 89, 188, 47, 45	89.188.47.71	ARP			: 00:21:5a:71:93:0f	
926 704,496584 89,188,47,45	89,188,47,71	ARP			00:21:5a:71:93:01	
920 704.490384 89.188.47.45 990 734.506609 89.188.47.45	89.188.47.71	ARP			: 00:21:5a:71:93:0f	
1107 764. 508632 89.188.47.45	89.188.47.71	ARP			00:21:5a:71:93:0f	
1994 796, 510787 89, 188, 47, 45	89,188,47,71	ARP			00:21:5a:71:93:0f	
2413 828, 512943 89, 188, 47, 45	89,188,47,71	ARP			: 00:21:5a:71:93:0f	
2804 860, 515101 89, 188, 47, 45	89,188,47,71	ARP			: 00:21:5a:71:93:0f	
3948 895, 517465 89, 188, 47, 45	89,188,47,71	ARP			: 00:21:5a:71:93:0f	
6099 1307.43723 89.188.47.45	89.188.47.71	ARP			00:21:5a:71:93:0f	
7478 1357, 43661 89, 188, 47, 45	89,188,47,71	ARP			: 00:21:5a:71:93:0f	
7478 1337.43001 89.188.47.43	89.188.47.71	ARP			00:21:54:71:93:06	
/33/ 1403.43384 89.188.4/.43	09.100.4/./1	Aller	42.07.100.47.	15 IS at	00121134171193101	
arrival Time: Dei 24, 2014 09 (Time shift or this packet: spoch time: 141941183, 808748 (Time sitta from previous cap (Time sitta from previous cap (Time sitta erference or firs: Frame Number: 139 Frame Length: 42 bytes (336 Decaytore Length: 42 bytes (336 Protocols in frame: etheche (Coloring Nule same: APP) (Coloring Nule string: arp) Etherment 17, Stre 80, 388, 47.43	0.00000000 seconds] 1000 seconds trured frame: 0.000037 splayed frame: 28.00088 tr frame: 120.917245000 sits) b bits) ertype:arp]	1000 seconds] 19000 seconds] 1 seconds]		7:56)		
	19:db:2e:57:56) 9:db:2e:57:56)					

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

PAKET BACK TRACK Decembar3 napad 2015.pcapng [Wireshark 1.12.2 (v1.12.2-0-g898fa22 from master-1.12)]

Filter:	i,	addr eq 8	9,188.47	.71						Exp	ression.		Clear	A	pply	Save	Filter
802.11 Chi	innel:	Chan	nel Offsi	e 14	FCS Filter:	All Fram	es	1	Wireshark	~	Wireless	s Sett	ings D	ecryptic	in Keys		
No.	Time		Source			Destin	nation				Protoc		Length	Info			
73322	754	.724246	89.18	8.47	45	89.1	188.47.	.71			ICMP			Echo	(ping)	reply	id=0x020
		775674				89.1	188.47.	45			ICMP					k request	id=0xb30
73324	754	883680	89.18	8.47	.71	89.1	188.47.	45			ICMP		78	Time	stamp r	equest	id=0xa5
73325	754	.991723	89.18	8.47	.71	89.1	188.47.	.45			ICMP					request	id=0xa40
73326	755	.099701	89.18	8.47	.71	89.1	188.47.	45			ICMP		74	Echo	(ping)	request	id=0x020
73327	755	.099819	89.18	8.47	.45	89.1	188.47.	.71			ICMP					reply	id=0x020
77263	997	186390	89.18	8.47	.71	89.1	188.47.	.45			LANMA	AN	165	Nets	hareEnu	m Request	
		3428730					188.47.				NBNS		92	Name	query	NESTAT **	00><00><
		3431110					188.47.				NBNS					response	
406	72.	3464540	89.18	8.47	.71		188.47.				NBNS		92	Name	query	NBSTAT *4	00><00><0
		3465290					L88.47.				NBNS					response	
		. 277823					188.47.				NBNS					NBSTAT **	
8150	327	.278046	89.18	8.47	.45	89.1	188.47.	.71			NBNS		217	Name	query	response	NBSTAT
		.281416					188.47.				NBNS					NESTAT #4	
8152	327	.281494	89.18	8.47	.45	89.1	188.47.	.71			NBNS		217	Name	query	response	NBSTAT

Ethernet II, Src: 0-Link_df:15:06 (00:05:5d:df:15:06), Dst: Hewlett-_71:93:0f (00:21:5a:71:93:0f) Destination: Hewlett-_71:93:0f (00:21:5a:71:93:0f) B Source: D-Link_df:13:0d (00:05:5d:df:15:06) Type: IP (00:000) Internet Protocol version 4, Src: 89.188.47.71 (89.188.47.71), Dst: 89.188.47.45 (89.188.47.45) version: 4

Header Length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport

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.

	tistics Telephony <u>T</u> ools Inter 2. 🏟 🕸 🚳 🐺 🦺 🗌		E 🖬 🖬 🥦 🐝 📖
	~~~~~Z		
Filter: ip.addr eq 89.188.47.71		<ul> <li>Expression</li> </ul>	Clear Apply Save Filter
2.11 Channel: Channel Offset: FCS Filte	er: All Frames 🛛 🗸 Wiresha	rk 👻 Wireless Sett	ings Decryption Keys
o. Time Source	Destination	Protoco + I	Length Info
77253 995, 684546 89, 188, 47, 45	89, 188, 47, 71	SMB	165 Negotiate Protocol Response
77255 996.277989 89.188.47.71	89,188,47,45	SMB	154 Session Setup AndX Request,
77256 996, 293914 89, 188, 47, 45	89.188.47.71	SMB	155 Session Setup Andx Response
77258 996.818196 89.188.47.71	89.188.47.45	SMB	131 Tree Connect Andx Request, Pa
77259 996.818433 89.188.47.45	89.188.47.71	SMB	116 Tree Connect AndX Response
77264 997.186542 89.188.47.45	89.188.47.71	SMB	105 Trans Response, Error: Out of
983 155.238232 89.188.47.71	89.188.47.45	SNMP	82 get-request 1.3.6.1.2.1.1.2.
14900 410.457084 89.188.47.71	89.188.47.45	SNMP	82 get-request 1.3.6.1.2.1.1.2.
71413 666. 504014 89. 188. 47. 71	89.188.47.45	SNMP	82 get-request 1.3.6.1.2.1.1.2.
77245 995.428327 89.188.47.71	89.188.47.45	TCP	74 48800-139 [SYN] Seq=0 win=58-
77246 995.428510 89.188.47.45	89.188.47.71	TCP	74 139-48800 [SYN, ACK] Seq=0 A
77247 995.428846 89.188.47.71	89.188.47.45	TCP	66 48800-139 [ACK] Seq=1 Ack=1 V
77250 995.456158 89.188.47.71	89.188.47.45	TCP	66 48800-139 [ACK] 5eq=77 Ack=5
77254 995.684727 89.188.47.71	89.188.47.45	TCP	66 48800-139 [ACK] Seg=245 Ack=
77257 996, 294245 89, 188, 47, 71	89,188,47,45	TCP	66 48800-139 [ACK] Seg=333 Ack=

■ thermat 11, Src: D-LINE_OTISIOB (00/05/S0/0713/500), UST: Hewlett-__1195/07 (00/21/S0/1195/07) B Ostination: Hewlett-_1193/07 (00/21/S0/17)/51/07) B Source: D-Link_df:15/06 (00/05/S0/df:15/d6) Type: IP (0x0800) B Internet Protocol Version 4, Src: 89.188.47.71 (89.188.47.71), Dst: 89.188.47.45 (89.188.47.45) Version: Version 4, Src: 89.188.47.71 (89.188.47.71), Dst: 89.188.47.45 (89.188.47.45) Comparison (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (1990) (19

Header Length: 20 bytes B Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport Fotal Length: 78 Edentification: 0x0000 (0)

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 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.

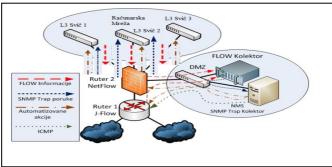


Fig. 9. Elements of control systems - Connecting switches, Flow collector

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.

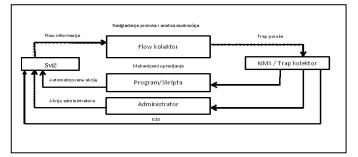


Fig. 10. Schematic representation of management process

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

<pre>set WshShell = WScript.CreateObject("WScript.Shell")</pre>
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 " <b>ctrl+e</b> "

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

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}"

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 -1 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}" 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}"

and all and a	NAPAD Iz MREŻE	le authorite le monte					
escription :	Detekcija nedozvoljenih protokol	a,apleacija iz mreze					
elect Source :	Interface     IP Groups	<ul> <li>Interface Group</li> </ul>					
Selected Interfac	es : All Interfaces (Modify Selection)	)					
efine Alert Criteria	IN Traffic OUT	Traffic					
Port/Protoco	I 🖉 DSCP 🖉 Application 🛛	IP Address					
Port/Protocol :	ICMP * 3020	(Ex: 1001 or 1200-1600)					
		(Ex: 1001 or 1200-1600)					
Port/Protocol : [ DSCP : 101001		(Ex: 1001 or 1200-1600)					
	۲	(Ex: 1001 or 1200-1600)					
DSCP : 101001 Application : IC	• MP_App •						
DSCP : 101001 Application : IC	• MP_App •		Packets				
DSCP : 101001 Application : IC	• MP_App •		Packets minutes; Severity	Warning •	SNMP Trap •	Server-Port:Community	
DSCP : 101001 Application : IC efine Thresholds a	MP_App     v  Ind Action : Utilization	Volume Speed .				Server:Port:Community admin@ac.me	
DSCP : 101001	MP_App	Volume Speed *	minutes; Severity				
DSCP : 101001 Application : IC three Thresholds : 98	MP_App	Volume Speed *	minutes; Severity				
DSCP : 101001 Application : IC fine Thresholds a \$98 \$98 Add More	MF_tep     MF_tep     md Action :     Utilization     Packets 5     Packets 10	Volume Speed *	minutes; Severity				

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.

lescription :	Detekcija nedozvoljenih protokol	a antiferentia in menina				
rescription :	Decekcija ned ozvoljenih procokol	ajapiikaoja iz mreze				
elect Source :	Interface     IP Groups	<ul> <li>Interface Group</li> </ul>				
Selected Inter	faces : All Interfaces (Modify Selection)					
efine Alert Crit	erla : 🛛 🔿 IN Traffic 🔍 OUT	Traffic				
Port/Prot	ocol 🗹 DSCP 🗹 Application 🛛	IP Address				
Port/Protocol	: TCP	(Ex; 1001 or 1200-1600)				
PorgPlotocol		(ex. 1001 0 1200-1800)				
DSCP : CS4	•					
Application :	http •					
		Volume     Speed	Packets			
efine Threshold		Volume     Speed     Jimes 5	Packets minutes; Severity (Wami	ing V SNMP Trap	Server:Port:Co	mmunity
lefine Threshold	fs and Action : Utilization				Server:Port:Co     admin@ac.me	
lefine Threshold	is and Action : Utilization	times 5	minutes; Severity Wami			
efine Threshold	Is and Action : Utilization	times 5	minutes; Severity Wami			
lefine Threshold X 1 X 5 Add More	Is and Action : Utilization	times 5 times 10	minutes; Severity [Wami minutes; Severity [Wami			
lefine Threshold X 1 X 5 Add More	is and Action : Utilization	times 5 times 10	minutes; Severity [Wami minutes; Severity [Wami			
lefine Threshold X 1 X 5 Add More E-mail Subje	is and Action : Utilization	times 5 times 10	minutes; Severity [Wami minutes; Severity [Wami			
Lefine Threshold X 1 X 5 Add More E-mail Subje Lusiness Hour A	Is and Action : Utilization UB • 4 UB • 10 Utilization UB • 1 UB • 1 UB • 10 Utilization UB • 10 Utilization UB • 10 Utilization UB • 10 Utilization	times (5 times (10 nalyzer \$deviceName\$ \$interfaceN	minutes; Severity Wami minutes; Severity Wami ame\$			
Lefine Threshold     X 1     X 5     Add More     E-mail Subje     tusiness Hour A     From : 00:00	Is and Action : Utilization Ut	bimes 5 bimes 10 nalyzer Sdeviceliames Sinterfacellu openca • TimeZo	minutes; Severity Wami minutes; Severity Wami ame\$			

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.

Alert Profile Name	NAPAD iz MREŻE 2		
Description :	Detekcija nedozvoljenih pro	otokola,aplikacija iz mreže	
Select Source :	Interface     IP Gr	roups 🕕 Interface Group	
Selected Interfac	ces : All Interfaces (Modify Sele	ection)	
Define Alert Criteri	a: ) IN Traffic ()	OUT Traffic	
P. Dart (Dertor)	al 🖲 DSCP 🖉 Application	IP Address	
· POLYPIOLOO	a coch c whitenou	<ul> <li>IP ADUTESS</li> </ul>	
Port/Protocol : [	UDP • 51413	Ex: 1001 or 1200-1600	1)
Port/Protocol : [	UDP • 51413	Ex: 1001 or 1200-1600	3)
Port/Protocol : [ DSCP : Default		Ex: 1001 or 1200-1600	3)
DSCP : Default	۲	Ex: 1001 or 1200-1606	3)
	۲	Ex: 1001 or 1200-1600	0)
DSCP : Default Application : L	<ul> <li>ARP_App</li> </ul>		
DSCP : Default Application : L	* ARP_A00 *		
DSCP : Default Application : L	<ul> <li>ARP_App</li> </ul>		
DSCP : Default Application : L Define Thresholds (	ARP_App  and Action :      () Utilization	ion 🕜 Volume 🔿 Speed	d 🕡 Padets
DSCP : Default Application : L Define Thresholds :	ARP_App	ion Volume O Speed	6 Podets minutes; Severty (Critical • Staff Ing •) Server-Port:Community

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 countermeasures 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.