Proposal of Software System for Automatic Reaction to Network Anomalies and Real Time Data Capturing

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

1. 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. [1]

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[2].

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. 2.Digital Forensic

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 [3].

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.

3. 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 gradific investigation depending on the

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.

4. 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.

	Reports Admin	Security Analytics	WAAS MediaTrao	IPSLA Cisco W	Deshboards Devices
Dicense Mana	QUser Management	Application / QoS Maps	Schedule 2	ps DAlert Profiles	ice Groups 🛛 🗿 IP Grou
	te danige trainigencies produce en	/ •			5 - Kancelarije javne adr.
		orsation	ion QoS Conv	Source Destinatio	Traffic Application
		12-24 10:13 Back	24 09:43 To: 2014	71 From: 2014-12-2	ource Report - 89.188.47
				lane 🔹	w IP Group by [
DSCP	Protocol	Port	Application	d IP	IP D
Default	TCP	80	at http	udpraxy405.sucuri.net	188.47.71 de
Default	TCP	443	t https	02s20-in-f7.1e100.net	188.47.71 fra
Default	TCP	443	t fillps	02s20-in-f3.1e100.net	188.47.71 fra
Default	TOP	443	t https	02s20-in-f5.1e100.net	188.47.71 fr
Default	TCP	80	.n http	m02s13-in-f24.1e100.r	100.47.71 ha
Default	TCP	80	http	owip.net	188.47.71 sh
		Traffic• Application	Τα		Traffic- Destination
c	<u>a</u>	i - Kancelarije javne adr.	CI	R	5 - Kancalarije javne odr.

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.

Dashboards Day	ices IPSLA Ci	sco WAAS MediaTrac	e Security Analytics	Reports Admin	
Device Groups	Groups Alert Pro	files 🛗 Schedule	Application / QoS Maps	S 🚨 User Management	Dicense Ma
CIS - Kancelarije javne	e adr. Switch To> [CIS - Kancelarije javne odr.	•		
Traffic Applicatio	n Source Dest	ination QoS Co	nversation		
op Destination Report - I	89.188.47.71 From: 2	014-12-24 08:17 To:	2014-12-24 14:17 Ba	ck	
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
143.127.139.110	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	CS4

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 coroon Ex	tit full screen (F11)
Src IP	Dst IP	Application	fou have gone	run screen.	at rull screen (F11)
cluster002.ovh.net	89.188.47.71	http	80	TCP	101001
cloudproxy405.sucuri.net	89.188.47.71	http	80	TCP	101001
173.194.6.78	89.188.47.71	https	443	TCP	101001
ra02s20-in-f7.1e100.net	89.188.47.71	https	443	TCP	101001
fra02s20-in-f0.1e100.net	89.188.47.71	https	443	TCP	101003
ra02s20-in-f3.1e100.net	89.188.47.71	https	443	TCP	101001
faces.eu	89.188.47.71	http	80	TCP	101001
ac2-54-214-249-27.us-wi	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	101003
ra02s20-in-f9.1e100.net	89.188.47.71	https	443	TCP	101001
duster002.ovh.net	89.188.47.71	http	80	TCP	101001
ra02s20-in-f10.1e100.ne	89.188.47.71	https	443	TCP	101001
ra02s20-in-f2.1e100.net	89.188.47.71	https	443	TCP	101001
ham02s13-in-f24.1e100.r	89.188.47.71	http	80	TCP	101001
addons-blocklist-single2.	89.188.47.71	https	443	TCP	CS4
showip.net	89.188.47.71	http	80	TCP	CS4
2.21.90.85	89.188.47.71	https	443	TCP	101001
op.lv	89.188.47.71	http	80	TCP	101001
NO DE N		4.45		an atom	

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	Statistics Telephony Iools Int	ternals Help	
0	۰ 🧉 🔳	1 / I B B X C	9, * * 9 7 2 1	I Q Q Q	. 🖸 📽 🗹 🥵 %
8	ilten			Expression	Clear Apply Save Filter
02.1	I Channel:	+ Channel Offset: + FCS	Filten All Frames 👻 Wiresha	urk 💌 Wireless Settin	igs Decryption Keys
о.	Time	Source	Destination	Protocol I	ength Info
-	1 0.000	00000 89.188.47.71	89.188.47.45	IGMP	98 Echo (ping) request id=0x6d16
	2 0,000	09700 89.188.47.45	89.188.47.71	ICMP	98 Echo (ping) reply id=0x6d16
	3 0.379	27500 fe80::3884:9a	70:f1fff02::1:ff77:7a23	ICMPV6	86 Neighbor Solicitation for fe80
	4 0.713	49800 69. 57. 168. 130	89.188.47.45	TCP	60 80-3000 [FIN, ACK] Seg=1 Ack=1
	5 0.713	58300 89.188.47.45	69.57.168,130	TCP	54 3000-80 [ACK] Seg=1 Ack=2 win=
	6 0.845	33400 89.188.47.45	89.188.32.44	DNS	132 Standard query 0xb5c1 PTR 9.9
	7 0.845	72700 89.188.32.44	89.188.47.45	DNS	167 Standard query response 0xb5c1
	8 0.846	32000 89.188.47.45	89.188.32.44	DNS	132 Standard query 0x134d PTR 3.2
	9 0, 846	58500 89.188.32.44	89.188.47.45	DNS	202 Standard guery response 0x134d
	10 0.847	10300 89.188.47.45	89.188.32.44	DNS	85 Standard guery 0x3825 PTR 45.
	11 0.847	33000 89.188.32.44	89.188.47.45	DNS	134 Standard query response 0x3825
	12 0.847	94800 89.188.47.45	89.188.32.44	DNS	85 Standard guery 0x55b5 PTR 71.
	13 0.848	18900 89.188.32.44	89.188.47.45	DNS	134 Standard guery response 0x55b5
	14 1.000	07800 89.188.47.71	89.188.47.45	ICMP	98 Echo (ping) request id=0x6d16
	15 1.000	20100 89.188.47.45	89.188.47.71	ICMP	98 Echo (ping) reply id=0x6d16
ie.	40 4 400	+		Trees.C	or visitin Filsiphieses for from

Ethernet II, SrC: Micro-St_2e:57:56 (00:19:db:2e:57:56), DSt: Hewlett-_71:93:0f (00:21:5a:71:93:0f) © Internet Protocol Version 4, SrC: 89.188.47.71 (89.188.47.71), DSt: 89.188.47.45 (89.188.47.45) © Internet Courtol Nessage 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.

0	0 🖌 🔳 🖉	(🖻 🛯 🗙 😂	Q & # # # # # !			0 👧 🖇		
Fil	ter: (ip.add	Ir eq 89.188.47.71)		Expression C	lear	Apply	Save	Filter
02.11	Channel:	Shannel Offset	Filter All Frames Vires	hark 💽 Wireless Settings.	Decryptic	n Keys		
10.	Time	Source	Destination	Protocol Leng				
		000 89.188.47.71		ICMP			request	1d=0x6d1
		700 89.188.47.45	89.188.47.71	ICMP	98 Echo			id=0x6d1
		800 89.188.47.71	89.188.47.45	ICMP			request	id=0x6d1
		100 89.188.47.45	89.188.47.71	ICMP	98 Echo			id=0x6d1
		500 89.188.47.71	89.188.47.45	ICMP			request	id=0x6d1
		900 89.188.47.45	89,188.47.71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	21 3.00022	200 89.188.47.71	89.188.47.45	ICMP	98 Echo	(ping)	request	id=0x6d1
	22 3.00037	400 89.188.47.45	89.188.47.71	ICMP	98 Echo			id=0x6d1
	23 4.00027	400 89.188.47.71	89,188,47.45	ICMP	98 Echo	(ping)	request	id=0x6d1
	24 4.00039	800 89.188.47.45	89,188,47.71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	25 5.00034	100 89.188.47.71	89, 188, 47, 45	ICMP	98 Echo	(ping)	request	id=0x6d1
	26 5.00046	600 89.188.47.45	89.188.47.71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	27 6.00041	900 89.188.47.71	89.188.47.45	ICMP	98 Echo	(ping)	request	id=0x6d1
	28 6.00055	300 89.188.47.45	89,188,47,71	ICMP	98 Echo	(ping)	reply	id=0x6d1
	35 7,00048	500 89.188.47.71	89,188,47,45	ICMP	98 Echo	(ping)	request	id=0x6d1
-	35 7 00000			Trun	00	1-1-1		33
()					m			

□ Source: MiCro-St_2ef3rise (U013940012ef3rise) Type: IP (0x0800) □ Internet Protocol Version 4, Src: 89,188,47,71 (59,188,47,71), 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

		9. * * * * * *		N 🖸 🖌 🖬 🕅 🥵 🐰	A	
Filter	eth.dst = = 00:19:db:2e57:56		Expression	Clear Apply	Save	Filter
211 Chan	net - Channel Offset - FCS F	ilten All Frames - Wires	herk 💌 Wireless Set6	Decryption Keys		
	ime Source	Destination	Protocol	Length Info		
142 9	2.9073560 89.188.47.45	89.188.47.71	ARP			00:21:5a:71:93:0f
1.59	20.917245 89.188.47.45	89.186.47.71	ARP			00:21:5a:71:93:0f
218 2	48.939016 89.188.47.45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
	51.802183 89.188.47.45	89.188.47.71	ARP			00:21:5a:71:93:0f
768 5	81.812198 89.188.47.45	89.188.47.71	ARP			00:21:5a:71:93:0f
926 7	04.496584 89.188.47.45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
990 7	34.506609 89.188.47.45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
1107 7	64. 508632 89. 188. 47. 45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
1994 7	96.510787 89.188.47.45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
2413 8	28. 512943 89. 188. 47. 45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:98:0f
2804 8	60. 515101 89. 188. 47. 45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
3948 8	195, 517465 89, 188, 47, 45	89.188.47.71	ARP	42 89, 188, 47, 45	is at	00:21:5a:71:93:0f
6099 7	307,43723 89,188,47,45	89,188,47,71	ARP	42 89, 188, 47, 45	is at	00:21:5a:71:93:0f
7478 3	357.43661 89.188.47.45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
7557.1	405.43984 89.188.47.45	89.188.47.71	ARP	42 89.188.47.45	is at	00:21:5a:71:93:0f
				19		
Inte	rface id: 0 (\Device\NP psulation type: Etherne	t (1)	814E-CGEEE83D3FE	9})		
Inte Enca Arri [Tin Epoc [Tin Fram Fram Capt [Fra [Fra [Fra [Fra [Fra [Col] Ethern © Col [Col] Ethern © Ad	rface id: 0 (\Device\NP	E	B146-C0688B303FE trtal Europe Stan 17000 seconds] 8889000 seconds] 000 seconds] 000 seconds] 001 seconds] 001 seconds] 001 seconds] 001 seconds] 001 seconds] 001 seconds]	9)) dard Time 71 (00:19:db:2e:57 (factory default) cast)	56)	

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

Filter: ip.addr eq 89.188.47.71		Expression Clear Apply Save Filt	er
2.11 Channel Channel Offset: V FCS	Filten All Frames Wireshart	k 🗸 Wireless Settings Decryption Keys	
o. Time Source	Destination	Protoco + Length Info	
73322 754.724246 89.188.47.45	89.188.47.71	ICMP 74 Echo (ping) reply id=	0x0
73323 754.775674 89.188.47.71	89,188,47,45	ICMP 60 Address mask request id=	0xb3
73324 754.883680 89.188.47.71	89.188.47.45	ICMP 78 Timestamp request id-	oxa!
73325 754.991723 89.188.47.71	89,188,47,45	ICMP 78 Information request id-	0xa4
73326 755.099701 89.188.47.71	89.188.47.45		Ox0
73327 755.099819 89.188.47.45	89.188.47.71	ICMP 74 Echo (ping) reply id=	0x02
77263 997.186390 89.188.47.71	89.188.47.45	LANMAN 165 NetShareEnum Request	
404 72.3428730 89.188.47.71	89.188.47.45	NBNS 92 Name query NBSTAT * <dd><</dd>	
405 72.3431110 89.188.47.45	89.188.47.71	NBNS 217 Name query response NBST	
406 72.3464540 89.188.47.71	89.188.47.45	NBNS 92 Name query NBSTAT *<00><	
407 72.3465290 89.188.47.45	89.188.47.71	NBNS 217 Name query response NBST	
8149 327.277823 89.188.47.71	89.188.47.45	NBNS 92 Name query NBSTAT *<00><	
8150 327.278046 89.188.47.45	89.188.47.71	NBNS 217 Name query response NBST	
8151 327.281416 89.188.47.71	89.188.47.45	NBNS 92 Name query NBSTAT *<00><	
8152 327.281494 89.188.47.45	89.188.47.71	NBNS 217 Name query response NBST	AT
	(The black on black over	red (736 bits) on interface 0	
Ethernet II, Src: D-Link_df:			

Version: 4 meader Length: 20 bytes B Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport

Total Length: 78 Identification: 0x0000 (0)

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.

0 6	1	1 A		×2	0.4	命日	3	2		10	R Q	Q	2		8	36	1		
Filte	c	p.addr.eq	89.188.47.1	n					v	Expe	ession	e	Clear		Apply		Save	Filter	
02.11 0	hannel:	Cha	nnel Offset	FCS	Filter: All F	rames		Wires	hark	~	Wireless	Setti	ngsill	Decr	yption K	eys			
ło.	Time		Source		D	estinatio	n				Protoco	+ 1	ength	In	fo				_
772	3 995	. 684 54	5 89, 188	.47.45	8	9.188	47.7	1			SMB		16	5 N	egotia	te Pr	otoco	1 Respon	se
772	5 996	. 27798	9 89.188	.47.71	8	9.188	47.4	5			SMB							X Reques	
772	6 996	. 29391	\$ 89.188	.47.45	8	9.188	47.7	1			SMB							x Respon	
772	8 996	. 81819	6 89.188	.47.71	8	9.188	47.4	5			SMB		13	1 T	ree co	nnect	Andx	Request	, Pa
772	9 996	. 81843	3 89.188	.47.45	8	9.188	47.7	1			SMB		11	6 T	ree Co	nnect	Andx	Respons	e
772	4 997	.18654.	2 89.188	.47.45	8	9.188	47.7	1			SMB		10	5 T	rans R	espor	nse, E	rror: Ou	c 01
91	3 155	. 23823	2 89.188	.47.71	8	9.188	47.4	5			SNMP		8	2 9	et-req	uest	1.3.6	.1.2.1.1	.2.0
1490	0 410	.45708	4 89.188	.47.71	8	9.188	47.4	5			SNMP		8	2 9	et-req	uest	1.3.6	.1.2.1.1	.2.1
7143	3 666	. 50401	4 89.188	.47.71	8	9.188	47.4	5			SNMP							.1.2.1.1	
772	5 995	.42832	7 89.188	.47.71	8	9.188	47.4	5			TCP		7	4 4	8800-1	39 [:	SYN] S	eq=0 win	=584
772	6 995	.42851	0 89.188	.47.45	8	9.188	47.7	1			TCP		7	4 1	39-488	00 [:	SYN, A	CK] Seq=	J A
7724	7 995	.42884	6 89.188	.47.71	8	9.188	47.4	5			TCP							eq=1 Ack	
			8 89.188		8	9.188	47.4	5			TCP							eq=77 Ac	
			7 89.188			9.188					TCP							eq=245 A	
772	7 996	. 29424	5 89.188	.47.71	8	9.188	47.4	5			TCP		6	64	8800+1	39 [/	ACK] 5	eq=333 A	ck=1
	mo 40	1. 02 1	mitter of	a wire	(736 bit	c) 0	hir	as rar	THEOR	1 (7	26 bit	(a)	on i	nt o	rfare	0			

Type: IP (DX0800) Internet Protocol Version 4, Src: 89.188.47.71 (89.188.47.71), Dst: 89.188.47.45 (89.188.47.45) Version: 4

Version: 4 Header Length: 20 bytes ⊡ Offferentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport Total Length: 78 Identification: 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.

5. 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.

6. 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[6] 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 а 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 bestknown 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:

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

2) 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:

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

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

5) <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.

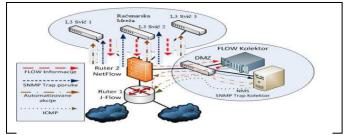


Figure. 9. Elements of control systems - Connecting switches, Flow collector

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.

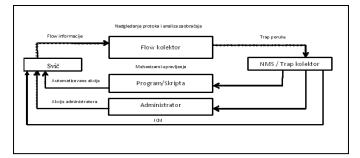


Figure. 10. Schematic representation of management process

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

set WshShell = WScript.Create	Object("WScrip	t.Sh	ell")	
WshShell.Run "C:\putty.exe	89.188.47.45	-1	user	-pw
password"				
WshShell.Run "C:\Program Fi	les\Wireshark>V	Wire	shark.	exe"
WshShell.AppActivate "89.188	8.47.45 - Wiresh	ıark	••	
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 -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 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 "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}"

lert Profile Name :	NAPAD IZ MREŻE						
escription :	Detekcija nedozvoljenih proto	kola,aplikacija iz mreže					
elect Source :	Interface IP Grou	ps 💿 Interface Group					
Selected Interfac	es : All Interfaces (Modify Selecti	on)					
Jefine Alert Criteria	t O IN Traffic O O	UT Traffic					
Port/Protoco	B DSCP 🖉 Application	IP Address					
Port/Protocol :	CMP • 3020	(Ex: 1001 or 1200-1600)					
		(Ex: 1001 or 1200-1600)					
Port/Protocol : DSCP : 101001		(Ex: 1001 or 1200-1600)					
	T	(Ex: 1001 or 1200-1600)					
DSCP : 101881	•] MP_App •						
DSCP : 101801 Application : 10	•] MP_App •		activits				
DSCP : 101001 Application : 10	•] MP_App •			Warning •	SNMP Trap	• Server:Part:Community	c]
DSCP : 101001 Application : 16 Define Thresholds a	MP_App Ind Action : Utilization	🖸 Volume 😔 Speed 🔹 9				Server:Port:Community admin@acme	t
DSCP : 101001 Application : 10 Define Thresholds a	MP_App	Volume © Speed & P times 0.4	minutes; Severity				6
DSCP : 101001 Application : 10 efine Thresholds 4 × 98 × 98 Add More	MF_App	Volume Speed # P time 0.4 time (0.4	minutes; Severity				
DSCP : 101001 Application : 10 efine Thresholds 4 × 98 × 98 Add More	MF_App	Volume © Speed & P times 0.4	minutes; Severity				

Figure. 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.

	NAPAD iz MREŻE 1				
Description :	Detekcija nedozvoljenih protoko	la,aplikacija iz mreže			
Select Source :	Interface IP Groups	📋 Interface Group			
Selected Interface	es : All Interfaces (Modify Selection)			
Define Alert Criteria	: O IN Traffic O UT	Traffic · Combined			
R Port/Protocol	€ 2 DSCP € Application □	IP Address			
Port/Protocol : 1	TCP # 80	(Ex: 1001 or 1200-1600)			
DSCP : CS4	۲				
Application : http://www.application.com					
Appacation ; ind	Þ •				
		Volume O Speed O	Packets		
		Volume Speed times 5		MMP Trag • Server:Port:Community	
Define Thresholds a	nd Action : O Utilization				
Define Thresholds a	nd Action : Utilization	times 5	minutes; Severity Warning 🔹 S		
Define Thresholds a X1 X5 Add More	Ind Action : Utilization	times 5 times 10	minutes; Severity (Warning •) S minutes; Severity (Warning •) E		
Define Thresholds a X1 X5 Add More	nd Action : Utilization	times 5 times 10	minutes; Severity (Warning •) S minutes; Severity (Warning •) E		
Define Thresholds a X 1 X 5 Add Hore E-mail Subject :	Ind Action : Utilization	times 5 times 10	minutes; Severity (Warning •) S minutes; Severity (Warning •) E		
Define Thresholds a X 1 X 5 Add More E-mail Subject : Business Hour Alerts	Ind Action : Utilization UB • 4 UB • 10 (Sseveritys) Mert From NetFlore A	times 5 times 20 nalyzer \$deviceName\$ \$interfaceN	minutes; Severitry Warning • B minutes; Severitry Warning • E minutes; Severitry Warning • E		

Figure. 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.

	: NAPAD iz MREŻE 2						
Description :	Detekcija nedozvoljenih pr	utokola,aplikacija iz mreže					
Select Source :	🖲 Interface 🛛 🕕 IP G	roups 📋 Interface Group					
Selected Interf	aces : All Interfaces (Modify Sele	ection)					
Define Alert Criter	ia : 🔵 IN Traffic 🛛 🔘	OUT Traffic					
Port/Proto	col 🗹 DSCP 🗷 Application	IP Address					
Port/Protocol :	UDP • 51413	(Ex: 1001 or 1200-1600	0				
Port/Protocol : DSCP : Defaul		(Ex: 1001 or 1200-1600	0				
		(Ex: 1001 or 1200-1600	0				
		(Ex: 1001 or 1200-1600	1				
DSCP : Defaul	L V						
DSCP : Defaul Application :	LARP_App And Action : Utilization	ion Volume Speed	1 💮 Fackets				
DSCP : Defaul Application :	L V			al • SNNP Tra	p • Server:PortsCo	mmunity	
DSCP : Defaul Application : lefine Thresholds	LARP_App And Action : Utilization	ion Volume Speed	1 💮 Fackets		p • Server:Port:Co • Jadmin@ac.me		
DSCP : Defaul	I V	ion Volume Speed times 3	Packets minutes; Severity Cminutes				

Figure. 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.

7.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.

Reference

- Reith M., Carr C., Gunsch G., An examination of digital forensics, International Journal of Digital Evidence, 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.

[6] https://www.manageengine.com/network-performancemanagement.html?MEtab Manage Engine NetFlow Analyzer

- [7] Gareth Knight, The Forensic Curator: Digital Forensics as a Solution to Addressing the Curatorial Challenges Posed by Personal Digital Archives, The International Journal of Digital Curation Volume 7, Issue
- [8] Carrier, B. & Spafford, E.H. (2003). Getting physical with the digital investigation process. International Journal of Digital Evidence, https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/2003-29.pdf
- [9] Craiger, J.P. (n.d.). Computer forensics procedures and methods. <u>http://ncfs.org/craiger.forensics.methods.procedures.final.pdf</u>
- [10] Eiland, E.E. (2006). Time line analysis in digital forensics. Retrieved from <u>http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.147.3864</u>