FORENSIC NETWORK ANALYSIS AND IMPLEMENTATION OF SECURITY ATTACKS ON VIRTUAL PRIVATE SERVERS

Naikson Fandier Saragih¹, Ridho Agus Wery Nanda Panjaitan², Mufria Jonatan Purba³ ^{1,2,3}Indonesian Methodist University Jalan Hang Tuah No. 8 Medan, North Sumatra 20152 E-mail :saragihnaikson@gmail.com

ABSTRACT- PT Kodinglab Integrasi Indonesia's Virtual Private Server (VPS) product requires good quality standards, including security. The challenge that arises is still frequent disruptions to the protection of PT Kodinglab's VPS customers, where it is difficult to identify the source of the attack. Network forensics in the form of dead forensics and live forensics using the NIST method with the stages of collection, examination, Analysis, and reporting are used to find the source of the attack. Data for dead forensics comes from snort tools, and data for live forensics comes from capture Wireshark. The collection stage involves collecting attack data from snort logs and wireshark for life forensics. While the examination dataset stages are further analyzed and mapped. Advanced check on the server via syslog snort.

From the attack testing carried out to obtain information in the form of the attacker's IP address, destination IP address, date of the attack, server time, and type of attack from testing the TCP Flooding and UDP Flooding attacks, all attacks on the customer's VPS can be identified. The information obtained regarding the attacker is in the form of the date and time the attack occurred, the attacker's IP address and the victim's IP address, and the protocol used.

Keywords: Network Forensic, Dead Forensic, Live Forensic, Virtual Private Server, DDos, TCP Flooding, UDP Flooding.

1. INTRODUCTION

PT Kodinglab Integrasi Indonesia, one of the services provided is a Virtual Private Server for customers. Standards are prepared with high quality, especially regarding security in providing information system development and development services, mobile applications, development, egovernment, and digital marketing. The Kodinglab server has now accommodated manv web applications from government and private agencies using the Docker VPS. However, throughout 2021 there will still be 20 percent security disturbances to VPS customers of PT Kodinglab, such as backdoor and DDoS attacks.

Virtual Private Server (VPS) is one of the virtual machines provided for Internet hosting service providers that have been widely used since 2010. VPS is widely used when website traffic is high and requires more resources when you need Private IP access and create applications based on file sharing, streaming, and MAP Server [1]. Therefore VPS is inseparable from the target of security attacks.

To find the source of the attack, Network Forensic can be used by monitoring and analyzing resource usage, amount of traffic, and user activity, whether the server is on or off. The results of research using Network Forensic, investigators can quickly detect an attack and identify the attacker. In addition, the data needed for the investigation process, among other things, the date of the attack, the time of the attack, the attacker's MAC Address, the attacker's IP Address, the Victim's MAC Address, and the Victim's IP Address, can be presented quickly and precisely so that they can assist investigators in making decisions on attacks that occur [2].

2. RESEARCH CONTENT 2.1 Method

1. Network Forensics

The Network forensic section is a science that focuses on an area of computer networks and devices connected to a network to find attacker information and to find evidence of an attack on a computer network [3]. Network forensic stages can be seen in the image below:



a. collection

At this stage, it is carried out to obtain evidence according to procedures approved by the authorities. **b. examination**

At this stage, it checks the activities that occur on the server.

c. Analysis

This stage is to analyze the data that has been collected to obtain estimates, possibilities, and hypotheses from the data that will potentially become evidence.

d. reporting

All of the steps above must be well documented to facilitate making reports easy for others to understand from the investigative activities carried out.

1. Live Forensic and Dead Forensic

Live forensics collects data/evidence of attack when the server is on, while Dead forensics collects data/evidence of attack when the server is down, which is generally stored in the snort log.

2. Research Flow

To carry out the research process to get the desired results, the researcher makes a research flow, as shown in Figure 2



Figure 2 Research Flow

3. System Implementation

This stage is carried out by hiring a VPS service from the herza.id provider. The rented system is a KVM VPS with the Ubuntu 20.04 operating system. The specifications for the VPS that are rented can be seen in Table 1

Table1. KVM VPS S	Specifications
CPU used	Cores 4
Ram	4GB
Hard drive	80GB
Bandwidth	Unlimited
Operating system	Ubuntu 20.04

Table1. KVM VPS Specifications

Next, install snort on the ubuntu server. Figure 3 shows that snort has been successfully installed.

Running in packet dump mode									
== Initializing Snort == Initializing Output Plugins! pcap DAQ configured to passive. Acquiring network traffic from "eth0". Decoding Ethernet									
== Initialization Complete ==									
-*> Snort! <*- o")> Version 2.9.7.0 GRE (Build 149) >>> Wartia Rosch & The Snort Team: http://www.snort.org/contact#team Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved. Copyright (C) 1998-2013 Sourcefire, Inc., et al. Using libpcap version 1.9.1 (with TPACKET_V3) Using PCRE version: 8.39 2016-06-14 Using ZLIB version: 8.39 2016-06-14									

Figure 3 Display snort that has been installed

The image above shows the version of snort that has been successfully installed, namely snort version 2.9.7.0.

In Snort, there are rules to Snort's intelligence. The set rules provide intelligence at the time of protection, so the investigation process depends on rule intelligence. The script below is a configuration for saving rules. When there is an attack on the server, the log of the attack will be stored in the rules.

Next, add the DDoS attack rules with the script below

=> Added snort rules

alert ICMP any any -> \$HOME_NET any (msg:"ICMP Flooding"; detection_filter:tracks by_src, count 30, second 60; sid1000006; rev2;) alert tcp any -> \$HOME_NET any (msg: there is a "TCP Flooding" attack; detection_filter: tracks by_src, count 30, second 60; sid1000006; rev2;) alert udp any any -> \$HOME_NET any (msg:"UDP Flooding"; detection_filter:tracks

by_src, count 30, second 60; sid1000003; rev1;) From the script above, the following results from changing the snort rules configuration script on the ubuntu server. Figure 4 is the snort rules



Figure 4 Configuring snort rules.

Next, we will install Wireshark on the client's PC. The following shows the Wireshark that has been successfully installed, shown in Figure 5

Welcome to Wireshark		
Capture		
using this filter: Enter a cap	ure filter	*
Wi-Fi: en0 awdl0	a.A.	
Thunderbolt Bridge: bridge0	_	
Thunderbolt 1: en1 Thunderbolt 2: en2		
p2p0		
Loopback: Io0		

Figure 5 Wireshark View

2.4 Attack Testing

After implementing the system and configuring it, we then carry out attack testing to see the snort's response in monitoring data packets. For testing, only use attacks*Distributed Denial of Service*(DDoS) TCP Flooding and UDP Flooding using the LOIC application. Previously carried out the following tests. The CPU performance conditions in the initial state on the VPS overview tab are shown in Figure 6

Q Overview	🛎 Graphs	Settings	Section 24 Applications	🛓 Install	
Disk Usage 6.6 % 4.48 / 80 GB		_	Bandwidth 0 % 1 / 👁 GB		_
CPU					
1% Used					

Figure 6 Initial CPU conditions

From the picture above, the results of the initial CPU condition data are obtained, which are shown in Table 2

Table 2Initial	CPU state	description
----------------	-----------	-------------

No	Tab	Information
1	Disk Usage	5.6%
2	Bandwidth	0%
3	CPUs	1% Used

To get network forensic results in finding evidence of security attacks, researchers conducted DDoS TCP Flooding and UDP Flooding attack scenarios, with information in Table 3

Table 3Attack Testing

No	Attack Type	Number of Tests
1	TCP Flooding	2
2	UDP Flooding	2

1. TCP Flooding

An attack that targets the TCP protocol takes advantage of the protocol connection connected to the

server by flooding it(*flooding*), Experiments in making TCP connections that are accommodated by the server in buffers are indeed different. Still, the capacity is no more than a few hundred connections. By sending many packets into TCP, the buffer location will be full and cause the server to not work properly. Figure 7 is a test of the TCP Flooding attack.

Figure 7 Types of TCP Flooding attacks

In the test above, the LOIC application sends many TCP packets so the server becomes full of these packets. With the "top" tools on Linux Ubuntu, network performance for the CPU and memory running on the server can be seen in Figure 8.

root@fikom-metho top - 21:32:14 u Tasks: 141 total %Cpu(s): 22.4 us MiB Mem : 3935 MiB Swap: 2047	dist: p 16 , 1 , 9. , 9. .7 to .0 to	~# to days, runn 4 sy, tal, tal,	p 12:10, ing, 13 0.0 n 1862. 2047.	1 use 8 sleep i, 66.9 3 free, 0 free,	r, load ing, 2 id, 0.0 496.1 0.0	average stopped wa, 6 used, used.	e: 3.0 d, 0 0.0 hi 157 311	0, 1.08, 0.40 zombie , 1.1 si, 0.3 st 7.3 buff/cache 2.0 avail Mem	^
PID USER	PR	NI	VIRT	RES	SHR S	%CPU	%MEM	TIME+ COMMAND	
85683 root			37548	13788	3348 S	4.7	0.3	4:00.36 snort	
85837 root			37548	13708	3264 S	4.7	0.3	4:04.30 snort	
86918 root			37548			4.7	0.3	4:05.05 snort	
118650 root			37548	13760	3320 S	4.7	0.3	2:26.87 snort	
118668 root			37548	13804		4.7	0.3	2:27.92 snort	
11735 root			37548	13700	3260 S	4.3	0.3	11:38.74 snort	
26050 root			37548		3316 S	4.3	0.3	7:24.69 snort	
26070 root			37548	13656	3216 S	4.3	0.3	7:21.46 snort	
31464 root			37548	13812	3380 S	4.3	0.3	7:05.42 snort	
33480 root			37548		3256 S	4.3	0.3	6:55.10 snort	
. 33549 root			37548	13652	3208 S	4.3	0.3	6:51.69 snort	
34196 root			37548	13804	3368 S	4.3	0.3	7:04.17 snort	
85802 root			37548	13780	3348 S	4.3	0.3	3:59.07 snort	
85813 root			37548	13840	3404 S	4.3	0.3	4:02.19 snort	
86009 root			37548		3280 S	4.3	0.3	3:59.33 snort	
694 www-data			56792	7176			0.2	0:56.23 nginx	
693 www-data			57892	8276			0.2	1:41.00 nginx	
696 www-data			56560	6944	4292 S		0.2	0:40.70 nginx	
695 www-data	20	0	56196	6492	4204 S	0.7	0.2	0:03.59 nginx	V

Figure 8 Server performance during a TCP Flooding attack

From the picture above, the CPU performance is at 22.4%

2. UDP Flooding

Testing the second attack is done by testing DDoS attacks with UDP-type *flooding* using the LOIC application. Figure 9 is a test of the UDP Flooding attack.



Figure 9 UDP Flooding type attack

The test above shows that LOIC floods User Datagram Protocol requests by sending many UDP packets so that the server becomes full. CPU performance and server memory with the "top" command on Linux ubuntu can be seen in Figure 10

I	root@fik	com-n	ethodis	t:~# t							
1	top - 21	L:38:	37 up 1	.6 days	;, 12:17,		r, load a	averag	e: 9.2	5, 2.78, 1.12	
ł	Tasks: 1	L43 t	otal.	1 rur	ning, 13	7 sleep	ing, 5 s	stoppe	d, 0	zombie	
1	%Cpu(s):	42.	3 us, l	.4.1 sy	/, 0.0 m	i, 40.8	id, 0.0		0.0 hi	, 2.7 si, 0.1 st	
1	MID Mem		2922.7	total,	1859.	5 free,	498.9	used,	157	7.3 buff/cache	
	MiB Swap		2047.0	total,	2047.	0 free,	0.0	used.	310	9.2 avail Mem	
									0/10000		
	PID	USER	P	R NI	VIRT	RES	SHR S	%CPU	%MEM	TIME+ COMMAND	
I	118743	root	: 2	0 0	37548	13696	3256 S	10.6	0.3	2:31.40 snort	
I	26050	root	: 2	0 0	37548	13752	3316 S	9.9	0.3	7:28.76 snort	
I	11401	root		0 0	37548	13804	3364 S	9.6	0.3	11:45.47 snort	
I	85837	root		0 0	37548	13708	3264 S	9.6	0.3	4:08.48 snort	
I	86009	root		0 0	37548	13724	3280 S	9.6	0.3	4:03.51 snort	
I	86918	root			37548			9.6	0.3	4:09.30 snort	
I	94484	root			37548	13796		9.6	0.3	4:20.97 snort	
I	85813	root			37548	13840	3404 S	9.3	0.3	4:06.41 snort	
I	118668	root			37548	13804		9.3	0.3	2:32.27 snort	
I	33549	root			37548	13652	3208 S	8.9	0.3	6:55.87 snort	
I	85671	root			37548	13708	3268 S	8.9	0.3	4:03.12 snort	
I	86867	root			37548	13864	3436 S	8.9	0.3	4:06.96 snort	
I	26085	root			37548	13916	3488 S	8.6	0.3	7:17.34 snort	
I	31464	root			37548	13812	3380 S	8.6	0.3	7:09.81 snort	
I	33480	root			37548	13692	3256 S	8.6	0.3	6:59.29 snort	
I	33994	root			37548	13696	3256 S	8.6	0.3	7:10.33 snort	
1	34196	root			37548	13804	3368 S	8.6	0.3	7:08.38 snort	
1	85683	root	: 2	0 0	37548	13788	3348 S	8.6	0.3	4:04.48 snort	
1	85802	root	: 2	0 0	37548	13780	3348 S	8.6	0.3	4:03.12 snort	

Figure 10 Server performance during a UDP Flooding attack

From the picture above, CPU performance has increased to 42.3%.

2.5 Network Forensics

Network forensics includes a collection stage to deal with previously committed crimes. The focus is to prevent future crimes.

2.3.1 Collections

Retrieval of attack data using snort tools, namely on*logs*snort and also on wireshark for life forensics From the attack, testing is carried out to obtain information in the form of the attacker's IP address, destination IP address, date of the attack, server time, and type of attack.

1. Log Data on Snort

a. Log Data for TCP Flooding attacks

Attack trials were carried out twice, each shown in Figure 11 and Figure 12. The first attack was carried out on June 9, and the second on June 16, 2022



Figure 11 Snort log testing the first TCP Flooding attack

From Figure 11 above, it can be seen that the attacker's IP address is 114.5.144.36, the attack date is June 6, the server was attacked at 09.50, with id 46407, and the type of attack is a TCP attack.



Figure 12 Snort log of TCP attack testing Second flooding

From figure 12 above, it can be seen that the attacker's IP address is 36.72.172.243, the attack date is June 16, the server was attacked at 16.58, with id 64318, and the type of attack is a TCP attack.

b. Log Data for UDP Flooding attacks

Attack trials were carried out twice, respectively shown in **figure 13** and figure 14. the first attack was carried out on June 16, and the second on June 18, 2022



Figure 13 Snort log of the first UDP Flooding attack test

From Figure 13 above, it can be seen that the attacker's IP address is 36.72.172.243, the attack date is June 16, the server was attacked at 17.19, with id 62484, and the type of attack is a UDP attack.



Figure 14 Snort log testing the second UDP Flooding attack

From Figure 14 above, it can be seen that the attacker's IP address is 114.5.147.54, the attack date is June 18, the server was attacked at 12.32, with id 20264, and the type of attack is a UDP attack.

2. Log Data on Wireshark

In this tool, you can see the time, source, destination, and also protocol used.

a. TCP Flooding attack logs

Attack trials were carried out twice, with the results shown in Figure 15 and Figure 16, respectively.

N	.	Time	Source	Destination	Protocol	Length Info
	50360	17.015343	103.160.63.239	192.168.1.82	TCP	54 80 + 50832 [ACK] Seq=1 Ack=5006 Win=501 Len=0
	50361	17.015344	103.160.63.239	192.168.1.82	TCP	66 80 + 50835 [ACK] Seq=1 Ack=4986 Win=501 Len=0 SLE=4841 SRE=4946
	50362	17.015344	103.160.63.239	192.168.1.82	TCP	54 80 + 50851 [ACK] Seq=1 Ack=4971 Win=501 Len=0
Г	50363	17.015660	192.168.1.82	103.160.63.239	TCP	89 50969 + 80 [PSH, ACK] Seq=596 Ack=329 Win=131072 Len=35 [TCP segment of a
	50364	17.015711	192.168.1.82	103.160.63.239	TCP	149 50002 + 80 [PSH, ACK] Seq=4991 Ack=1 Win=512 Len=95 [TCP segment of a reas
	50365	17.015740	192.168.1.82	103.160.63.239	TCP	89 50832 + 80 [PSH, ACK] Seq=5071 Ack=1 Win=512 Len=35 [TCP segment of a reas
	50366	17.015771	192.168.1.82	103.160.63.239	TCP	89 50835 → 80 [PSH, ACK] Seq=5051 Ack=1 Win=512 Len=35 [TCP segment of a reas
	58367	17.015883	103.160.63.239	192.168.1.82	TCP	54 80 + 50930 [ACK] Seq=329 Ack=2046 Win=64128 Len=0
	50368	17.015883	103.160.63.239	192.168.1.82	TCP	54 80 + 50781 [ACK] Seq=1 Ack=4976 Win=501 Len=0
	50369	17.015884	103.160.63.239	192.168.1.82	TCP	66 80 + 50976 [ACK] Seq=329 Ack=196 Win+64128 Len+0 SLE=6 SRE=151
Т	- 50370	17.015884	103.160.63.239	192.168.1.82	TCP	54 80 + 50781 [RST, ACK] Seq=1 Ack=4976 Win=501 Len=0
	50371	17.015884	103.160.63.239	192.168.1.82	TCP	54 80 + 50849 [ACK] Seq=1 Ack=4991 Win=501 Len=0
	50372	17.015885	103.160.63.239	192.168.1.82	TCP	54 80 + 50890 [ACK] Seq=329 Ack=3551 Win=64128 Len=0
	50373	17.015885	103.160.63.239	192.168.1.82	TCP	54 80 + 58834 [ACK] Seq=1 Ack=4991 Win=501 Len=0
	50374	17.015885	103.160.63.239	192.168.1.82	TCP	54 80 + 50929 [ACK] Seq=329 Ack=2041 Win=64128 Len=0
	58375	17.015886	103.160.63.239	192.168.1.82	TCP	54 80 + 50920 [ACK] Seq=329 Ack=2296 Win=64128 Len=0
	50376	17.015886	103.160.63.239	192.168.1.82	TCP	66 80 + 50910 [ACK] Seq=329 Ack=2866 Win=64128 Len=0 SLE=2681 SRE=2861
	58377	17.016538	192.168.1.82	103.160.63.239	TCP	149 50930 → 80 [PSH, ACK] Seq=2046 Ack=329 Win=131072 Len=95 [TCP segment of a
	50378	17.016588	192.168.1.82	103.160.63.239	TCP	89 58976 + 80 [PSH, ACK] Seq=261 Ack=329 Win=131072 Len=35 [TCP segment of a

Figure 15 Data packet for testing the TCP Flooding attack first

From Figure 15, it can be seen that the attacker's IP number 50363 is 192.168.1.82, the destination IP is 103.160.63.239, and the protocol is TCP.

No.	Time	Source	Destination	Protocol	Length Info
	638 9.614911	103.160.63.239	192.168.252.6	TCP	54 80 → 65407 [ACK] Seq=329 Ack=9729 Win=64128 Len=0
	639 9.615086	192.168.252.6	103.160.63.239	TCP	246 65410 → 80 [PSH, ACK] Seq=6817 Ack=329 Win=131328 Len=192 [TCP segme
	640 9.615219	192.168.252.6	103.160.63.239	TCP	214 65407 → 80 [PSH, ACK] Seq=9729 Ack=329 Win=131328 Len=160 [TCP segment
	641 9.615368	103.160.63.239	192.168.252.6	TCP	54 80 → 65411 [ACK] Seq=329 Ack=3713 Win=64128 Len=0
	642 9.615445	192.168.252.6	103.160.63.239	TCP	246 65411 → 80 [PSH, ACK] Seq=3713 Ack=329 Win=131328 Len=192 [TCP segment
	643 9.615586	103.160.63.239	192.168.252.6	TCP	54 80 → 65406 [ACK] Seq=329 Ack=9729 Win=64128 Len=0
	644 9.615667	192.168.252.6	103.160.63.239	TCP	246 65406 + 80 [PSH, ACK] Seq=9729 Ack=329 Win=131328 Len=192 [TCP segmer
	645 9.654831	103.160.63.239	192.168.252.6	TCP	54 80 → 65408 [ACK] Seq=329 Ack=9857 Win=64128 Len=0
-	646 9.654961	192.168.252.6	103.100.03.239	TCP	
	647 9.661753	192.168.252.6	103.160.63.239	TCP	1078 [TCP Retransmission] 65412 + 80 [PSH, ACK] Seq=33 Ack=329 Win=131328
	648 9.666117	103.160.63.239	192.168.252.6	TCP	54 80 → 65409 [ACK] Seq=329 Ack=9889 Win=64128 Len=0
	649 9.666247	192.168.252.6	103.160.63.239	TCP	214 65409 → 80 [PSH, ACK] Seq=9889 Ack=329 Win=131328 Len=160 [TCP segment
	650 9.672635	103.160.63.239	192.168.252.6	TCP	54 80 → 65410 [ACK] Seq=329 Ack=7009 Win=64128 Len=0
	651 9.672766	192.168.252.6	103.160.63.239	TCP	214 65410 → 80 [PSH, ACK] Seq=7009 Ack=329 Win=131328 Len=160 [TCP segment
	652 9.672957	103.160.63.239	192.168.252.6	TCP	54 80 → 65411 [ACK] Seq=329 Ack=3905 Win=64128 Len=0
	653 9.673033	192.168.252.6	103.160.63.239	TCP	214 65411 → 80 [PSH, ACK] Seq=3905 Ack=329 Win=131328 Len=160 [TCP segment
	654 9.673143	103.160.63.239	192.168.252.6	TCP	54 80 → 65406 [ACK] Seq=329 Ack=9921 Win=64128 Len=0
	655 9.673216	192.168.252.6	103.160.63.239	TCP	214 65406 + 80 [PSH, ACK] Seq=9921 Ack=329 Win=131328 Len=160 [TCP segment
	656 9.673338	103.160.63.239	192.168.252.6	TCP	54 80 → 65407 [ACK] Seq=329 Ack=9889 Win=64128 Len=0
۲					

Figure 16 Data packet testing the second TCP Flooding attack

From the picture above, it can be seen that the attacker's IP at number 647 is 192.168.252.6, the destination IP is 103.160.63.239, and the protocol is TCP.

b. UDP Flooding Attack Data Logs

Attack trials were carried out twice, with the results shown in Figure 17 and Figure 18, respectively.

	Current filter: udp									
No.		Time	Source	Destination	Protocol	Length	Info			
	1	0.000000	192.168.1.48	103.160.63.239	UDP	74	52187 → 80 Len=32			
	2	0.000003	192.168.1.48	103.160.63.239	UDP	74	52145 -> 80 Len=32			
r	3	0.000015	192.168.1.48	103.160.63.239	UDP	74	52215 → 80 Len=32			
P	4	0.000018	192.168.1.48	103.160.63.239	UDP	/4	52200 → 80 Len=32			
	5	0.000124	192.168.1.48	103.160.63.239	UDP	74	52185 → 80 Len=32			
	6	0.000126	192.168.1.48	103.160.63.239	UDP	74	52147 → 80 Len=32			
	7	0.000129	192.168.1.48	103.160.63.239	UDP	74	52223 → 80 Len=32			
	8	0.000193	192.168.1.48	103.160.63.239	UDP	74	52198 → 80 Len=32			
	9	0.001265	192.168.1.48	103.160.63.239	UDP	74	52177 → 80 Len=32			
	10	0.001268	192.168.1.48	103.160.63.239	UDP	74	52131 → 80 Len=32			
	11	0.001271	192.168.1.48	103.160.63.239	UDP	74	52212 → 80 Len=32			
	12	0.001274	192.168.1.48	103.160.63.239	UDP	74	52143 → 80 Len=32			
	13	0.001380	192.168.1.48	103.160.63.239	UDP	74	52189 → 80 Len=32			
	14	0.001381	192.168.1.48	103.160.63.239	UDP	74	52139 → 80 Len=32			
	15	0.001384	192.168.1.48	103.160.63.239	UDP	74	52175 → 80 Len=32			
	16	0.001444	192.168.1.48	103.160.63.239	UDP	74	52164 → 80 Len=32			
	17	0.001446	192.168.1.48	103.160.63.239	UDP	74	52160 → 80 Len=32			
	18	0.001449	192.168.1.48	103.160.63.239	UDP	74	52182 → 80 Len=32			
	19	0.001507	192.168.1.48	103.160.63.239	UDP	74	52178 → 80 Len=32			

Figure 17 First UDP Flooding attack test data packet

From the picture above, it can be seen that the attacker's IP at number 3 is 192.168.1.48, the destination IP is 103.160.63.239, and the protocol is UDP.

Apply a display filter <ctrl-></ctrl->									
No.	Time	Source	Destination	Protocol	Length Info				
Г	1 0.000000	192.168.252.6	103.160.63.239	UDP	74 61561 → 80 Len=32	Ī			
	2 0.000030	192.168.252.6	103.160.63.239	UDP	/4 63/0/ → 80 Len=32	1			
	3 0.001299	192.168.252.6	103.160.63.239	UDP	74 56799 → 80 Len=32				
	4 0.001301	192.168.252.6	103.160.63.239	UDP	74 63940 → 80 Len=32				
	5 0.001309	192.168.252.6	103.160.63.239	UDP	74 65449 → 80 Len=32				
	6 0.001310	192.168.252.6	103.160.63.239	UDP	74 52130 → 80 Len=32				
	7 0.001364	192.168.252.6	103.160.63.239	UDP	74 52128 → 80 Len=32				
	8 0.001366	192.168.252.6	103.160.63.239	UDP	74 65453 → 80 Len=32				
	9 0.001366	192.168.252.6	103.160.63.239	UDP	74 52126 → 80 Len=32				
	10 0.001403	192.168.252.6	103.160.63.239	UDP	74 64330 → 80 Len=32				
	11 0.001403	192.168.252.6	103.160.63.239	UDP	74 52463 → 80 Len=32				
	12 0.001404	192.168.252.6	103.160.63.239	UDP	74 61543 → 80 Len=32				
	13 0.002390	192.168.252.6	103.160.63.239	UDP	74 60748 → 80 Len=32				
	14 0.002391	192.168.252.6	103.160.63.239	UDP	74 52129 → 80 Len=32				
	15 0.002401	192.168.252.6	103.160.63.239	UDP	74 63933 → 80 Len=32				
	16 0.002402	192.168.252.6	103.160.63.239	UDP	74 62796 → 80 Len=32				
	17 0.002453	192.168.252.6	103.160.63.239	UDP	74 49425 → 80 Len=32				
	18 0.002454	192.168.252.6	103.160.63.239	UDP	74 61558 → 80 Len=32				
	19 0.002455	192.168.252.6	103.160.63.239	UDP	74 63709 → 80 Len=32				
/									

Figure 18 Data packet testing the second UDP Flooding attack

From the picture above, it can be seen that the attacker's IP at number 1 is 192.168.252.6, the destination IP is 103.160.63.239, and the protocol is TCP.

2.3.2 Examination

The data obtained forms a dataset and can be analyzed and mapped. Inspection is not only carried out on attack data obtained previously using tools but also checking activities carried out on the server via Syslog, as shown in Figure 19.



Figure 19 syslog snort

From the picture above, the activities carried out on the server will be recorded by snort, which is stored in the syslog, so from the information obtained, the server user can see irregularities if at any time there is a misuse of the server.

2.3.2 Analysis

From stages *collection*, evidence of an attack has been obtained, and a series of information obtained is evidence that there was an attack on the server, the server's IP address. The Analysis was carried out in the form of a comparison of CPU performance in the initial state before the attack was carried out and when the attack was carried out.

1. Server conditions on ubuntu in an initial state

To see the condition of the CPU on the server, you can do it with the "top" command. After the command is entered, we will see the results, as shown in figure 20 below.

root@fikom-methodist:~# top											
top - 1			days,			rs, load	avera	ge: 0.	00, 0.36,	0.59	
Tasks:	145 total,	1		ing, 14	1 sleep:	ing, 3	stoppe	d, 0	zombie		
%Cpu(s)	: 0.0 us,	0	1 sy,	0.0 n	i, 99.8	id, 0.0		0.0 hi	, 0.0 si	, 0.1 st	
MiB Mem	: 3935.	7 to	otal,	1810.	1 free,	544.9	used,	158	0.6 buff/		
MiB Swa	p: 2047.0	ð to	otal,	2047.	0 free,	0.0	used.	306	2.6 avail	Mem	
PID	USER	PR	NI	VIRT	RES	SHR S	%CPU	%MEM	TIME+	COMMAND	
11401	root	20	0	37548	13804	3364 S	0.3	0.3	12:35.08	snort	
37957	Debian-+			34764	13300	8136 S	0.3	0.3	10:25.89	snmpd	
85837				37548	13708	3264 S				snort	
86009	root			37548	13724	3280 S	0.3	0.3	4:52.09	snort	
118668				37548	13804		0.3	0.3		snort	
137396	root					0 I	0.3	0.0	0:34.75	kworker+	
141758	root	20	0	11832	3644	3144 R	0.3	0.1	0:00.07	top	
1	root			168464	12676	8384 S	0.0	0.3	0:43.91	systemd	
2	root					0 S	0.0	0.0	0:00.55	kthreadd	
3							0.0	0.0	0:00.00	rcu gp	
4	root					0 I	0.0	0.0	0:00.00	rcu par+	
6							0.0	0.0	0:00.00	kworker+	
8	root					0 I	0.0	0.0	0:00.00	mm perc+	
9						0 S	0.0	0.0	0:00.28	ksoftir+	
10	root					0 1	0.0	0.0	24:01.31	rcu sch+	
11	root					0 S	0.0	0.0	0:07.88	migrati+	

Figure 20 The initial condition of the server on ubuntu

From the picture above, it can be seen that the initial condition of the CPU is at 0.0%, and it can be concluded that the performance of the CPU is in a normal state.

2. Server condition on ubuntu after an attack

After the attack was carried out, the CPU performance condition increased, namely at 43.6%, which can be seen in Figure 21 below.

top - 07:39:46	up 23 days	s, 22:18,	, 2 use	rs, load	averag	e: 12	.29, 9
Tasks: 175 tot	al, 30 ru	nning, 1	39 sleep	ing, 6	stopped	I, 0	zombie
%Cpu(s): 43.6	us, 12.9 s	/, 0.0 r	ni, 39.9	id, 0.0	wa, 0	.0 hi	, 3.4
MiB Mem : 39	35.7 total	, 1625.	.9 free,	647.3	used,	166	2.5 but
MiB Swap: 20	47.0 total	, 2047.	0 free,	0.0	used.	295	6.1 ava
PID USER	PR NI	VIRT	RES	SHR S	%CPU	%MEM	TIN
31464 root	20 0	37548	13812	3380 R	9.0	0.3	17:16.
208947 root	20 0	37548	13692	3248 R	9.0	0.3	2:39.
94484 root	20 0	37548	13796	3352 R	8.6	0.3	14:31.
118650 root	20 0	37548	13760	3320 R	8.6	0.3	12:35.
11410 root	20 0	37548	13772	3336 R	8.3	0.3	21:37
33480 root	20 0	37548	13692	3256 R	8.3	0.3	16:55.
34196 root	20 0	37548	13804	3368 R	8.3	0.3	17:02.
85813 root	20 0	37548	13840	3404 R	8.3	0.3	13:52.
85837 root	20 0	37548	13708	3264 R	8.3	0.3	13:55.
86918 root	20 0	37548	13792	3352 R	8.3	0.3	13:55.
118668 root	20 0	37548	13804	3372 R	8.3	0.3	12:45.
118743 root	20 0	37548	13696	3256 R	8.3	0.3	12:41.
195353 root	20 0	37548	13804	3372 R	8.3	0.3	3:14.
11401 root	20 0	37548	13804	3364 R	8.0	0.3	21:37
11735 root	20 0	37548	13700	3260 R	8.0	0.3	21:31.
33540 root	20 0	37548	13760	3324 R	8.0	0.3	16:43.
33549 root	20 0	37548	13652	3208 R	8.0	0.3	16:42.
Figure 21	Server (onditi	on on	uhuntu	after	• test	ing

Figure 21 Server condition on ubuntu after testing the attack

Thus it can be concluded that the increase in CPU conditions before and after the attack was carried out was 43.6%.

2.3.2 Reports

1. Snort data source

The results of the investigation with the snort data source have occurred on the PVS server from several computers whose complete information can be seen in Table 4

in Table 4										
	Table 4 Details of data packets from Snort									
N	Attacke	The	Destinati	Serv	ID	Prot				
	r Ip	date	on IP	er		ocol				
		of		Tim		s				
		the		e						
		attac								
		k								
1	114.5.1	June	103.160.	09:5	46	TCP				
	44.36	9	63.239	0:00	40					
					7					
2	36.72.1	June	103.160.	16:5	64	ТСР				
	72.243	16	63.239	8:45	31					
					8					
3	36.72.1	June	103.160.	17:1	62	UDP				
	72.243	16	63.239	9:55	48					
					4					
4	114.5.1	June	103.160.	12:3	20	UDP				
	17 51	18	63 230	2.03	26					

2. Wireshark data source

The results of the investigation with the Wireshark packet capture data source have been an attack on the PVS server from several computers whose complete information can be seen in Table 5

Table 5 Details of data packets from Wireshark

Ν	time	Attacker IP	Destination	Protoc
0			IP	ols
1	17.0158	192.168.1.	103.160.63.	ТСР
	84	82	239	

2	9.66.17	192.168.25	103.160.63.	TCP
	53	2.6	239	
3	0.00001	192.168.1.	103.160.63.	UDP
	5	48	239	
4	0.00000	192.168.25	103.160.63.	UDP
	0	2.6	239	

3 CONCLUSION

Conclusion

Some conclusions are as follows:

- 1. *Network forensics*, in collecting evidence of attacks in the snort log, has been able to identify the attacker in the form of the date of the attack, the time of the attack, the attacker's IP address, the victim's IP address, ID, the protocol can be presented quickly and precisely so that it can assist investigators in making decisions about the attack that occurred on a Virtual Private Server (VPS).
- 2. Collecting evidence of attacks on wireshark log data in the form of the time of the attack, attacker's IP address, victim's IP address, and protocol.

4 CLOSING

In this study, the authors provide suggestions that are necessary for the next development, which are as follows:

- 1. The research is also expected to provide recommendations for tools used in detecting and identifying attacks and attackers.
- 2. Future research also examines how to handle and prevent security attacks on Virtual Private Servers (VPS).

BIBLIOGRAPHY

- Dewi, EK (2017). Snort Log Analysis Using Network Forensic. JIPI (Scientific Journal of Informatics Research and Learning), 2(2), 72–79. https://doi.org/10.29100/jipi.v2i2.370
- [2] Eka, R., Rachman, A., & Wahyu, T. (2010). Virtual Private Server (VPS) as an Alternative to a Dedicated Server. Seminar on Intelligent Technology and Its Applications, SITIA, 2–7.
- [3] Fahana, J., Umar, R., & Ridho, F. OUERY: Journal (2017). of Information Systems Volume: 01, Number: 02, October 2017 ISSN 2579-5341 (online) Utilization of Telegram as an Attack Notification for Network Forensics Purposes QUERY: Journal of Information Systems Volume: 01, Number: 02, October 2017. 5341(October), 6-14.
- [4] Hae, Y. (2021). Network Security Analysis on the Web From Sniffing

Attacks With Experimental Methods. JATISI (Journal of Informatics Engineering and Information Systems), 8(4), 2095–2105. https://doi.org/10.35957/jatisi.v8i4.11 96

- [5] Hafizh, MN, Riadi, I., & Fadlil, A. (2020). Network Forensics Against ARP Spoofing Attacks using the Live Forensic Method. Journal of Telecommunications and Computers, 10(2), 111. https://doi.org/10.22441/incomtech.v1 0i2.8757
- [6] Hunt, R., & Zeadally, S. (2012). Network Forensics: An Analysis of. 36–43.
 Information, FT (2017). Network Forensic On Cloud-Based Networks Scientific Articles. 672011212.
- [7] Kamajaya, GEA, Riadi, I., Prayudi, Y., & Dahlan, UA (2020). STATIC FORENSICS INVESTIGATION ANALYSIS OF MAN ATTACK IN THE ARP POISONING BASED ON MAN IN THE MIDDLE ATTACK IN STATIC. 3(1), 6–12. https://doi.org/10.33387/jiko
- [8] Muhyidin, A., & Yogyakarta, UN (2019). Ubuntu Server Fundamentals book (Ubuntu Camp 2016). march. Nasution, MAH, & Laksono, AT (2020). Investigation of Backdoor Remote Access Trojan (RAT) Attacks Against Smartphones. 7(4), 505–510. https://doi.org/10.30865/jurikom.v7i4. 2301
- [9] Netsec. (2020). Implementation of Snort as a detection tool. https://netsec.id/snortnids/%0Ahttps://media.neliti.com/med ia/publications/142056-IDimplementasi-snort-As-alatpendetek.pdf%0A
- [10] Supervisor, D., Hidayanto, BC, & Information, DS (2018). NETWORK FORENSIC ANALYSIS OF DOS ATTACKS ON UBUNTU SERVER OPERATING SYSTEMS 16. 04 DAN MICROSOFT WINDOWS SERVER 2016 NETWORK FORENSICS ANALYSIS ON DOS ATTACK AT UBUNTU SERVER 16. 04 AT MICROSOFT WINDOWS SERVER.
- [11] <u>http://blog.iweb.com/en/2010/06/why-should-i-care-about-sql-server/4772.html</u>, March 17, 2011, 19.30