The Effectiveness of Snort in DDoS Attack Scenarios

Tartila Izzatillah[1*], Fania[2], Zulfata[3]
[1]Faculty of Engineering, Universitas Serambi Mekkah, Aceh, Indonesia
[2]Faculty of Engineering, Universitas Serambi Mekkah, Aceh, Indonesia
[3] Faculty of Engineering, Universitas Serambi Mekkah, Aceh, Indonesia

*Corresponding Author: tarizzatt@gmail.com

Abstract. *Secure communication in the digital era has become important, while on the other side, the attackers are consistently moving into more sophisticated ways to acquisition their target. DDoS was one of the common techniques used with the aim to overwhelm and make their target fail to function. To detect and limit this threat, many organizations have begun to employ various solutions. However, it was discovered that several solutions are ineffective while others require high cost for the implementation, and this has become a challenge for medium to low-sized organizations to meet their business strategy. For that, in this study we aim to introduce Snort as an open source solution for the network detection and prevention system to determine how it undertakes and performs analysis especially in terms of accuracy and speed in relation to DDoS attacks. This study will involve a few steps: first, to establish the simulation environment; secondly, to perform the simulation with a DDoS attack to allow Snort to capture the traffic and respond according to the pre-established rules; and finally, to measure and evaluate the result. Snort as an open-source product, in nature allows the public to contribute, and that becomes the advantage compared with other commercial products especially in detecting anomalies. This will help the administrators to react more quickly by having an accurate information with an earlier warning system. Additionally, Snort is affordable, making it good choice for the organization.*

Keywords: *Network, security, DDoS, Snort.*

1. Introduction

In this digital era, the internet has become an essential part of communities. With the rapid development of the internet, make it suitable to be accessed from everywhere through various platforms and devices, and this creates opportunity for the perpetrators to gain advantage through the network for their criminal activities, making it very likely to become unsecure, and this is not only an effect on the individual users but even more on a bigger scale, such as an organization or business, thus it becomes necessary to have an adequate protection on the network.

Today network security is essential for safeguarding data and systems, preserving data availability and integrity, and guaranteeing privacy and data protection (Iftikhar et al., 2023; Nandy et al., 2024). In order to safeguard their data and systems from cyberattacks and other security risks, businesses must pay close attention to network security as technology advances and the number of cyberthreats consistently increas. Various types of attacks have been demonstrated by the attacker that resulted in huge loses to their target. Distributed Denial of Service (DDoS) attacks are one of the common methods that are widely used with the intention to create a serious disruption to the services or make a network system amalfunction (Akhir et al., 2016). On September 6 and 7, 2019, Wikipedia had a noteworthy DDoS attack that made the website unavailable in Germany and numerous other European countries. Users reported having trouble accessing Wikipedia pages as a result of the severe service outages produced by this attack (Cavanagh, 2019). This incident demonstrates how susceptible international information platforms are to hacks and how they affect users all over the world.

Derived from such a situation, it has become essential to have an appropriate defence to detect DDoS using methods like Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS) (Nalayini et al., 2024). These systems will act as a major barrier prior to successfully entering the target network. IDS as a native solution has the main function of identifying threats within a network. This system monitors the traffic and forwards the alerts to the respective authorities for any suspicious activities detected. However, IDS have the drawback that they could not take immediate action to block threats, therefore, manual intervention is still required to address the detected issues (Nandy et al., 2024). In contrast, IPS not only identifies the threats but is also able to automatically block them. IPS actively works to stop attacks before they can cause damage, thereby reducing the need for user intervention during detection (Kurniawan & Prakoso, 2020). Here are various solutions available in the market offering network protection with anti- DDoS as one of their key features, where this solution often requires a high cost of investment, which may not be suitable particularly for medium-low organization due to their resources limitations (Iftikhar et al., 2023).

Snort, then introduce into the industry as an open-source software which very well-known among cybersecurity professionals (Adiwal et al., 2023). It offers free tools with solid capability to analyze the network traffic and identify any suspicious packet according to the pre-defined signature, including DDoS. Snort is the world's leading open-source Intrusion Prevention System (IPS). Snort IPS uses a series of rules that help define harmful network activities and employs these rules to find matching packets and generate alerts for users (Awal & Gusman, 2023).

With the above background, this study aims to assess and analyze the accuracy of Snort, including the response time against such events. To offer alternative solutions that can be easily adopted not only for network protection but also can be enhanced to provide a new experience towards the notification mechanism.

2. Method

In this study, we will conduct a simulation on the isolated network to minimize the impact. We employed Kali Linux to deliver the attack, Ubuntu with Snort installed to act as middleman to capture and analyze the logs, and Windows Server 2019 as the target. In this simulation we enabled the notifications by embedding the Telegram - API to send the notification during the events.

2.1 Building Environment

In the context of this study, we developed a simple network diagram as displayed in “Figure. 1,”. To visualize the system architecture and data flow as well as the relationships between components.

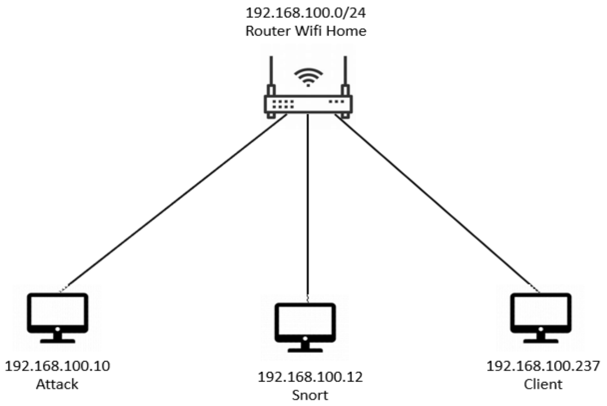
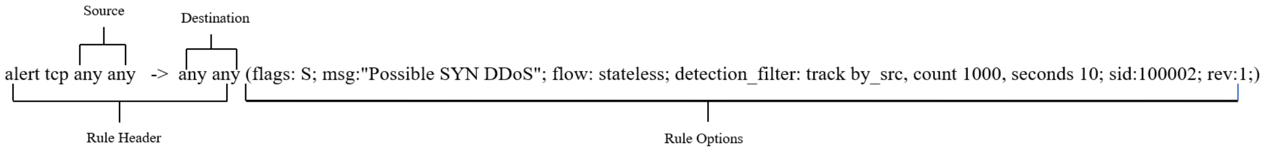


Figure 1. Diagram Network

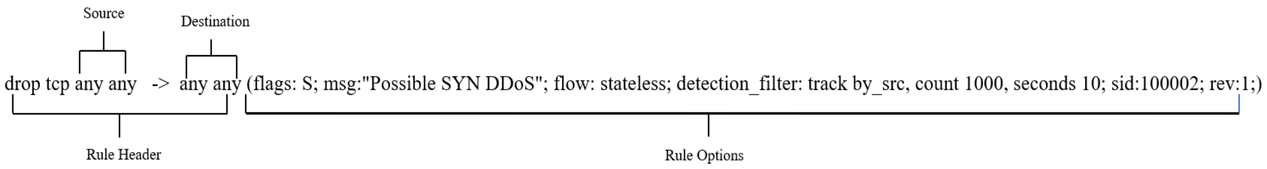
As our simulation basis, various rules have been applied to maximize and gain a proper throughput from Snort. These rules are designed to specifically identify various patterns to match the flag defined and respond to it accordingly. The rule header is a key component of every Snort rule, serving to define the basic characteristics of the traffic to be analyzed. Within this header, there are several important elements, including the rule action, which determines the action to be taken when certain conditions are met, as well as the protocol that indicates the type of communication being monitored, such as TCP. Furthermore, the source and destination sections in the rule header are also crucial as they indicate the source and destination IP addresses of the target analysis. This helps Snort determine where an attack may originate from and where it is targeted. After the header, there are rule options that provide further details on how data packets should be examined. These options allow users to set specific criteria, such as matching content in the payload or configuring other parameters relevant to DDoS attack detection (Turner & Joseph, 2017). This explanation will provide deeper insights into how network monitoring systems can be optimized to protect infrastructure from cyber threats. Below is an example of a rule header using an alert "Figure. 2".



```
alert tcp any any  ->  any any (flags: S; msg:"Possible SYN DDoS"; flow: stateless; detection_filter: track by_src, count 1000, seconds 10; sid:100002; rev:1;)
```

Figure 2. Snort Rule for SYN

Alert rules are an important instrument used as an IDS by Snort. These rules are designed to quickly and effectively identify and respond to abnormal conditions in network traffic "Figure. 3.".



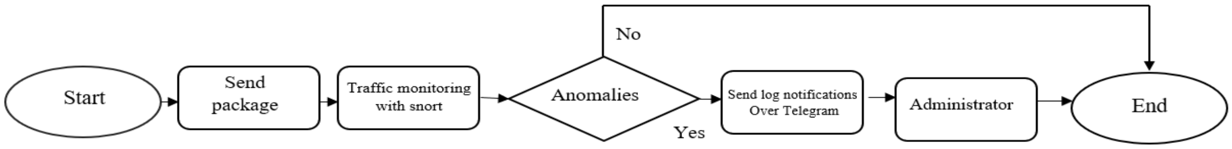
```
drop tcp any any  ->  any any (flags: S; msg:"Possible SYN DDoS"; flow: stateless; detection_filter: track by_src, count 1000, seconds 10; sid:100002; rev:1;)
```

Figure. 2 .Snort Rule for Drop SYN DDoS

In contrast to “Alert”, the implementation of “Drop” in Snort makes it behave as an IPS, which can effectively prevent malicious packets from reaching their destination. This enables organizations not only to respond quickly to the threats but also to maintain the integrity and availability of their services. Other rules exist in addition to “Alert” and “Drop”, such as “Block”, which is used to stop the current packet and all subsequent unwanted packets in the stream. There is also a log for recording incoming packets or terminating the session with a specific protocol such as TCP reset or ICMP unreachable message (Pradipta, 2017).

2.2 Simulation

For the simulation, Snort will first be configured as an IDS, before switching to IPS to analyze the differentiation on how effectively it captures and react according to the predefined rules. When these rules are met, Snort will generate alerts and sending of messages using the Telegram bot. The notification messages will include important information such as the type of attack detected, the IP address of the attacker, and the time of the incident, allowing administrators to have comprehensive information before aking action. As visualized in “Figure. 4,” explaining how the data flow works, it begins when a packet enters the network, passes through Snort for analysis to determine whether the traffic is legitimate or not. If not, it will block and send an alarm to the administrator.



Start
Send package
Traffic monitoring with snort
Anomalies
No
Yes
Send log notifications Over Telegram
Administrator
End

Figure. 3 Data flow from Snort to Telegram

2.3 Data Collection and Measurement

Snort is not only capable of logging traffic based on predefined rules but also can detect various network activities and anomalies. In this context, we present a traffic log diagram after running Snort in duration for one hour. This diagram visually illustrates the types of traffic detected and shows how Snort analyzes and categorizes network packets over time “Figure. 5.”.

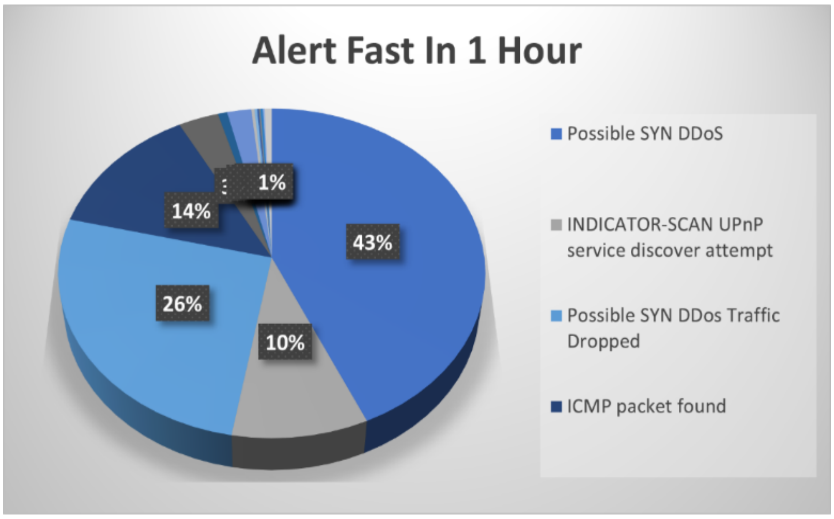
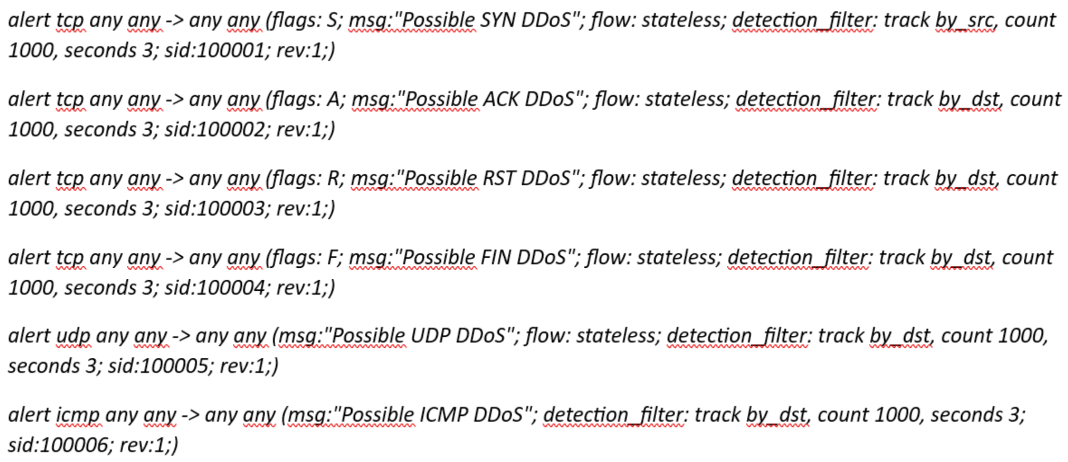


Figure. 4. Network Traffic

From the simulation result, the largest portion (43%) indicates a potential SYN DDoS attack, where the attacker sends numerous SYN packets in an attempt to open connections to the server without completing them, which can lead to resource exhaustion on the server. A total of 26% of the diagram shows SYN DDoS traffic that may have been blocked, meaning the detection system successfully identified and removed some suspicious SYN packets. Additionally, 14% indicates detected ICMP packets, which are often used for network diagnostics, such as ping commands. Suspicious ICMP packets may indicate network scanning activity or attacks. Finally, 10% represents attempts at UPnP service discovery through indicator scanning. UPnP allows network devices to communicate with each other without manual configuration, but it can become an attack vector if not properly secured.

We obtained this data while running Snort with the command hping3 -S -p 80 -i u10 192.168.100.237 using the following rules “Figure. 6.”. After simulating numerous attacks and monitoring the log results, Snort indicates that it successfully achieved a 100% success rate in detecting and sending notifications, with an average time of 2 seconds after the attack was detected. It is proven that Snort is capable of acting within a short time, which can be influenced based on the characteristics of the attack performed and the network architecture.



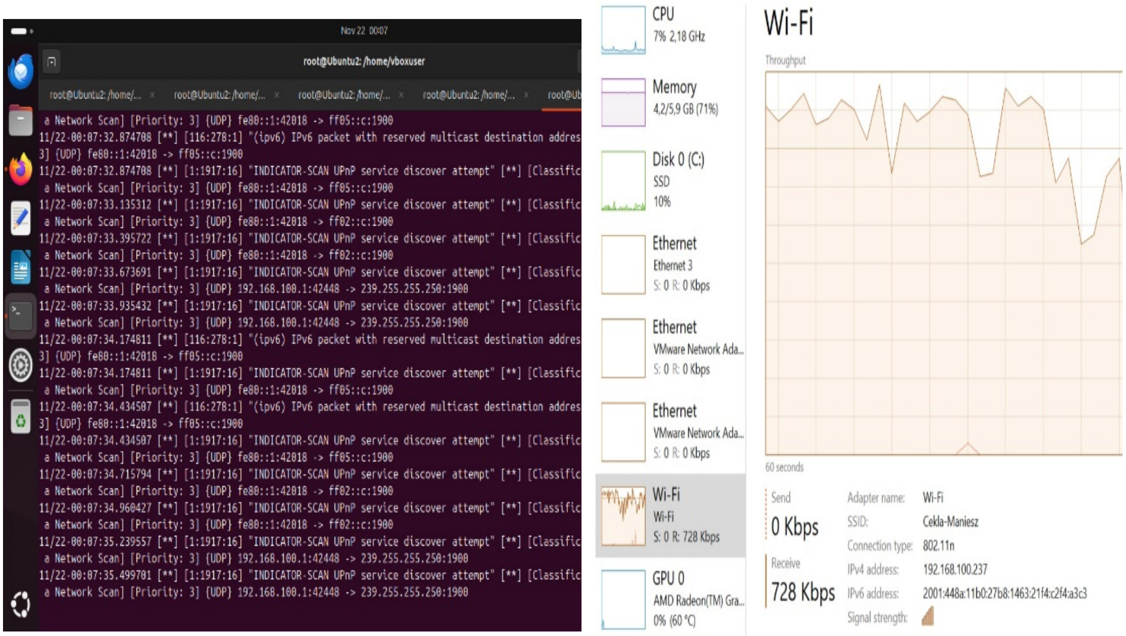
```
alert tcp any any -> any any (flags: S; msg:"Possible SYN DDoS"; flow: stateless; detection_filter: track by_src, count 1000, seconds 3; sid:100001; rev:1;)

alert tcp any any -> any any (flags: A; msg:"Possible ACK DDoS"; flow: stateless; detection_filter: track by_dst, count 1000, seconds 3; sid:100002; rev:1;)

alert tcp any any -> any any (flags: R; msg:"Possible RST DDoS"; flow: stateless; detection_filter: track by_dst, count 1000, seconds 3; sid:100003; rev:1;)

alert tcp any any -> any any (flags: F; msg:"Possible FIN DDoS"; flow: stateless; detection_filter: track by_dst, count 1000, seconds 3; sid:100004; rev:1;)

alert udp any any -> any any (msg:"Possible UDP DDoS"; flow: stateless; detection_filter: track by_dst, count 1000, seconds 3; sid:100005; rev:1;)

alert icmp any any -> any any (msg:"Possible ICMP DDoS"; detection_filter: track by_dst, count 1000, seconds 3; sid:100006; rev:1;)
```

Figure 5. Snort Protocol Rules

3. Results and Discussions

The results of the DDoS simulation are visualized in this section, along with the abnormalities that were discovered by Snort that serving as an IDS and IPS. Overall, Snort has been demonstrated and proven effective in identifying various kinds of anomalies, particularly on DDoS. It also shows how the integration with Telegram can help to improve the network administration's responsiveness to the threat.

Following "Figure. 7," showing the alerts triggered from Snort as an IDS to generate the notification without dropping the traffic, along with the throughput display on Task Manager from the target machine before and during a DDoS attack. This is showing a clear result of changes in network activity by comparing the data collected before and during the attack as displayed on "Figure. 8."..

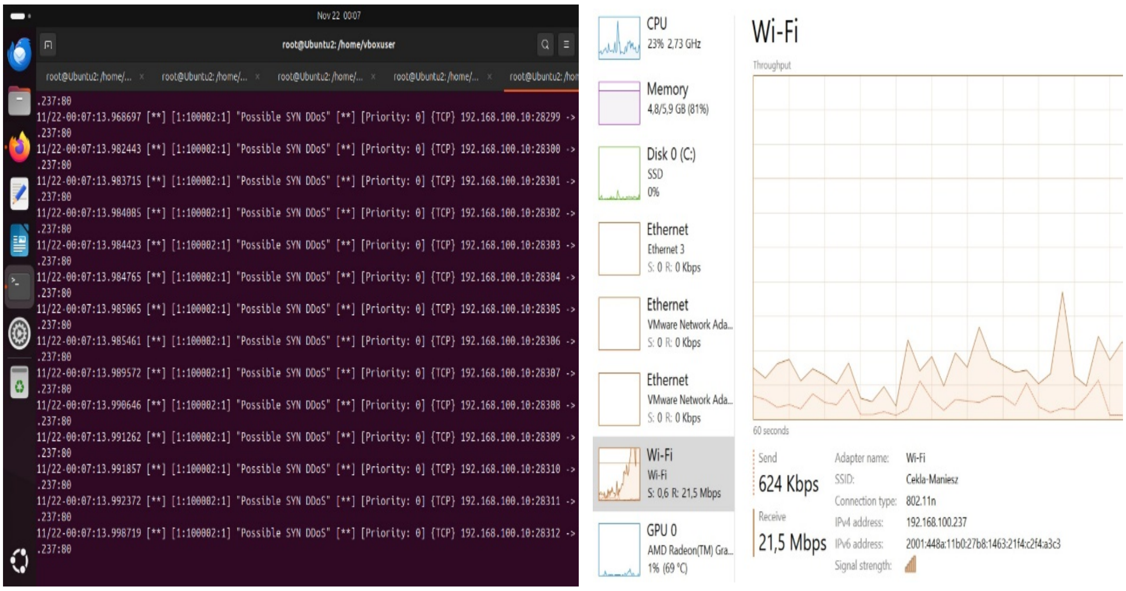


```
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:07:32.874708 [**] [116:278:1] "(ipv6) IPv6 packet with reserved multicast destination addres
3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:07:32.874708 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:07:33.135312 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:07:33.395722 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:07:33.673691 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
11/22-00:07:33.935432 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
11/22-00:07:34.174811 [**] [116:278:1] "(ipv6) IPv6 packet with reserved multicast destination addres
3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:07:34.174811 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:07:34.434507 [**] [116:278:1] "(ipv6) IPv6 packet with reserved multicast destination addres
3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:07:34.434507 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:07:34.715794 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:07:34.960427 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:07:35.239557 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
11/22-00:07:35.499701 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classific
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
```

Send
0 Kbps
Receive
728 Kbps

Adapter name: Wi-Fi
SSID: Cekla-Maniesz
Connection type: 802.11n
IPv4 address: 192.168.100.237
IPv6 address: 2001:448a:11b0:27b8:1463:21f4:c2f4:a3c3
Signal strength:

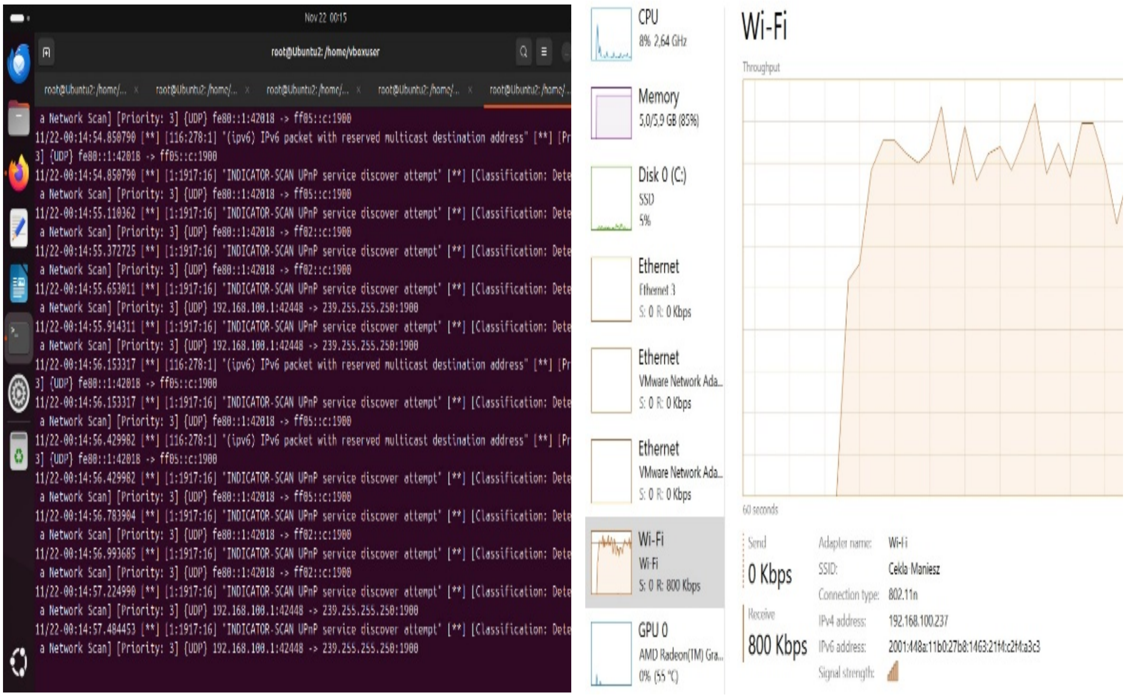
Figure. 6 Before SYN DDoS Attack



```
.237:80
11/22-00:07:13.968697 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28299 ->
.237:80
11/22-00:07:13.982443 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28300 ->
.237:80
11/22-00:07:13.983715 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28301 ->
.237:80
11/22-00:07:13.984085 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28302 ->
.237:80
11/22-00:07:13.984423 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28303 ->
.237:80
11/22-00:07:13.984765 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28304 ->
.237:80
11/22-00:07:13.985065 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28305 ->
.237:80
11/22-00:07:13.985461 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28306 ->
.237:80
11/22-00:07:13.989572 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28307 ->
.237:80
11/22-00:07:13.990646 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28308 ->
.237:80
11/22-00:07:13.991262 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28309 ->
.237:80
11/22-00:07:13.991857 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28310 ->
.237:80
11/22-00:07:13.992372 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28311 ->
.237:80
11/22-00:07:13.998719 [**] [1:100002:1] "Possible SYN DDoS" [**] [Priority: 0] {TCP} 192.168.100.10:28312 ->
.237:80
```

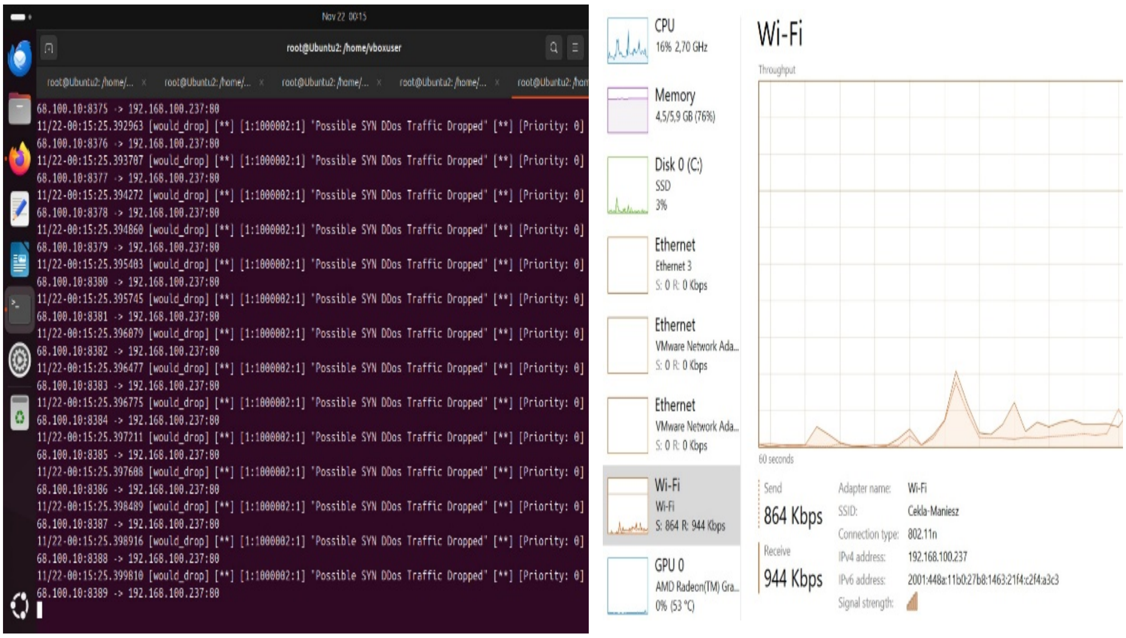
Figure. 7 After SYN DDoS Attack

In the second scenario, Snort is setup to function as an IPS. It shows the fact that in "Figure. 9" and "Figure. 10," the graphic of the network performance on the target remains consistent without experiencing any spikes this is indicates the capability of Snort to stop the malicious packets before reaching the target machine.



```
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:14:54.850790 [**] [116:278:1] "(ipv6) IPv6 packet with reserved multicast destination address" [**] [Pr
3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:14:54.850790 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:14:55.110362 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:14:55.372725 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:14:55.653011 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
11/22-00:14:55.914311 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
11/22-00:14:56.153317 [**] [116:278:1] "(ipv6) IPv6 packet with reserved multicast destination address" [**] [Pr
3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:14:56.153317 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:14:56.429982 [**] [116:278:1] "(ipv6) IPv6 packet with reserved multicast destination address" [**] [Pr
3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:14:56.429982 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff05::c:1900
11/22-00:14:56.783904 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:14:56.993685 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} fe80::1:42018 -> ff02::c:1900
11/22-00:14:57.224990 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
11/22-00:14:57.484453 [**] [1:1917:16] "INDICATOR-SCAN UPnP service discover attempt" [**] [Classification: Dete
 a Network Scan] [Priority: 3] {UDP} 192.168.100.1:42448 -> 239.255.255.250:1900
```

Figure. 8 Before SYN DDoS Drop



```
68.100.10:8375 -> 192.168.100.237:80
11/22-00:15:25.392963 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8376 -> 192.168.100.237:80
11/22-00:15:25.393707 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8377 -> 192.168.100.237:80
11/22-00:15:25.394272 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8378 -> 192.168.100.237:80
11/22-00:15:25.394860 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8379 -> 192.168.100.237:80
11/22-00:15:25.395403 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8380 -> 192.168.100.237:80
11/22-00:15:25.395745 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8381 -> 192.168.100.237:80
11/22-00:15:25.396079 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8382 -> 192.168.100.237:80
11/22-00:15:25.396477 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8383 -> 192.168.100.237:80
11/22-00:15:25.396775 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8384 -> 192.168.100.237:80
11/22-00:15:25.397211 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8385 -> 192.168.100.237:80
11/22-00:15:25.397608 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8386 -> 192.168.100.237:80
11/22-00:15:25.398489 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8387 -> 192.168.100.237:80
11/22-00:15:25.398916 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8388 -> 192.168.100.237:80
11/22-00:15:25.399810 [would_drop] [**] [1:1000002:1] 'Possible SYN DDos Traffic Dropped" [**] [Priority: 0]
68.100.10:8389 -> 192.168.100.237:80
```

CPU
16% 2,70 GHz

Memory
4,5/5,9 GB (76%)

Disk 0 (C:)
SSD
3%

Ethernet
Ethernet 3
S: 0 R: 0 Kbps

Ethernet
VMware Network Ada...
S: 0 R: 0 Kbps

Ethernet
VMware Network Ada...
S: 0 R: 0 Kbps

Wi-Fi
Wi-Fi
S: 864 R: 944 Kbps

GPU 0
AMD Radeon(TM) Gra...
0% (53 °C)

Wi-Fi

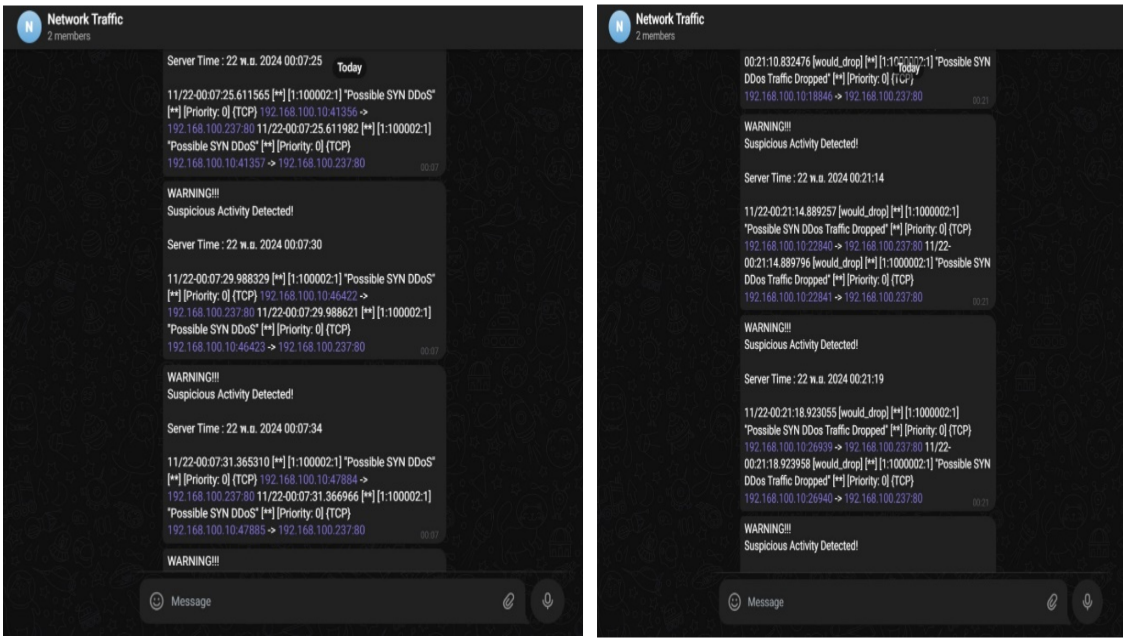
Send
864 Kbps

Receive
944 Kbps

Adapter name: Wi-Fi
SSID: Cekla-Maniesz
Connection type: 802.11n
IPv4 address: 192.168.100.237
IPv6 address: 2001:448a:11b0:27b8:1463:21f4:c2f4:a3c3
Signal strength:

Figure. 9 After SYN DDoS Drop

Additionally, as illustrated in "Figure. 11," the system's notification allows to instantly alerting network administrators for suspicious activity or possible threats is demonstrated by the integration of Snort with a Telegram bot, for a notification system.



```
Server Time : 22 พ.ย. 2024 00:07:25

11/22-00:07:25.611565 [**] [1:100002:1] "Possible SYN DDoS"
[**] [Priority: 0] {TCP} 192.168.100.10:41356 ->
192.168.100.237:80 11/22-00:07:25.611982 [**] [1:100002:1]
"Possible SYN DDoS" [**] [Priority: 0] {TCP}
192.168.100.10:41357 -> 192.168.100.237:80                 00:07

WARNING!!!
Suspicious Activity Detected!

Server Time : 22 พ.ย. 2024 00:07:30

11/22-00:07:29.988329 [**] [1:100002:1] "Possible SYN DDoS"
[**] [Priority: 0] {TCP} 192.168.100.10:46422 ->
192.168.100.237:80 11/22-00:07:29.988621 [**] [1:100002:1]
"Possible SYN DDoS" [**] [Priority: 0] {TCP}
192.168.100.10:46423 -> 192.168.100.237:80                 00:07

WARNING!!!
Suspicious Activity Detected!

Server Time : 22 พ.ย. 2024 00:07:34

11/22-00:07:31.365310 [**] [1:100002:1] "Possible SYN DDoS"
[**] [Priority: 0] {TCP} 192.168.100.10:47884 ->
192.168.100.237:80 11/22-00:07:31.366966 [**] [1:100002:1]
"Possible SYN DDoS" [**] [Priority: 0] {TCP}
192.168.100.10:47885 -> 192.168.100.237:80                 00:07

WARNING!!!
```

```
00:21:10.832476 [would_drop] [**] [1:1000002:1] "Possible SYN
DDos Traffic Dropped" [**] [Priority: 0] {TCP}
192.168.100.10:18846 -> 192.168.100.237:80                 00:21

WARNING!!!
Suspicious Activity Detected!

Server Time : 22 พ.ย. 2024 00:21:14

11/22-00:21:14.889257 [would_drop] [**] [1:1000002:1]
"Possible SYN DDos Traffic Dropped" [**] [Priority: 0] {TCP}
192.168.100.10:22840 -> 192.168.100.237:80 11/22-
00:21:14.889796 [would_drop] [**] [1:1000002:1] "Possible SYN
DDos Traffic Dropped" [**] [Priority: 0] {TCP}
192.168.100.10:22841 -> 192.168.100.237:80                 00:21

WARNING!!!
Suspicious Activity Detected!

Server Time : 22 พ.ย. 2024 00:21:19

11/22-00:21:18.923055 [would_drop] [**] [1:1000002:1]
"Possible SYN DDos Traffic Dropped" [**] [Priority: 0] {TCP}
192.168.100.10:26939 -> 192.168.100.237:80 11/22-
00:21:18.923958 [would_drop] [**] [1:1000002:1] "Possible SYN
DDos Traffic Dropped" [**] [Priority: 0] {TCP}
192.168.100.10:26940 -> 192.168.100.237:80                 00:21

WARNING!!!
Suspicious Activity Detected!
```

Figure. 10 Notification Result on Telegram

4. Conclusions

This study, provides a simulation of using Snort to detect suspicious packet over the network, mainly for DDoS. Based on the result obtained, it highlights Snort's capability not only to detect various types of attacks and its effective response, but also demonstrates how the Telegram-based notification system can be integrated to improve the network administrator’s responsiveness to potential threats. Thus, this conclusion emphasizes the importance of combining detection technology and communication to strengthen overall network security.

Due to the limited resources, the simulation only involved a limited number of servers and DDoS attacks, suggesting that larger-scale trials could be developed to ensure the validity of this result. Performance analysis, such as response time and overhead, could also be an area for further development to ensure the efficiency of this system.

5. References

Adiwal, S., Rajendran, B., D., P. S., & Sudarsan, S. D. (2023). DNS Intrusion Detection (DID) — A SNORT-based solution to detect DNS Amplification and DNS Tunneling attacks. *Franklin Open*, *2*, 100010. https://doi.org/10.1016/j.fraope.2023.100010

Akhir, T., Kuliah, M., Informasi, K., Jaringan, D., Najib, M., Satria, D., Rahardjo, I. B., & Elektro, T. (2016). *Bentuk Serangan DoS (Denial of Service) dan DDoS (Distributed Deial of Service) pada Jaringan NDN (Named Data Network).*

Awal, H., & Gusman, A. P. (2023). Implementasi Intrusion Detection Prevention System Sebagai Sistem Keamanan Jaringan Komputer Kejaksaan Negeri Pariaman Menggunkan Snort Dan Iptables Berbasis Linux. In *Jurnal Sains Informatika Terapan (JSIT) E-ISSN* (Vol. 2, Issue 2). Bulan Juni.

Cavanagh, M. (2019). “Malicious attack” takes Wikipedia offline in Germany. *Deutsche Welle*. https://www.dw.com/en/malicious-attack-takes-wikipedia-offline-in-germany/a-50335521

Nalayini, C. M., Katiravan, J., Geetha, S., & Christy Eunaicy, J. I. (2024). A novel dual optimized IDS to detect DDoS attack in SDN using hyper tuned RFE and deep grid network. *Cyber Security and Applications*, *2*. https://doi.org/10.1016/j.csa.2024.100042

Iftikhar, A., Qureshi, K. N., Shiraz, M., & Albahli, S. (2023). Security, trust and privacy risks, responses, and solutions for high-speed smart cities networks: A systematic literature review. *Journal of King Saud University - Computer and Information Sciences*, *35*(9), 101788. https://doi.org/10.1016/j.jksuci.2023.101788

Javanmardi, S., Ghahramani, M., Shojafar, M., Alazab, M., & Caruso, A. M. (2024). M-RL: A mobility and impersonation-aware IDS for DDoS UDP flooding attacks in IoT-Fog networks. *Computers and Security*, *140*. https://doi.org/10.1016/j.cose.2024.103778

Kurniawan, R., & Prakoso, F. (2020). *Implementasi Metode IPS (Intrusion Prevention System) dan IDS (Intrusion Detection System) untuk Meningkatkan Keamanan Jaringan Program Studi Teknik Informatika 1)2) Sekolah Tinggi Manajemen Informatika dan Komputer Indo Daya Suvana* (Vol. 2, Issue 02).

Nandy, T., Md Noor, R., Kolandaisamy, R., Idris, M. Y. I., & Bhattacharyya, S. (2024). A review of security attacks and intrusion detection in the vehicular networks. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 36, Issue 2). King Saud bin Abdulaziz University. https://doi.org/10.1016/j.jksuci.2024.101945

Pradipta, Y. W. (2017). *Implementasi Intrusion Prevention System (IPS) Mengggunakan IPTABLES Linux IMPLEMENTASI INTRUSION PREVENTION SYSTEM (IPS) MENGGUNAKAN SNORT DAN IP TABLES BERBASIS LINUX*. www.snort.org.

Turner, C., & Joseph, A. (2017). A Statistical and Cluster Analysis Exploratory Study of Snort Rules. *Procedia Computer Science*, *114*, 106–115. https://doi.org/10.1016/j.procs.2017.09.023