New Snort rule for detection and prevention of SMTP e-mail bomb attacks

Andrei-Daniel Tudosi, Doru Gabriel Balan, Alin Dan Potorac
Department of Computers, Electronics and Automation
Stefan cel Mare University of Suceava
Suceava, Romania
andrei.tudosi1@student.usv.ro

Abstract— **Cyberattacks on networks are launched in every moment nowadays. Due to the advancement of digitalization and technology, these types of attacks are present in our lives, whether we are aware of them or not, they still exist. E-mail bomb attacks are listed as cyberattacks, which can create difficulties in the telecommunication sector because they target services that will be disrupted and be harder to access. In definition, an e-mail bomb involves the process of sending a large number of e-mails to a specific server or person. A very powerful open-source tool that can handle this situation is Snort [1], which is used as a solution to identify this network attack. Additionally, with Snort, this paper presents a custom rule proposed to show promising results in detecting this kind of attack.**

Keywords— Virtual machining , Network servers, Routing, Network interfaces, Snort, pfSense, E-mail bomb, SMTP, Wireshark, Scapy.

I. INTRODUCTION

E-mail bomb attacks can be considered an abuse because they can cause a lot of temporary damage by blocking activities. Usually, attackers collect e-mail addresses via web forms and then start to attack the receivers. This huge amount commonly fills up the receiver's disk space on the system or server [2] and the process can cause the machine to crash and stop functioning. Having this issue causes inaccessibility to other users who use the service and because of virtualization, on that machine there can be more than one server installed. If the e-mail server is down, both sending and receiving e-mail services are unusable. This cyberattack can do a lot of harm, being an automated attack that sends thousands of e-mails per second. Because of the volume of data, servers or systems are flooded and become slower or unavailable. Signs that e-mail bomb attacks are occurring are many and visible. Usually e-mail activity is increased and abnormally, this can be observed in the inbox of our e-mail address. We will discover that there are numerous e-mails that have similar template like unusual sender, subject, content or attachment. E-mail bomb attacks are often used for some reasons. It can be used, for example, to give a hard time for an IT specialist in the background of the network or to disorientate a normal user by mixing spam e-mails with normal and important ones. This can also be a distraction from other attacks in the same network. Spammers can infect numerous computers inside a network to obtain certain information [3]. Regardless of the reason for the attack, the result is to block or at least aggravate the entire activity of the network, because inside a local network frequently there are multiple servers interconnected, and the inbound traffic is limited to a certain speed.

If the network is not well configured, and it doesn't have a traffic prioritization configured, then there is a high chance that flooding with e-mails will collapse the entire network. These problems usually appear because of reasons like the following: fragmented IT architecture, the lack of right security tools or accurate threat intel, internal cyber skills shortage and low awareness. An extra security measure can be the integration of a "honeypot" in the network for virtual traps [4]. A honeypot is an easy and ideal target because it looks like a genuine computer inside a local network. The IT team used this technique to observe how the attacker proceeds for the future defenses to protect the network. It is crucial for companies to permanently adjust their business models and to explore the digitization opportunities that arise. Every day, new risks appear and the cybersecurity threats grow, making companies to become more vulnerable. Technology cannot be neglected because it is indispensable and an integral part of nowadays business area.

In this paper, we have proposed a solution to an experimental design consisting of a virtual network where an e-mail bomb affects the entire system by flooding a server. Our improvements to the network were made using open-source software with custom configurations. After analyzing the network and finding the source of the problem, we have solved it with a solution consisting of the help of an IDS. We have proposed Snort to deal with the heavy traffic generated by the network simulator. The custom configuration of Snort has shown a huge impact on this network issue, which will be presented in the future sections.

II. RELATED WORK

The authors of [5] proposed a system to protect an environment from distributed denial-of-service (DdoS) attacks using an intrusion prevention system (IPS). In this case, Snort is used to block the attack traffic and is a bridge between network devices forwarding network data through it. The method used for protection is blacklisting attackers IPs, which is a good solution. Furthermore, their approach brings new technology like blockchain and smart contracts. A mix of

978-1-6654-8162-5/22/$31.00 ©2022 IEEE

algorithms is used so that the transactions are faster, and the entire system doesn't use high resources in activity. Smart contracts have the feature of maintaining the attack source information. For better results, as the authors said, networks containing private blockchain used in their study need to be replaced by public Ethereum blockchain.

An interesting approach is presented in [6], where the authors used tools to provide an active monitoring technique. The operating principle is by sending from the monitoring agent different packets to the server or the service with the purpose of measuring the network performance, and to expose network failures. Usually, this technique is used with few packets, because using a large number of packets can overload the network. They are capable of diagnosing problems related to the Simple Mail Transfer Protocol (SMTP) in a real working environment. This process is good for the process of monitoring the availability of services and protocol functionality using services. The monitoring protocol algorithm is used and executed to calculate the response time, the functioning of the SMTP protocol and different averages.

Snort [7], the intrusion detection system (IDS) and IPS that we have included in our research, is used in many cases and situations. It can be an important component along deep packet inspection for extra security, being a high-end solution in detecting and dropping cyberattacks hidden in many forms. This method was proposed in [8] by authors, and it is used in an industrial automation control system. The solution was tested to protect systems from being exploited via the protocol Modbus/TCP, which is a vulnerable protocol designed without security features. Three different IDS and IPS were tested in this situation, because the main challenge is to have the lowest latency and jitter in traffic flow due to the real time communication. Deep packet inspection is a security measure that can scale down Modbus/TCP vulnerabilities that are taken advantage of in industrial area.

A. E-mail bomb attack

The main reason that e-mail bomb attack is launched is that an unauthorized person wants to have access to an e-mail server. They try to distract the persons who use that server in any way possible, so they can try to bypass the login into the e-mail server. Once they realize to have access to the e-mail server, they can use information from the existing emails to gain money or to cause damage. Having access to the database of the e-mail server means that the attacker can make online shopping using the credentials of the users. This action can be hidden through other attacks of the same kind, because users of the e-mail server cannot see this thing happening, for example, one order confirmation email is received at the same time with other 100 emails. If these situations occur, then the users need to check and secure their bank and e-mail accounts and payment options. Studies presented in [9] showed that in current year 2021 94% of malware is delivered via e-mail and 48% of malicious e-mail attachments are office files. At the beginning of the Covid pandemic, Google blocked almost 18 million daily malware and phishing e-mails related to the Coronavirus. Communication is an essential thing in business. Without communication between a seller and a customer, it cannot exist a business. One method for cutting this connection is via e-mail bomb attacks, where an unauthorized person sends spam e-mails using bots to a specific e-mail server trying to slow it or even shut it down completely. This causes the loss of valuable time, which could result in loss of income in case of business. It is vital for a business to maintain communication via e-mails, so if e-mail bomb attack is successful, then the ability to communicate can be lost entirely. In some cases where the attacker gains full access into the network, the e-mail server can also become blacklisted. The domain name needs to be checked by e-mail focused on monitoring to ensure that the communication of the business is working in normal parameters. SMTP is a process by which e-mails are sent via the Internet between two clients [10]. An important thing regarding computer ports is the way that devices connect to a network and how they complete tasks containing electronic processes. A SMTP port is a mixture of both connectivity and fulfillment of tasks, being a port made to send e-mail through a network and to its recipient. Primarily, for SMTP it is used port 25. But, in many cases, e-mail clients nowadays don't operate via this port, because it is restricted by ISPs and providers of Cloud Hosting technology to reduce the quantity of spam that is released from devices infected with malware. As the default e-mail submission port, we use port 587 when an e-mail client is sending an e-mail to be directed by a proper server. The main advantage of this port that it supports Transport Layer Security (TLS) encryption, which ensures that e-mail is sent in a protected manner, and it follows the rules made by the Internet Engineering Task Force (IETF). If something happens to port 587, and it can't be used, then the next alternative is port 2525, which is commonly used and support by providers. An e-mail server has several similarities with a web server, the communication is made by IP addresses, but to facilitate the process, it is used a domain name [11]. Communication is made using ports, a computer is identified by an IP address and a port is used to identify a specific application or service running on that computer.

III. PROPOSED RULE DESIGN

The authors of research presented in [12] proposed a set of rules designed for improving different types of attacks inside a network. They realize a rule that focuses on e-mail bomb attacks with offers protection against them. Regarding DDoS attacks, they created a custom rule in Snort, which filters packets containing thousands of mail messages that are received or sent to a certain user in a period of time. In their case, the rule becomes active after the traffic on port 25 exceeds 1000 emails per second. In their research, having a custom experimental setup, Snort had an efficiency of 50% based on the rule.

Rules are an important part of Snort, which analyzes and has a real time alerting system that manages the traffic data network. Multiple components work together in the process of detecting particular network attacks, components such as preprocessors, logging and alerting systems, detection engine, packet decoder and output modules. A rule used by Snort has two logical parts, the rule header and the rule option. In the header section, it is described attributes of a packet and instructions of commands if the packet matches the rule. In close connection is the rule options, which will follow the rule

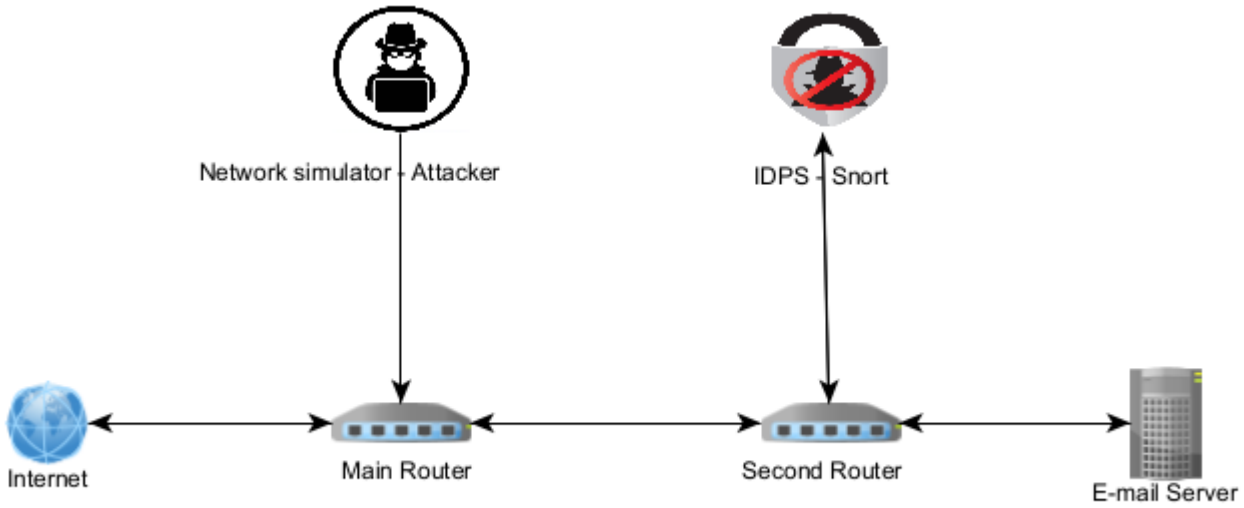


Fig. 1. Experimental setup

header and has the permission to create alert messages. The rule header has a structure composed by the following: action, protocol, source address, source port, direction, destination address and destination port.

As a result of our research regarding e-mail bomb attacks, we have improved the rule against this type of attack by changing the flag type. In this rule, “S” stands for SYN TCP flag. SYN means Synchronization, which is used as a first step in connection establishment or in three-way handshake process between two devices that want to connect with each other. The “S” flag is used only for the first packet from the sender or receiver, realizing the initiation of the connection. This topic is discussed in paper [13]. Beneath we presented the custom rule:

alert tcp any any -> $SMTP_SERVERS 25 (msg: "Possible E-mail Bomb attack occurring "; flags: S; threshold: type threshold, track by_dst, count 2, seconds 1; sid:1000003; rev:001;)

If we want to split the rule into logical parts, we can do the following:

Rule header:

Action: alert

Protocol: tcp

Source address and Source port: any

Direction: ->

Destination address: $SMTP_SERVERS

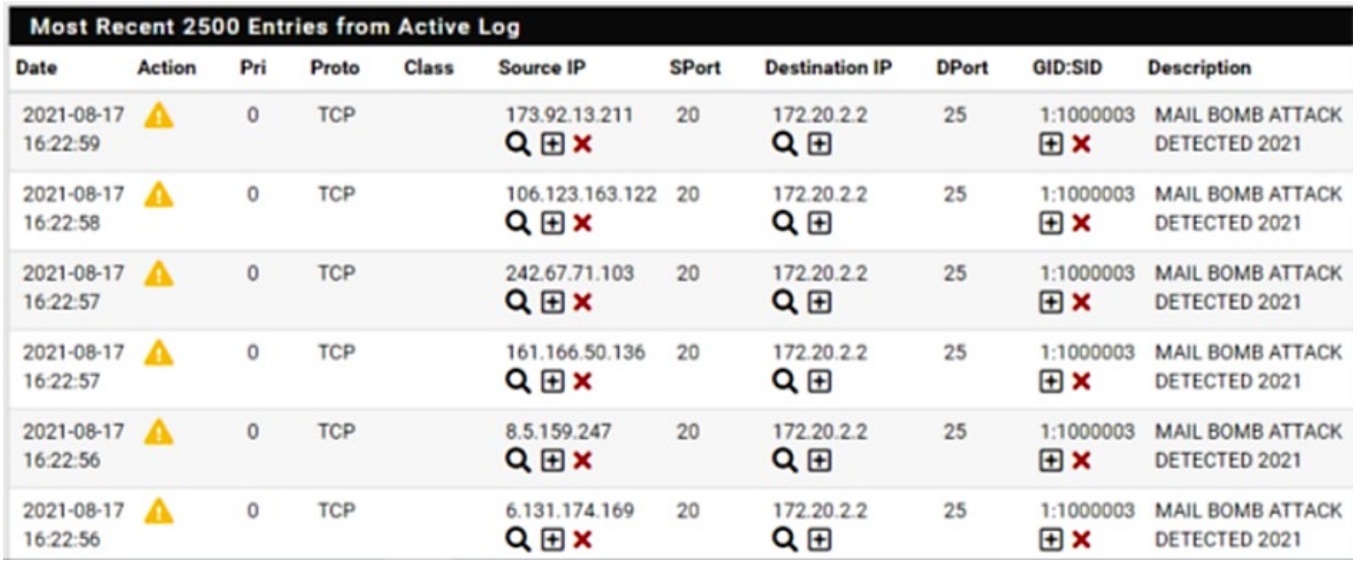
Destination port: 25

Rule option: (msg: "Possible E-mail Bomb attack occurring "; flags: S; threshold: type threshold, track by_dst, count 2, seconds 1; sid:1000003; rev:001;)

Snort is using rules in high level processing stages. During the traffic analysis, packets are received via libpcap from the network. After the capture of the packets, they are filtered through a process that uses decoding. In this process, Snort determines the protocol of the packets, and after decoding the packets are sent into the preprocessing stage. In this stage, packets are analyzed and then reassembled. The next step after preprocessing is the content normalization. After this step, Snort’s detection engine uses a string searching algorithm to correlate the packet payload with rules. This process is the most time and resource consuming because of the matching. Upon scanning, if a malicious signature is matched, then the alert engine is notified and actions are taken. The rules used by Snort [14] consist of the following: a description of network traffic, needed signatures, information of the threat, and a description of what action to take as a response of detection.

IV. EXPERIMENT

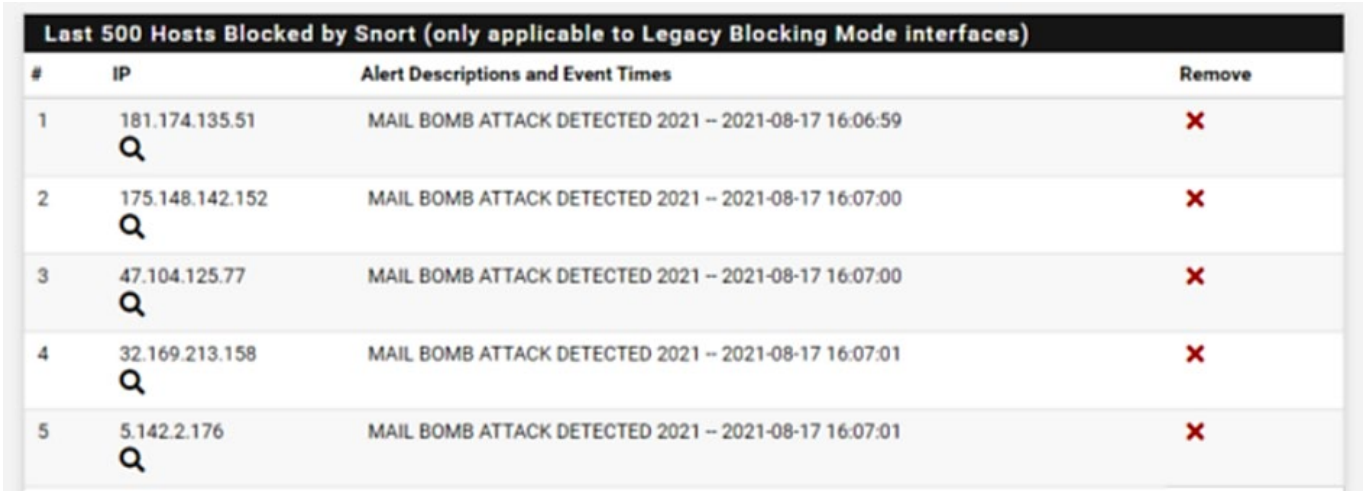
In this section, we will evaluate the performance of the proposed rule designed for email bomb attacks. Based on the diagram in Fig. 1, we have simulated a local network inside XCP-NG, which is a cloud virtual infrastructure based on XenServer, the software mentioned in Table I. In our virtual network, we have inbound and outbound traffic, as a common setup. Traffic comes from the Internet inside our network through routers. The connection between the e-mail server and the Internet is made via routers using port forward, firewall rules and custom gateway. In our setup, the attacker only sends traffic to the e-mail server, due to its purpose of flooding. Inside the network we have four hosts: the attacker, which is traffic simulated by Scapy [15] installed on Ubuntu [16] client, then we have two hosts with pfSense [17], and the target machine with Ubuntu Server configured as an e-mail server. In this experimental setup, having two active firewalls means that



Most Recent 2500 Entries from Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2021-08-17 16:22:59	⚠	0	TCP		173.92.13.211	20	172.20.2.2	25	1:1000003	MAIL BOMB ATTACK DETECTED 2021
2021-08-17 16:22:58	⚠	0	TCP		106.123.163.122	20	172.20.2.2	25	1:1000003	MAIL BOMB ATTACK DETECTED 2021
2021-08-17 16:22:57	⚠	0	TCP		242.67.71.103	20	172.20.2.2	25	1:1000003	MAIL BOMB ATTACK DETECTED 2021
2021-08-17 16:22:57	⚠	0	TCP		161.166.50.136	20	172.20.2.2	25	1:1000003	MAIL BOMB ATTACK DETECTED 2021
2021-08-17 16:22:56	⚠	0	TCP		8.5.159.247	20	172.20.2.2	25	1:1000003	MAIL BOMB ATTACK DETECTED 2021
2021-08-17 16:22:56	⚠	0	TCP		6.131.174.169	20	172.20.2.2	25	1:1000003	MAIL BOMB ATTACK DETECTED 2021

Fig. 2. Snort alerts



Last 500 Hosts Blocked by Snort (only applicable to Legacy Blocking Mode interfaces)

#	IP	Alert Descriptions and Event Times	Remove
1	181.174.135.51	MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:06:59	✕
2	175.148.142.152	MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:00	✕
3	47.104.125.77	MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:00	✕
4	32.169.213.158	MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:01	✕
5	5.142.2.176	MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:01	✕

Fig. 3. Snort blocking IP addresess

the incorrect traffic is blocked by default and the good traffic is allowed to be analyzed further by Snort.

TABLE I. LIST OF SOFTWARE USED FOR TESTS WITH SPECIFICATIONS

Software Applications	Version	Resources
XCP-NG	20.04.01	16 threads/ 32GB Ram
pfSense	2.5.2	4 threads/ 2GB Ram
Snort	3.2.9.10	-
Ubuntu client	20.04.2.0 LTS	4 threads/ 2GB Ram
Ubuntu server	18.04	4 threads/ 2GB Ram
Scapy	2.4.3	integrated in Ubuntu client
Wireshark	3.4.7	integrated in Ubuntu client

Table I. shows the software programs with which we made this installation, versions of these programs were the most recent.

During this research, the different phases are the following: START -> Launch traffic via Scapy with E-mail server as victim -> Capture simulated traffic -> Identify features of the attack traffic -> Inspect traffic packets -> Design Snort rule -> Test Snort rule -> Analyze improvement -> END

The command used in Ubuntu terminal to generate traffic using Scapy is send (IP (dst = “172.20.2.2”, src = RandIP()) / TCP(dport = 25, flags = “S”), count=x), where x means the number of the generated packets depending on the scenario. The destination target is IP 172.20.2.2 , which is the IP address of the second pfSense router. We have created a firewall rule inside second pfSense to redirect all the traffic on port 25 to the IP address of e-mail server, which is 172.21.2.10. As a result, all the traffic from the IP 172.20.2.2 is analyzed and send via a static route to IP 172.21.2.10.

The traffic between the attacker and the email server is captured by: Wireshark [18], which is installed on the same machine as Scapy as source, Packet Capture which is configured on pfSense as transit, and tcpdump on the e-mail server as destination. Multi-point traffic monitoring gives us a better view of what's going on inside the network [19].

The e-mail server is receiving SYN packets from random IP addresses as shown in Wireshark. Traffic is redirected from pfSense to e-mail server IP using the port 25 extension. As we mentioned, incorrect traffic is blocked by the two pfSense routers, then the remaining is analyzed by Snort. Figures 2 and 3 present the results of the rule that we proposed.

V. RESULTS

In this section, we present the results of analyzing traffic produced by Scapy. We also present our proposed rule along with the results obtained from testing it. A key identifier of an attack is the traffic rates, because an e-mail bomb usually involves in transmitting a huge number of packets, or depending on the situation, a large volume of traffic in a short period of time. In many situations, this indicates an attack. In our situation, the attack is based on TCP, exploiting the TCP three-way handshake. This method opens several connections and closes them without sending data, or send a small amount of data and close the connection, or close the opened connection without sending data. In our research, we inspected the captured packets from attack traffic to determine the form and the type of the packets. The three-way handshake was used for establishing a connection and the payload. Simulated packets by Scapy are using SYN flag for establishing the number of connections requested by the attacker.

For testing purposes, we have elaborated six scenarios. The first scenario was consisted in sending 10 packets from Scapy to the e-mail server. In the next scenarios, we're exponential growth to demonstrate the scalability of the experiment. Each reported graph is the result of the average of five experiments of each scenario. Our aim was to analyze the performance of the e-mail server where the attack of the e-mail bomb is happening. In the following section, we will investigate the efficiency of the custom rule and observe the usage of the resources that were allocated to the e-mail server.

In Figure 4, we can observe how the rule behaves under different situations. In every scenario, the number of Snort alerts is the same as the blocked IPs. After starting the experiment with sending the simulated traffic from Scapy, in short time Snort starts to alert. One of our objectives of this experiment, was to analyze the resource utilization for improvement. In our scenarios, we created the attack to be sent without other traffic for better determination of the resource usage. Under the attack traffic, we measured the usage of the resources and the results are shown in Figure 5. As shown in Figure 6, CPU usage increases with the number of packets transmitted over the network. Starting with scenario number 5, where 100000 packets were sent from Scapy, the e-mail server CPU uses 99% of the allocated resources. Also, pfSense is using a lot of power because of the huge number of packets incoming in the network. We can see that in scenario number 6, where 1000000 packets were sent to the e-mail server, that pfSense uses 37% of the CPU for this experiment. Using the custom rule that we created, we managed to improve the resource usage as shown in Figure 8. In scenario number 5, using our custom rule, we managed to run the e-mail server with 37% CPU usage, instead of 99%, which means a huge improvement. The same case is in scenario number 6, where the CPU usage dropped from 99% to 39%. This gain comes with a low extra usage from pfSense, in scenario 5 it uses extra 7% and in scenario 6 extra 5.8%.

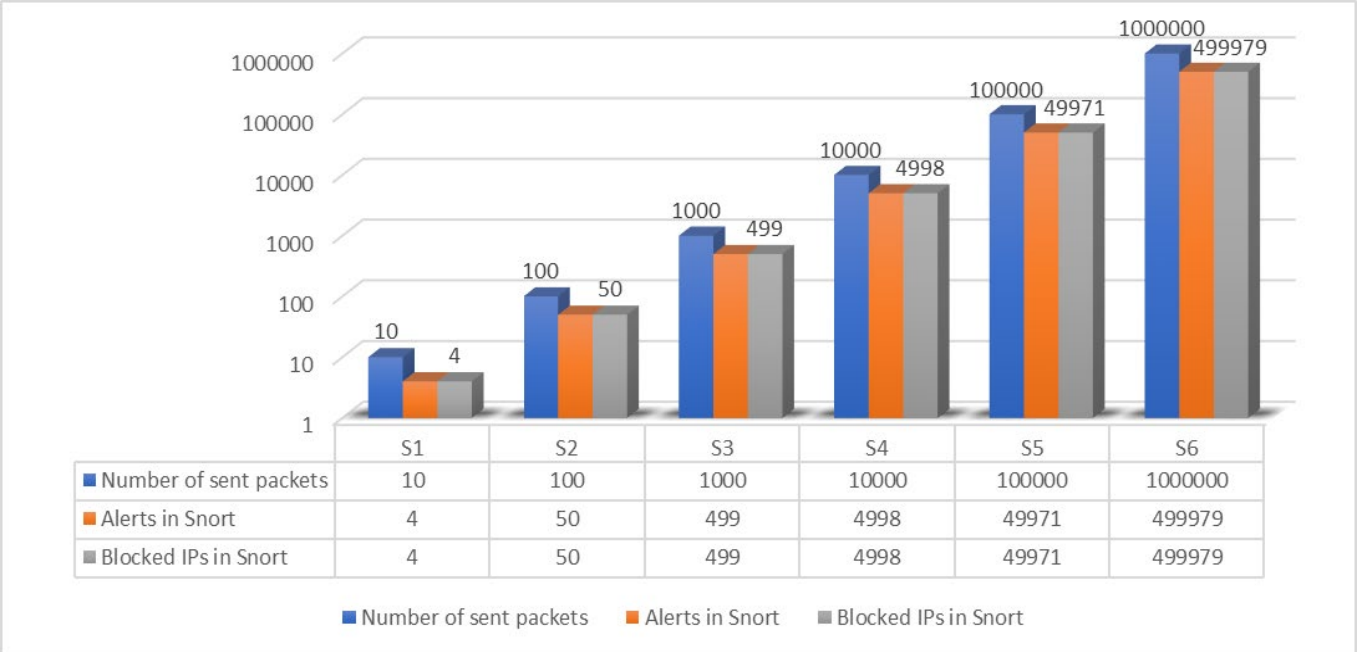
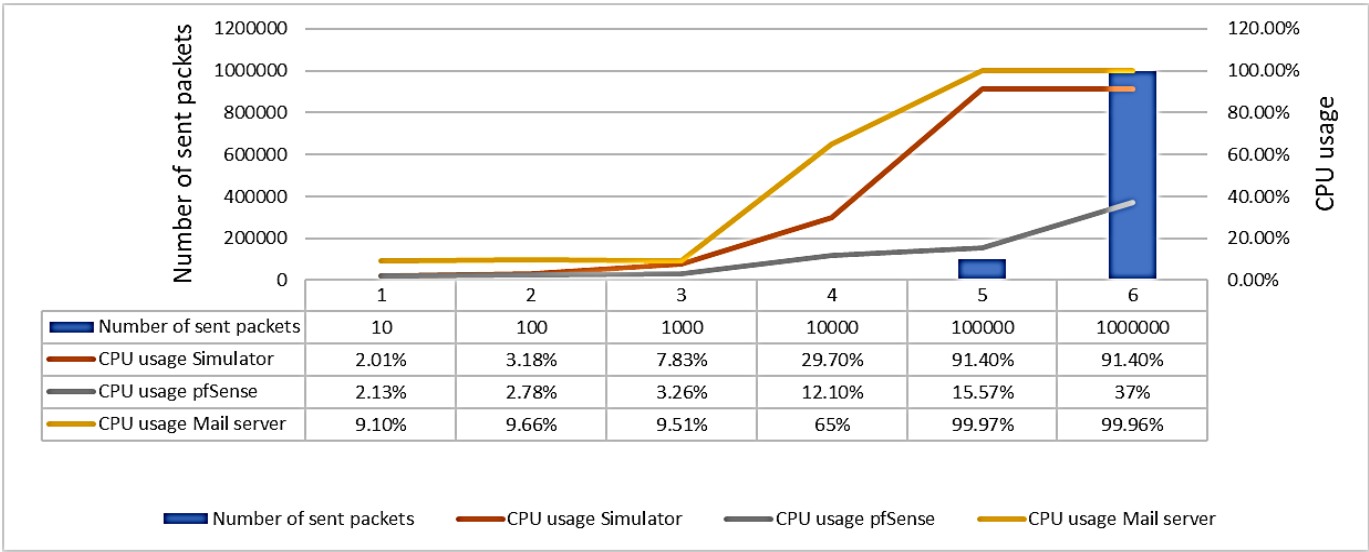
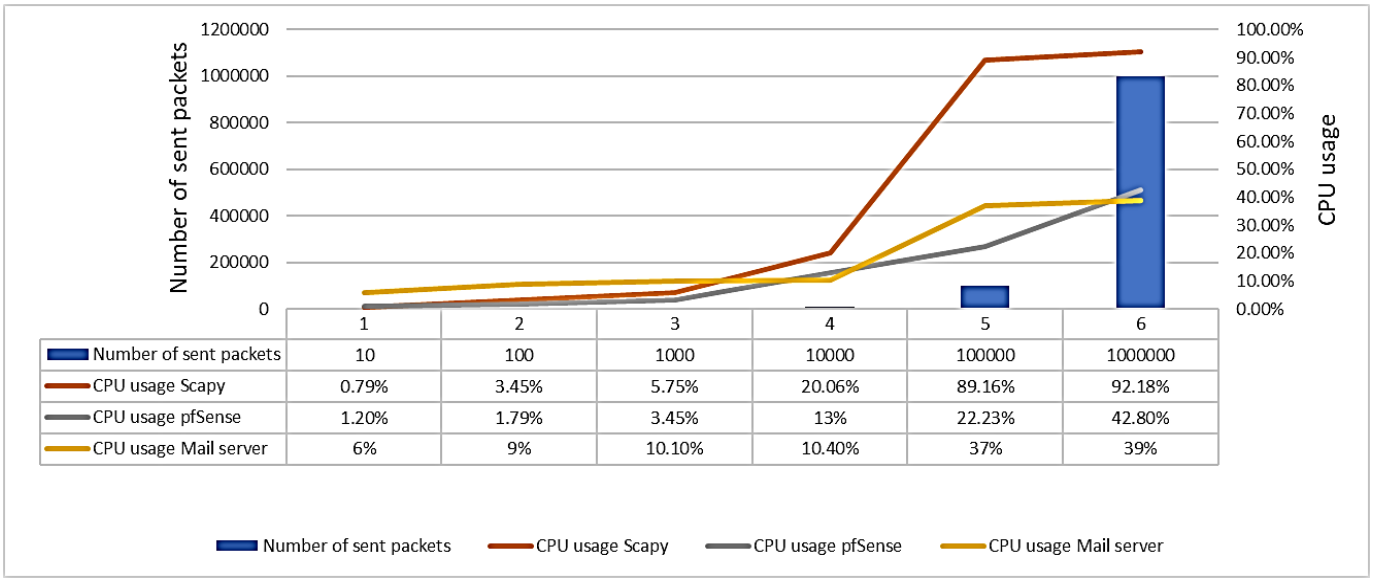


Fig. 4. Alerts and blocked IPs in Snort GUI



	1	2	3	4	5	6
Number of sent packets	10	100	1000	10000	100000	1000000
CPU usage Simulator	2.01%	3.18%	7.83%	29.70%	91.40%	91.40%
CPU usage pfSense	2.13%	2.78%	3.26%	12.10%	15.57%	37%
CPU usage Mail server	9.10%	9.66%	9.51%	65%	99.97%	99.96%

Fig. 5. CPU usage of VM's without Snort and custom rule



	1	2	3	4	5	6
Number of sent packets	10	100	1000	10000	100000	1000000
CPU usage Scapy	0.79%	3.45%	5.75%	20.06%	89.16%	92.18%
CPU usage pfSense	1.20%	1.79%	3.45%	13%	22.23%	42.80%
CPU usage Mail server	6%	9%	10.10%	10.40%	37%	39%

Fig. 6. CPU usage of VM's with Snort and custom rule

The main reason we chose Scapy as the network simulator is because it supports random IP as the source of the attack. Usually, for high efficiency, denial-of-service attacks are distributed using botnets, which are computers that work together to launch multiple attacks on the same target. The feature that we used from Scapy is advantageous, offering more computing power to the experiment. If we have used only one source of attack, Snort would have blocked the IP and the level of the attack was effortless.

VI. Conclusion

In this research article, we analyzed the simulated e-mail traffic and proposed a rule which detects and blocks unwanted traffic. Snort had an improved rule for the given situation, and the result reduced the incoming SYN traffic that was intended to flood the e-mail server. The testing scenarios have shown that adding this custom rule is a true benefit regarding the security of the network. The results of the experiment show that an e-mail server cannot be 100% protected, and there are always paths to improve the security. This raises many challenges, though e-mail service providers claim to offer good protection, we should have a strong and well configured firewall, but also well-trained staff in our organization for extra protection. In an e-mail bomb attack, the attacker compromises several devices into a centralized network known as a botnet network. Botnets are controlled by a central server and are programmed to make synchronized request to a specific server. The result of their actions is to overwhelm a server and make it useless for production. Having a server from the network taken down means that we have a traffic jam in the services and damage of all kinds.

Having an IDS/IPS configured in a network has a huge positive impact into the security, but everything has limitations. Some of these are rare, but they still exist. The

effectiveness can be limited by noises of the external network, which consists in bad packets that are generated by bugs, local packets that can create false alarms and corrupt DNS data. Usually, it is frequent that the number of real attacks to be lower than the number of false alarms. A huge advantage is the existence of IDS/IPS, but they don't compensate for everything, having a weak identification and authentication or other weaknesses is a big problem. The information provided by an IDS/IPS is based on network address associated with the IP packet received, and sometimes the IP attribution can be faked or scrambled. Another drawback can be the signature-based attacks, which create a delay for discovery and its signature to be applied to the IDS/IPS, during this time the system is unable to discover the threat.

For further research, as an improvement, we recommend using QoS to properly control the incoming traffic and detected packets. A system that can improve the security of a network is honeypot. Combined with Snort, it can provide a well performed security system, being able to analyze real time malicious attacks. These systems can reduce the false positive alarm level, which is a critical disadvantage of IDS. Another additional security measure can be the contribution of Machine Learning, a technique used widely for network intrusion detection applications [20]. Different machine learning methods can be used for intrusion detection, depending on the situation and the resources allocated. Machine learning techniques produce a positive impact on the improving of overall performance of the intrusion detection system, increasing the accuracy and lowering the false negatives detected. In these techniques, multiclass classification provides more informative results because of the differentiating different types of attacks.

VII. REFERENCES

[1] "Snort," Snort. https://www.snort.org/documents (accessed Mar. 29, 2022).

[2] "Election Security Spotlight – Email Bombs," https://www.cisecurity.org/insights/spotlight/cybersecurity-spotlight-email-bombs, Aug. 22, 2021.

[3] M. G. Kaplan, "Automatic authentication of email servers and personal computers independent of the active participation of server administrators or personal computer users," in Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference on - CEAS '11, 2011, pp. 38–45. doi: 10.1145/2030376.2030381.

[4] Vetterl Alexander, "Honeypots in the age of universal attacks and the Internet of Things," Cambridge, 2020.

[5] B. A. A. Al'Aziz, P. Sukarno, and A. A. Wardana, "Blacklisted IP distribution system to handle DDoS attacks on IPS Snort based on Blockchain," in Proceeding - 6th Information Technology International Seminar, ITIS 2020, Oct. 2020, pp. 41–45. doi: 10.1109/ITIS50118.2020.9320996.

[6] R. Sureswaran, H. al Bazar, O. Abouabdalla, A. M. Manasrah, and H. El-Taj, "Active e-mail system SMTP protocol monitoring algorithm," in Proceedings of 2009 2nd IEEE International Conference on Broadband Network and Multimedia Technology, IEEE IC-BNMT2009, 2009, pp. 257–260. doi: 10.1109/ICBNMT.2009.5348490.

[7] V. da S. Faria, J. A. Gonçalves, C. A. M. da Silva, G. de B. Vieira, and D. M. Mascarenhas, "SDToW: A Slowloris Detecting Tool for WMNs," Information, vol. 11, no. 12, p. 544, Nov. 2020, doi: 10.3390/info11120544.

[8] O. N. Nyasore, P. Zavarsky, B. Swar, R. Naiyeju, and S. Dabra, "Deep Packet Inspection in Industrial Automation Control System to Mitigate Attacks Exploiting Modbus/TCP Vulnerabilities," in Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, May 2020, pp. 241–245. doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00051.

[9] Rob Sobers, "134 Cybersecurity Statistics and Trends for 2021." https://www.varonis.com/blog/cybersecurity-statistics (accessed Feb. 20, 2022).

[10] W. Goralski, The Illustrated Network (Second Edition), 2nd Edition. 2017.

[11] D. Fadhilah and M. I. Marzuki, "Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines against Dos/DDoS Attacks," in 2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering, BCWSP, Sep. 2020, pp. 157–162. doi: 10.1109/BCWSP50066.2020.9249449.

[12] A. Gupta and L. sen Sharma, "Mitigation of DoS and Port Scan Attacks Using Snort," International Journal of Computer Sciences and Engineering, vol. 7, no. 4, pp. 248–258, Apr. 2019, doi: 10.26438/ijcse/v7i4.248258.

[13] P. Kumar, M. Tripathi, A. Nehra, M. Conti, and C. Lal, "SAFETY: Early Detection and Mitigation of TCP SYN Flood Utilizing Entropy in SDN," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1545–1559, Dec. 2018, doi: 10.1109/TNSM.2018.2861741.

[14] Martin Roesch, "Snort - Lightweight Intrusion Detection for Networks," in Proceedings of LISA '99: 13th Systems Administration Conference, Nov. 1999, pp. 229–238.

[15] Scapy, "Scapy." Accessed: Mar. 29, 2022. [Online]. Available: https://scapy.readthedocs.io/en/latest/

[16] Ubuntu, "Ubuntu." Accessed: Mar. 29, 2022. [Online]. Available: https://help.ubuntu.com/

[17] pfSense, "pfSense." Accessed: Mar. 29, 2022. [Online]. Available: https://www.pfsense.org/getting-started/

[18] Wireshark, "Wireshark." Accessed: Mar. 29, 2022. [Online]. Available: https://www.wireshark.org/docs/

[19] W. Song et al., "A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection," Sensors, vol. 20, no. 6, p. 1637, Mar. 2020, doi: 10.3390/s20061637.

[20] K. Fotiadou, T.-H. Velivassaki, A. Voulkidis, D. Skias, S. Tsekeridou, and T. Zahariadis, "Network Traffic Anomaly Detection via Deep Learning," Information, vol. 12, no. 5, p. 215, May 2021, doi: 10.3390/info12050215.