Network packet analysis in real time traffic and study of snort IDS during the variants of DoS attacks

Nilesh Kunhare[1], Ritu Tiwari[2], and Joydip Dhar[3]

Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior, India
nilesh954@gmail.com

Abstract. This paper discusses the functionality of port scanning techniques used for accessing the IP addresses of vulnerable hosts present in the network. These techniques usually perform for network monitoring and troubleshooting purposes. On the other hand, the attackers use this utility to find the vulnerabilities in the network, gain unauthorized access, and penetrate the network system. The primary step taken by the attacker to bombard a targeted cyber-attack is the port scanning technique. Nowadays, port scanning becomes highly dispersed, sophisticated, compound, and stealthy, hence the detection techniques are unachievable. We also discuss the working mechanism of snort intrusion detection system (IDS) tool used for intrusion detection, architecture, its installation, the configuration of files, and detection techniques. In our experiment, we have installed, configured snort IDS with rule files in one machine and the traffic monitored for other machines connected in the network. This research work demonstrates the implementation of denial of service attack (DoS) variants in the real-time network traffic and ramifications of the attacks using snort IDS tool.

Keywords: port scanning, DoS attack, snort, detection techniques, network forensics.

1 Introduction

Information security has become a significant research area due to the expansion of computation power, the enormous speed of data transfer and expansion of computer networks. The exchange and sharing of information through the internet result in a compromise of the data because of the presence of malicious activities and threats over the network [1]. A secured system should possess confidentiality, integrity and availability in it [2].

1. *Confidentiality:* It includes encryption, security tokens, and biometric verification methods of the data to ensure the confidentiality information should not be accessible to unauthorized users. It includes encryption, security tokens, and biometric verification methods of the data to ensure confidentiality.
2. *Integrity:* The data should not be altered and modified by unauthorized users during the transmission. Integrity ensures the consistency, trustworthiness, and accuracy of the data. Checksums and access controls used for the verification of integrity.
3. *Availability:* The information must be available to the authorized users for access.

The information transmits through the network in the form of data packets. Therefore data packets considered as the basic entities in network communication systems. The information transmits from source to destination in the form of streamlined flows, including infinite duplicates of the data packet [3]. The data packet encompassed in the segment of the data, which includes the information of the protocol used during the transmission, the physical address of the destination, time to live, and other relevant information. Hence the security of a network depends on the surveillance of the network packets. The vulnerability of the hosts compromised by the hackers through information gathering includes port scanning of the victim's machine [4]. The process of port scanning defined as identifying the services available on the target hosts or network with the help of observing the response to connecting attempts. The number of 'ports' or 'doors' available by which the intruders unauthorized gain access to the resources of the network. The hackers use port scanning as the first step to look for the number of ports accessible on the target network and detect the malicious scans to exploit the vulnerability on the network, analyze the network traffic and collect credentials. Packet sniffing is the study of examining and observing the contents of the data segment, and their packets and log details collected with this process termed as packet logging. The packet capturing operates in a promiscuous

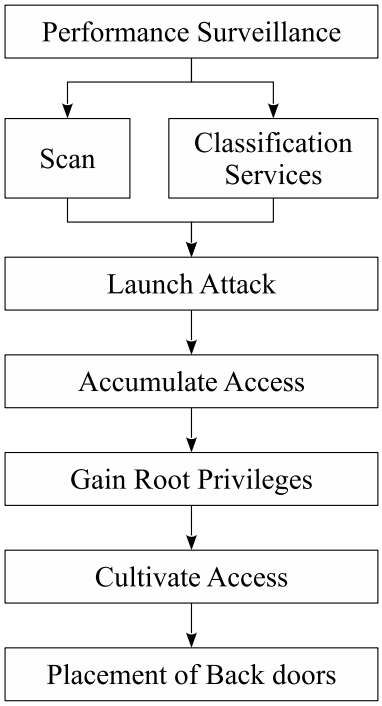

Fig. 1. Steps to perform attack

mode, which means that entire traffic passes through the Network Interface Card (NIC) are read whether it is transmitted or not to other machines. This paper illustrates the process of packet capturing passes through the network and also includes the installation of Snort IDS, the configuration of rules files and its observations for the malicious and normal traffic in the network. Figure 1 represents the steps pursued by an attacker for performing the attack in the system.

2 Related Work

Many organizations deployed NIDS for the cyber-security to prevent malicious activities from different layers of networks [4,5,6,7]. Snort is a network-based IDS used for detecting various intrusions and attacks. The authors discussed the protocol standards, inspection mechanisms, including signature matching, application control, and anomaly detection. Furthermore, analysis of application-level vulnerabilities including cross-site scripting, SQL injection attacks have been performed [8,9,10,11,12]. It uses various pattern matching algorithms[13] for the configuration, installation and designing rules of this tool. The biggest feature of snort is the ability to drop packets when handling with high speed, a gigantic quantity of traffic or massive packet size. The performance of snort analyzed on different processors Celeron, Pentium with contrasting operating systems Windows 7, XP, and Vista using network speed of 100 Mbps in [14]. The comparison of snort and suricata represented over 10 Gbps network speed. They concluded that snort good in detection accuracy and suricata can handle high-speed network [15]. The rule sets of both the IDS are common, the difference reflects in the designing architecture. Snort is single-threaded whereas Suricata is multithreaded. The experimental evaluation states that Suricata requires high processing power as comparing to Snort. The paper also concluded about the detection accuracy of both IDS in real-time environment [16]. Distinctive types of port scanning approach based on types, condition, and mechanism of detection techniques described approaching various datasets in [17]. An extensive survey of DDoS flooding attacks, detection, and prevention mechanisms discussed along with the counters measures in [18]. A semi-supervised approach proposed on KDD99 dataset using snort based statistical algorithm to improve the detection rate in [19].

3 Research gap

The snort IDS was configured and implemented on linux based system, the performance of the system analyzed using data mining techniques [19]. The alerts generated through base analysis security engine. The system configured with WinPcap packet capturing tool. However, the snort rules were not effective. Khamphakdee *et al.* [20] analyzed MIT- DARPA99 dataset for improving network probe attacks during four and five weeks. The wireshark tool used for the analysis of the dataset and the detection performance of network probe attacks correlated with detection scoring truth. The analysis of data took additional time to generate the pattern. The several patterns matching algorithms compared between malicious traffic and the standard dataset in [21]. The performance criteria were cpu utilization, throughput, and memory utilization. The algorithms do not give satisfactory results when performed on the dataset. However, the algorithms outperform for malicious traffic. In [22] performed stealth port scanning in the network and designed snort rules to identify the attacks and triggered alerts. However, the performance of Snort missing when increasing the number of systems in the network.

4 Research Methodology

In this paper, we have installed snort in one machine M1 and monitored the network packets passes over other machines. Figure 2 represents the proposed methodology. We performed the variants of DoS attack including ping of death attack, TCP-SYN flood attack, UDP flood attack in the network lab and observes the network pattern in snort. The details of the network traffic are captured in the log file of the snort and also captured by the wireshark tool. The algorithm used for capturing and filtering packets based on the protocol is mentioned below:
Variables:

$\text{Pkt}_i(\text{flag})$ = Return_Flag
$\text{Pkt}_i(\text{prot})$ = Return_Protocol

Inputs:
Arriving_Packets
Outputs:
Correlated_PacketVector

Step 1. Initialize:
Correlated_PacketVector $[pv_1, pv_2, \ldots pv_n] \rightarrow [0, 0, \ldots 0]$
Step 2. Process Arriving_Packets
Step 3. **if** (Packet i (prot) equals to TCP)
Go to Step 4
else go to Step 2
Step 4. **if** (Packet i (flag) equals to ACK or RST or ACK
Go to Step 2
else go to Step 5
Step 5. Correlated_PacketVector $\rightarrow$ Packet i // Summate packet to vector
Go to step 2

---

5 Types of port scanning

Many services are running in the machine, including TCP and UDP when it connects to the network. The TCP and UDP ports are used for communications between machines. There are total 65,536 ports available in a machine [23]. The attackers use these ports to gain access over the system. Table 1 show the types of ports with their ranges.

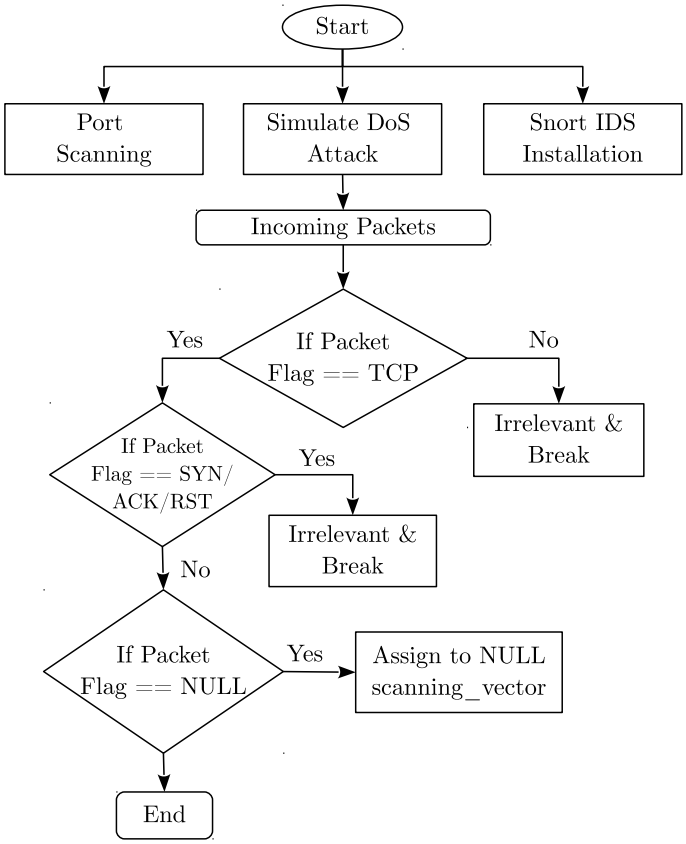

Fig. 2. Research Methodology

Table 1. Types of port scanning

Serial No.	Type of port	Range of port
1	Conventional ports	0–1023
2	Cataloged ports	1024–49151
3	Private ports	49152–65535

5.1 Denial of Service attack

These type of attacks become very harmful for legitimate users, and the attackers try to block the services by sending excessive requests to the server or the network. These attacks intend to slow down the services, bandwidth and as well as the network. To implement these attacks, the sender sends millions of requests contains a large number of packets with invalid data, flooding the target system in an attempt to slow down the network. The most intense form of this type of attack is Distributed Denial of Service (DDoS) attack, which makes the services unavailable to the users maliciously. Ping of death, TCP SYN flood attack, UDP floods, GET/POSTS floods, and fragmented packet attacks are the variants of these attacks[24,25]. DDoS is a form of attack where a single victim is targeted by multiple attackers (systems), causing a denial of service of the victim system. The target of the DDoS attack is to consume the availability of services providers in an attempt to make the systems unavailable for legitimate users. The DDoS attack divided into three parts:

1. *Volume based attacks:* The bandwidth of the network saturated by sending packet storm and the magnitude is deliberate in bits per second.
2. *Protocol attacks:* This type of attacks dissipates server resources, communication devices such as load balancers, firewalls, routers, switches, and is deliberate in packets per second.
3. *Application layer attacks:* The target of such type of attack is to clatter the webserver, and the magnitude is deliberate in solicitations per second.

5.2 Implementation of DoS attacks

The working mechanism of the variants of DoS attacks discussed and implemented below:
TCP-SYN flood attack: The attacker takes advantage of three-way handshake connection to allocate memory for the victim machine that never used and the legitimate users deny to access it. Whenever a TCP connection established a session is needed to be created by the host for communication. It is the starting phase for a three-way handshake. The SYN (synchronize sequence number) flag is set to 1 whenever the source node sends TCP packets to the destination. The packet comprehends source IP address associated port number, a destination IP address associated port number ,and many other associated fields required in the TCP packet. The destination node reply with SYN and ACK flags for the TCP connection set to 1. One more TCP packet is dispatched by the source machine to the destination machine using the ACK flag set to 1. These steps complete the three-way handshake, and the transfer of data takes place after this. The TCP-SYN flood attack executes when the sender not able to complete the last step of communication. The following commands are used to percolate the TCP-SYN attack:
hping3 —S —p 80 —flood —rand —source 192.168.40.66
S indicates SYN flag is set.
P is the destination port.
The attack is exploited from the machine 192.168.40.22 to the machine 192.168.40.66.

Ping of death attack: A large number of ping request with maximum packet limit are sent to the target machine in order to keep busy the target system in responding to the ICMP echo replies. The attacker deliberately sends IP packets larger than 65,536 bytes to the opponent. The command to perform ping of death:
ping 192.168.40.66 —t —l 65500.
t indicates the packets sent to the destination till the end of program.
l is the size of the packet.

UDP flood attack: this type of attack is performed by the attacker by sending floods of UDP packets to the victim machine.
The commands for performing UDP flood attacks.
hping3 2 —S —p 80 —flood 192.168.40.66.

No.	Time	Source	Destination	Protocol	Length	Info
66417	631.751078	192.168.40.66	3.116.248.23	TCP	58	80 → 40968 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66418	631.751090	192.168.40.66	183.244.108.219	TCP	58	80 → 40969 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66419	631.751102	192.168.40.66	39.178.130.178	TCP	58	80 → 40970 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66420	631.751114	192.168.40.66	148.182.132.90	TCP	58	80 → 40971 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66421	631.751126	192.168.40.66	29.23.46.12	TCP	58	80 → 40972 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66422	631.751139	192.168.40.66	139.6.87.71	TCP	58	80 → 40973 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66423	631.751152	192.168.40.66	20.92.48.176	TCP	58	80 → 40974 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66424	631.751164	192.168.40.66	143.213.121.78	TCP	58	80 → 40975 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66425	631.751177	192.168.40.66	151.204.121.42	TCP	58	80 → 40976 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
66426	631.751190	192.168.40.66	94.4.135.146	TCP	58	80 → 40977 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0

Fig. 3. TCP-SYN Flood attack with random source

No.	Time	Source	Destination	Protocol	Length	Info
45	18.385532	192.168.40.107	192.168.40.66	ICMP	60	Echo (ping) request id=0x450a, seq=0/0, ttl=64 (reply in 48)
48	18.386415	192.168.40.66	192.168.40.107	ICMP	42	Echo (ping) reply id=0x450a, seq=0/0, ttl=128 (request in 45)
50	19.385558	192.168.40.107	192.168.40.66	ICMP	60	Echo (ping) request id=0x450a, seq=256/1, ttl=64 (reply in 51)
51	19.385628	192.168.40.66	192.168.40.107	ICMP	42	Echo (ping) reply id=0x450a, seq=256/1, ttl=128 (request in 50)
55	20.385808	192.168.40.107	192.168.40.66	ICMP	60	Echo (ping) request id=0x450a, seq=512/2, ttl=64 (reply in 56)
56	20.385878	192.168.40.66	192.168.40.107	ICMP	42	Echo (ping) reply id=0x450a, seq=512/2, ttl=128 (request in 55)
59	21.385938	192.168.40.107	192.168.40.66	ICMP	60	Echo (ping) request id=0x450a, seq=768/3, ttl=64 (reply in 60)
60	21.386004	192.168.40.66	192.168.40.107	ICMP	42	Echo (ping) reply id=0x450a, seq=768/3, ttl=128 (request in 59)
64	22.386085	192.168.40.107	192.168.40.66	ICMP	60	Echo (ping) request id=0x450a, seq=1024/4, ttl=64 (reply in 65)
65	22.386156	192.168.40.66	192.168.40.107	ICMP	42	Echo (ping) reply id=0x450a, seq=1024/4, ttl=128 (request in 64)

Fig. 4. Ping of death attack

No.	Time	Source	Destination	Protocol	Length	Info
397	238.898087	196.132.195.165	192.168.40.66	UDP	60	1810 → 80 Len=0
398	238.898087	148.158.86.108	192.168.40.66	UDP	60	1811 → 80 Len=0
399	238.898088	172.123.149.55	192.168.40.66	UDP	60	1812 → 80 Len=0
400	238.898089	172.190.191.212	192.168.40.66	UDP	60	1813 → 80 Len=0
401	238.898090	104.27.166.200	192.168.40.66	UDP	60	1814 → 80 Len=0
402	238.898091	182.194.200.171	192.168.40.66	UDP	60	1815 → 80 Len=0
403	238.898091	221.81.21.96	192.168.40.66	UDP	60	1816 → 80 Len=0

Fig. 5. UDP flood attack

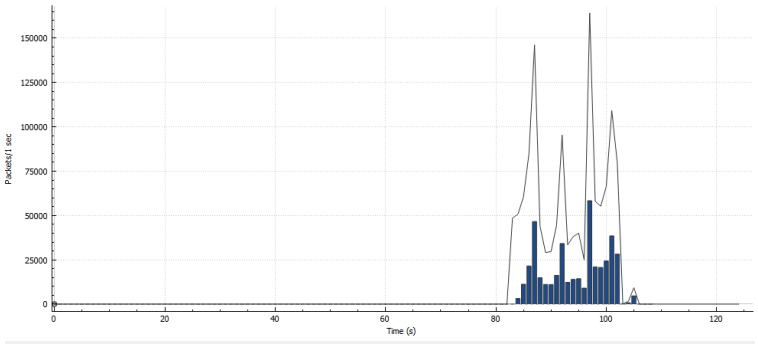

Fig. 6. Increase in consumption of network bandwidth during the DoS attacks

Figure 3, 4, and 5 represents the TCP-SYN flood attack, Ping of death and UDP flood attack respectively captured by the wireshark tool which is an open source packet sniffing and analyzer tool for the network [26]. Figure 6 represents the observations of network bandwidth during the attacks performed.

6 Snort and its components

The snort IDS is configured and deployed in the network for capturing the packet passes through the network. Snort is an open source network interference detection system refined by Martin Roesch have the capabilities to capture real-time network traffic and notify for any intrusions and alert the administrator. The snort IDS can perform protocol analysis, detect various types of attacks including buffer overflow, denial of service attack, port scans, OS fingerprinting and many more probes. The snort IDS can be configured in the following way:

1. *Packet Sniffer:* In this method, the incoming and outgoing packets pass across the network is captured by the Snort and all the details of the packets display on a console.
2. *Packet logger:* In this method, the packet details are logged and captured in the text file.
3. *Honeypot Monitor:* The snort have the ability to deceive the malevolent party.
4. *Network Intrusion Detection:* The snort performs analysis based on the signature rules on the network traffic to detect the intrusions and suspicious activities in the network.

The primary purpose of snort is to analyze the incoming and outgoing packet passes across the network, drop packets if it does not match with the signature rules and generate the report which includes information – packet drops, packet analyses, the packet received and other alerts including attacks and intrusions in the network. The architecture of snort represented in Figure 7. The major components of snort described as follows:
Packet decoder: The task of the packet decoder is to capture the packets passes across the network from the different network interface and prepare for preprocessing of the packets.

Preprocessor: The arrangement and modification of the packets take place in the preprocessing phase before it is dispatch for the analysis to the detection engine.
Detection engine: The function of this engine is to identify intrusions based on predefined definitions of the attacks. The packets are compared with the signature rules for the match if found, appropriate actions are suggested to discard or drop the packets.
Log and alert system: The log records are generated based on the results of the detection engine in the pattern of the text file or TCP-dump format. The alerts and logs can be modified using –l command.
Output modules: This module includes functions like log reports generation, database logging (MySQL), reporting to the server log.

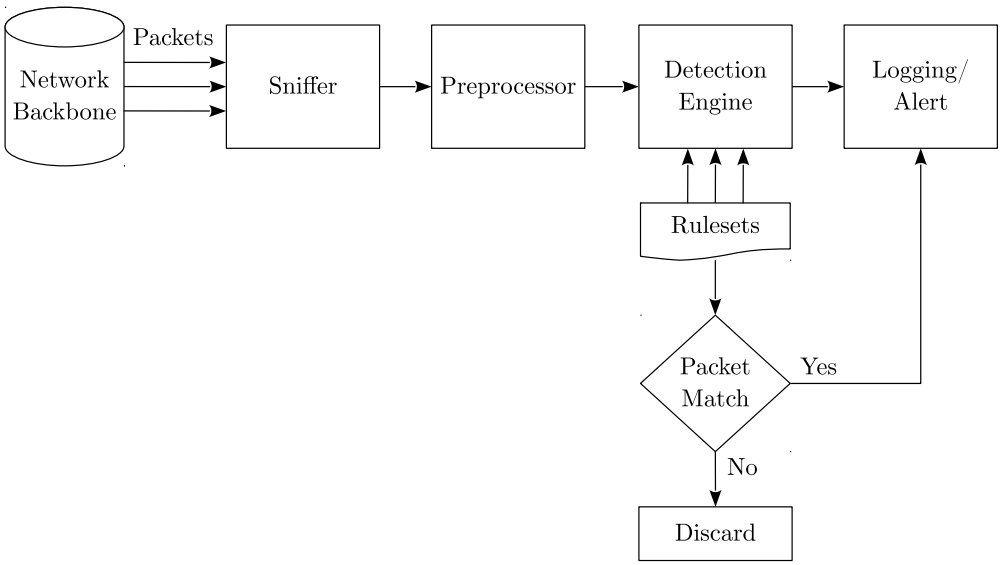

Fig. 7. Snort Architecture

6.1 Experimental setup and demonstration of snort IDS

The snort is an open source network intrusion detection system which can be deployed in any plate form (Windows and Linux). To set up the snort IDS we need winpcap, nmap, wireshark tools to be installed in the system. The snort can be operated in the following modes -

1. Snort as sniffer mode: This form of snort generates the network traffic summary captured during the packet transmission through the network. The network administrator can use this command in the command line prompt with the following syntax:
   # snort —v —d —e.
   —v displays the packet header with standard output.
   —d displays the packet payload information including UDP, TCP and ICMP packets.
   —e displays the link layer information.

2. Snort as packet logger mode: Once the packets are captured, the next step is to make a log record of these packets, which is performed by packet logger by using —l option in the command. The log details are stored in the /snort/log directory by default.
   # snort —l.
   To log the record of the subnet IP 192. 168.68.121 can be achieved by following syntax:
   # snort —vde —l C:\snort \log —h 192.168.68.121.

3. Snort as network intrusion detection mode: In this form, the Snort does not capture log file, instead, it performs detection based on the signature definition of the rules and generates the alert for any match found in the network. The command to start snort in NIDS mode is: # snort —c C:\Snort\etc\snort.conf

We observed the performance of snort for TCP, UDP and ICMP packets. The Snort triggers alert whenever any TCP packet passes across the network. The alerts also generate any ICMP and UDP packets. We have examined the collection of network packets passes through the network lab and observed the behaviour of snort IDS whenever the suspicious activity triggers an alert is generated based on the signature definitions in the rules files. The network topology represents the number of machines used for implementing the variants of DDoS attacks.

7 Results and Analysis

The port scanning is performed in the network by any machine and snort IDS is installed in the machine M1 to capture the traffic represented in the Figure 8. The malicious traffic is exploited from machine 2 to other systems and all the TCP, UDP, ICMP and other protocol supported packets are captured in the log records. The snort generates alert when any malicious traffic passes through the network. The snort triggers an alert based on the definitions of rules specified in the rule file. Some of the definitions configured in the rules are mention below.

Rule 1: alert icmp any any ->any any (msg: "ICMP packet alert"; sid : 1000001;).
Rule 2: alert tcp any any ->any any (msg: "TCP packet alert"; sid : 1000002;).
Rule 3: alert udp any any ->any any (msg: "UDP packet alert"; sid : 1000003;).
Rule 4: alert udp any any ->any any (msg: "FTP File access alert"; sid : 1000004;).
Rule 5: alert tcp any any ->any any (msg: "SYN Messages";flags: S; sid : 1000005;).
Rule 6: alert tcp any any ->any any (msg: "Scan Attack"; flow: to_server, not_established; threshold: type threshold, track by_src, count 15, seconds 30; flags: S; sid: 1000006;).

The rule file can be configured based on the definitions and the system triggers alerts whenever it matches the schema. Every packet transmits through the network is compared with the rule sets, if any match found the alert is stored in the log file. The parameters of the log file include a timestamp, alert message, source IP, destination IP, source port and destination port. The main objective of creating the network lab with Snort IDS is to collect the data packets passes across the network which includes malicious and normal traffic packets of TCP, UDP, ICMP and another relevant format. The malicious traffic is passed from machine 2 and snort triggers alert for this activity. Table 2 represents the statistics of the packet captured by the snort.

Table 2. Packets I/O Total

Received:	25056729	
Analyzed:	1786013	7.128%
Dropped:	23270716	48.152%
Filtered:	0	0.000%
Outstanding:	23270716	92.872%
Injected:	0	0%

8 Conclusion and Future Work

In this paper, we have discussed the port scanning techniques used by the attacker in order to access the information of the machines connected in the network. This paper demonstrates the denial of service attack

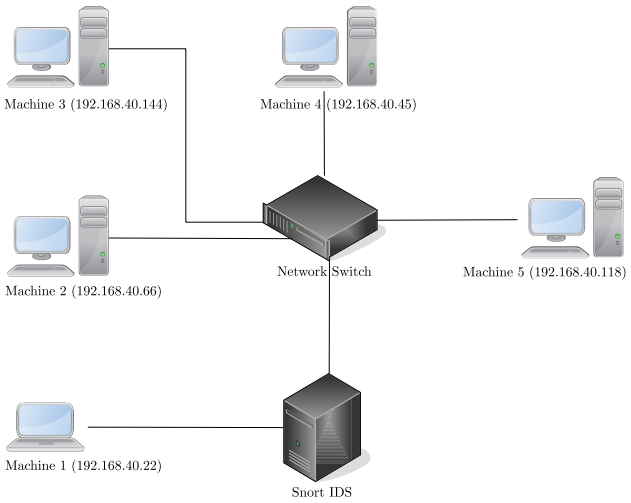

Fig. 8. Network of the Lab

Table 3. Details of Intrusion with corresponding machines

SN	Alert	Src_IP	Dst_IP	S_Port	D_Port	Intrusion
1.	TCP	192.168.40.22	192.168.40.66	445	3389	Y
2.	TCP	192.168.40.22	192.168.40.66	80	139	N
3.	FTP	192.168.40.22	192.168.40.45	1025	135	Y
4.	UDP	192.168.40.22	192.168.40.45	1028	1029	Y
5.	FTP	192.168.40.22	192.168.40.144	1026	1030	Y
6.	UDP	192.168.40.22	192.168.40.118	3107	3106	N
7.	UDP	192.168.40.22	192.168.40.25	500	138	N
8.	ICMP	192.168.40.22	192.168.40.66	85401	3094	Y
9.	SYN	192.168.40.22	192.168.40.45	1027	21	Y

and its variants. The simulation of DoS attacks in real time environment including TCP-SYN flood attack, Ping of death attack. We also discussed the architecture of snort IDS, installation and configuration of rule sets for the detection of intrusions in the network. The exploitation of malicious activities and normal traffic in real time systems are captured by snort IDS and alerts are triggered based on the signature definition and stored in the log file which can be used as a dataset having the information of TCP, UDP, ICMP and other relevant packet formats including the alerts for suspicious activities. The future work includes the categorization of machine learning algorithms in the snort IDS to indentify the detection rate and preciseness of the system.

References

1. M. F. Umer, M. Sher, and Y. Bi, "Flow-based intrusion detection: Techniques and challenges," *Computers & Security*, vol. 70, pp. 238–254, 2017.
2. S. William, "Cryptography and network security: principles and practice," *Prentice-Hall, Inc*, pp. 23–50, 1999.
3. W. Stallings, *Network Security Essentials: Applications and Standards, 4/e*. Pearson Education India, 2000.
4. Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, and S. Anwar, "Intrusion response systems: Foundations, design, and challenges," *Journal of Network and Computer Applications*, vol. 62, pp. 53–74, 2016.
5. E. Guillen, D. Padilla, and Y. Colorado, "Weaknesses and strengths analysis over network-based intrusion detection and prevention systems," in *Communications, 2009. LATINCOM'09. IEEE Latin-American Conference on*. IEEE, 2009, pp. 1–5.
6. L. Schaelicke, T. Slabach, B. Moore, and C. Freeland, "Characterizing the performance of network intrusion detection sensors," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2003, pp. 155–172.

7. N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, "Network attacks: Taxonomy, tools and systems," *Journal of Network and Computer Applications*, vol. 40, pp. 307–324, 2014.
8. A. R. Baker and J. Esler, "Snort intrusion detection and prevention toolkit," *Andrew Williams*, vol. 1, 2007.
9. W. Bul'ajoul, A. James, and M. Pannu, "Improving network intrusion detection system performance through quality of service configuration and parallel technology," *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 981–999, 2015.
10. K. Salah and A. Kahtani, "Performance evaluation comparison of snort nids under linux and windows server," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 6–15, 2010.
11. Y. Meng and L.-F. Kwok, "Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection," *Journal of Network and Computer Applications*, vol. 39, pp. 83–92, 2014.
12. I. Kim, D. Oh, M. K. Yoon, K. Yi, and W. W. Ro, "A distributed signature detection method for detecting intrusions in sensor systems," *Sensors*, vol. 13, no. 4, pp. 3998–4016, 2013.
13. A. V. Aho and M. J. Corasick, "Efficient string matching: an aid to bibliographic search," *Communications of the ACM*, vol. 18, no. 6, pp. 333–340, 1975.
14. W. Bulajoul, A. James, and M. Pannu, "Network intrusion detection systems in high-speed traffic in computer networks," in *e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on*. IEEE, 2013, pp. 168–175.
15. S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.
16. E. Albin and N. C. Rowe, "A realistic experimental comparison of the suricata and snort intrusion-detection systems," in *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*. IEEE, 2012, pp. 122–127.
17. M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," *The Computer Journal*, vol. 54, no. 10, pp. 1565–1581, 2011.
18. S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.
19. G. Nadiammai and M. Hemalatha, "Handling intrusion detection system using snort based statistical algorithm and semi-supervised approach," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 16, pp. 2914–2922, 2013.
20. N. Khamphakdee, N. Benjamas, and S. Saiyod, "Improving intrusion detection system based on snort rules for network probe attack detection," in *2014 2nd International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2014, pp. 69–74.
21. A. Mahajan, A. Gupta, and L. S. Sharma, "Performance evaluation of different pattern matching algorithms of snort," *International Journal of Advanced Networking and Applications*, vol. 10, no. 2, pp. 3776–3781, 2018.
22. R. R. Singh and D. S. Tomar, "Network forensics: detection and analysis of stealth port scanning attack," *scanning*, vol. 4, p. 8, 2015.
23. M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Ieee communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336.
24. M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection," *Pattern Recognition Letters*, vol. 51, pp. 1–7, 2015.
25. H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.
26. A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.