# An Approach for Network Based Intrusion Detection System using Snort

Pavithra P S, Durgadevi P

SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India

Corresponding author: Pavithra P S, Email: pp2616@srmist.edu.in

The newest technology, the Internet of Things, uses IoT devices to transmit data through networks. The biggest challenge with IoT is ensuring data transfer security. An intrusion detection system (IDS) is suggested as a solution to this problem. An essential network security tool for protecting computers and network systems is the IDS. It is capable of detecting and watching network activity. To find the unusual activity, we used the Snort IDS programme. An open source network security tool is Snort IDS. Both recognized and unidentified hazards might be found. To find attacks and produce alerts, it may search and compare the rules with network traffic data. This article examines protocol risks, attacks, and security problems related to network security. It also includes a plan to mitigate these risks. The MIT-DARPA 1999 data collection was used to produce the experiment's findings. The Snort IDS is used to identify anomalous behavior data since behaviour pattern datasets can be both normal and aberrant. The effectiveness of Snort's rule was assessed and put to the test in this article.

**Keywords**: Alert Correlation, Intrusion Detection system (IDS), NIDS, Security, Snort rules

2023. In Satyasai Jagannath Nanda & Rajendra Prasad Yadav (eds.), *Data Science and Intelligent Computing Techniques*, 481–491. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.56155/978-81-955020-2-8-44

# 1. Introduction

To protect their data and network security, numerous enterprises require a reliable security tool. The highest level of customer data security is especially necessary for businesses that offer internet banking, goods sales, and communication services. The business may suffer severe harm if the data is stolen by somebody with bad intentions. Additionally, network communication technology is rapidly evolving and becoming more complex. If the company lacks effective security solutions for data protection and network security, the attacker will be able to destroy the network in this way. A network intrusion detection system is a software tool that detects and keeps track of internet message flow. When it discovers abnormal packet data indicative of an attack technique, the system will issue a warning [9]. IDS are becoming a vital tool for securing data and network platforms as a result. Additionally, this is one of the most fascinating security study subjects for researchers worldwide. The two methods of detection are intrusion detection for anomalies and intrusion detection for breaches. The two most important classifications of intrusion detection systems are host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS), which include Snort. Protocols are necessary for an IDS to perform the different phases of intrusion detection. For all IDS kinds and methodologies, a significant amount of programmers are available. In the section headed "IDSs Developed for IoT Systems," a few of these IDS techniques will be briefly explored. Many of these IDS methods can also be applied to a variety of alternative detection methods. In accordance with their difficulty, implementation time, and time to detection specifications, compact anomaly-based intrusion detection system algorithms are the subject of this section. These techniques can be employed in IoT-based systems. The following discussion will utilize the PCA (principal component analysis) algorithm as an instance of representation because it is a simple approach that can be used for a variety of IDS detection techniques [14]. In accordance with Mori et al. [8], "principal component analysis (PCA) is a frequently employed description multimodal technique for managing qualitative information that can be expanded to handle combined measurement-level information." PCA has so been extensively used in a variety of disciplines.[10] According to, based on the variance-covariance pattern of the underlying factors, PCA produces a collection of variables[15]. These additional variables, which are less numerous than the initial factors, are linear amalgamations of the initial variables. PCA is a dimensionality-reduction and detection method utilized by IDSs. Elrawy et al.'s [7] anomaly-based statistics and information analysis IDS relies on the split of the principle elements into the most important and least significant main elements, and it was created using the PCA technique. The main primary component value and the smaller principal component score are used in this the system's detection stage [17]. Additionally, PCA has been utilised in intrusion detection methods that rely on algorithms for learning, statistical modelling, payloads modelling, and information mining. Detecting intrusions based on misuse. Utilising an archive of well-known fingerprints and behaviours associated with harmful software and breaches, misuse-based intrusion detection techniques can identify well-known attacks. Three drawbacks of misuse-based IDSs include system traffic exhaustion, the costly nature of trademark matched, and a frequent occurrence of false alerts. Fig1. Additionally [11], since misuse-based IDSs must retain an extensive collection of malicious signatures, the high memory limits in particular types of systems, such as WSNs, have a negative impact on their efficiency. Additionally, patterns-matching IDSs and identity-based IDSs also require ongoing updates to their signature and trends collections. These misuse-based IDSs are made to find malicious hacking and threats based on the past. identification of intrusions based on anomalies An anomaly-driven intrusion detection method creates an average data structure based on information from regular users, which is then contrasted online with the current data structures to find anomalies[19]. These occurrences are brought on by noise or other occurrences that may, in theory, be caused by malicious software. Abnormalities are hence uncharacteristic actions brought on by intrusions that leave traces in the computing environment[12]. ones, especially unidentified ones, are recognised by these tracks. An anomaly-based IDS functions by building an ongoing model of typical user activity in the computer context from information provided by everyday users, and utilising this model to identify any divergence from typical conduct. The benefits and drawbacks of several anomaly-based intrusion detection methods.

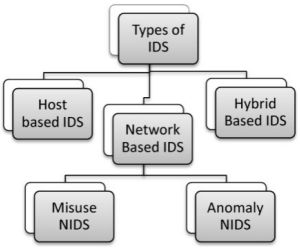


**Figure 1**. Types of Intrusion detection systems

## 2. IDS-Snort

Popular Intrusion Detection and Protection Systems (IDS/IPS) Snort is used to reduce the danger of an attack on the system. It is a simple open-source programme that Martin Roesch created in the C programming language in 1998. Snort can be installed on nearly every computer architecture and operating system environment Fig2. Real-time notifications are also generated by Snort-IDS [18]. In order to discover odd data packet traffic, it looks up each incoming packet in the internet traffic and analyses it according to specifications. One line is used to describe each Snort-IDS rule [13]. It is adaptable and easy to read and understand. The core components of Snort-IDS are the network device, pre-processor, monitoring system, monitoring and notification system, and efficiency module.

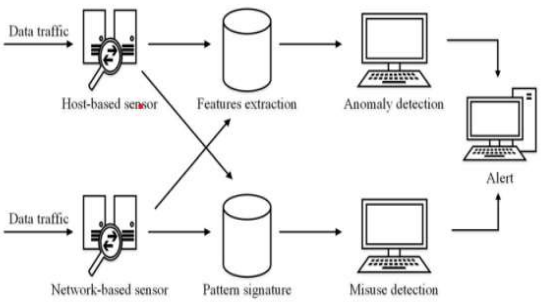


**Figure 2**. Different stage s of IDS operations

The Snort-IDS employs criteria that are appropriate for the lost packets incoming traffic. The basic structure of Malware guidelines is composed of the rule folder and the rule alternative, which are separated into two logical parts. Each item in the Snort-IDS frameworks' rule header. It includes an explanation of the criteria used to match rules to data packet traffic networks [2]. You can also define the form of action, such as approve, publish alarm, etc., in the rule header's action field. Following the rule header come the rule alternatives, which are separated by a pair of parenthesis. An application and an identifier are the typical components of each condition alternative Fig 3. The phrases pull from sentences that include a semicolon and a symbol. The term parameter is encased in double quotes, and each rule is followed by a symbol period.

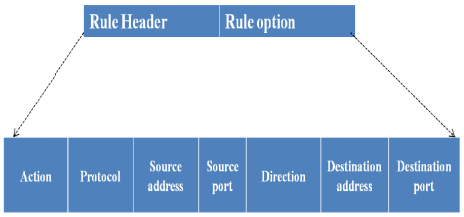


**Figure 3**. Structure of Snort Rule

The alert will be generated by this Snort-IDS rule. If the udp protocol is used, the target port number is 46, and the source IP address number 167.14.1.50 is identified from any port delivered to any destination Node (DNS). Additionally, it displayed the text "DNS request effort," and the rule's Sid number was 1010101010.

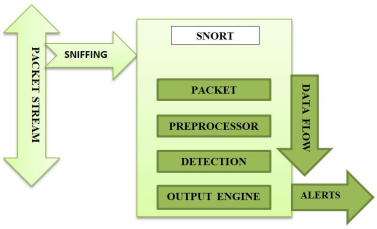


**Figure 4**. Component of Snort

**A.Packet Decoder**

While transferring the information to a pre-processing or monitoring machine, the protocol decoder collects data via various network connections. It is possible to use network device types such as the Internet, SLIP, PPP, and others.

**B. Preprocessor**

In order to apply an operation to a packet if it is corrupted, it interacts with Snort to arrange or change the packet before the detection engine. When anomalies are discovered in a packet, they will occasionally also create an alert. In essence, it matches the structure of the full string. The pre-processor organises the text, which allows the IDS to identify the thread; therefore, an intrusion can fool the IDS by modifying the sequence or by adding additional entries. Data redundancy is one crucial operation that the pre-processor performs. Due to the fact that attackers occasionally split the signature into two packets Fig 4. Therefore, both packets must be defragmented before the signature can be checked, and the pre-processor performs this step.

**C. Detection Engine**

With the aid of Snort rules, its primary task is to discover intrusion activity that escapes a packet. If this activity is discovered, the proper rule is applied; if not, the transmission is lost. Response times to different protocols vary depending on the strength of the technology and the number of declared criteria in the software.

**D. Logging and Alerting System**

This system is responsible for communication, packet monitoring, and alarm generation. Depending on what the security model finds inside a packet, including whether it should be used to track actions or generate alarms, the function of the message will be determined. By design, all log files are stored in a specific place. This address can be changed using command-prompt options. The type and level of

information that is logged by the logging and alerting system can be changed using a variety of command-line arguments. By default, all log files are kept in the C: Snort log folder, but the location can be modified by using the -l command-line option.

## 3. Releted Work

The Snort-IDS attacker detection tool is of interest to researchers from all around the world. The programme described allows the network manager to use graphical interfaces to construct Snort IDS rules and alarms. (GUI) [1]. Furthermore, we built a Snort-based signature-based improvement for detecting the atypical interconnection. The alarm results generated by the Snort-IDS are also shown using the Basic Analysis and Security Engine (BASE). However, the laws were not changed as a result of these investigations to make threat identification more efficient.

For the campus network, a decentralised detection and prevention model was developed using Snort-IDS. This study's main objective In order to increase the precision and effectiveness of the intrusion detection system, it was necessary to compare the analysis approach with the rule-based technique. They assessed the Snort-IDS notification system's functionality and examined unusual internet activity behaviours [2]. The capability of Snort-IDS to swiftly identify hazards on campuses was also evaluated. The performance showed that most Snort-IDS warnings were caused by ICMP PING attacks. These experiments just analysed the effectiveness of the Snort-IDS rules; no modifications to them were made.

The Snort Lab was presented by Jinsgeng Xu et al[3]. as a tool for teaching students how to create Snort-IDS rules. The students were given six questions to answer using the Snort-IDS criteria for each threat that was detected. Additionally, they employed electronic communication replication to evaluate the Snort-IDS protocols. The Snort-IDS investigation employing personalities was performed by Sagar N. Shah, Purnima Singh, and WinPcap. On the Windows operating system, they developed and tested In their research, they wish to examine the strange actions on the internet. The findings demonstrate that Snort-IDS are operating system-supported. Additionally, it can be set up to function as a firewall Fig 5. Mohammad Dabbour et al[4]. Further investigated the characteristics of Snort-IDS rules to construct and alter Snort-IDS rules for identifying and defending against threats on three different types of websites, especially SQL injection, XSS, and command execution. They did, however, show how the Snort-IDS rules' estimated performance had increased and provided instructions on how to create Snort rules. But it can only detect specific invasions [6].Some research made it easier for the network manager to look at the threats based on the prior study. Some research [5], however, only assessed the Snort IDS's capabilities. In this paper, new Snort rules for network probe assault identification are presented in an effort to enhance the intrusion detection system. We also classify a group of internet probe threats based on their characteristics.

## 4. Proposed System

This section explains the enhanced Snort-IDS rule procedure, which consists of the enhanced Snort-IDS rule method and the analytical data packet procedure. We therefore suggest a few Snort-IDS rule.

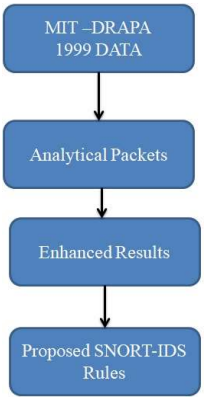


**Figure 5.** System Architecture

Several open-source and for-profit network products, such as packet analysts (packet sniffers), networking analyzers, network intrusion detection systems, bandwidth producers, and internet testers, use the datagram and processing engines of Win cap. Additionally, it allows for the reduction of packet headers to files and the reading of data containing retrieved packets. Implementations can be created using WinPcap that can either read saved captures and analyse them using the same analyse code, or they can be capable of capturing network traffic and analysing it. Programmes that comprehend that protocol, such as tcpdump, Wireshark, and CA Net Master, can read a detect that exists to protect in the WinPcap format.

Including viruses, most incursion behavior has an identifier. Data about these identities is used to create Snort rules. However, we may find out what invaders are doing, along with specifics about their tools and techniques, by using honeypots. Additionally, intruders intend to use databases of security breaches. These well-known exploits are also used as warning signs to check ifanyone is trying to exploit vulnerabilities. These identifiers might be present in a packet's header or contents. Snort bases its identification process on rules. These rules are built on the concept of attacker identities. Using Snort rules, a number of packets can be examined several times. Application layer methods cannot be examined by Snort 1.x generations, although level 3 and 4 identifiers may. The upcoming Snort version 2 is intended to provide support for application-level protocols. The rules are enforced identically for every payload, regardless of type. Rules can be used to monitor a response, produce an alert signal, pass the incoming packets, or secretly delete them in the instance of Snort. Pass has a different meaning in this context than it does in network firewalls [6]. On network firewalls, the pass and delete functions counter each other. The syntax for writing Snort criteria is easy to understand. Most of the rules are written in a single phrase. You can still extend rules across several lines by inserting a line break at the end of every line. Normally, rules are kept in the snort.conf file system. You can also use them by including extra files in the principal file name. The many categories of laws and how they work are discussed in this chapter. You'll discover a good number of references to standard rules for intrusion prevention activity towards the end of this book. After completing this chapter, the two that came before it, and this one, you should be able to configure Snort as a basic intrusion detection system.

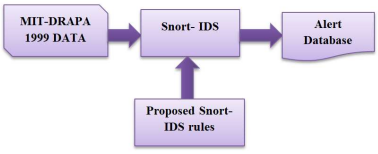


**Figure 6** Snort-IDS rules tested procedure

**A. Data packets procedure**

The enhanced analysis of the Snort-IDS rules method In this study, we make use of the MIT-DARPA 1999 dataset, which was used by the MIT Lincoln Laboratory to test and assess detection capability. The dataset comprises connections that are both normal and aberrant and was captured in a variety of file formats. In this paper, we make use of the tcpdump file format by selecting two files in weeks 4 and 5: inside.tcpdump and outside. tcpdump. Then we use Wireshark to read datasets from files that display the connections between each packet, such as the source and destination IP addresses, source and destination ports, flags, window size, data, and so on. These facts are crucial for determining the nature of attacks and enhancing Snort-IDS rules. It will decrease falsealerts and improve the accuracy of the detection rules. Most network administrators, however, are not represented in this data.

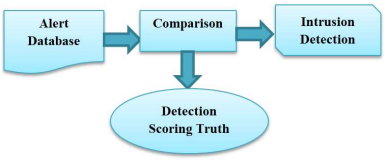


**Figure 7**. The procedure of the attack detection comparison

**B. Improved Snort-IDS Rules**

The analysed data packets between each network association in the dataset and the attacking event in Identification Assessment Truth are compared in this chapter to determine the targeting event. Examples of data comparisons include attack name, ID num, date, and start time. In this study, we only look at internet probe threats.

**TABLE 1.** RULE TYPES

| No. | Rules type | Total |
|---|---|---|
| 1 | probe portsweep.rules | 21 |
| 2 | \sprobeipsweep.rules | 7 |
| 3 | \sprobe Satan. Rules | 5 |
| 4 | \sprobe ls domain. Rules | 3 |

Each Snort-IDS rule is enhanced before even being saved as a text document. Network probing attacks come in a variety of forms, though. Therefore, we classify the internet probing threat into six types, as given in Table I. When the findings are ordered by rule type, the probe queso. Rule, which contains 28 regulations overall and the most rules in this article, comes in first. In contrast, probing the LS domain Rules only list one rule.

# 5. Snort-IDS Rules

Here, we go through the specifics of a few Snort Id criteria that are used to spot probing attacks.
notify udp any $HOME NET -> $100 $HOME NET Ftp scan attempts were made (msg: "Tcp Probe performed"; circulate; fragoffset:0; fragbits:!D; ack:0; signals; window:2058; classtype: connectivity; priority:3; sid:1010071;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "TCP Scan Echo"; itype:7; icode:0; fragbits:!D; content:"|00 01 10 01 10 01 10 01 10 01|"; depth:16; icmp_id:0; icmp_seq:0; ttl:274; classtype: attempted-recon; priority:5; sid:1010124)

Table I displays the rules for Snort-IDS. The attacking identification rules, known as the "portsweep attack," were enhanced in Example No. 1. The internet threat can be detected by the Snort IDS rules thanks to this rule. When a hacker tries to scan a machine with an open HTTP service, Snort-IDS will issue an alert (port number 80). Additionally, we use the criteria of windows, flags, and optional fragments when comparing the protocols. We enhanced the "ip sweep attack" attacking classifier. When an attacker attempts to find a potential IP address by sending icmp ping packets to a network from outside the system, Snort-IDS will issue an alert. We employ the criteria icmp id and icmpseq in the rule for comparing the packets in order to accurately identify the attack. However, for the information rule parameter, we define the corresponding binary values message packet: 01 10 01 10 01 10 01 10 01 10 01 10 01 which can improve the effectiveness of the Snort-IDS threat detection.

# 6. Performance Evaluation

The efficiency of identification is contrasted in this part using an exploratory method of the Snort-IDS rules. The evaluation system consists of four steps: the checked Snort-IDS rules process, the tested internet testing attacks method [18], the checked Snort-IDS policies procedure's comparison efficiency in detecting attacks, and the proposed Snort-final IDS's report of detecting attacks.

**A.Threats on categorized network probe**

The Dos, U2R, R2L, and Probe assault events make up Detection Scoring Truth. The amount of typical true positive alarms that the three IDSs reported throughout the course of an 18-hour period, during which the various attacks were injected. For the five threats, Snort's average performance was DoS (95%), probe (97.4%), U2R (96%) and R2L (97%).

The average number of false positive alarms generated by Snort IDS throughout the 18-hour period, which was divided into 6 hour blocks, was 3.7%. The average number of false positive alarms generated by Snort IDS throughout the 18-hour period Table II, which was divided into 6 hour blocks [17], U2L attack false positive rate for Snort was whereas for scan attack, Snort reported a false positive rate of 7.0%. Finally, it was discovered that of the alerts generated by Snort IDS for U2R attacks were false positives.

A false negative's effects Snort had a false negative rate. Snort did not raise a false negative alarm in response to a probing attack, had a 1.0% false negative rate for a U2L attack, and had an averageof false negatives. For a U2R attack, Snort raised a false negative alert percentage.

**TABLE 2**. NETWORK TRAFFIC ANALYSIS

| Attacks | True Positive | False Positive | False Negative |
|---|---|---|---|
| Probe | 97.4 | 3.7 | 1.7 |
| Dos | 95 | 9 | 0 |
| U2R | 96 | 2.3 | 1 |
| R2L | 97 | 7 | 4 |


**Figure 8.** Detection of threats

**B.Normal Traffic Classification by IDS**

In this discussion, the effects of processing typical traffic, such as TCP, UDP, and ICMP through the three IDSs, are studied. For TCP packets Table III, Snort detected a 9% false positive rate, but no false negative or true positive rates for the same packet type. Snort generated 12% false positive rate alarms, 1% false negative alarms, and no alarm for real positive when processing the UDP packets. Snort had 5% false positives on ICMP packets [16]. Snort had 0% false negative alerts, therefore there were none; nevertheless, it 1% real positive alarms. The classification of typical traffic using Snort IDS findings shown in Figure 9.

**TABLE 3**. PROTOCOL TRAFFIC ANALYSIS

| Normal Traffic | Snort IDS | | |
|---|---|---|---|
| | FPR (%) | FNR (%) | TPR (%) |
| TCP | 9 | 1 | 1 |
| UDP | 11 | 2 | 1 |
| ICMP | 4 | 0 | 2 |

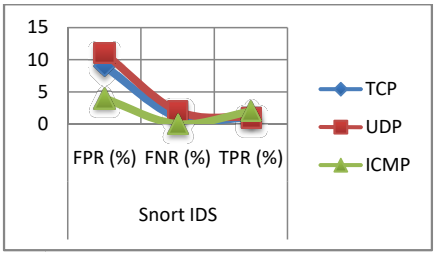


**Figure 9.** Protocol Traffic Analysis

## 7. Conclusion

In this article, many Snort-based intrusion detection strategies are explored in order to maintain an institution's security against threats after becoming familiar with IDS and its categories. Effective rules and procedures can be used by Snort-based IDS to protect against aberrant activities. Different issues are highlighted and explored that need to be taken into account when creating effective IDS for network Layer. This study suggests a design to boost Snort IDS's effectiveness. The effectiveness of an intrusion detection system based on Snort can still be improved in a variety of ways. Future work will involve integrating the suggested architecture into the Snort tool and testing it for improved detection rates with fewer false alarms.

## References

[1] Mahfouz A, Abuhussein A, Venugopal D, Shiva S. 2020. Ensemble classifiers for network intrusion detection using a novel network attack dataset. Future Internet 12(11):1–19 DOI 10.3390/fi12110180.

[2] McIntosh T, Jang-Jaccard J, Watters P, Susnjak T. 2019. The inadequacy of entropybased ransomware detection. Communications in Computer and Information Science 1143 CCIS:181–189 DOI 10.1007/978-3-030-36802-9_20.

[3] Mirsky Y, Doitshman T, Elovici Y, Shabtai A. 2018. Kitsune: an ensemble of autoencoders for online network intrusion detection. 18–21 DOI 10.14722/ndss.2018.23204.

[4] Papastergiou S, Mouratidis H, Kalogeraki E-M. 2020. Cyber security incident handling. Warning and response system for the european critical information infrastructures (CyberSANE). Communications in Computer and Information Science 1000:476–487 DOI 10.1007/978-3-030-20257-6_41.

[5] Park JH. 2019. Advances in future internet and the industrial internet of things. Symmetry 11(2):244 DOI 10.3390/sym11020244.

[6] Saleh AI, Talaat FM, Labib LM. 2019. A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. Artificial Intelligence Review 51(3):403–443 DOI 10.1007/s10462-017-9567-1.

[7] Sarker IH, Kayes ASM, Badsha S, Alqahtani H, Watters P, Ng A. 2020. Cybersecurity data science: an overview from machine learning perspective. Journal of Big Data 7(1):1–20 DOI 10.1186/s40537-020-00318-5

[8] Sommestad T, Holm H, Steinvall D. 2021. Variables influencing the effectiveness of signature-based network intrusion detection systems. Information Security Journal: A Global Perspective DOI 10.1080/19393555.2021.1975853.

[9] Vaiyapuri T, Binbusayyis A. 2020. Application of deep autoencoder as an one-class classifier for unsupervised network intrusion detection: a comparative evaluation. PeerJ Computer Science 6:1–26 DOI 10.7717/peerj-cs.327.

[10] Verma A, Ranga V. 2020. Machine learning based intrusion detection systems for IoT applications. Wireless Personal Communications 111(4):2287–2310 DOI 10.1007/s11277-019-06986-8.

[11] Zhang K, Zhao F, Luo S, Xin Y, Zhu H. 2019. An intrusion action-based IDS alert correlation analysis and prediction framework. IEEE Access 7:150540–150551 DOI 10.1109/ACCESS.2019.2946261.

[12] Zhou Y, Cheng G, Jiang S, Dai M. 2020. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks 174:107247 DOI 10.1016/j.comnet.2020.107247.

[13] Xiaojin Hong, Changzhen Hu, Zhigang Wang, Guoqiang Wang and Ying Wan, "VisSRA: Visualizing Snort Rules and Alerts," In Proc. Of Fourth International Conference on Computational Intelligence and Communication Networks (CICN), pp.441-444, 2012.

[14] XinyuGeng, Bing Liu and Xiaoyan Huang, "Investigation on Security System for SNORT-Based Campus Network," In Proc. Of 1st International Conference on Information Science and Engineering (ICISE), pp.1756-1758, 2009.

[15] Suman Rani and Vikram Singh, "SNORT: An Open Network Security Tool for Intrusion Detection in Campus Network Environment," International Journal of Computer Technology and Electronics Engineeering (IJCTEE), 2012.

[16] Jinsheng Xu, Jinghua Zhang, TriveniGadipalli, Xiaohong Yuan and Huiming Yu, "Learning Snort Rule By Capturing Intrusions in Live Nerwork Traffic Replay," Proceedings of the 15th Colloquium for Information Systems Security Education (CISSE), pp.145-150, 2011.

[17] Mohammad Dabbour, IzzatAlsmadi and Emad Alsukhni, "Efficient Assessment and Evaluation for Websites Vulnerabilities Using SNORT," International Journal of Security and its Applications, pp.7- 16, 2013.

[18] Haroon A., Naeem W., Shah M. A., Kamram M., Asim Y. &Javaid Q. "Constraints in the IoT: The World in 2020 and Beyond". International Journal of Advanced Computer Science and Applications,Vol. 7, No. 11, 2016