SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 19, No. 1, February 2022, 67-78
UDC: 004.7:004.056.53 **DOI: https://doi.org/10.2298/SJEE2201067G**

Snort IDS System Visualization Interface for Alert Analysis

Nađa Gavrilović[1], Vladimir Ćirić[1], Nikola Lozo[1]

Abstract: Over the past decades, the rapid Internet development and the growth in the number of its users have raised various security issues. Therefore, it is of great importance to ensure the security of the network in order to enable the safe exchange of confidential data, as well as their integrity. One of the most important components of network attack detection is an Intrusion Detection System (IDS). Snort IDS is a widely used intrusion detection system, which logs alerts after detecting potentially dangerous network packets. A major challenge in network monitoring is the high volume of generated IDS alerts. A necessary step in successful network protection is the analysis of the great amount of logged alerts in search of deviations from normal traffic that may indicate an intrusion. The goal of this paper is to design and implement a visualization interface for IDS alert analysis, which graphically presents alerts generated by Snort IDS. Also, the proposed system classifies the alerts according to the most important attack parameters, and allows the users to understand evolving network situations and easily detect possible traffic irregularities. An environment in which the system has been tested in real-time is described, and the results of attack detection and classification are given. One of the detected attacks is analyzed in detail, as well as the method of its detection and its possible consequences.

Keywords: Intrusion Detection, Alert Analysis, Snort, Visualization.

1 Introduction

Recent technological advances have led to the use of technology in very important areas, such as e-commerce, banking, insurance, health systems, etc. The unlimited possibilities of the Internet and the ease of communication also bring a significant risk of various attacks on users and their data. One of the primary requirements of network users has become the design of a secure network infrastructure which will enable safe data transmission and storage. Despite the existence of different systems for detecting and preventing attacks, the problem is still present today. Malicious attacks are becoming more sophisticated, which makes their detection even more difficult [1, 2].

[1]Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, Nis, Serbia;
E-mails: nadja.gavrilovic@elfak.ni.ac.rs; vladimir.ciric@elfak.ni.ac.rs; nikolalozo@elfak.rs

Within large companies and organizations, the danger comes not only from the outside network, but also from the inside, sometimes by employees who can use their access for malicious purposes, but even more often as a result of social engineering attacks.

Intrusion detection is an ability to detect any kind of unauthorized access to a computer system or network. An intrusion into a system can be defined as an attempt by an intruder to gain access to a computer or network by bypassing the security mechanisms. Also, intrusion often has the intention of compromising the CIA (Confidentiality, Integrity and Availability). An intrusion detection system (IDS) is a system used to address the problem of intrusions by monitoring network traffic and events, analyzing them in detail and detecting unauthorized intrusions [1, 3].

IDSs can be classified based on the monitoring scope into host-based IDS (HIDS) and network-based IDS (NIDS). Host-based intrusion detection systems monitor events happening within single host. They analyze process identifiers, the system calls they make and operating system specific logs in order to detect evidence of suspicious activity. On the other hand, the monitoring scope of a network-based intrusion detection system is a whole network. NIDS are responsible for detecting network traffic that may be considered unauthorized and harmful [3, 4].

One of the most widely used network intrusion detection systems is Snort IDS. Its simple configuration and efficiency make it the preferred option in most environments in need of protection [4]. Snort is often the subject of research in the field of network security. Recently, there have been various papers studying the implementation of Snort in different environments [5, 6]. Also, authors have studied ways to improve the network attack detection rate of the Snort intrusion detection system [7, 8]. In most environments, Snort IDS is configured to log alerts after detecting potentially dangerous network packets. Successful network protection requires a detailed analysis of those logged alerts. Part of the logs is usually normal traffic, but of interest is the part that can indicate an intrusion.

One of the biggest challenges in monitoring is the high volume of generated alerts. IDS systems are constantly becoming more advanced, but also more sensitive to different types of attacks. One of the consequences is a high rate of false positive alerts, which makes detecting real network danger a daunting task. Also, inability to find a connection between a large number of alerts makes the process of protection and prediction the attacker's next step even more difficult. Therefore, successful extraction of significant information from the high-volume IDS data is a task of great importance.

The goal of this paper is to design and implement a visualization interface for alert analysis, which graphically presents alerts generated by Snort IDS. The implemented system allows the users to visually analyze generated traffic logs

and easily identify attack patterns. Furthermore, it shows the most common source addresses, classes and dates of attacks, as well as the most common alert priorities. Such IDS alert analysis makes detecting network irregularities quick and straightforward. The results of real-time attack detection and classification in an appropriate environment are shown in detail. An example of an attack is analyzed, as well as the method of its detection and its possible consequences.

The paper is organized as follows. Section 2 gives a review of related work in the area of network security visualization systems. Section 3 gives a brief introduction to intrusion detection systems in general, and the Snort IDS. Section 4 gives the proposed system's interface design and architecture. In Section 5 the implementation results are presented, while the concluding remarks are given in Section 6.

2 Related Work

Significant amount of work has been recently published in the area of network security visualization systems and tools. Visual network data analysis helps network administrators to identify traffic patterns and trends, but also notice possible security deviations. Also, by analyzing generated traffic logs, network event visualization makes planning of the necessary security actions and steps faster and more straightforward [9, 10].

Visualization systems can have different input data, which include raw network data, network events from network devices (routers, switches, etc.), security events from IDS/IPS/firewall systems, and even the application logs [10].

Hao, et al. [11] implemented a web-based visualization solution which uses different types of user-configurable charts to do network traffic analysis. As input data, they used netflow data and Snort IDS alerts. In their system, the emphasis is on a detailed traffic flow analysis based on 2D charts, while in the system implemented in this paper the idea is to allow the system administrator to quickly and easily see what types of attacks occur in the network, and then what are the attributes of these malicious packets. Also, unlike ours, their system requires user interaction and customization. Another system that also uses both raw data and Snort alerts is presented by Dasireddy, et. al [12]. They created two separate models. First model gives the logical topology of the network in which the nodes contain information about their related alerts. The second, flocking model, presents the visual representation of IDS data, where alerts with maximum similarity are clustered together. The goal of visualizing alerts in their system is similar to that presented in this paper, because their system also displays the most common groups of attacks. However, the method of implementation differs, because their system visually connects correlated alerts, while the system

proposed in this paper processes information from alerts, sorts them and displays them in text form.

Visualization system which as an input source has IDS alert logs is presented by Shi, et. al [13]. They presented a radial visualization system with an interesting and novel approach. The principal visual components they used correspond to planet's ring systems, crust, and core. In the results discussion section, they emphasize that real-time monitoring requires carefully planned optimization of the preprocessing strategy, which is a conclusion of great importance for the implementation of the system proposed in this paper.

With constantly growing quantity of network event traces, it is necessary to pay attention to the monitoring system robustness and the possibility of processing a significant amount of data. A lot of popular open-source network visualization solutions are too robust and require the whole stack [14 – 16]. Instead, our motive was to make a light-weight solution for environments where basic assistance to an administrator in SOC (Security Operations Center) is desired, but without the need for the whole stack. Also, the advantage of the proposed solution is that the visual response is immediate, without any intermediate components for alert preprocessing.

3 Background on IDS and Snort

IDSs can be classified into signature matching and anomaly-based IDSs, based on the classification technique used to divide the network packets in two groups - regular and malicious. Signature-based systems use the detected properties of previous attacks for detection. A signature is a pattern of a known attack or threat, which is previously identified and stored in a database. Pattern matching IDSs compare network traffic to malicious attempt patterns in order to recognize possible intrusions. Aside from being efficient, signature-based detection technique can experience problems with detecting new and previously unknown threats [1, 17].

Anomaly-based detection is the process of comparing network traffic against the definitions of normal network activity in order to recognize significant deviations. A network traffic anomaly is considered to deviate from known traffic behavior so significantly, that it raises the suspicion of being a malicious attempt. These systems can detect zero-day attacks, but can also experience false positive alerts [1, 3, 17]. Also, in recent IDS systems, artificial intelligence is often used, most often in conjunction with the aforementioned techniques, in order to further improve system detection [18].

In terms of NIDS components, a typical NIDS gathers data from the network, distributes it to the network sensors and further to network analyzers, which classify data as either safe or malicious and determine the threat level. NIDS also includes an alert notifier, which generates on-screen, audible or e-mail alerts,

SNMP messages, etc. Furthermore, the command manager is a component that acts as a central command authority. Database servers usually include both behavioral and misuse statistics and other data [3].

Most NIDS implementations use multiple sensors, which have to be carefully placed on the key points of the network. They can be deployed in one of two modes. An inline sensor is typically placed at the network border, in order to monitor all the traffic which passes through the network. A passive (tap) sensor monitors mirrored network traffic, instead of actual traffic. They typically monitor network traffic from the key network locations [17]. In this paper, we will focus on pattern matching based IDSs, which have a network sensor configured in passive (tap) mode.

Snort is a widely used, highly configurable and portable, open-source network intrusion detection system based on pattern matching. Snort is easily deployed on a variety of network nodes. Also, its operation is efficient and does not take much memory and processor time [4]. Snort uses a set of signatures, which define what constitutes an attack and thus enable detection of attacks and malicious activities. Snort's signature sets are called Snort rules. A rule is formally defined as [7]:

```
<rule action><protocol><source ip><source port>
 <direction><dest ip><dest port><rule options>
```

Rule action field defines the type of Snort rule (alert, log, drop). The most common are alert rules, which store alert data for further analysis and later retrieval. The rest of the fields describe the main attributes of network packets. The rule options field defines one or more key-value pairs that further describe the rule (class type, msg, flags, etc.). Each rule also assigns priority to the alert, according to the alert class. A priority of 1 indicates the most serious threat, and priority of 4 indicates the least severe one [7]. Example of a Snort rule:

```
   Alert tcp $EXTERNAL NET any ->$HOME NET any (msg:
'SCAN SYN FIN' flags: SF, 12; reference: arachnids, 198;
          classtype: attempted-recon;) [4].
```

The main components of Snort architecture are given in Fig. 1. The packet sniffer collects network traffic and directs it to the decoder, which processes captured packets in order to isolate protocol headers at each of the OSI layers. The actual intrusion detection is done in the detection engine unit. This module analyzes each packet and checks it against all of the rules. The action (logging and/or alerting) specified by the rule is triggered every time a packet is detected which meets the condition defined by the rule [4, 19].

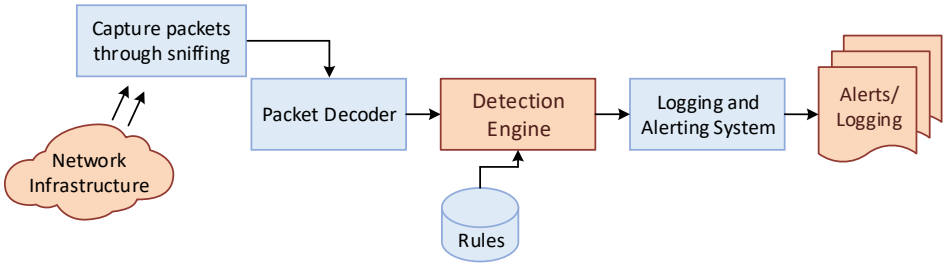

Fig. 1 – *Snort Architecture.*

4 System Overview

The proposed Snort IDS system visualization interface is implemented as a client-server application, which structures and graphically presents traffic alerts logged by Snort. The proposed interface allows users to visually analyse the traffic logs in real-time and easily detect deviations from normal traffic that may indicate an intrusion.

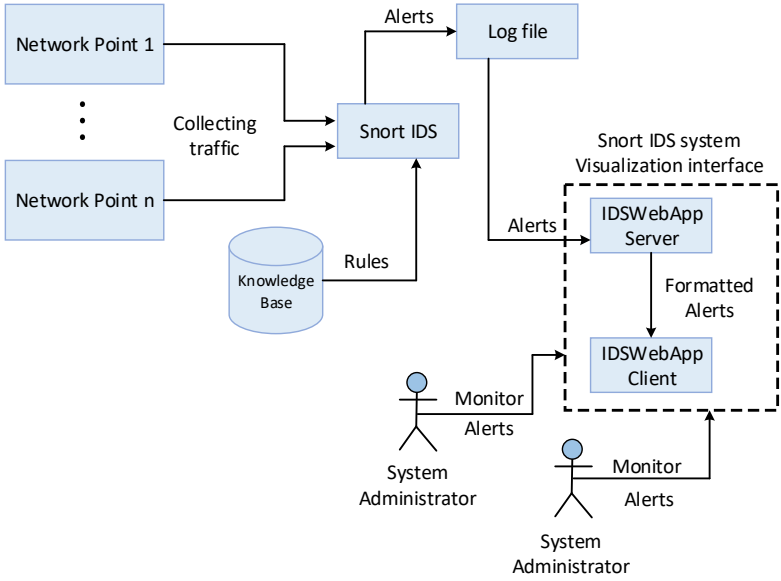

Fig. 2 – *Architecture of the Proposed System.*

In order to demonstrate and evaluate the proposed solution, the Snort visualization interface has been integrated into the system whose components are given in Fig. 2. From the carefully selected key points on the network, the traffic

is sent to the machine on which the Snort IDS is executing. For signature matching Snort uses an open-source registered rule base. In order for the detection to be as accurate as possible, it is important to refresh the database regularly. Snort IDS logs alerts on the machine's file system in JSON format. The implemented Snort graphical interface reads the data from the log file, processes it, and displays it to the user [20].

By using the client-server model, the proposed visualization interface allows the efficient graphical presentation of alerts generated by Snort IDS. The server side of the application (IDSWebApp server) reads the Snort IDS log file at application startup (an initial read), as well as each time that file changes, that is, when Snort generates a new alert. The IDSWebApp server also formats these alerts so that they can be sent to the client properly. Finally, the IDSWebApp server sends collected alerts to the client using the web socket.

The client side (IDSWebApp client) receives data through the web socket and reads the alerts sent by the server. Its main function is to organize and display Snort alerts to the system administrator in real-time, by refreshing the interface with every new alert. This results in an instant display of new alerts to the user. In addition, the client sorts the alerts received from the server by four criteria and displays the most common source addresses of the attack, the most common alert priorities, the most common classes of attacks, and the most common dates of attacks. Alert statistics are regularly updated, thus showing the most common attack attributes. Also, if the user selects a particular alert, the IDSWebApp client will display detailed information about it. In that way, the detection of possible traffic irregularities becomes quick and straightforward.

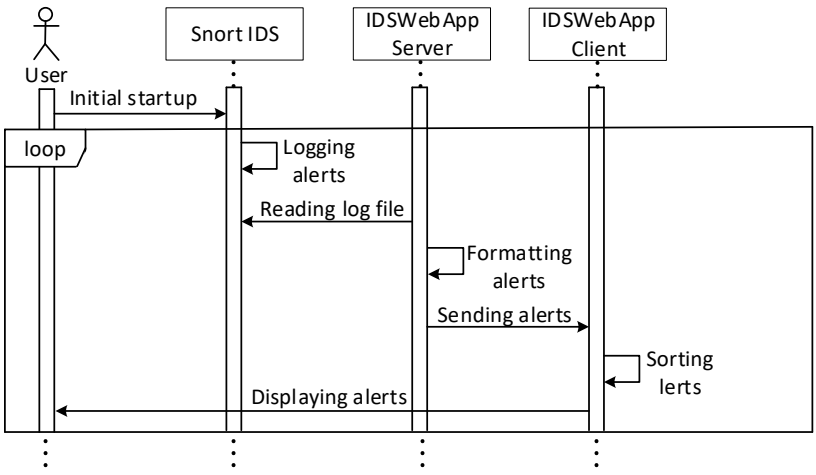

Fig. 3 – *Sequential diagram of the Snort IDS visualization interface operation.*

The operation of the graphical interface and client-server communication in order to display the warnings to the user is given in the sequential diagram in Fig. 3.

5 Implementation Results

The client side of the Snort IDS system visualization interface (IDSWebApp client) was developed in the TypeScript programming language and the Angular framework. JavaScript programming language, Node.js runtime environment and the Express.js framework were used to develop the server side of the application (IDSWebApp server). Client-server communication between them is done through the Socket.io library. Snort IDS, version 3.0, is running on a Linux Ubuntu 20.04 server, with 4 processor units and 16GB of RAM.

The proposed Snort IDS visualization interface is shown in Fig. 4. The evaluation of the interface, integrated into the system from Fig. 2, was performed over two days, in two different environments. The goal of such testing was to evaluate the implemented solution in different situations and environments in order to assess the behavior of the system in different conditions and types of traffic. The first part of the testing was done on the public network and publicly available servers during the time when the amount of traffic (and attacks) is greatest, while the second part was performed on the private network with simulated attacks. In both cases, traffic was observed from the two servers on the network which have the highest access rate, process the largest amount of data, and are most vulnerable to attacks.

The first part of the testing was performed during the working hours from 11 am to 2 pm. During that period, the monitored network points were completely opened to the Internet, without any protection in the form of firewalls.

During the second part of the system evaluation, the system was tested on an internal network protected from outside intrusions by a firewall, for the duration of one hour, when attacks were simulated using the Kali Linux. The purpose of this testing was to simulate an attack coming from a local network, possibly as a result of a social engineering attack. For attack simulation purposes Vega vulnerability scanner was used, which can execute different attack attempts in order to find and validate SQL Injection, Cross-Site Scripting (XSS), and other vulnerabilities. The obtained results are shown in Fig. 4.

At the top of Fig. 4, in the Live alert log section, a list of all network packets that Snort IDS has logged is displayed. The bottom four sections show the most common source addresses, classes and dates of attacks, as well as the most common alert priorities. In the Top classifications section, it can be noticed that the highest number of packets is classified with the class “none”, which is of low priority. However, three classes of attacks are also shown – Attempted

Administrator Privilege Gain, Web Application Attack, and Misc Attack, with medium and high priority.

Fig. 4 indicates a large number of packets (~100k) that generated the alert, most of them with low priority. The administrator should pay attention to this type of warnings, and check if there is a reason for further packet investigation. However, packets that generate a higher priority and a specific attack class are those that certainly require more detailed analysis and further action.

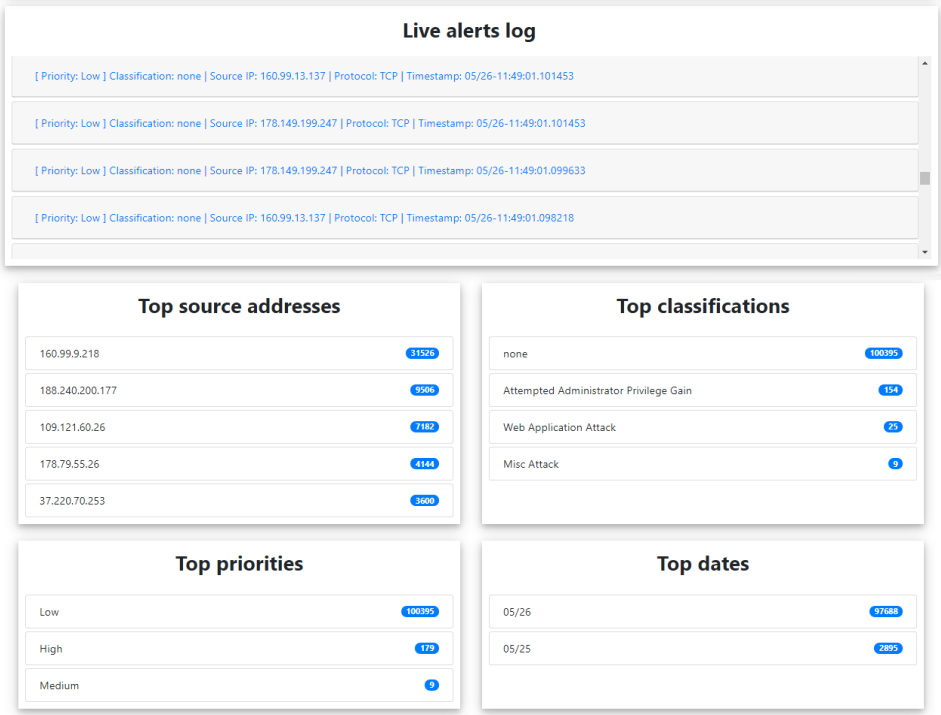

Fig. 4 *Snort IDS visualization interface.*

In this regard, the proposed graphical interface can also display the alert attributes, by clicking on a specific alert. The attribute values of one of the detected attacks are shown in Fig. 5.

The “Misc (miscellaneous) attack” alert displayed in the figure provides detailed information about the source and destination IP address of the logged packet, protocol, required service, TCP port, timestamp, attack class, etc. The attributes shown reveal very important information that can help detect intrusions and take the necessary measures promptly to prevent or stop the attack. This warning displays an SQL injection attack attempt, which can be seen by the

message field. Detection of such a packet indicates that there is a rule in the Snort database that marked this network packet as an attack by the pattern matching process. The Snort rule that caused this particular warning is:

```
alert tcp $EXTERNAL_NET any ->$HOME_NET $HTTP_PORTS ( msg:"SQL
union select - possible sql injection attempt - GET parameter";
  flow:to_server,established; http_uri; content:"union",fast
          pattern,nocase; content:"select",nocase;
  pcre:"/nunionns+(allns+)?selectns+/i"; metadata:policy max-
    detect-ips drop,policy security-ips drop; service:http;
        classtype:misc-attack; sid:13990; rev:26; )
```

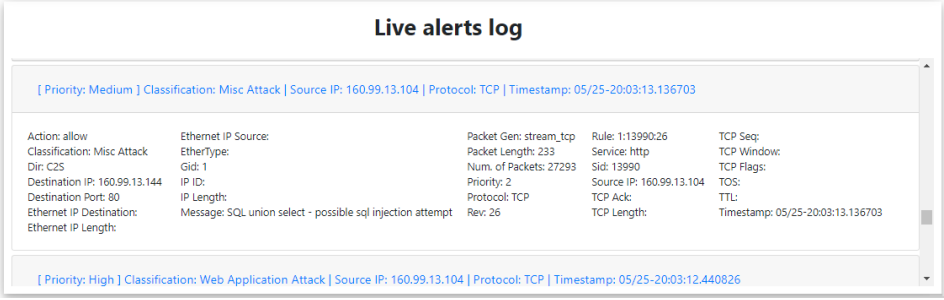

Fig. 5 – *Misc attack alert example.*

The rule is shortened for better display, but all relevant attributes are shown. It can be noticed that this rule recognizes packets which can be a case of an SQL injection attack, due to the characteristic content that contains the words "union" and "select". SQL injection is very popular among intruders. It is a kind of web security vulnerability which allows an unauthorized user to interfere with the database queries. The consequences of such an attack are great and can include unauthorized access, modifying, or deleting sensitive data.

The implemented system provides an intuitive and fast way to detect attacks. It also helps to classify important warnings from those that are negligible, thus speeding up the analysis of a large number of logs by administrators and allowing a more detailed investigation of attacks with higher priority.

6 Conclusion

In this paper we designed and implemented a visualization interface that graphically presents alerts generated by Snort IDS and shows the most common source addresses, classes and dates of attacks, as well as the most common alert priorities, thus allowing the users to easily detect possible traffic irregularities. The system has been tested in an appropriate environment in real-time. The results of attack detection and classification were given. It is shown that the Snort

IDS visualization interface makes detecting network irregularities quick and straightforward. Unlike a firewall that usually monitors external traffic to the network, the proposed system monitors traffic on any number of selected machines, so it can also detect attacks from the local network, which pose a great danger. The great advantage of the system is its light-weight architecture, which enables an immediate visual response.

7 Acknowledgment

This work was supported by the Serbian Ministry of Education, Science and Technological Development [grant number TR32012].

8 References

[1] H.- J. Liao, C.- H. R. Lin, Y.- C. Lin, K.- Y. Tung: Intrusion Detection System: A Comprehensive Review, Journal of Network and Computer Applications, Vol. 36, No. 1, January 2013, pp. 16 – 24.

[2] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman: Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges, Cybersecurity, Vol. 2, July 2019, pp. 20-1 – 20-22.

[3] J. M. Kizza: Guide to Computer Network Security, 4th Edition, Springer, Chattanooga, 2017.

[4] W. Stallings, L. Brown: Computer Security: Principles and Practice, 4th Edition, Pearson Education, Inc., Hoboken, 2018.

[5] G. Ahmed, M. N. A. Khan, M. S. Bashir: A Linux-Based IDPS Using Snort, Computer Fraud and Security, Vol. 2015, No. 8, August 2015, pp. 13 – 18.

[6] Z. Hassan, Shahzeb, R. Odarchenko, S. Gnatyuk, A. Zaman, M. Shah: Detection of Distributed Denial of Service Attacks Using Snort Rules in Cloud Computing & Remote Control Systems, Proceedings of the IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC), Kiev, Ukraine, October 2018, pp. 283 – 288.

[7] U. Aickelin, J. Twycross, T. Hesketh-Roberts: Rule Generalisation in Intrusion Detection Systems Using Snort, International Journal of Electronic Security and Digital Forensics, Vol. 1, No. 1, October 2007, pp. 10 – 16.

[8] N. Khamphakdee, N. Benjamas, S. Saiyod: Improving Intrusion Detection System based on Snort Rules for Network Probe Attacks Detection with Association Rules Technique of Data Mining, Journal of ICT Research and Applications, Vol. 8, No. 3, March 2015, pp. 234 – 250.

[9] T. Zhang, X. Wang, Z. Li, F. Guo, Y. Ma, W. Chen: A Survey of Network Anomaly Visualization, Science China Information Sciences, Vol. 60, December 2017, pp. 121101-1–121101-17.

[10] H. Shiravi, A. Shiravi, A. A. Ghorbani: A Survey of Visualization Systems for Network Security, IEEE Transactions on Visualization and Computer Graphics, Vol. 18, No. 8, August 2012, pp. 1313 – 1329.

[11] L. Hao, Ch. G. Healey, S. E. Hutchinson: Flexible Web Visualization for Alert-Based Network Security Analytics, Proceedings of the 10th Workshop on Visualization for Cyber Security (VizSec'13), Atlanta, USA, October 2013, pp. 33 – 40.

[12] S. Dasireddy, W. Gasior, X. Cui, L. Yang: Alerts Visualization and Clustering in Network-Based Intrusion Detection, Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '10), Oak Ridge, USA, April 2010, pp. 1 – 4.

[13] Y. Shi, Y. Zhao, F. Zhou, R. Shi, Y. Zhang, G. Wang: A Novel Radial Visualization of Intrusion Detection Alerts, IEEE Computer Graphics and Applications, Vol. 38, No. 6, November 2018, pp. 83 – 95.

[14] V. Sharma: Getting Started with Kibana, Beginning Elastic Stack, Apress, Berkeley, 2016.

[15] E. Salituro: Learn Grafana 7.0: A Beginner's Guide to Getting Well Versed in Analytics, Interactive Dashboards, and Monitoring, 1st Edition, Packt Publishing, Birmingham, 2020.

[16] S. Wexler, J. Shaffer, A. Cotgreave: The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios, 1st Edition, John Wiley & Sons, Inc., Hoboken, 2017.

[17] K. Scarfone, P. Mell: Guide to Intrusion Detection and Prevention Systems (IDPS), National Institute of Standards and Technology, Gaithersburg, 2007.

[18] S. Gamage, J. Samarabandu: Deep Learning Methods in Network Intrusion Detection: A Survey and an Objective Comparison, Journal of Network and Computer Applications, Vol. 169, November 2020, pp. 1 – 47.

[19] V. Ciric, D. Cvetkovic, N. Gavrilovic, N. Stojanovic, I. Milentijevic: Input Splits Design Techniques for Network Intrusion Detection on Hadoop Cluster, Facta Universitatis, Series: Electronics and Energetics, Vol. 34, No. 2, June 2021, pp. 239 – 257.

[20] N. Gavrilovic, V. Ciric, N. Lozo: Snort IDS System Visualization Interface, Proceedings of the 8th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN), Ethno Village Stanišići, Republic of Srpska, September 2021, pp. 513.