Snort IDS system visualization interface

Nadja Gavrilovic, Vladimir Ciric, Nikola Lozo
University of Nis, Faculty of Electronic Engineering, Nis, Serbia

***Abstract*—Over the past decades, the rapid Internet development and the growth in the number of its users have raised various security issues. Despite numerous available security tools, the exchange of data over the Internet is becoming increasingly insecure. For this reason, 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. The next step in successful network protection is the analysis 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 that graphically presents alerts generated by Snort IDS, classifies them according to the most important attack parameters, and allows the users to 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.**

***Index Terms*—IDS, snort, network intrusion detection, visualization interface**

I. 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 that 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].

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 a technique of detecting unauthorized access to a computer system or a computer network. An intrusion into a system can be defined as an attempt by an intruder to bypass the security mechanisms of a computer or network and illegally gain access. 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 the events occurring in a computer system or network, analyzing them and detecting unauthorized intrusions [1], [3].

IDSs are often classified based on the scope of monitoring and type of data analyzed, into host-based IDS (HIDS) and network-based IDS (NIDS). Host-based intrusion detection systems monitor the characteristics of a single host and the events occurring within it. They analyse process identifiers, the system calls they make and operating system specific logs (including system, event, and security logs on Windows systems and syslog in Unix environments), in order to detect evidence of suspicious activity. On the other hand, network-based intrusion detection systems have the whole network as the monitoring scope. They 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 various research in the field of network security. Recently, there have been numerous papers studying the implementation of Snort in different environments [5], [6]. Also, many 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 involves a detailed analysis of recorded 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 that graphically presents alerts generated by Snort IDS. The implemented system allows the users to visually analyze generated traffic logs. Furthermore, it shows the most common source addresses, classes and dates of attacks, as well as the most common alert priorities. Such traffic analysis makes detecting possible 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.

There are popular network visualization solutions [9], but

Nadja Gavrilovic is with the Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, Nis, Serbia (e-mail:nadja.gavrilovic@elfak.ni.ac.rs).

Vladimir Ciric is with the Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, Nis, Serbia (e-mail:vladimir.ciric@elfak.ni.ac.rs).

Nikola Lozo is with the Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, Nis, Serbia (e-mail:nikolalozo@elfak.rs).

all of these solutions are too robust and require the whole stack. Our motive was to make a light-weight solution for environments where there is no need for the whole stack.

The paper is organized as follows. Section 2 gives a brief introduction to intrusion detection systems in general, and the Snort IDS. Section 3 is the main section and presents the design and architecture of the proposed Snort IDS system visualization interface. In Section 4 the implementation results will be presented, while the concluding remarks are given in Section 5.

II. Background on IDS and Snort

Based on the technique used to assess the network packets as regular or malicious, IDSs can be classified into signature (or pattern) matching and anomaly-based IDSs. 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. Signature-based detection is very effective at detecting known threats but can experience problems with detecting new and previously unknown threats [1], [10].

Anomaly-based detection is the process of comparing network traffic and observed events against the definitions of the activity that is considered normal in order to identify 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 are capable of detecting zero-day attacks, but with a drawback of possible false positives [1], [3], [10]. 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 [11].

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 deployed so that the network traffic it is monitoring must pass through it. They are typically placed at the network border. A passive (tap) sensor is deployed so that it monitors a copy of the actual network traffic. They typically monitor network traffic from the key network locations [10]. 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.). Defining classifications for rules provides a way to better organize the event data Snort produces. Each rule also assigns priority to the alert, according to the alert class. A priority of 1 (high) is the most severe and 4 (very low) is the least severe [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 each captured packet to identify and isolate protocol headers at the data link, network, transport, and application 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 first rule that matches the decoded packet triggers the action specified by the rule. For each packet that matches a rule, the rule specifies what logging and alerting options are to be taken [4].

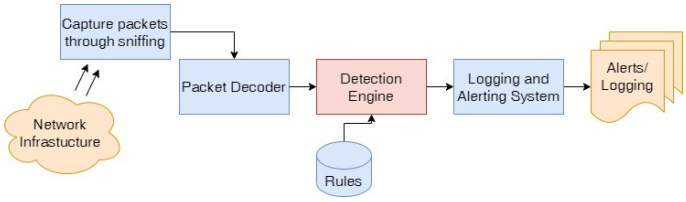

Fig. 1. Snort Architecture

III. 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 and easily detect deviations from normal traffic that may indicate an intrusion.

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.

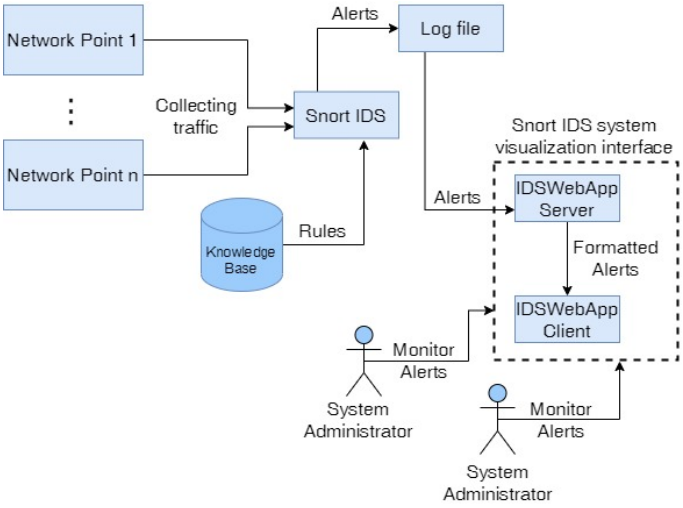

Fig. 2. System Architecture

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.

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.

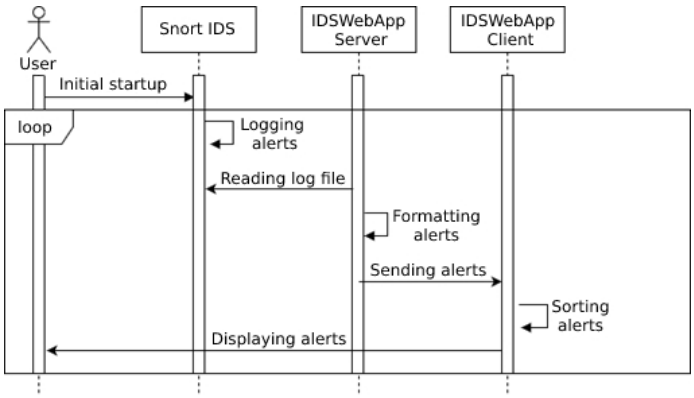

Fig. 3. Sequential diagram of the Snort IDS visualization interface operation

IV. 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. 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

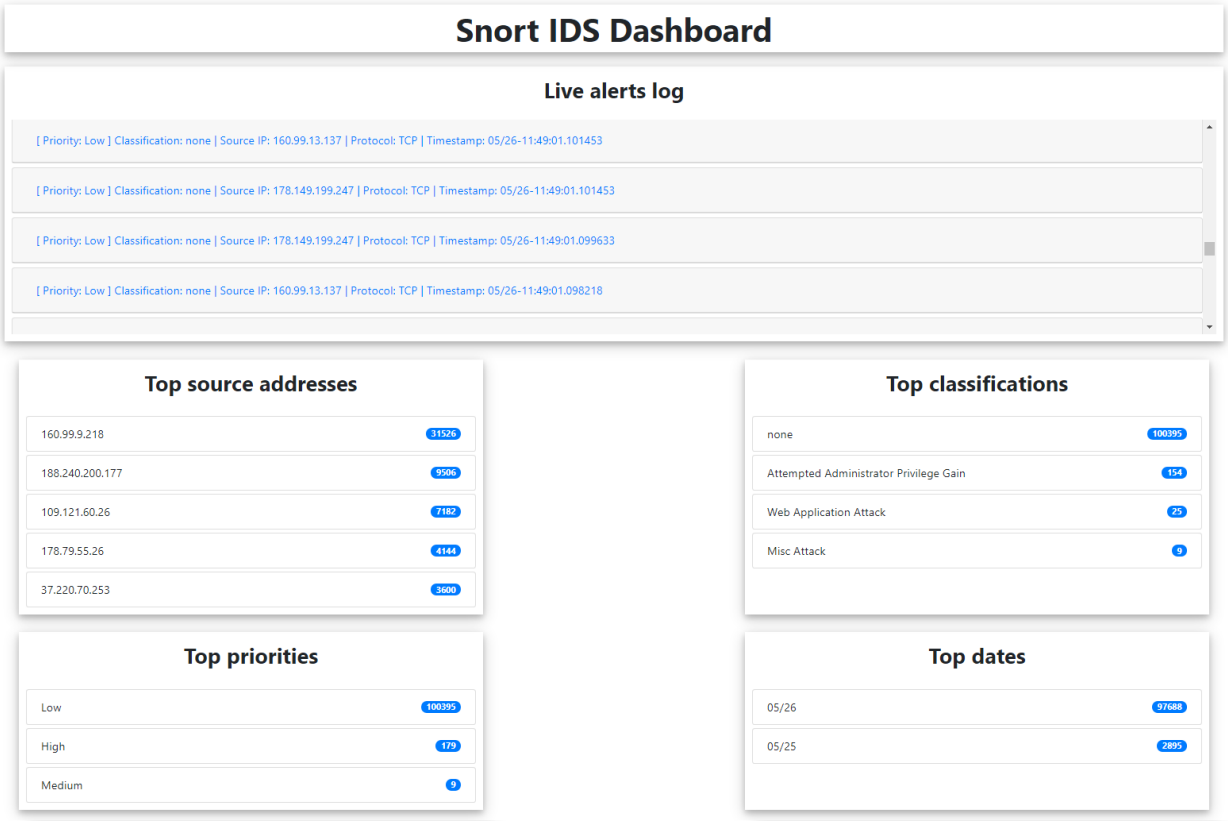

Fig. 4. Snort IDS visualization interface

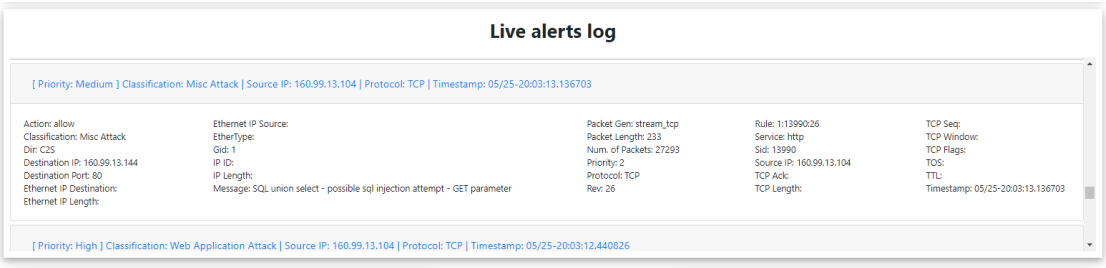

Fig. 5. Misc attack alert example

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.

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; con-

tent:"select",nocase; pcre:"\union\s+(all\s+)?select\s+/i"; metadata:policy max-detect-ips drop,policy security-ips drop; service:http; classtype:misc-attack; sid:13990; rev:26;)

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 a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. The consequences of such an attack are great and can include unauthorized access, modifying, or deleting sensitive data. In some situations, an attacker can escalate an SQL injection attack to compromise the underlying server or other back-end infrastructure or perform a denial-of-service attack. 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.

V. 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 possible 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 great danger. The great advantage of the system is that it could be implemented on any location on the network and it could collect logs from any IDS, which makes it a very efficient and portable solution.

Acknowledgment

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

References

[1] H. J. Liao, C. R. Lin, Y. C. Lin, K. Y. Tung, "Intrusion detection system: A comprehensive review", Journal of Network and Computer Applications, Vol. 36, Issue 1, pp. 16-24, 2013.

[2] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges", Cybersecurity, Vol. 2, 2019.

[3] J. M. Kizza, *Guide to Computer Network Security*, 4th. ed. Springer Publishing Company, Incorporated, 2017.

[4] W. Stallings, L. Brown, *Computer Security Principles and Practice*, Hoboken, New Jersey, Pearson Education, 2018.

[5] G. Ahmed, M.N.A. Khan, M. S. Bashir, "A Linux-based IDPS using Snort", Computer Fraud and Security, Vol. 2015, Issue 8, pp. 13-18, 2015.

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

[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, 2008.

[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, pp. 234-250, 2015.

[9] V. Sharma, Getting Started with Kibana, In Beginning Elastic Stack, Apress, Berkeley, CA, 2016.

[10] K. Scarfone, P. Mell, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, NIST Special Publication, 2007.

[11] S. Gamage, J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison", Journal of Network and Computer Applications, Volume 169, 2020.