Detecting Unusual Activities in Local Network Using Snort and Wireshark Tools

Naif Alsharabi 1,2,*, Maha Alqunun 1, and Belal Abdullah Hezam Murshed 2,3

1 Department of Computer Engineering, College of Computer Science and Engineering, University of Ha’il, Ha’il 55476, Saudi Arabia; Email: s20200324@uoh.edu.sa (M.A.)
2 Department of Computer Science, College of Engineering and IT, Amran University, Amran 00967, Yemen
3 Department of Studies in Computer Science, Mysore University, Mysore-570006, Karnataka, India;
Email: belal.a.hezam@gmail.com (B.A.H.M.)
*Correspondence: n.sharabi@uoh.edu.sa (N.A.)

Abstract—Many organizations worldwide encounter security risks on their local network caused by malware, which might result in losing sensitive data. Thus, network administrators should use efficient tools to observe the instantaneous network traffic and detect any suspicious activity. This project aims to detect incidents in local networks based on snort and Wireshark tools. Wireshark and snort tools combine their advantages to achieve maximum benefit, enhance the security level of local networks, and protect data. Snort Intrusion Detection System (Snort-IDS) is a security tool for network security. Snort-IDS rules use to match packet traffic. If some packets match the rules, Snort-IDS will generate alert messages. First, this project uses a virtual dataset that includes normal and abnormal traffic for the performance evaluation test. In addition, design local rules to detect anomalous activities. Second, use Wireshark software to analyze data packets. Second, use Wireshark software to analyze data packets. This project categorizes the detected patterns into two groups, anomaly-based detection, and signature-based detection. The results revealed the efficiency of the snort-IDS system in detecting unusual activities in both patterns and generating more information by analyzing it by Wireshark, such as source, destination, and protocol type. The promoted experience was tested on the virtual local network to ensure the effectiveness of this method. Keywords: network, intrusion detection system, Wireshark, snort, anomaly-based detection, signature-based detection, packet traffic, alert.

Keywords—network, intrusion detection system, wireshark, snort, anomaly-based, detection, signature-based detection, packet traffic, alert

I. INTRODUCTION

A. Local Network Developments

Local area networks are the interconnection of different computers and devices, which give the user freedom and flexibility to move around and ease and speed in using network resources [1]. Reliance on information systems and the computer network is increasing day by day, which a sensitive topic, which is the protection of data and systems, especially with the increase in cases of penetration and the development of the capabilities of attackers. At present, it is challenging to build a system capable of repelling all attacks, so hence the need to use intrusion detection systems of all kinds to protect networks from external and internal attacks [2, 3].

B. Intrusion Detection System (IDS)

An intrusion detection system is defined as the process of monitoring and analyzing events in a network or system to detect any indication of a system intrusion. These infiltrations and suspicious activities are usually from attackers or users from within the facility to gain higher privileges or abuse their powers [4, 5].

C. Types of Intrusion Detection Systems

1) Anomaly-based detection system

which is a system that works by defining the expected behavior of the system in statistical ways and then considering any deviation from this behavior as an indication of malicious activities.

2) Signature-based detection system

Relies on a set of predefined patterns of possible attacks on the system and a group of measures that must be applied when one of these patterns in the system is detected [5, 6].

D. Detection of Intrusion by A Network Monitoring Tool

Wireshark and Snort Software are used in this study for intrusion detection. Below are brief descriptions of each instrument. Wireshark is an open-source network packet analyzer that captures data packets passing over the network and logically displays them [7]. Snort is a modern security tool, and can be used as a packet sniffer, a packet logger, or a Network-based Intrusion Detection System, among other things. There are also a variety of Snort add-on packages that provide alternative means of recording and managing Snort log files, fetching, and maintaining current Snort rule sets, and alerting to notify your admin when potentially malicious traffic is detected [8].
II. LITERATURE REVIEW

Afzal and Murugesan [9] demonstrate that traffic analysis or a long time and is a useful technique for monitoring networks and identifying anomalies. A simulated SS7 attack traffic data set was used to deploy the Snort and Wireshark tools to demonstrate that the deployment of Snort is a workable solution to the network security issue. Snort IDS rules enable the detection of common attack types.

Evaluating the effectiveness of free rule sets for Snort network detection analyzed the attacks that occur frequently. Many of them have substantial ramifications that interfere with people’s and enterprises’ routine operations and may even result in lasting harm. Defensive tools are used to track down and stop attacks Snort is one of these tools. When Snort detects malicious data packets, it alerts the user and blocks the resulting attack. To determine what is harmful, Snort uses a list of attack signatures known as a rule set.

Explaining the implemented Snort-Based IDS is lightweight, scalable, and capable of monitoring and detecting suspicious behavior based on defined customized rules. Its key benefit is that it supports a wide range of programming languages for the generation of rules. It is also capable of determining whether a Denial-of-Service (DoS) attack is based on Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) [10].

Ghaﬁr et al. [11] have been focusing on the importance of network monitoring, as it is one of the main tasks of the network administrator. Due to the essential services carried out through the networks, their disruption decreases the company’s productivity.

In this study, the authors discussed a set of techniques and tools for monitoring the company (Wireshark, Tshark, Suricata, snort), in addition to analyzing the shortcomings of each tool. The study aims to provide the most appropriate solutions for monitoring networks depending on the company’s environment and needs.

Jain and Anubha [12] debate the importance of an intrusion detection system without interference from the network administrator. The goal of the intrusion detection system is to provide a safe data transmission environment. They discovered that the most well-known tool for sniffing and analyzing packets, Wireshark, is less effective at detecting intrusion. In addition to the Wireshark tool, the recommends using intrusion detection tools, where it works to create a file with data packets and an alert and export to the Wireshark. They demonstrated the significance of using intrusion detection tools in conjunction with Wireshark to provide a more secure network. The critical importance of administrators being aware of the traffic that passes through their networks is discussed in [13], to troubleshoot and solve problems more effectively. The paper examined three network monitoring tools (SNMP, RMON, and Cisco Netflow) as well as two new monitoring methods (WREN, SCNM) that use a combination of passive and active techniques. According to the study, institutions should maintain the network’s health so that it does not affect productivity by determining whether a more effective system or a newer system is required.

Through packet sniffer tools, Pallavi and Patel explained the importance of intrusion detection systems in maintaining the network and improving economic efficiency [14]. There are numerous tools, but their functionality is limited. They take up a lot of memory, only track IP packets, and some only capture network traffic without analyzing it. As a result, the researcher must employ several tools in order to achieve the required level of intrusion detection system.

Yuppala and Parik [15] describe numerous ways to combine an intrusion detection and prevention system to aid in the protection of an organization from threats and attacks. Sourcefire’s IPS is the best option for a company because it has the highest ratings for intrusion detection and prevention on the market. Sourcefire IPS got its start with Snort, an open-source intrusion detection and prevention program. Although the product is a closed source, it can receive Snort warnings. Sourcefire IPS provides intrusion detection, blocking, and Snort base editing. With all these benefits, the study concludes that Sourcefire is the best choice for a LINUX server. Snort IDS is put to the test with a variety of attacks [16]. Snort was put to the test with a variety of send and attack rates. When the transmission rate exceeded the rate at which Snort IDS could detect all packets transferred over the network, it was unable to detect them all. In the second experiment, fragment-sized assaults were sent, demonstrating that Snort could identify all fragmented packets. Snort failed to detect fragmentation packets in high-speed attacks in the third set of testing, which gathered fragmentation packets and transmission rates. It can be deduced that attackers may be able to assault Snort IDS using software such as Scapy and TCPReplay without being noticed by Snort.

Iqbal and Naaz [17] discuss the importance of Wireshark as a sniffer tool in a computer network, as well as how can detect a variety of LAN attacks such as ARP poisoning, DoS attacks, MAC flooding, and DNS spoofing, as well as mitigation measures for these attacks. It’s an excellent tool for monitoring network traffic. Wireshark, on the other hand, is unable to warn users of an impending attack. As a result, network analysts must consider the techniques that will enable Wireshark to forecast and develop data flows.

Intrusion occurs when someone tries to access a normal user and exploits the attack across the network. According to the study of Nadiaamnai and Hemalatha, Snort is a network packet that capture software program can be used to detect intrusion [18]. It preprocesses without the assistance of security specialists. With the use of internal rules, it also generates an alarm whenever any abnormal package is discovered. In this study, the strengths of both the abuse and anomaly algorithms are merged to create a more effective intrusion detection system. According to the findings, Snort + NETAD detects 133 out of 180 threats (73.88%). Various statistical methods could be used with snoring in the future to improve performance.
According to the study of Ghafir and Prenosil [19], network monitoring is a set of tools that allows network administrators to keep track of the current state and long-term trends of a complex computer network. This study discusses the current state of network monitoring. There are three approaches to traffic analysis. Flow monitoring, packet capture, and deep automated packet inspection are all possible. Each strategy has benefits and drawbacks. The purpose of this study is to give readers an overview of contemporary network monitoring technologies, including their architectures, features, and characteristics. It also provides a comparison of such strategies.

The intrusion detection system is the first and most reliable method in network security that relies on gathering data from a computer network, according to the study of Ibrahim et al. [20]. The demand for data traffic monitoring, auditing, and analysis solutions has increased. Packet sniffing refers to the methods used to capture data and convert it into a usable format. Three sniffer software, TCPDump, Wireshark, and Colasoft, were compared based on a variety of criteria, including detection ability, filtering, availability, supported OS systems, open source and graphical user interface, as well as their characteristics and features. This publication intends to provide new researchers with an introduction, basics, and comprehension of packet inhalation techniques.

III. METHODOLOGY

The project’s four-stage methodology is summarized in Fig. 1, which includes data collection, detection, analysis, testing, and rule creation. The experiment will be modulated in a virtual machine using the SNORT and WIRESHARK tools to distinguish and analyze a worm’s warmth alarm. Fig. 1 shows the mechanism models. The model is divided into several stages. Data collection, detection, analysis, testing, and rule creation are all part of the process.

A. Phases of the Methodology

1) Data collection phase

Determine the requirements for implementing the experiment using virtual data from the training site at this stage (traffic and malware analysis). This website provides malware files to test the detection and analysis capabilities of monitoring tools. This website contains various types of malwares, and a set of malicious software was chosen to suit the research’s purpose (Initials and Name n.d.).

2) Detection and analyzing phase

This experiment will be carried out in a virtual environment using the Virtual Box (VB) software. This software assists in the installation of virtual systems and the creation of a complete virtual network without jeopardizing the platform. To test the experience, install VB’s snort and Wireshark tools. Since snort is a highly efficient intrusion detection tool, and Wireshark provides an in-depth view of what is happening inside the network, this proposed method combines these two tools to achieve the required level of security. SNORT is an open-source intrusion detection tool that sniffs and captures all network packets and logs them in case of intrusion or malicious activity. The SNORT record is exported to WIRESHARK, which scans the packets. WIRESHARK displays all packet details such as source, destination, and protocol type [12].

B. Snort

Martin Roesh coded and designed Snort tool in 1998 as a free and open-source Intrusion Detection System (IDS) or Intrusion Prevention System (IPS). In 2009, Snort was named one of the best open-source software for information security. This tool works with all major operating systems (Windows, Linux, Unix, and Mac) and comes in three forms:

- Use it to eavesdrop on networks (Sniffer Mode), which causes the tool to read network packets and display them on the screen in front of you.
- It is used to save and store packets (Packet Logger) so that packets sent over the network can be saved to a file on the device (LogFile).
- The tool’s most common application is as an Intrusion Detection/Prevention System (IDS/IPS). Snort has a set of rules that define intrusion signatures by protocol type, IP address, and port number in order to detect malicious behavior and generate alerts. Because the tool is open source, users can write and configure their
own rules. The rules are divided into two sections: the header and the options.

Fig. 3 shows the structure of Snort rule which is divided into:

1. Header: It consists of the following fields
   - Action: This field specifies the action that the system will take when this rule is met.

2. Options: Rule options in Snort can be categorized into four basic types:
   - General Rule Option: gives us information about the rule but does not affect the matching process, like msg, sid, and class type.
   - Payload Detection Rule Options: Looks for parameters within the package’s content, such as content, offset, and distance.
   - Non-payload detection Rule options: To search for data not in the content, such as flow, ack, or TTL.
   - Post-Detection Rule Options: Specifies what should be fired by snort after a matching, such as a session, resp, reaction, or tag.

C. Wireshark

Wireshark is a network analysis tool that informs administrators about what is happening on a microscopic level with the network to provide you with the most accurate information about the internal network. Wireshark is the best program for analyzing internal network protocols.

Wireshark, formerly known as Ethereal, has an easy-to-use interface that can display data from hundreds of different protocols on all major network types. These data packets can be viewed in real time or analyzed offline, and dozens of capture/tracking file formats, including CAP and ERF, are supported. Many popular protocols, such as WEP and WPA/WPA2, can be viewed using integrated decoders. The Wireshark interface was shown in Fig. 4.

D. Testing Phase

Snort’s warnings are tested for accuracy at this stage by extracting suspicious software and examining it in two steps. The first step is Kali Linux.

Kali Linux is an operating system that includes tools for penetration testing. Kali Linux is open source and contains over 600 tools, each with a specific use and purpose that serves the hacker’s goals and allows him to test the effectiveness of security systems. The hash value of suspicious files is extracted by this tool during the final phase.

The second step is a virus Total. The hash value will be checked on the virus Total through the previous step. Virus Total is a Google service that allows you to scan any file or link for free. The “Virus Total” website will begin checking the hash’s value by sending it and scanning it through numerous antivirus programs [21].

E. Creating Rule Phase

The snort tool can be used to create custom rules for any type of traffic. This section will cover how to create rules, or rather the signatures that Snort uses to detect various types of network activity. Focusing on detecting unusual activities that do not conform to the snort rules.

IV. IMPLEMENTATION

The study intends to integrate network monitoring tools to achieve a high level of security. Any suspicious network activity is detected using the Wireshark and Snort tools. There are two methods for detecting suspicious activity in this study. The first is that SNORT, an open-source program, will be used to extract live data packets from the network. This utility will compare captured packets to predefined signatures and send alert messages to the user. To gain access to any type of malware, this procedure must be followed. Following the
start of the capture process, this utility saves each captured packet in a log file in its directory. This log file contains the data packets and alert messages.

The SNORT log file is then sent to Wireshark for analysis of the network packets collected. Wireshark generates all log file package information.

The second type is the detection of suspicious activities through the establishment of facility rules. Snort has the advantage of producing grammar that is comprehensive, adaptable, and not overly difficult. Snort and other websites have a wealth of excellent pre-written rule sets. An administrator can write their own rules or combine rules from other sets. Snort rules’ rule header and rule options sections are divided into two logical sections. The rule header contains the rule action, protocol, IP addresses, source and destination, network masks, and source and destination port information. The rule option section includes both alarm messages and information on which parts of the packet should be evaluated to determine whether the rule should be implemented.

A. Signature-Based Detection

This section presents two snort alert cases. In each case, Snort detects the presence of a malicious file on the network and issues an alert. After obtaining the snort’s initial information, such as the IP address and port number, then, using Wireshark, track packets and gather information about the infected device. After gathering the data, suspicious files will be examined by extracting the hash value and running them through a virus total service. This chapter will provide illustrations of how to carry out the steps that the network administrator takes to protect the infected device and prevent the spread of malicious software within the network.

1) Case 1

LAN segment data:
- LAN segment range: 10.8.21.0/24 (10.8.21.0 through 10.8.21.255)
- Domain: tec solutions.info
- Domain controller: 10.8.21.8 - Pizza-Bender-DC
- LAN segment gateway: 10.8.21.1
- LAN segment broadcast address: 10.8.21.255

2) Case 2

- LAN segment range: 10.0.0.0/24 (10.0.0.0 through 10.0.0.255)
- Domain: pascalpig.com
- Domain controller: 10.0.0.10 - Pascalpig-DC
- LAN segment gateway: 10.0.0.1
- LAN segment broadcast address: 10.0.0.255

3) Case 1: Snort alert on port 80

As shown in Fig. 5, the snort tool generates a suspicious trojan backdoor alert for the IP address 45.12.4.190 on port 80. Backdoor Trojans are malicious software programs that allow remote attackers to gain unauthorized computer access.

The log file is sent to Wireshark, which deeply analyzes the captured packet. When the file is opened, all the data collected is visible. The source and destination address of the packet, as well as the protocol and packet information, are displayed. Wireshark includes two filtering languages for packet capture and packet inspection. According to the Snort tool, the IP is 192.168.1.96 and the port is 80, which is an HTTP protocol port. So the packets will be searched using http.request and ip.addr == 45.12.4.190.

Fig. 6 shows malicious HTTP traffic at 45.12.4.190 port 80 - ncznw6a.com. GET/djuker/keyv/1?=rane11.cab. A set of information about the infected device was accessed by opening and examining the packet. The IP address is 10.8.21.163, the MAC address is 10:0:c:3:7:b0:a:0:2:85 (ASUSTekC 0a:0:0:0:0:0:0:0:2:85), the host name is DESKTOP-OFF4E8A, and the user account name is matthew.jones.
4) Case 2: Snort alert on port 587

As in the previous case, snort issued an alert about the presence of malware, as well as an alert via the SMTP port. SMTP port numbers as an old and widely used protocol, SMTP provides a wide range of port numbers for a variety of purposes and use cases, as shown in Fig. 7.

Wireshark was used to analyze traffic after the logs were exported. After examination a large amount of data about the Victim device was extracted. MAC address: 00:0c:6e:34:b2:d0 (ASUSTekC34:b2:d0), Host name: DESKTOP-M1JC4XX, and User account name: ronaldo.paccione. Fig. 8 depicts the packets are filtered and chosen. Malicious HTTP traffic can be found at 185.61.152.63 port 587 - mail.big3.icu - SMTP traffic with data stolen from the infected Windows host and 198.12.66.108 port 80 - GET /jojo.exe. The EXE acronym stands for executable, which means that the file can be run as a program through the computer’s operating system.

B. Test Alarms Comparison between Case 1 and Case 2

Previous alerts can be checked by computing the hash value of suspicious files in Cases 1 and 2 and comparing it to a virus database.

Fig. 9 illustrates the MD5 algorithm being used to compute the hash value of suspicious files. To generate a hash, the MD5 hashing algorithm employs a complex mathematical formula. It divides data into varying-sized blocks and manipulates it several times. Meanwhile, the algorithm adds a unique value to the computation and converts the result into a small signature or hash (Walia and Thapar, 2014).

The Virus Total service can be used to confirm that suspicious files contain malicious software after calculating their hash value. Virus Total is a web-based scanner that extracts signals from uploaded content using over 70 antivirus scanners, URL/blacklisting services, and other technologies. Virus Total is a search engine for files and URLs that accepts both files and URLs. Submissions are received via a public internet interface, a desktop uploader, a browser extension, or a programmatic API. After submitting content to Virus Total, basic results are delivered to the submitter and shared with examination partners who use the results in their own systems. Submissions directly benefit the Virus Total security community [22].
C. Anomaly-Based Detection

A network traffic anomaly detection system examines network traffic and compares it to a predefined baseline to determine what typical network bandwidth, protocols, ports, and other devices are. Machine learning is frequently used in this type to create a baseline and supporting security policy. Suspicious behavior and policy violations are then reported to IT staff. By using a broad model to detect threats rather than specific signatures and traits.

In this section, a new rule will be written to detect any network-based attempts to connect to the ftp protocol. Because FTP does not support data encryption, data transmissions are vulnerable to a variety of issues. Users use proxy FTP to establish a direct connection between two servers when the network is slow. A hacker can gain access to ports and data by impersonating a middleman and using the PORT command. FTP is also vulnerable to brute force attacks, which can be used to crack long-used passwords. Using packet capture techniques, hackers can capture and decode sent data packets because the ftp protocol sends packets as TCP packets through port number 21, the special rule will be as follows: $HOME\text{ NET} \ 21$ (msg:"FTP connection attempt”; sid:1000002; rev: 1). Explain the rule: Alert: This specifies the action taken by Snort, which is to display communication alerts. Other measures include log: Log connection data.

- Pass: Ignore the connection.
- Reject: Reject the connection with the log data and send the rejection message to the sender.
- $\text{sdrop}$: Refused to connect without logging in.

Protocol TCP UDP or ICMP, any IP address of the source – The port of the source.

The packet direction, $\text{SHOME} \ _\text{-home networks}$: The message that appears when the alert appears.

As shown in Figs. 10 and 11, if a user uploads a malicious file, Virus Total notifies them and displays the detection label for each engine. Some engines will reveal additional information, such as whether a given URL is part of a botnet or which brand a phishing site is targeting, and so on. Virus Total detects malware using the most recent signature sets.

Table I below summarizes the most critical information gleaned from analyzing snort alerts with the packet tracer and Wireshark. The table summarizes the essential information about device addresses and port numbers, as well as the virus Total result.

---

**Table I. Summaries Result of Analysis**

<table>
<thead>
<tr>
<th>Case No</th>
<th>IP address</th>
<th>Port no</th>
<th>MAC Address</th>
<th>Host name</th>
<th>Suspicious file</th>
<th>Hash value of suspicious file(DMS)</th>
<th>VirusTotal result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert1</td>
<td>10.8.21.1</td>
<td>63</td>
<td>10:c3:7b:0a:f2:85(ASU ST ekC_0a2:85)</td>
<td>DESKTOP-QF4FE8A</td>
<td>Keyvl.php%3fiber aneel.cab</td>
<td>A52ae151b4b993e00f87e8f87b3780d731</td>
<td>Malware</td>
</tr>
<tr>
<td>Alert2</td>
<td>10.18.20.97</td>
<td>587</td>
<td>00:06:34:52:00(ASU ST ekC_34:52:00)</td>
<td>DESKTOP-M1JC4XX</td>
<td>Jojo.exe</td>
<td>Lad6f564701054b692b5fe6324a2</td>
<td>Malware</td>
</tr>
</tbody>
</table>

---

sid:100001: means the base number of the gadget since numbers less than “million” are already reserved by the tool.

Rev:1: Used to document the rule and to facilitate its processing and tracking in the future [23].

Fig. 12 illustrates the experiment’s rule written in the local rule. To run the rule, open the tool’s rules file, which is located at /etc/snort/rules/local.rules. The command sudo gedit /etc/snort/rules/local.rules opens it in the gedit text editor. Use this tool as an Intrusion Detection System now (IDS). To accomplish this, use the command line to run sudo snort -A console -I eth1 -u snort -g snort -c /etc/snort/snort.conf. After launching Snort as an intrusion detection system, test the rule by attempting to connect to port 21 from another device on the network (ftp). Fig. 13 illustrates the development of a snort tool that detects the presence of an ftp connection attempt.

---

Figure 11. Virus total result (Case 2).

Figure 12. Create new rule in snort.
Snort stores log files for any warnings or messages in the following directory: /var/log/snort/. As a result, these files can be checked and tracked professionally by launching the Wireshark network analysis program, then selecting File, then Open, and then returning to the previous path and selecting any desired file. In packet capture, the specifics of each packet as they were transmitted across the LAN will be displayed. Fig. 14 depicts a screenshot of a packet capture window. The top panel of the window displays the source and destination nodes for each packet, as well as the protocol and metadata for each packet.

![Figure 13. FTP connection alert.](image1)

![Figure 14. Snort logs tracking.](image2)

V. DISCUSSION

Combining the benefits of the Wireshark and Snort tools in this study resulted in maximum benefit and an increase in the security level of local networks. The results demonstrated the system’s ability to detect and alert the administrator in the event of any unusual network activity, whether it was a signature or an anomaly, as well as the ability to examine the captured data packets to provide accurate details. These findings assist the administrator in quickly responding to and mitigating damage, thereby achieving the research’s main goals and increasing the level of security in local networks.

VI. CONCLUSION

Network security is the most pressing concern for all organizations and businesses in this digital era. Network monitoring has long been an important part of thwarting attacks. The research questions, such as how to make networks more secure and the tools that must be used, were answered through this project. A local network can be made more secure by employing an intrusion detection system (IDS). Thanks to the Wireshark and Snort tools, Intrusion Detection Systems (IDS) play an important role in network monitoring. Using these technologies, intrusive digital data on network processes can be graphically tracked. The use of an IDS network management system improves efficiency and security. In this research, look at the Snort Network Intrusion Detection System properties, which are used for intrusion detection using predefined rules, creating local rules, and informing the user via alert messages. SNORT generates a log file with data packets and alert messages. Wireshark is used to examine the captured data packets that were exported from the recorded log file. All of the packet details from the log file are generated by this Wireshark utility. It goes into great detail about frames, internet protocols, sources, and destinations. Furthermore, to avoid security breaches, prior protection rules must be followed. These security precautions have previously been discussed in Alsharabi and Alshammeri et al.’s scientific paper [24].

VII. FUTURE WORK

Wireshark has previously proven its value as a Network Protocol Analyzer in all necessary areas. There is, however, room for improvement in terms of alarm generation and heuristic development can be working on incorporating specific tools into the Wireshark source code to address flaws by making Wireshark alert capable. This research suggests using the Snort tool as an Intrusion Prevention System (IPS) as well as an intrusion detection system.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, N. A., and M.A.; methodology, M. A. and N. A.; software, M.A.; validation, N.A., and B.M.; formal analysis, M. M., N. A. and B.M.; investigation, M. A. and N. A.; resources, M. A; data curation, M. A.; writing—original draft preparation, M. A., and N. A.; writing—review and editing, N. A. and B.M.; visualization, B.M.; Supervision, N. A.; project administration, N. A.; funding acquisition, N. A., and M. A.; All authors have read and agreed to the published version of the manuscript.
FUNDING

This research has been funded by Scientific Research Deanship at University of Ha’il, Saudi Arabia through project number GR-22 017.

REFERENCES


Copyright © 2023 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.