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].

Manuscript received October 27, 2022; revised December 9, 2022; accepted March 31, 2023; published July 5, 2023.

doi: 10.12720/jait.14.4.616-624

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, Tcpdump, 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.

Vuppala and Farik [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 sniffing 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 Nadiammai 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 warm 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.

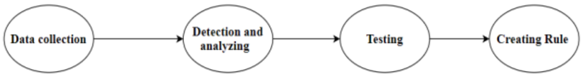


Figure 1. Phases of the methodology.

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 WIRSHARK, which scans the packets. WIRSHARK displays all packet details such as source, destination, and protocol type [12].

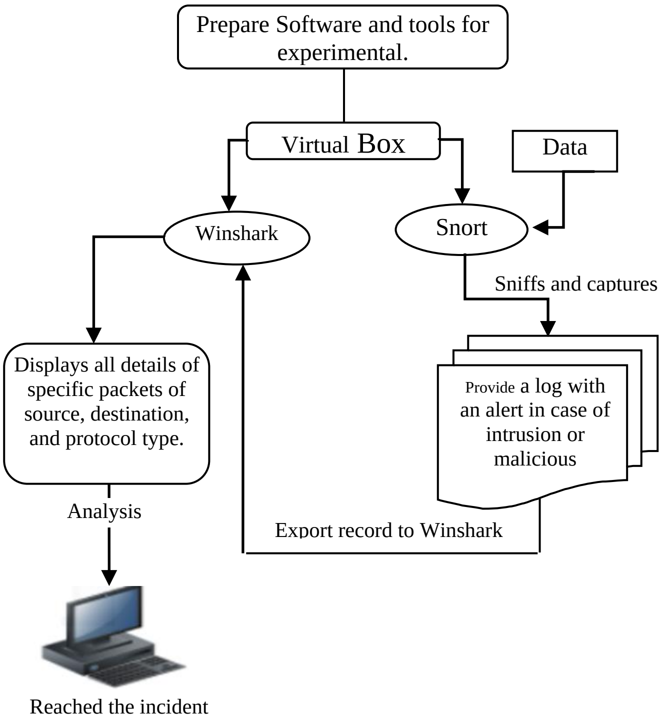

Prepare Software and tools for experimental.
Virtual Box
Data
Winshark
Snort
Sniffs and captures
Displays all details of specific packets of source, destination, and protocol type.
Provide a log with an alert in case of intrusion or malicious
Analysis
Export record to Winshark
Reached the incident

Figure 2. Detection and analyzing phase.

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.

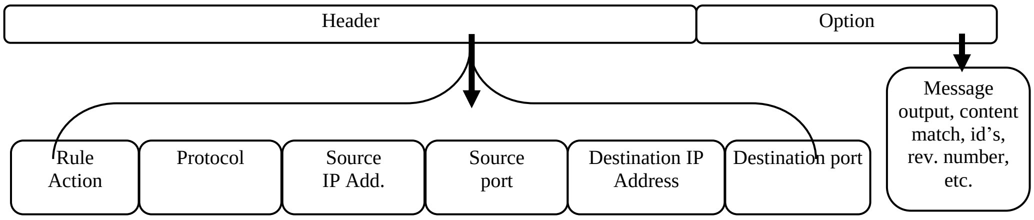


Figure 3. Snort rule structure.

- Protocol: used to apply the rule to a specific protocol (UDP, TCP, IP.)
- IP destination and Source: Specifies the route for the packets to which the rule should apply.
- Source and destination ports: these two fields are used for TCP, UDP protocols.
- Flow: used to express the direction of the package to which the rule will be applied.

2. Options: Rule options in Snort can be categorized into four basic types:

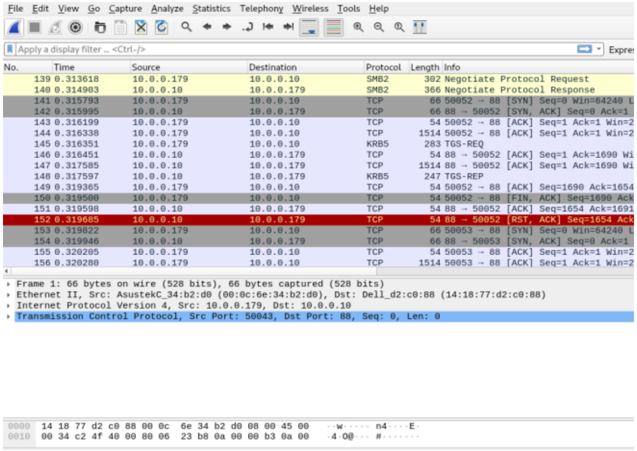
- General Rule Option: gives us information about the rule but does not affect the matching process, like msg, sid, and class type.
- Payload Detection Rule Options: Looks for parameters within the package's content, such as content, offset, and distance.

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.



```
No.     Time        Source        Destination   Protocol Length Info
    139 0.313618    10.0.0.179    10.0.0.10     SMB2      302 Negotiate Protocol Request
    140 0.314903    10.0.0.10     10.0.0.179    SMB2      366 Negotiate Protocol Response
    141 0.315793    10.0.0.179    10.0.0.10     TCP        66 50052 → 88 [SYN] Seq=0 Win=64240 L
    142 0.315995    10.0.0.10     10.0.0.179    TCP        66 88 → 50052 [SYN, ACK] Seq=0 Ack=1
    143 0.316199    10.0.0.179    10.0.0.10     TCP        54 50052 → 88 [ACK] Seq=1 Ack=1 Win=2
    144 0.316338    10.0.0.179    10.0.0.10     TCP      1514 50052 → 88 [ACK] Seq=1 Ack=1 Win=2
    145 0.316351    10.0.0.179    10.0.0.10     KRB5      283 TGS-REQ
    146 0.316451    10.0.0.10     10.0.0.179    TCP        54 88 → 50052 [ACK] Seq=1 Ack=1690 Wi
    147 0.317585    10.0.0.10     10.0.0.179    TCP      1514 88 → 50052 [ACK] Seq=1 Ack=1690 Wi
    148 0.317597    10.0.0.10     10.0.0.179    KRB5      247 TGS-REP
    149 0.319365    10.0.0.179    10.0.0.10     TCP        54 50052 → 88 [ACK] Seq=1690 Ack=1654
    150 0.319500    10.0.0.179    10.0.0.10     TCP        54 50052 → 88 [FIN, ACK] Seq=1690 Ack
    151 0.319598    10.0.0.10     10.0.0.179    TCP        54 88 → 50052 [ACK] Seq=1654 Ack=1691
    152 0.319685    10.0.0.10     10.0.0.179    TCP        54 88 → 50052 [RST, ACK] Seq=1654 Ack
    153 0.319822    10.0.0.179    10.0.0.10     TCP        66 50053 → 88 [SYN] Seq=0 Win=64240 L
    154 0.319946    10.0.0.10     10.0.0.179    TCP        66 88 → 50053 [SYN, ACK] Seq=0 Ack=1
    155 0.320205    10.0.0.179    10.0.0.10     TCP        54 50053 → 88 [ACK] Seq=1 Ack=1 Win=2
    156 0.320280    10.0.0.179    10.0.0.10     TCP      1514 50053 → 88 [ACK] Seq=1 Ack=1 Win=2

▸ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: AsustekC_34:b2:d0 (00:0c:6e:34:b2:d0), Dst: Dell_d2:c0:88 (14:18:77:d2:c0:88)
▸ Internet Protocol Version 4, Src: 10.0.0.179, Dst: 10.0.0.10
▸ Transmission Control Protocol, Src Port: 50043, Dst Port: 88, Seq: 0, Len: 0

0000  14 18 77 d2 c0 88 00 0c  6e 34 b2 d0 08 00 45 00   ··w····· n4····E·
0010  00 34 c2 4f 40 00 80 06  23 b8 0a 00 00 b3 0a 00   ·4·O@··· #·······
```

Figure 4. Wireshark interface.

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:

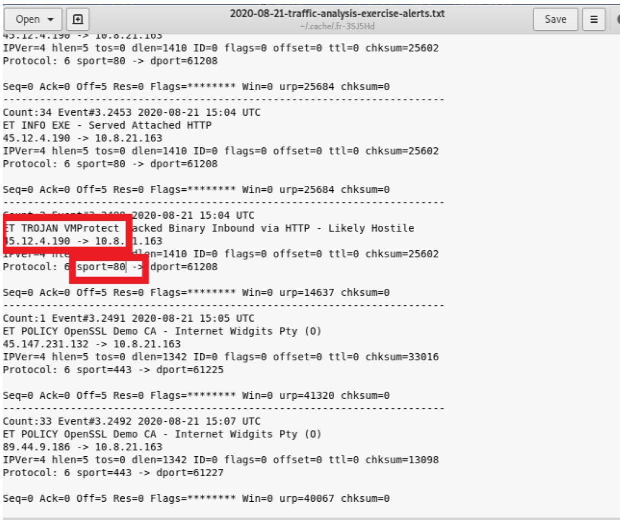
- LAN segment range: 10.8.21.0/24 (10.8.21.0 through 10.8.21.255)
- Domain: tecsolutions.info
- Domain controller: 10.8.21.8 - Pizza-Bender-DC
- LAN segment gateway: 10.8.21.1
- LAN segment broadcast address: 10.8.21.255

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: 10.00.10 - 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

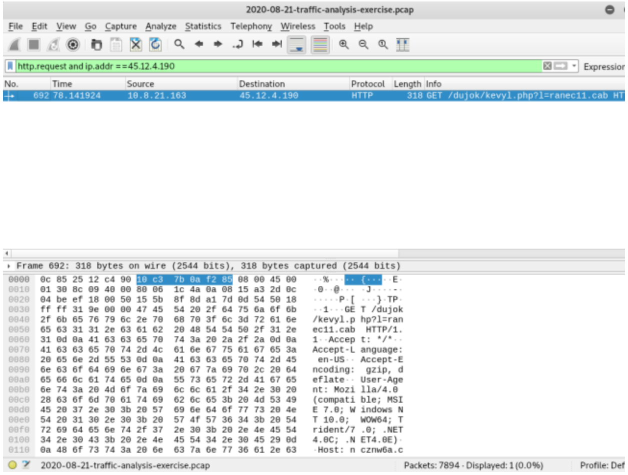
As shown in Fig. 5, the snort tool generates a suspicious trojan backdoor alert for the IP address 45.12.4.190 on port 80. Backdoor Trojans are malicious software programs that allow remote attackers to gain unauthorized computer access.



```
45.12.4.190 -> 10.8.21.163
IPVer=4 hlen=5 tos=0 dlen=1410 ID=0 flags=0 offset=0 ttl=0 chksum=25602
Protocol: 6 sport=80 -> dport=61208

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=25684 chksum=0
---------------------------------------------------------------------
Count:34 Event#3.2453 2020-08-21 15:04 UTC
ET INFO EXE - Served Attached HTTP
45.12.4.190 -> 10.8.21.163
IPVer=4 hlen=5 tos=0 dlen=1410 ID=0 flags=0 offset=0 ttl=0 chksum=25602
Protocol: 6 sport=80 -> dport=61208

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=25684 chksum=0
---------------------------------------------------------------------
[illegible] 2020-08-21 15:04 UTC
ET TROJAN VMProtect Packed Binary Inbound via HTTP - Likely Hostile
45.12.4.190 -> 10.8.21.163
IPVer=4 hlen=5 tos=0 dlen=1410 ID=0 flags=0 offset=0 ttl=0 chksum=25602
Protocol: 6 sport=80 -> dport=61208

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=14637 chksum=0
---------------------------------------------------------------------
Count:1 Event#3.2491 2020-08-21 15:05 UTC
ET POLICY OpenSSL Demo CA - Internet Widgits Pty (O)
45.147.231.132 -> 10.8.21.163
IPVer=4 hlen=5 tos=0 dlen=1342 ID=0 flags=0 offset=0 ttl=0 chksum=33016
Protocol: 6 sport=443 -> dport=61225

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=41320 chksum=0
---------------------------------------------------------------------
Count:33 Event#3.2492 2020-08-21 15:07 UTC
ET POLICY OpenSSL Demo CA - Internet Widgits Pty (O)
89.44.9.186 -> 10.8.21.163
IPVer=4 hlen=5 tos=0 dlen=1342 ID=0 flags=0 offset=0 ttl=0 chksum=13098
Protocol: 6 sport=443 -> dport=61227

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=40067 chksum=0
```

Figure 5. Snort alert Case 1.

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 == for 45.12.4.190.


2020-08-21-traffic-analysis-exercise.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression

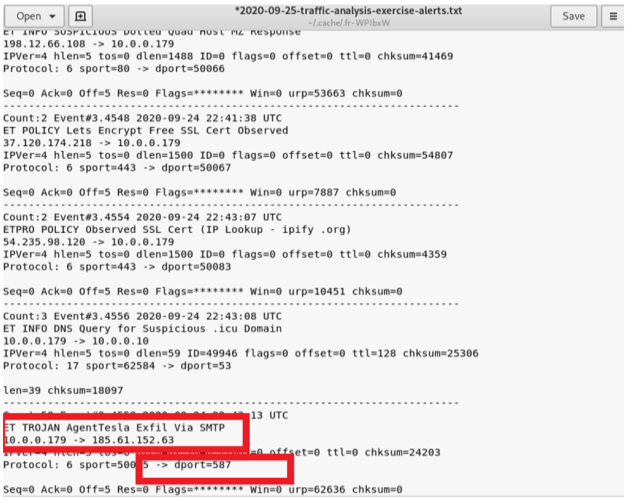
```
http.request and ip.addr ==45.12.4.190

No.   Time       Source       Destination  Protocol Length Info
  692 78.141924  10.8.21.163  45.12.4.190  HTTP        318 GET /dujok/kevyl.php?l=ranec11.cab HT
```

```
Frame 692: 318 bytes on wire (2544 bits), 318 bytes captured (2544 bits)
0000  0c 85 25 12 c4 90 10 c3  7b 0a f2 85 08 00 45 00   ··%···· {·····E·
0010  01 30 8c 09 40 00 80 06  1c 4a 0a 08 15 a3 2d 0c   ·0··@··· ·J····-·
0020  04 be ef 18 00 50 15 5b  8f 8d a1 7d 0d 54 50 18   ·····P·[ ···}·TP·
0030  ff ff 31 9e 00 00 47 45  54 20 2f 64 75 6a 6f 6b   ··1···GE T /dujok
0040  2f 6b 65 76 79 6c 2e 70  68 70 3f 6c 3d 72 61 6e   /kevyl.p hp?l=ran
0050  65 63 31 31 2e 63 61 62  20 48 54 54 50 2f 31 2e   ec11.cab  HTTP/1.
0060  31 0d 0a 41 63 63 65 70  74 3a 20 2a 2f 2a 0d 0a   1··Accep t: */*··
0070  41 63 63 65 70 74 2d 4c  61 6e 67 75 61 67 65 3a   Accept-L anguage:
0080  20 65 6e 2d 55 53 0d 0a  41 63 63 65 70 74 2d 45    en-US·· Accept-E
0090  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 20 64   ncoding:  gzip, d
00a0  65 66 6c 61 74 65 0d 0a  55 73 65 72 2d 41 67 65   eflate·· User-Age
00b0  6e 74 3a 20 4d 6f 7a 69  6c 6c 61 2f 34 2e 30 20   nt: Mozi lla/4.0
00c0  28 63 6f 6d 70 61 74 69  62 6c 65 3b 20 4d 53 49   (compati ble; MSI
00d0  45 20 37 2e 30 3b 20 57  69 6e 64 6f 77 73 20 4e   E 7.0; W indows N
00e0  54 20 31 30 2e 30 3b 20  57 4f 57 36 34 3b 20 54   T 10.0;  WOW64; T
00f0  72 69 64 65 6e 74 2f 37  2e 30 3b 20 2e 4e 45 54   rident/7 .0; .NET
0100  34 2e 30 43 3b 20 2e 4e  45 54 34 2e 30 45 29 0d   4.0C; .N ET4.0E)·
0110  0a 48 6f 73 74 3a 20 6e  63 7a 6e 77 36 61 2e 63   ·Host: n cznw6a.c

2020-08-21-traffic-analysis-exercise.pcap          Packets: 7894 · Displayed: 1 (0.0%)          Profile: Def
```

Figure 6. Filtering traffics based on port and IP (Case 1).

Fig. 6 shows malicious HTTP traffic at 45.12.4.190 port 80 - ncznw6a.com. GET/dujok/kevyl.php?l=ranec11.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:c3:7b:0a:f2:85 (ASUSTekC 0a:f2:85), the host name is DESKTOP-OF4FE8A, and the user account name is matthew.jones.

4) *Case 2: Snort alert on port 587*



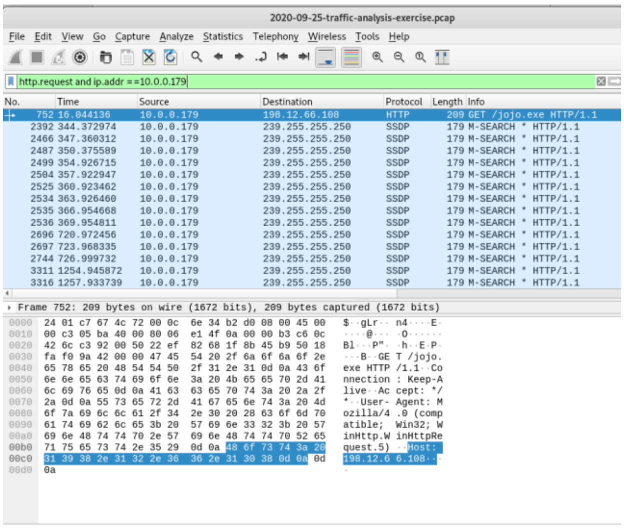
Open ▾ | *2020-09-25-traffic-analysis-exercise-alerts.txt | ~/.cache/.fr-WPIbxW | Save | ≡

```
ET INFO Suspicious Dotted Quad Host MZ Response
198.12.66.108 -> 10.0.0.179
IPVer=4 hlen=5 tos=0 dlen=1488 ID=0 flags=0 offset=0 ttl=0 chksum=41469
Protocol: 6 sport=80 -> dport=50066

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=53663 chksum=0
-----------------------------------------------------------------------
Count:2 Event#3.4548 2020-09-24 22:41:38 UTC
ET POLICY Lets Encrypt Free SSL Cert Observed
37.120.174.218 -> 10.0.0.179
IPVer=4 hlen=5 tos=0 dlen=1500 ID=0 flags=0 offset=0 ttl=0 chksum=54807
Protocol: 6 sport=443 -> dport=50067

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=7887 chksum=0
-----------------------------------------------------------------------
Count:2 Event#3.4554 2020-09-24 22:43:07 UTC
ETPRO POLICY Observed SSL Cert (IP Lookup - ipify .org)
54.235.98.120 -> 10.0.0.179
IPVer=4 hlen=5 tos=0 dlen=1500 ID=0 flags=0 offset=0 ttl=0 chksum=4359
Protocol: 6 sport=443 -> dport=50083

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=10451 chksum=0
-----------------------------------------------------------------------
Count:3 Event#3.4556 2020-09-24 22:43:08 UTC
ET INFO DNS Query for Suspicious .icu Domain
10.0.0.179 -> 10.0.0.10
IPVer=4 hlen=5 tos=0 dlen=59 ID=49946 flags=0 offset=0 ttl=128 chksum=25306
Protocol: 17 sport=62584 -> dport=53

len=39 chksum=18097
-----------------------------------------------------------------------
                                          13 UTC
ET TROJAN AgentTesla Exfil Via SMTP
10.0.0.179 -> 185.61.152.63
                                   =0 offset=0 ttl=0 chksum=24203
Protocol: 6 sport=500 5 -> dport=587

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=62636 chksum=0
```

Figure 7. Snort alert Case 2.

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 (ASUSTekC 34: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.



2020-09-25-traffic-analysis-exercise.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

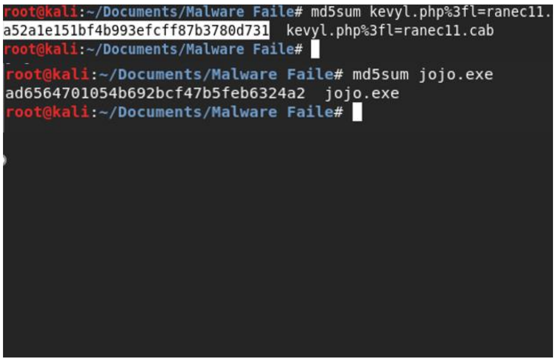
http.request and ip.addr ==10.0.0.179

```
No.   Time          Source        Destination       Protocol  Length Info
  752 16.044136     10.0.0.179    198.12.66.108     HTTP      209 GET /jojo.exe HTTP/1.1
 2392 344.372974    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2466 347.360312    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2487 350.375589    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2499 354.926715    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2504 357.922947    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2525 360.923462    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2534 363.926460    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2535 366.954668    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2536 369.954811    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2696 720.972456    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2697 723.968335    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 2744 726.999732    10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 3311 1254.945872   10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1
 3316 1257.933739   10.0.0.179    239.255.255.250   SSDP      179 M-SEARCH * HTTP/1.1

› Frame 752: 209 bytes on wire (1672 bits), 209 bytes captured (1672 bits)

0000  24 01 c7 67 4c 72 00 0c  6e 34 b2 d0 08 00 45 00   $··gLr·· n4····E·
0010  00 c3 05 ba 40 00 80 06  e1 4f 0a 00 00 b3 c6 0c   ····@··· ·O······
0020  42 6c c3 92 00 50 22 ef  82 68 1f 8b 45 b9 50 18   Bl···P"· ·h··E·P·
0030  fa f0 9a 42 00 00 47 45  54 20 2f 6a 6f 6a 6f 2e   ···B··GE T /jojo.
0040  65 78 65 20 48 54 54 50  2f 31 2e 31 0d 0a 43 6f   exe HTTP /1.1··Co
0050  6e 6e 65 63 74 69 6f 6e  3a 20 4b 65 65 70 2d 41   nnection : Keep-A
0060  6c 69 76 65 0d 0a 41 63  63 65 70 74 3a 20 2a 2f   live··Ac cept: */
0070  2a 0d 0a 55 73 65 72 2d  41 67 65 6e 74 3a 20 4d   *··User- Agent: M
0080  6f 7a 69 6c 6c 61 2f 34  2e 30 20 28 63 6f 6d 70   ozilla/4 .0 (comp
0090  61 74 69 62 6c 65 3b 20  57 69 6e 33 32 3b 20 57   atible;  Win32; W
00a0  69 6e 48 74 74 70 2e 57  69 6e 48 74 74 70 52 65   inHttp.W inHttpRe
00b0  71 75 65 73 74 2e 35 29  0d 0a 48 6f 73 74 3a 20   quest.5) ··Host: 
00c0  31 39 38 2e 31 32 2e 36  36 2e 31 30 38 0d 0a 0d   198.12.6 6.108···
00d0  0a                                                 ·
```

Figure 8. Filtering traffics based on port and IP (Case 2).

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.

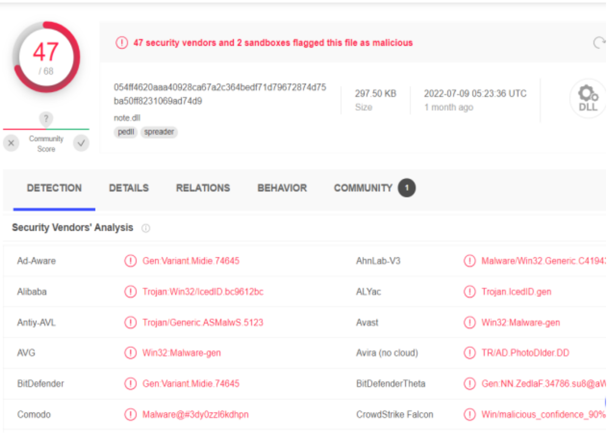


```
root@kali:~/Documents/Malware Faile# md5sum kevyl.php%3fl=ranec11.
a52a1e151bf4b993efcff87b3780d731  kevyl.php%3fl=ranec11.cab
root@kali:~/Documents/Malware Faile# 
root@kali:~/Documents/Malware Faile# md5sum jojo.exe
ad6564701054b692bcf47b5feb6324a2  jojo.exe
root@kali:~/Documents/Malware Faile# 
```

Figure 9. MD5 logarithm.

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].



47 security vendors and 2 sandboxes flagged this file as malicious

054ff4620aaa40928ca67a2c364bedf71d79672874d75
ba50ff8231069ad74d9
note.dll
pedll spreader

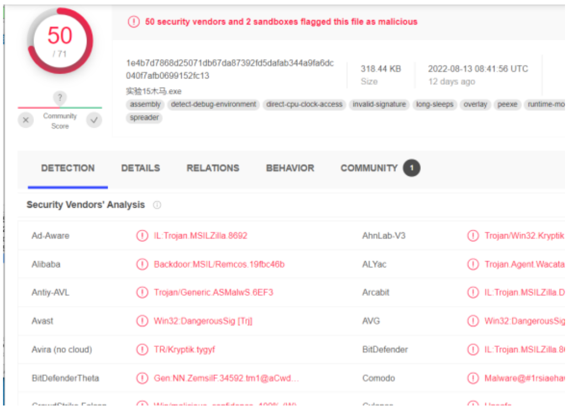
297.50 KB
Size

2022-07-09 05:23:36 UTC
1 month ago

Security Vendors' Analysis

Ad-Aware	Gen:Variant.Midie.74645	AhnLab-V3	Malware/Win32.Generic.C4194
Alibaba	Trojan:Win32/IcedID.bc9612bc	ALYac	Trojan.IcedID.gen
Antiy-AVL	Trojan/Generic.ASMalwS.5123	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Avira (no cloud)	TR/AD.PhotoDlder.DD
BitDefender	Gen:Variant.Midie.74645	BitDefenderTheta	Gen:NN.ZedlaF.34786.su8@aW
Comodo	Malware@#3dy0zzl6kdhpn	CrowdStrike Falcon	Win/malicious_confidence_90%

Figure 10. Virus total result (case 1).



50 security vendors and 2 sandboxes flagged this file as malicious

1e4b7d7868d25071db67da87392fd5dafab344a9fa6dc040f7afb0699152fc13
实验15木马.exe

318.44 KB
Size

2022-08-13 08:41:56 UTC
12 days ago

assembly detect-debug-environment direct-cpu-clock-access invalid-signature long-sleeps overlay peexe runtime-mo spreader

Security Vendors' Analysis

Ad-Aware	IL:Trojan.MSILZilla.8692	AhnLab-V3	Trojan/Win32.Kryptik
Alibaba	Backdoor:MSIL/Remcos.19fbc46b	ALYac	Trojan.Agent.Wacata
Antiy-AVL	Trojan/Generic.ASMalwS.6EF3	Arcabit	IL:Trojan.MSILZilla.D
Avast	Win32:DangerousSig [Trj]	AVG	Win32:DangerousSig
Avira (no cloud)	TR/Kryptik.tygyf	BitDefender	IL:Trojan.MSILZilla.8
BitDefenderTheta	Gen:NN.ZemsilF.34592.tm1@aCwd...	Comodo	Malware@#1rsiaeha

Figure 11. Virus total result (Case 2).

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 tracker 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

Case No	IP address	Port no	MAC Address	Host name	Suspicious file	Hash value of suspicious file(DM5)	VirusTotal result
Alret1	10.8.21.163	80	10:c3:7b:0a:f2:85(ASUST ekC_0af2:85)	DESKTOP-QF4FE8A	Kevyl.php%3fl=ranecll.cab	A52ale151bf4b993efcff87cff87b3780d731	Malware
Alret2	10.18.20.97	587	00:0c:6e:34:b2:d0(ASUST ekC_34:b2:d0)	DESKTOP-M1JC4XX	Jojo.exe	Lad6564701054b692bcf47b5feb6324a2	Malware

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 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.
- sdrop: Refused to connect without logging in.

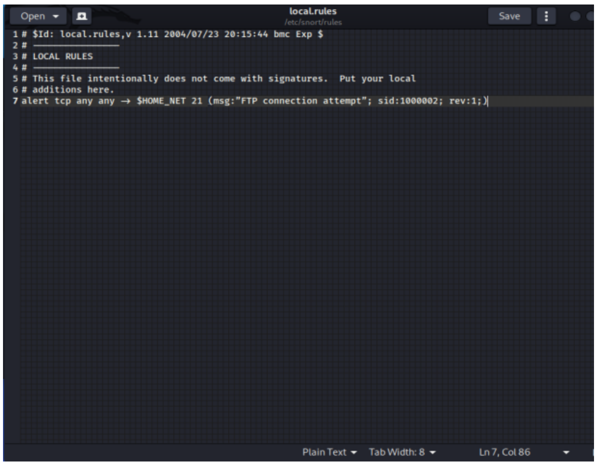
Protocol :(TCP UDP or ICMP), any – IP address of the source – The port of the source.

The packet direction, $HOME_NET -home networks: The message that appears when the alert appears.

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

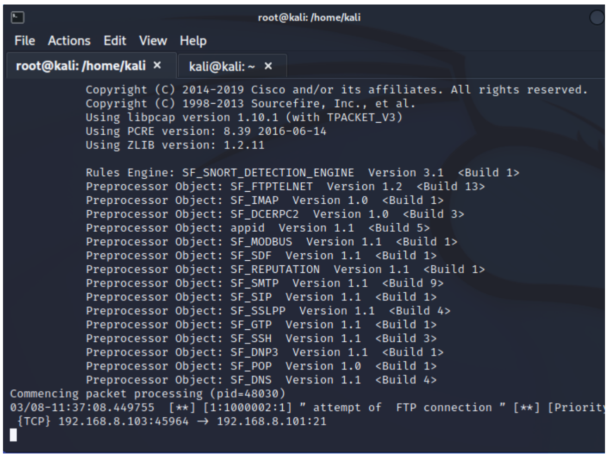
Rev:1: Used to document the rule and to facilitate its processing and tracking in the future [23].

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.rule 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.



```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ————————————————
# LOCAL RULES
# ————————————————
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert tcp any any → $HOME_NET 21 (msg:"FTP connection attempt"; sid:1000002; rev:1;)
```

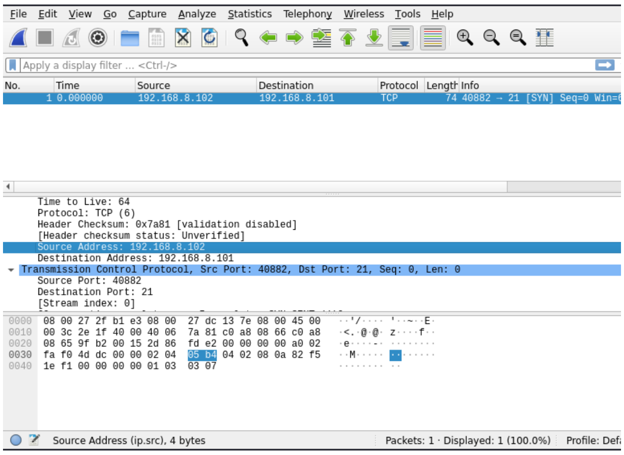
Figure 12. Create new rule in snort.



```
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
Commencing packet processing (pid=48030)
03/08-11:37:08.449755  [**] [1:1000002:1] " attempt of  FTP connection " [**] [Priority
 {TCP} 192.168.8.103:45964 -> 192.168.8.101:21
```

Figure 13. FTP connection alert.

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.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.      Time          Source           Destination      Protocol Length Info
       1 0.000000      192.168.8.102    192.168.8.101    TCP          74 40882 → 21 [SYN] Seq=0 Win=
```

```
      Time to Live: 64
      Protocol: TCP (6)
      Header Checksum: 0x7a81 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 192.168.8.102
      Destination Address: 192.168.8.101
  Transmission Control Protocol, Src Port: 40882, Dst Port: 21, Seq: 0, Len: 0
      Source Port: 40882
      Destination Port: 21
      [Stream index: 0]
```

```
0000  08 00 27 2f b1 e3 08 00  27 dc 13 7e 08 00 45 00   ··'/···· '··~··E·
0010  00 3c 2e 1f 40 00 40 06  7a 81 c0 a8 08 66 c0 a8   ·<.·@·@· z····f··
0020  08 65 9f b2 00 15 2d 86  fd e2 00 00 00 00 a0 02   ·e····-· ········
0030  fa f0 4d dc 00 00 02 04  05 b4 04 02 08 0a 82 f5   ··M····· ········
0040  1e f1 00 00 00 00 01 03  03 07                     ········ ··
```

```
Source Address (ip.src), 4 bytes          Packets: 1 · Displayed: 1 (100.0%)     Profile: Def
```

Figure 14. Snort logs tracking.

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

[1] R S Aarthee and E. Devarasan, “A logistic model for the population of virus growth in local area network,” *International Journal of Pure and Applied Mathematics*, vol. 115, no. 9, pp. 401–408, August 2017

[2] N. Khamphakdee, N. Benjamas, and S. Saiyod, “Improving intrusion detection system based on snort rules for network probe attack detection,” in *Proc. 2014 2nd International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2014, pp. 69–74, doi: 10.1109/ICoICT.2014.6914042.

[3] M. V. Pawar and J. Anuradha, “Network security and types of attacks in network,” *Procedia Computer Science*, vol. 48, 2015, pp. 503–506, doi: 10.1016/j.procs.2015.04.126

[4] S. Gopal, G. Sachin, and A. Ratish, “Intrusion detection using network monitoring tools,” *SSRN*, 2014, doi: 10.2139/ssrn.2426105.

[5] M. A. Qadeer, A. Iqbal, M. Zahid, and M. R. Siddiqui, “Network traffic analysis and intrusion detection using packet sniffer,” in *Proc. 2010 2nd International Conference on Communication Software and Networks*, 2010, pp. 313–317, doi: 10.1109/ICCSN.2010.104.

[6] J. Jabez and B. Muthukumar, “Intrusion Detection System (IDS): Anomaly detection using outlier detection approach,” *Procedia Computer Science*, vol. 48, 2015, doi: 10.1016/j.procs.2015.04.191.

[7] K. Amanpreet and M. Saluja, “Study of network security along with network security tools and network simulators,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp.88–99, 2014.

[8] S. Chakrabarti, M. Chakraborty, and I. Mukhopadhyay, “Study of snort-based IDS,” in *Proc. the International Conference and Workshop on Emerging Trends in Technology*, 2010, pp. 43–47.

[9] A. Rafia and R. Kumar, “Implementation of a malicious traffic filter using snort and wireshark as a proof of concept to enhance mobile network security,” *Journal of Telecommunications and Information Technology*, issue 1, pp. 64–71, 2022, doi: 10.26636/jtit.2022.155821.

[10] A. Singh, “Implementation of open-source ids (snort) to secure docker container,” Master’s thesis, National College of Ireland, 2020.

[11] J. Svoboda, I. Ghafir, and V. Prenosil, “Network monitoring approaches: An overview,” *Int. J. Adv. Comput. Netw. Secur.*, vol. 5, no. 2, pp. 88–93, 2015.

[12] G. Jain and Anubha, “Application of snort and wireshark in network traffic analysis,” *IOP Conference Series: Materials Science and Engineering*, 012007, 2021, doi: 10.1088/1757-899X/1119/1/012007.

[13] C. Alisha. 2012. A summary of network traffic monitoring and analysis techniques. [Online]. Available: http://www.cse.wustl.edu/~jain/cse567-06/ftp/net_monitoring.pdf

[14] A. Pallavi and P. Hemlata, “Analysis of various packet sniffing tools for network monitoring and analysis,” *International Journal of Electrical, Electronics and Computer Engineering*, pp. 55–58, 2012.

[15] V. Rajesh and M. Farik, “Intrusion detection amp prevention systems — Sourcefire snort,” *International Journal of Scientific & Technology Research*, vol. 4, no. 8, pp. 220–223, 2015.

[16] F. Tian and T. Chou, “An analysis of packet fragmentation attacks vs Snort intrusion detection system,” *International Journal of Computer Engineering Science*, vol. 2, no. 5, pp. 63–74, 2012.

[17] I. Haroon and S. Naaz, “Wireshark as a tool for detection of various LAN attacks,” *International Journal of Computer Sciences and Engineering*, vol. 7, no. 5, pp. 833–837, 2019.

[18] G. V. Nadiammai and M. Hemalatha, “Snort based network traffic anomaly detector to improve the performance of intrusion detection system,” *International Journal of Advanced Research in Computer Science*, vol. 3, no. 7, pp. 9–13, 2012, doi: 10.26483/ijarcs.v3i7.1402.

[19] I. Ghaffir, V. Prenosil, J. Svoboda, and M. Hammoudeh, “A survey on network security monitoring systems,” in *Proc. 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops*, 2016, pp. 77–82.

[20] I. Diyeb, I. Ali, A. Saif, and N. Al-shaibany, “Ethical network surveillance using packet sniffing tools: A comparative study,” *International Journal of Computer Network and Information Security*, vol. 10, no. 7, pp. 12–22, 2018.

[21] J. Canto, M. Dacier, E. Kirda, and C. Leita, “Large scale malware collection: Lessons learned,” in *Proc. 27th International Symposium on Reliable Distributed Systems*, vol. 52, no. 1, pp. 35–44, 2008.

[22] M. Rima and M. Aldwairi, “Automated malicious advertisement detection using,” in *Proc. 2017 8th International Conference on Information and Communication Systems*, 2017, pp. 336–341.

[23] V. Gurven, P. S. Patheja, and G. Ghai, “Intrusion detection a challenge: Snort the savior,” *International Journal of Computer Trends and Technology*, vol. 45, no. 1, pp. 1–3, 2017.

[24] N. Alsharabi, M. Alshammeri, and Y. Alharabi, “Analysis of ransomware using reverse engineering techniques to develop effective countermeasures,” *Journal of Advances in Information Technology*, vol. 14, no. 2, April 2023.