e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

NETWORK TRAFFIC ANALYSIS FOR INTRUSION WITH ZERO-TRUST

Nitin Ravi Gautam*[1], Harsh Mishra*[2], Dr. Arun Kumar Singh*[3]

*[1,2,3]Greater Noida Institute Of Technology, Greater Noida, India.

DOI : https://www.doi.org/10.56726/IRJMETS56206

ABSTRACT

In the contemporary landscape of cybersecurity, the proliferation of sophisticated cyber threats necessitates advanced approaches to safeguarding networks and data. This research paper explores the integration of Network Traffic Analysis (NTA) with Intrusion Detection Systems (IDS) to enhance the efficacy of Zero Trust security. The Zero Trust model framework assumes a posture of perpetual distrust, requiring continuous verification of entities and devices attempting to access resources within a network. The synergy of NTA and IDS within a Zero Trust architecture provides a comprehensive defense mechanism against evolving cyber threats.

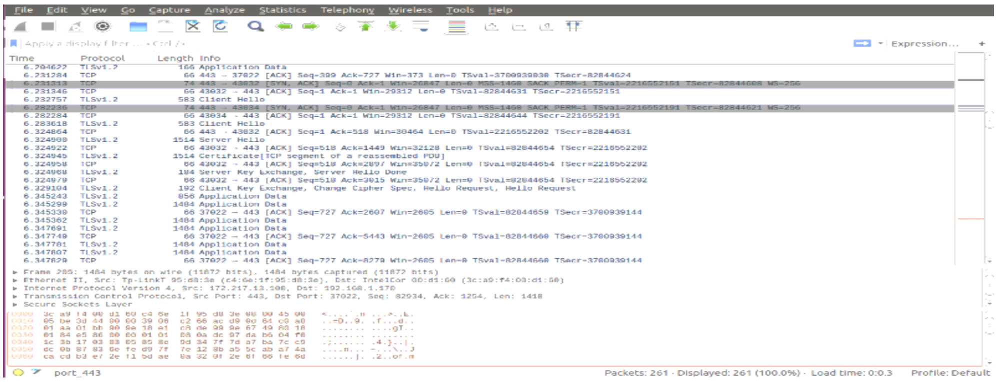
Keywords: Traffic Analysis, Intrusion Detection System, Network Traffic Analysis, Zero-Trust, Wireshark, Monitors Network, Cybersecurity And Security Analytics.

I. INTRODUCTION

Modern cybersecurity strategies must include Network Traffic Analysis (NTA), which gives organisations the capacity to track, examine, and react to network activity. NTA is essential for maintaining the integrity and security of networks in a world that is becoming more digitally connected and interconnected and where cyber threats are always changing. In order to guarantee a prompt and efficient response, this thorough method entails closely examining the data flow within a network in order to spot trends, abnormalities, and potential security threats.Using tools for network traffic analysis, you can keep an eye on both inbound and outgoing network data packets to gain important insights about things like bandwidth usage, security, and network performance.[1]

Network traffic analysis (NTA) tools, also known as real-time network traffic analyzers, are essential for tracking and evaluating network activity in real-time. With the use of these tools, organisations can spot irregularities, recognise possible security risks, and react quickly to network issues.[1] Organisations should take into account features like scalability, ease of use, compatibility with various protocols, and security and monitoring requirements when selecting a real-time network traffic analyzer. Several noteworthy instances of real-time network traffic analyzers are as follows:[3]

1.1 Wireshark: Wireshark is a widely used open-source network protocol analyzer. It allows users to capture and interactively browse the traffic running on a computer network in real-time.



```
Time        Protocol   Length Info
6.204622    TLSv1.2       166 Application Data
6.231284    TCP            66 443 → 37822 [ACK] Seq=399 Ack=727 Win=373 Len=0 TSval=3700939030 TSecr=82844624
6.231313    TCP            74 443 → 43032 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM=1 TSval=2216552151 TSecr=82844608 WS=256
6.231346    TCP            66 43032 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=82844631 TSecr=2216552151
6.232757    TLSv1.2       583 Client Hello
6.282236    TCP            74 443 → 43034 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM=1 TSval=2216552191 TSecr=82844621 WS=256
6.282284    TCP            66 43034 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=82844644 TSecr=2216552191
6.283618    TLSv1.2       583 Client Hello
6.324864    TCP            66 443 → 43032 [ACK] Seq=1 Ack=518 Win=30464 Len=0 TSval=2216552202 TSecr=82844631
6.324908    TLSv1.2      1514 Server Hello
6.324922    TCP            66 43032 → 443 [ACK] Seq=518 Ack=1449 Win=32128 Len=0 TSval=82844654 TSecr=2216552202
6.324945    TLSv1.2      1514 Certificate[TCP segment of a reassembled PDU]
6.324958    TCP            66 43032 → 443 [ACK] Seq=518 Ack=2897 Win=35072 Len=0 TSval=82844654 TSecr=2216552202
6.324968    TLSv1.2       184 Server Key Exchange, Server Hello Done
6.324979    TCP            66 43032 → 443 [ACK] Seq=518 Ack=3015 Win=35072 Len=0 TSval=82844654 TSecr=2216552202
6.329104    TLSv1.2       192 Client Key Exchange, Change Cipher Spec, Hello Request, Hello Request
6.345243    TLSv1.2       856 Application Data
6.345299    TLSv1.2      1484 Application Data
6.345330    TCP            66 37022 → 443 [ACK] Seq=727 Ack=2607 Win=2605 Len=0 TSval=82844659 TSecr=3700939144
6.345362    TLSv1.2      1484 Application Data
6.347691    TLSv1.2      1484 Application Data
6.347749    TCP            66 37022 → 443 [ACK] Seq=727 Ack=5443 Win=2605 Len=0 TSval=82844660 TSecr=3700939144
6.347781    TLSv1.2      1484 Application Data
6.347807    TLSv1.2      1484 Application Data
6.347829    TCP            66 37022 → 443 [ACK] Seq=727 Ack=8279 Win=2605 Len=0 TSval=82844660 TSecr=3700939144
▸ Frame 285: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits)
▸ Ethernet II, Src: Tp-LinkT 95:d8:3e (c4:6e:1f:95:d8:3e), Dst: IntelCor 00:d1:60 (3c:a9:f4:00:d1:60)
▸ Internet Protocol Version 4, Src: 172.217.13.108, Dst: 192.168.1.170
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 37022, Seq: 82934, Ack: 1254, Len: 1418
▸ Secure Sockets Layer
0000  3c a9 f4 00 d1 60 c4 6e  1f 95 d8 3e 08 00 45 00   <....`.n ...>..E.
0010  05 be 3d 44 00 00 39 06  c2 66 ac d9 0d 6c c0 a8   ..=D..9. .f...d..
0020  01 aa 01 bb 90 9e 18 e1  c8 de 99 9e 67 49 80 18   ........ ....gI..
0030  01 84 e5 86 00 00 01 01  08 0a dc 97 da b6 04 f8   ........ ........
0040  1c 3b 17 03 03 05 85 8c  9d 34 7f 7d a7 ba 7c c9   .;...... .4.}..|.
0050  dc 0b 87 83 6e fe d9 7f  7e 12 8b a5 5c ab a7 4a   ....n... ~...\..J
0060  ca cd b3 e7 2e f1 5d ae  0a 32 01 2e 61 66 fe 6d   ......]. .2..af.m
port_443          Packets: 261 · Displayed: 261 (100.0%) · Load time: 0:0.3     Profile: Default
```

Fig. 1. Analysis of Network with Wireshark

Wireshark supports a vast array of protocols and provides detailed packet-level analysis. While it is a versatile tool, it may require a certain level of expertise for in-depth usage.

1.2 NetFlow Analyzer: NetFlow Analyzer is a tool designed for real-time network traffic monitoring and analysis. It supports NetFlow, is Flow, IPFIX, and other flow technologies to collect and analyze traffic data.

NetFlow Analyzer provides insights into applications, conversations, and devices consuming network resources in real-time.

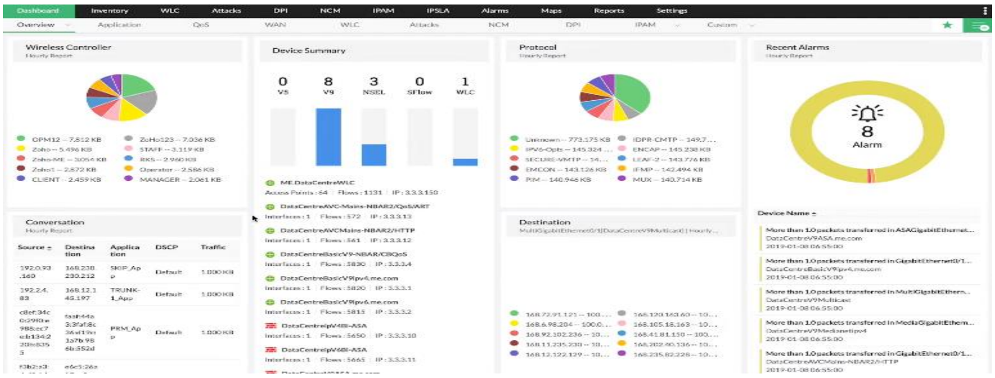


Fig. 2. Analysis of Network with NetFlow Analyzer

1.3 Cisco-Stealthwatch: Cisco Stealthwatch is an enterprise-grade network traffic analysis solution that offers real-time visibility into network activities. It utilizes NetFlow data and advanced security analytics to detect and respond to potential threats. Stealthwatch provides insights into user and device behaviors, aiding in the identification of anomalies

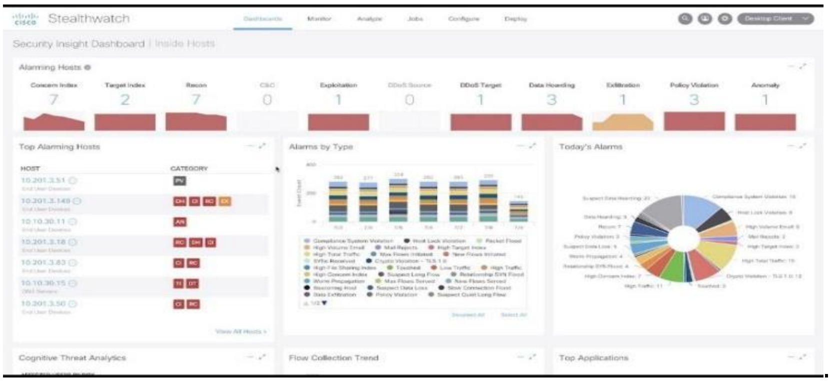


Fig. 3. Analysis of Network with Cisco Stealth-watch

1.4 Flowmon: Flowmon is a real-time traffic analysis tool for network security and monitoring. It uses flow data (IPFIX, Net-Flow, etc.) to provide information about network behaviour, application performance, and security incidents. Flowmon offers customisable dashboards and alerting features.

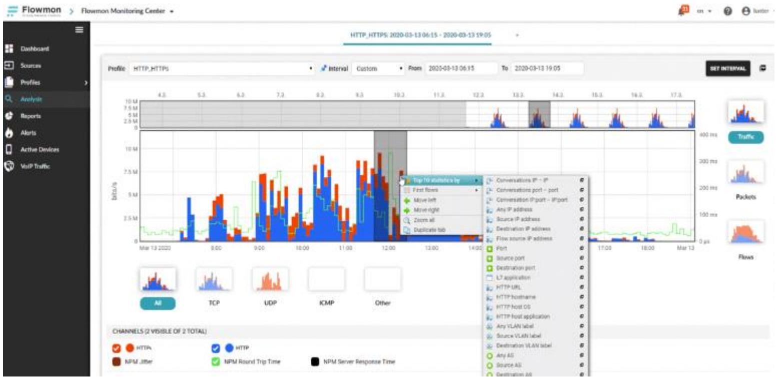
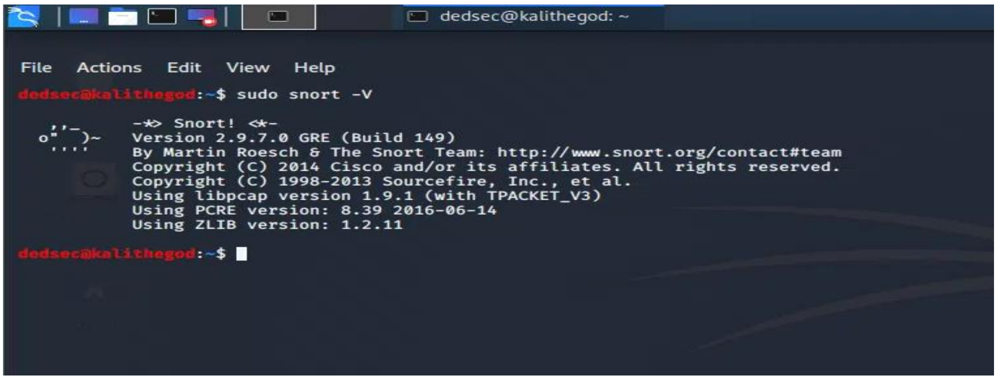


Fig. 4. Analysis of Network with Flowmon

Intrusion detection systems (IDS) are cybersecurity tools designed to monitor network or system activity and search for signs of malicious or unauthorised activity. These are the tools you need to identify security threats and take the necessary countermeasures. Additionally, they provide an additional line of defence to protect businesses' digital assets. Intrusion detection systems fall into two main categories: host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). Some well-known intrusion detection techniques that fit into these categories are as follows:

1.5 Snort:



```
dedsec@kalithegod:~$ sudo snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

dedsec@kalithegod:~$
```

Fig. 5. Analysis of Network with Snort

Snort is an open-source NIDS known for its flexibility and extensive community support. It uses a signature-based detection approach and can also perform protocol analysis and content matching.

1.6 Suricata:5

Fig. 6. Analysis of Network with Suricata

Suricata is an open-source IDS/IPS engine that emphasizes high-performance and extensive protocol support. It supports signature-based detection, anomaly detection, and has multi-threading capabilities.

1.7 Zeek: Zeek is an open-source framework for network analysis with a protocol analysis emphasis. Zeek is helpful for intrusion detection and network monitoring even though it isn't just an IDS because it offers deep insights into network activity.specific technologies, tools, and approaches to quickly detect and address security events.

Fig. 7. Analysis of Network with Zeek

1.8 Components of Intrusion Detection Systems:

IDS uses agents or sensors that are placed strategically throughout a network to gather information about system and network activity. These sensors send pertinent data to the central analysis engine while keeping an eye on system activity and traffic. The central unit responsible for processing the data gathered by sensors is the analysis engine. It finds patterns suggestive of possible security threats using behavioural analysis techniques, signatures, and predefined rules. The analysis engine sends out alerts to security staff when it finds suspicious activity. Alerts give details about the systems that are impacted, the type of intrusion, and the seriousness of the event.[12]

Intrusion Detection Systems may include automated or manual response mechanisms. Automated responses can include blocking malicious IP addresses, isolating affected systems, or triggering other preventive measures. Manual responses involve human intervention based on the alerts generated. In the dynamic landscape of cybersecurity, where threats constantly evolve and traditional security models are increasingly challenged, the Zero Trust paradigm has emerged as a foundational principle for safeguarding digital assets. Integrating Zero Trust with Network Traffic Analysis (NTA) for intrusion detection represents a powerful synergy that enhances an organization's ability to detect and respond to potential threats in real-time. This comprehensive approach not only challenges the traditional perimeter-based security model but also aligns with the principles of continuous verification and least privilege access, crucial elements of the Zero Trust framework.[4]

II. LITERATURE REVIEW

Even though Wireshark is a strong and popular tool for network traffic analysis, whether or not to use it over alternative tools will depend on your needs, preferences, and the analysis's objectives. For network traffic analysis, Wireshark may be favoured over other tools for the following reasons: Since Wireshark is an open-source programme, anyone can use it without restriction. For businesses or individuals on a tight budget or who favour open-source solutions, this is a huge benefit. A cross-platform utility for Linux, macOS, and Windows is called Wireshark. This guarantees adaptability and user-friendliness on various operating systems.

Because it can analyse a wide range of network traffic types, Wireshark is a versatile tool for network traffic analysis. It can handle complicated as well as common protocols, enabling thorough analysis. With Wireshark, one can examine recorded capture files or record network traffic in real time. This adaptability is useful for reviewing historical data as well as troubleshooting current issues. Wireshark provides a comprehensive packet-level analysis with an easy-to-use interface. It's simple to use and navigate for both novice and expert users because of its color-coded display, intuitive layout, and abundance of filtering options. Network traffic analysis intrusion detection tools come in a variety of forms, each with special applications and advantages. The ideal tool will vary depending on a number of factors, including the particular needs of the organisation, the network infrastructure, and financial constraints.[3]

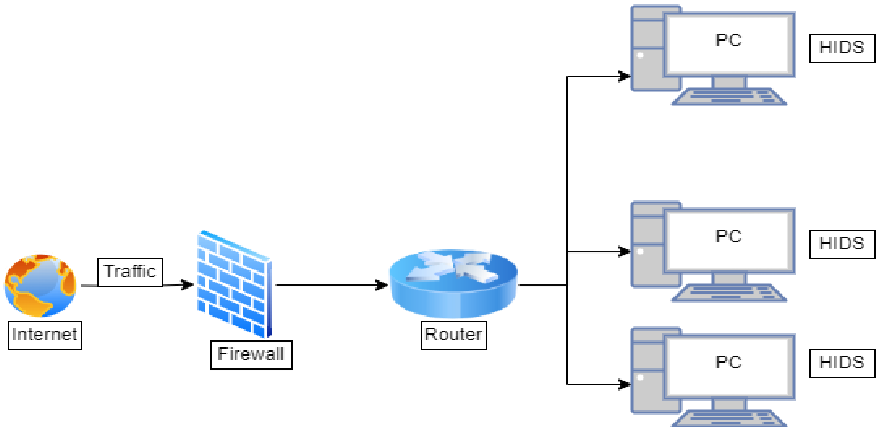


Fig. 8. Working of IDS

When selecting an intrusion detection tool, it's important to consider factors like compatibility with other security solutions in your environment, scalability, ease of use, and community support. A few instances of the various ways that intrusions can happen are malware infections, denial-of-service attacks, unauthorised access attempts, and data breaches. Intrusion detection systems aim to find patterns associated with these intrusions and promptly notify security teams of such findings. Intrusion detection systems fall into two main categories: host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS).

2.1 Snort:

Snort is an open-source intrusion detection and prevention system (IDS/IPS) known for its flexibility and community support. It uses signature-based detection, protocol analysis, and anomaly detection to identify potential threats.

Key Features: Signature-based detection., Protocol analysis., Anomaly detection., Extensive community rule sets.

2.2 Suricata:

Suricata is an open-source IDS/IPS engine designed for high performance. It supports multi-threading, signature-based detection, and has features for protocol analysis and anomaly detection. Suricata is suitable for high-speed networks.

Key Features: Signature-based detection., Anomaly detection., multi-threading support., Extensive protocol support.

2.3 Zeek (formerly Bro):

Zeek is a powerful network analysis framework that goes beyond traditional IDS functionality. It focuses on protocol analysis and metadata extraction, making it valuable for detailed network traffic analysis.

Key Features: Protocol analysis., Rich metadata extraction., High-level scripting language.

2.4 Snort with Snorby:

Snorby is a web-based front end for Snort that enhances the user interface and reporting capabilities. It provides a user-friendly dashboard for visualizing alerts and offers additional features for managing and analyzing intrusion detection data.

Key Features: Web-based dashboard., Alert visualization., Reporting capabilities.

These tools contribute to building a robust defense against cyber threats by monitoring network and host activities, detecting potential intrusions, and providing organizations with the information needed to respond effectively to security incidents. When implementing IDS solutions, organizations should consider factors such as scalability, ease of use, customization options, and the specific needs. Integrating Zero Trust with Network Traffic Analysis (NTA) for intrusion detection represents a powerful synergy that enhances an organization's ability to detect and respond to potential threats in real-time. This comprehensive approach not only challenges the traditional perimeter-based security model but also aligns with the principles of continuous verification and least privilege access, crucial elements of the Zero Trust framework.

2.5 Zero Trust Principles in Network Traffic Analysis:

Zero Trust, at its core, emphasizes continuous monitoring. When applied to Network Traffic Analysis, this principle ensures that every communication within the network is scrutinized in real-time. Instead of relying on static trust levels, continuous monitoring allows organizations to dynamically assess the trustworthiness of ongoing network activities.In a Zero Trust NTA framework, entities such as users, devices, and applications are continually verified. This verification extends beyond initial authentication, ensuring that entities maintain their legitimacy throughout their interactions with the network. Any deviation from established baselines triggers alerts, indicating potential security threats.

Zero Trust advocates for the principle of least privilege access, limiting user and device access to only what is necessary for their specific roles. When applied to NTA, this principle ensures that network traffic is scrutinized based on the context of the user or device, allowing organizations to detect and respond to any unusual or unauthorized access patterns. Micro-segmentation, a key element of Zero Trust, involves dividing the network into small, isolated segments. In the context of NTA, this segmentation allows for granular analysis of traffic within each segment. By isolating and monitoring segments, organizations can quickly identify abnormal communication patterns, restricting the lateral movement of potential threats. Zero Trust NTA mandates the inspection and logging of all network traffic, including encrypted communications. This ensures that malicious activities are not concealed within encrypted channels, aligning with the Zero Trust principle of assuming a breach and maintaining visibility into all network activities.

III. METHODOLOGY

Wireshark uses several phases for network traffic analysis. The steps are as follows for capturing the network packet:

1. Select the network on which you want to perform traffic analysis from various available options like Ethernet, Wi-Fi, LAN connection etc.
2. If you wish to apply filters before starting the capture you can do so by typing the same in the text field provided by the Wireshark. Filters like choosing a specific protocol or port and many more can be applied.
3. Traffic would be captured and be available for analysis. We can also save the captured traffic for future work if we wish to do so.
4. The packets are then decoded, examined, and analyzed by the users

3.1 TCP Header:

- Source Port: It includes the source or sender's unique port number. The field has 16 bits
- Destination Port: It includes the receiver's or destination's precise port number. The field has 16 bits.
- Sequence Number: This identifies the amount of data transferred throughout the TCP Session. It's a 32-field here. Initial sequence number for a new connection is a random 32-bit value. Utilizing this sequence number, the recipient replies with an acknowledgment to the sender. To make it simpler to comprehend, Wireshark utilizes relative sequence numbers beginning with 0.
- DO (Data Offset): The header length is another name for the 4-bit data offset field. It provides the length of the TCP header to identify when the actual data starts.
- RSV: The reserved field's three bits are set to 0 and not used.
- Flags: There are nine flag bits, are also known as control bits. This field is used to create connections, transmit data, and break connections:
- Urgent Pointer: (URG) The data should be viewed as having priority over other data when this bit is set, ACK: abbreviation for acknowledgement.
- PSH: This stands for PUSH. This instructs a program to send the data right away rather than waiting for it to fill the whole TCP segment.
- RST: This resets the connection. We must immediately cut off the connection if we receive it. This is not the typical technique to close a TCP connection; it is only done when there are irrecoverable faults.
- SYN: This is used to establish the first sequence number and for the initial three-way handshake.
- FIN: The TCP connection is ended using this finish bit. TCP is full duplex, so to terminate a connection; both parties must use the FIN bit. This is how we typically cut off a connection.
- Window: The window field is of 16-bit. It is used to represent the maximum length of bytes that a receiver will accept. It does this by informing the number of bytes after the sequence number in the acknowledgment field.
- Checksum: Checksum is a 16-bit field which is used to tell whether the TCP header is in a good situation or not.
- Urgent Pointer: URG is a 16-bit field and when this bit is set then it acts as a marker for the end of the urgent data.
- Options: This is an optional field. Its size lies between 0 to 320 bits.

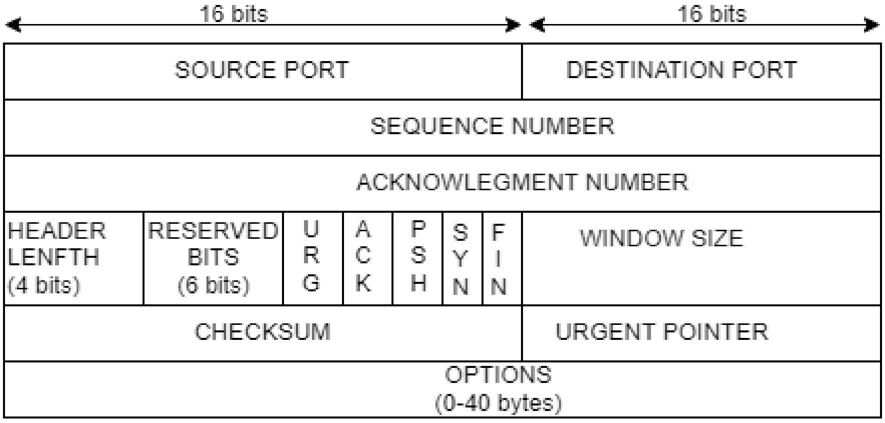
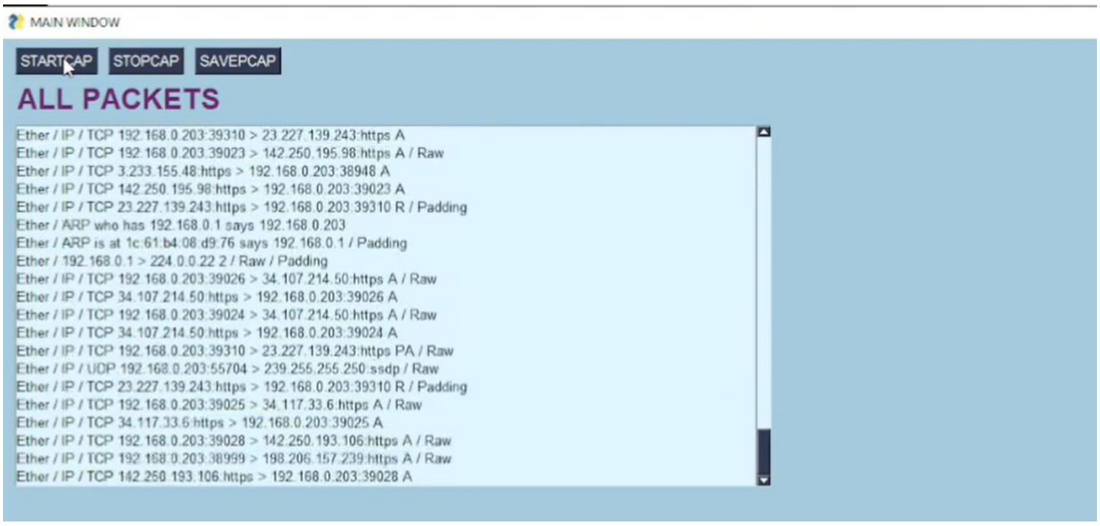


Fig. 9. Transmission Control Protocol Header Format

The Wireshark has easy to use interface. Users can easily capture the packets and analyze those packets. It is available for both UNIX and Windows. Captured packets can also be exported and imported in several capture file formats for further analysis. Color coding is also provided by Wireshark, helping the user to analyze the packets. Below table shows the various coloring rules.[12]

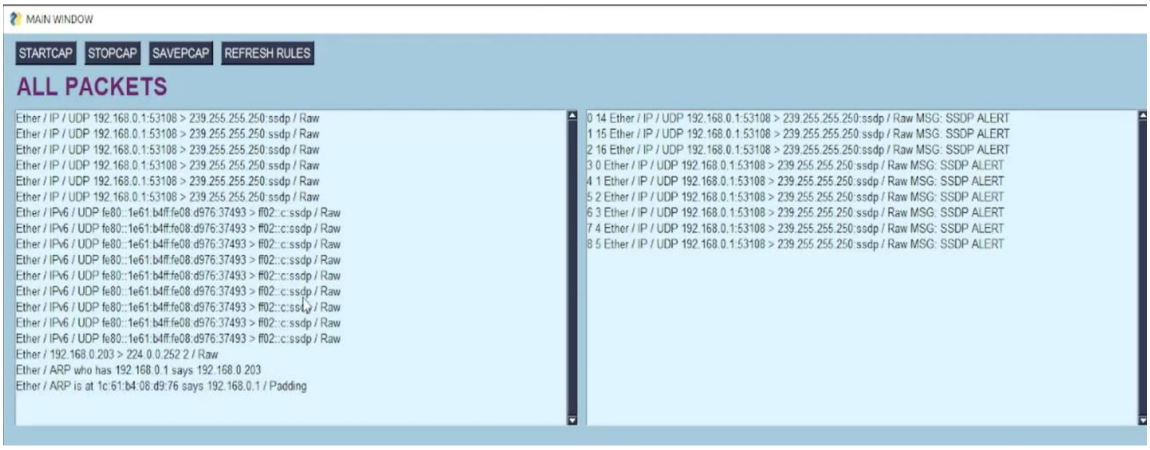
3.2 Applying filters and analyzing the packets:

In this traffic, protocol filter of HTTP was applied, which resulted in showing the packets related to http protocol only as shown in below figure.



```
Ether / IP / TCP 192.168.0.203:39310 > 23.227.139.243:https A
Ether / IP / TCP 192.168.0.203:39023 > 142.250.195.98:https A / Raw
Ether / IP / TCP 3.233.155.48:https > 192.168.0.203:38948 A
Ether / IP / TCP 142.250.195.98:https > 192.168.0.203:39023 A
Ether / IP / TCP 23.227.139.243:https > 192.168.0.203:39310 R / Padding
Ether / ARP who has 192.168.0.1 says 192.168.0.203
Ether / ARP is at 1c:61:b4:08:d9:76 says 192.168.0.1 / Padding
Ether / 192.168.0.1 > 224.0.0.22 2 / Raw / Padding
Ether / IP / TCP 192.168.0.203:39026 > 34.107.214.50:https A / Raw
Ether / IP / TCP 34.107.214.50:https > 192.168.0.203:39026 A
Ether / IP / TCP 192.168.0.203:39024 > 34.107.214.50:https A / Raw
Ether / IP / TCP 34.107.214.50:https > 192.168.0.203:39024 A
Ether / IP / TCP 192.168.0.203:39310 > 23.227.139.243:https PA / Raw
Ether / IP / UDP 192.168.0.203:55704 > 239.255.255.250:ssdp / Raw
Ether / IP / TCP 23.227.139.243:https > 192.168.0.203:39310 R / Padding
Ether / IP / TCP 192.168.0.203:39025 > 34.117.33.6:https A / Raw
Ether / IP / TCP 34.117.33.6:https > 192.168.0.203:39025 A
Ether / IP / TCP 192.168.0.203:39028 > 142.250.193.106:https A / Raw
Ether / IP / TCP 192.168.0.203:38999 > 198.206.157.239:https A / Raw
Ether / IP / TCP 142.250.193.106:https > 192.168.0.203:39028 A
```

Fig. 10. HTTP Protocol Packets Captured



```
Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw
Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / IPv6 / UDP fe80::1e61:b4ff:fe08:d976:37493 > ff02::c:ssdp / Raw
Ether / 192.168.0.203 > 224.0.0.252 2 / Raw
Ether / ARP who has 192.168.0.1 says 192.168.0.203
Ether / ARP is at 1c:61:b4:08:d9:76 says 192.168.0.1 / Padding
```

```
0 14 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
1 15 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
2 16 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
3 0 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
4 1 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
5 2 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
6 3 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
7 4 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
8 5 Ether / IP / UDP 192.168.0.1:53108 > 239.255.255.250:ssdp / Raw MSG: SSDP ALERT
```

Fig. 11. HTTP Protocol Packet Details

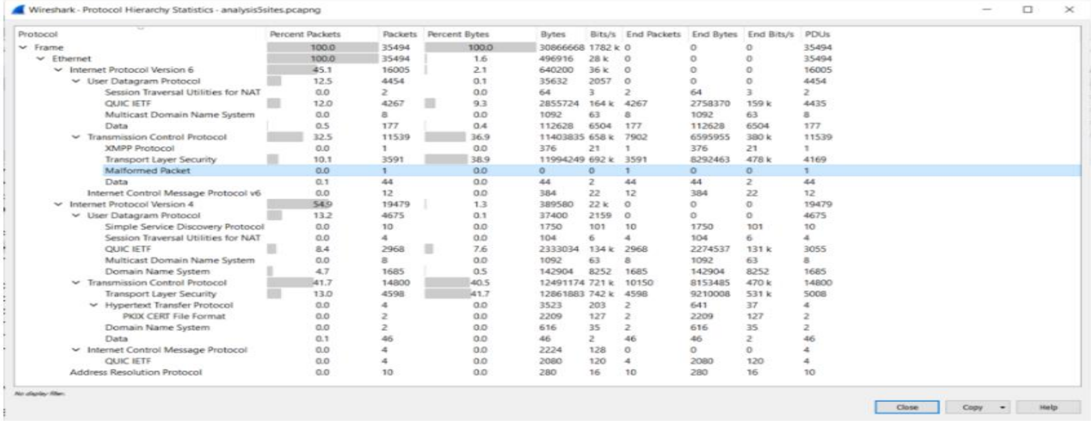
The HTTP header details are displayed when the Hypertext Transfer Protocol is selected in the Packet Details section for packet number 23173. As can be seen from the image above, this packet includes information about the certificate. The information below can be deduced from the above: Attackers may misuse such a field, leading to a variety of other attacks. This demonstrates how an attacker with access to the user's network could easily trace these packets and alter the data within them, leading to a Man-in-the-Middle or redirecting attack, if any user uses any http site.

3.3 Exploring the Statistics on the Captured Traffic:

Protocol Hierarchy: Protocol Hierarchy display the number of packets and number of bytes in those packets for various protocols that were captured during for network analysis all the protocols are arranged in the same hierarchy as they were found in the traffic. It provides the count of packets in which the protocol is present and the packet in which it is the last protocol in the stack. These last-protocol counts let you know how many packets—along with the corresponding byte count—ended in a certain protocol. They are listed under "End Packets" and "End Bytes" in the table.[13]

Flow Graph: Flow graph shows the connection between the hosts. For each connection that was captured it shows the packet timing, direction, ports, and comments. It provides filters like ICMP (Internet Control Message Protocol) flows, ICMPv6 flows, UIM flows, and TCP flow. The flow graph window provides different controls based on that.[5]

With the help of flow graph you can easily figure out various port numbers and IP addresses and thus can easily get to get know if any unusual port number or IP address occurs in the traffic. The below figure shows the flow graph for the captured traffic.

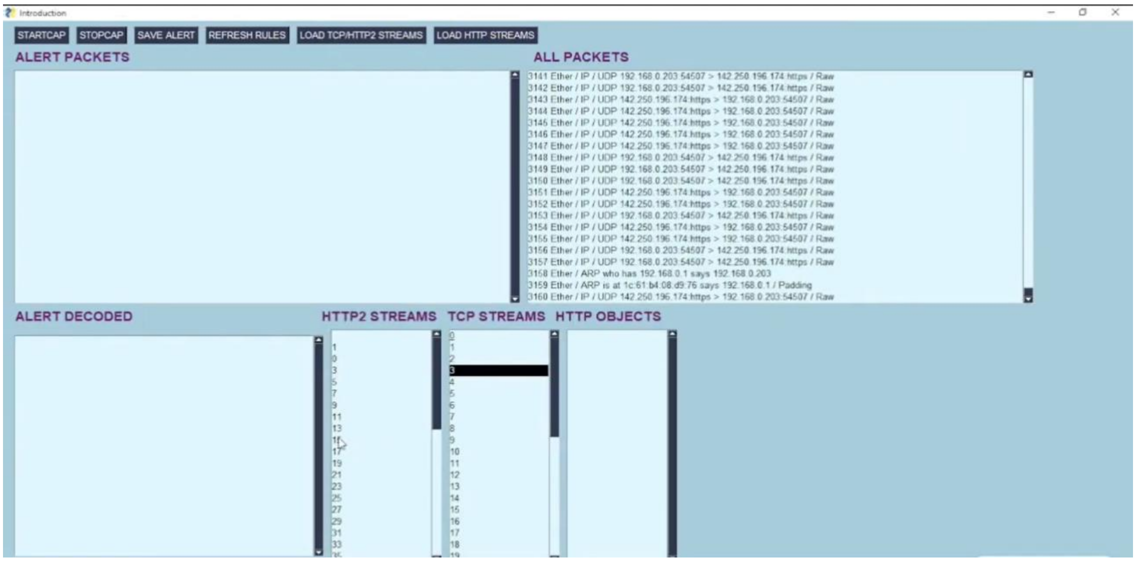


Wireshark · Protocol Hierarchy Statistics · analysis5sites.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	35494	100.0	30866668	1782 k	0	0	0	35494
Ethernet	100.0	35494	1.6	496916	28 k	0	0	0	35494
Internet Protocol Version 6	45.1	16005	2.1	640200	36 k	0	0	0	16005
User Datagram Protocol	12.5	4454	0.1	35632	2057	0	0	0	4454
Session Traversal Utilities for NAT	0.0	2	0.0	64	3	2	64	3	2
QUIC IETF	12.0	4267	9.3	2855724	164 k	4267	2758370	159 k	4435
Multicast Domain Name System	0.0	8	0.0	1092	63	8	1092	63	8
Data	0.5	177	0.4	112628	6504	177	112628	6504	177
Transmission Control Protocol	32.5	11539	36.9	11403835	658 k	7902	6595955	380 k	11539
XMPP Protocol	0.0	1	0.0	376	21	1	376	21	1
Transport Layer Security	10.1	3591	38.9	11994249	692 k	3591	8292463	478 k	4169
Malformed Packet	0.0	1	0.0	0	0	1	0	0	1
Data	0.1	44	0.0	44	2	44	44	2	44
Internet Control Message Protocol v6	0.0	12	0.0	384	22	12	384	22	12
Internet Protocol Version 4	54.9	19479	1.3	389580	22 k	0	0	0	19479
User Datagram Protocol	13.2	4675	0.1	37400	2159	0	0	0	4675
Simple Service Discovery Protocol	0.0	10	0.0	1750	101	10	1750	101	10
Session Traversal Utilities for NAT	0.0	4	0.0	104	6	4	104	6	4
QUIC IETF	8.4	2968	7.6	2333034	134 k	2968	2274537	131 k	3055
Multicast Domain Name System	0.0	8	0.0	1092	63	8	1092	63	8
Domain Name System	4.7	1685	0.5	142904	8252	1685	142904	8252	1685
Transmission Control Protocol	41.7	14800	40.5	12491174	721 k	10150	8153485	470 k	14800
Transport Layer Security	13.0	4598	41.7	12861883	742 k	4598	9210008	531 k	5008
Hypertext Transfer Protocol	0.0	4	0.0	3523	203	2	641	37	4
PKIX CERT File Format	0.0	2	0.0	2209	127	2	2209	127	2
Domain Name System	0.0	2	0.0	616	35	2	616	35	2
Data	0.1	46	0.0	46	2	46	46	2	46
Internet Control Message Protocol	0.0	4	0.0	2224	128	0	0	0	4
QUIC IETF	0.0	4	0.0	2080	120	4	2080	120	4
Address Resolution Protocol	0.0	10	0.0	280	16	10	280	16	10

No display filter.

Fig. 12. Protocol Hierarchy Statistics



```
3141 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3142 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3143 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3144 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3145 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3146 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3147 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3148 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3149 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3150 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3151 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3152 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3153 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3154 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3155 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
3156 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3157 Ether / IP / UDP 192.168.0.203:54507 > 142.250.196.174:https / Raw
3158 Ether / ARP who has 192.168.0.1 says 192.168.0.203
3159 Ether / ARP is at 1c:61:b4:08:d9:76 says 192.168.0.1 / Padding
3160 Ether / IP / UDP 142.250.196.174:https > 192.168.0.203:54507 / Raw
```

HTTP2 STREAMS:

```
1
0
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
```

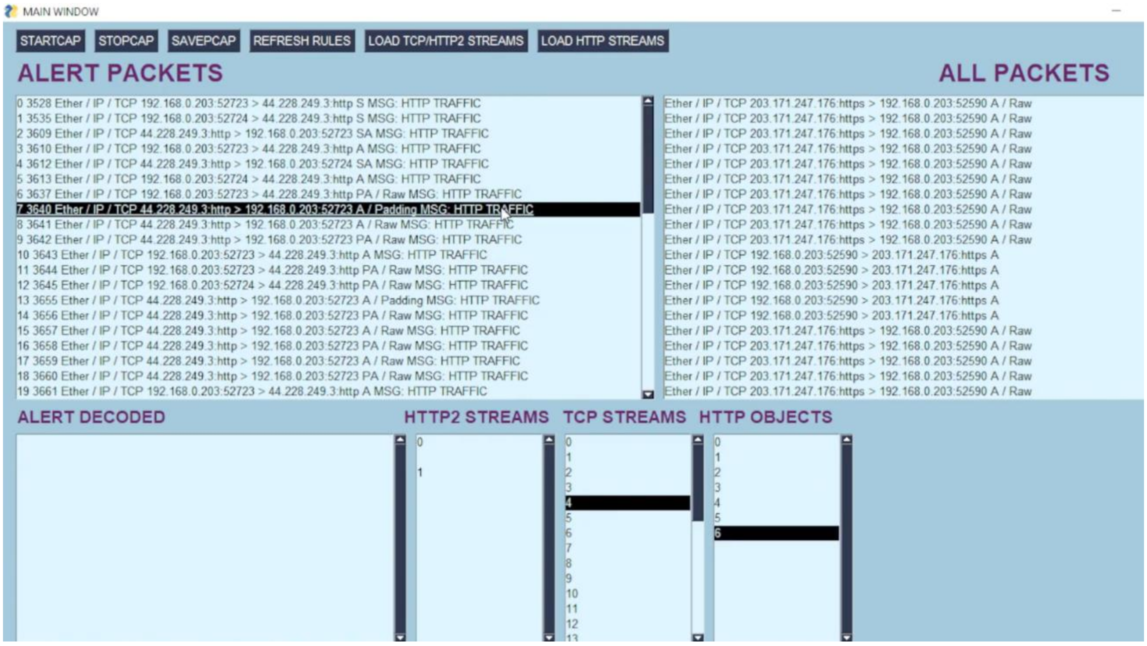
TCP STREAMS:

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

Fig. 13. Protocol Packet with Streams

IO GRAPH: Display the number of packets or the number of bytes per second for all packets that match the chosen filter. By default, only one graph displaying the number of packets per second will be shown.

HTTP Packet Counter: The packet counter the data regarding the HTTP packets. From here, we can analyze if there was any redirection or any kind of error. It helps in knowing if any attack like DDOS attack took place, or were any packets redirected to any other unusual address. From the below figure, we analyzed that all the packets have 2xx Response, indicating that all packets were transmitted successfully, and no packet was dropped.



```
0 3528 Ether / IP / TCP 192.168.0.203:52723 > 44.228.249.3:http S MSG: HTTP TRAFFIC
1 3535 Ether / IP / TCP 192.168.0.203:52724 > 44.228.249.3:http S MSG: HTTP TRAFFIC
2 3609 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 SA MSG: HTTP TRAFFIC
3 3610 Ether / IP / TCP 192.168.0.203:52723 > 44.228.249.3:http A MSG: HTTP TRAFFIC
4 3612 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52724 SA MSG: HTTP TRAFFIC
5 3613 Ether / IP / TCP 192.168.0.203:52724 > 44.228.249.3:http A MSG: HTTP TRAFFIC
6 3637 Ether / IP / TCP 192.168.0.203:52723 > 44.228.249.3:http PA / Raw MSG: HTTP TRAFFIC
7 3640 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 A / Padding MSG: HTTP TRAFFIC
8 3641 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 A / Raw MSG: HTTP TRAFFIC
9 3642 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 PA / Raw MSG: HTTP TRAFFIC
10 3643 Ether / IP / TCP 192.168.0.203:52723 > 44.228.249.3:http A MSG: HTTP TRAFFIC
11 3644 Ether / IP / TCP 192.168.0.203:52723 > 44.228.249.3:http PA / Raw MSG: HTTP TRAFFIC
12 3645 Ether / IP / TCP 192.168.0.203:52724 > 44.228.249.3:http PA / Raw MSG: HTTP TRAFFIC
13 3655 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 A / Padding MSG: HTTP TRAFFIC
14 3656 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 PA / Raw MSG: HTTP TRAFFIC
15 3657 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 A / Raw MSG: HTTP TRAFFIC
16 3658 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 PA / Raw MSG: HTTP TRAFFIC
17 3659 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 A / Raw MSG: HTTP TRAFFIC
18 3660 Ether / IP / TCP 44.228.249.3:http > 192.168.0.203:52723 PA / Raw MSG: HTTP TRAFFIC
19 3661 Ether / IP / TCP 192.168.0.203:52723 > 44.228.249.3:http A MSG: HTTP TRAFFIC
```

```
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 192.168.0.203:52590 > 203.171.247.176:https A
Ether / IP / TCP 192.168.0.203:52590 > 203.171.247.176:https A
Ether / IP / TCP 192.168.0.203:52590 > 203.171.247.176:https A
Ether / IP / TCP 192.168.0.203:52590 > 203.171.247.176:https A
Ether / IP / TCP 192.168.0.203:52590 > 203.171.247.176:https A
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
Ether / IP / TCP 203.171.247.176:https > 192.168.0.203:52590 A / Raw
```

Fig. 14. Packet Counter

3.4 Intrusion Detection

Intrusion Detection (ID) is one of the security mechanisms to identify authorized or unauthorized user behavior in the system or network. An unauthorized user is known as an intruder and it might be insider or outsider. Models of Intrusion Detection System (IDS) are of two types: Anomaly adjunct and misuse-signature adjunct. Of the two, anomaly adjunct identifies attacks based on deviations in behaviour, while misuse detection detects known attacks.[3]

The idea of user unauthorised behaviour is based on authorised behaviour observed by observers, and this leads to the development of an intrusion detection system. A user's behaviour that deviates from expectations is deemed intrusive. An intrusion detection system is used to identify LQWUXVLYHEHKDYLRUXVHUDFWLYLW \ORJDQGXVHU\VFXUUHQW activity. New techniques developed by researchers can observe and detect the present position of the user if there is a difference between the user's current state with stored profile an alert will be generated.

IDS based on signatures has high true positive rates for recognised patterns. It is simple to use and light-weight. Another name for the detected pattern is a signature. The signature-based intrusion detection system (IDS) merely recognises the known pattern by storing patterns in the database and comparing the user's current behaviour with a stored pattern to identify authorised or unauthorised users. We have proposed signature-based intrusion detection based on user behavior using pattern matching technique.[3]

Pattern or Signature detection for intrusion detection systems is rule-based techniques. Such systems are built on numerous conditional if-then rules for their detection. System matches current contextual information with the given list of rules. System allows to continue access if rules are matched otherwise any violation in rule matching leads to an ID.

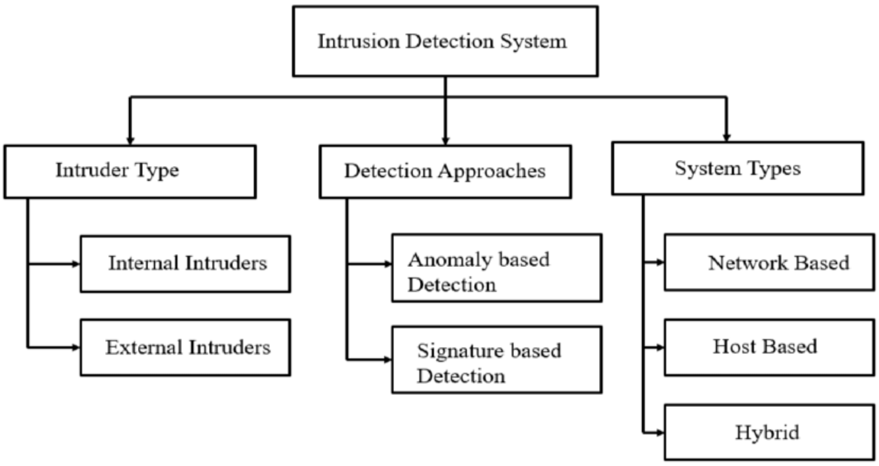


Fig. 15. Types in Intrusion Detection

Rules are developed or coined by analyzing all possible intrusions or any malicious activities by experts and then transferring them into conditional rules which are later used by inference modules of IDSs to compare against logs (monitoring data) to recognize any kind of incursion. Researchers have used statistical techniques such as time series and Markov chains to implement a method that can identify an unauthorized user based on user activity. [5]

3.5 Implementation Challenges and Considerations:

One of the challenges with IDS is the occurrence of false positives (incorrectly identifying normal behavior as an intrusion) and false negatives (failing to detect actual intrusions). Striking a balance between sensitivity and specificity is crucial to minimizing these occurrences. As networks grow in complexity and volume, IDS solutions must be scalable to handle increased traffic without sacrificing performance. Scalability considerations are vital to ensure that the system can effectively monitor and analyze all relevant network activities.

Encrypted traffic poses a challenge for IDS, as it may hide malicious activities. Solutions that can decrypt and inspect encrypted traffic without compromising user privacy are essential for comprehensive intrusion detection. IDS signatures, rules, and detection mechanisms require regular updates to remain effective against evolving threats. Organizations must establish processes for timely updates and ensure that the IDS is equipped to detect the latest attack vectors.

3.6 Intrusion Detection in a Zero Trust Environment:

The Zero Trust security model, which assumes that threats can come from both external and internal sources, aligns closely with the objectives of intrusion detection. In a Zero Trust environment, where trust is never assumed and continuous verification is paramount, IDS becomes a crucial component for monitoring and responding to potential security incidents.[4]

Intrusion Detection aligns with the continuous monitoring principles of Zero Trust by providing real-time visibility into network and system activities. This constant vigilance helps organizations detect and respond to potential threats promptly. Integrating User and Entity Behavior Analytics with IDS enhances the capability to detect insider threats within a Zero Trust framework. Monitoring user activities and access patterns helps identify anomalies that may indicate compromised accounts or malicious intent.[4]

Intrusion Detection complements micro-segmentation strategies within a Zero Trust architecture. By monitoring traffic within isolated segments, organizations can quickly identify and respond to abnormal communication patterns, limiting the impact of security incidents. In a Zero Trust environment, automated incident response is crucial. IDS plays a key role in triggering automated responses, such as isolating compromised devices, blocking malicious traffic, or initiating other predefined actions to contain and mitigate security incidents.

3.7 Future Trends in Intrusion Detection:

The integration of machine learning and artificial intelligence is a growing trend in intrusion detection. These technologies enhance the ability to detect complex, evolving threats by analyzing patterns and anomalies in large datasets. Behavioral biometrics, such as analyzing typing patterns and mouse movements, can be incorporated into IDS to enhance user authentication and detect anomalies in user behavior. This adds an additional layer of security to traditional intrusion detection methods.[2]

As organizations increasingly migrate to cloud environments, there is a growing need for cloud-based intrusion detection solutions. These solutions offer scalability, flexibility, and the ability to monitor and protect assets in dynamic cloud environments. Enhanced integration with threat intelligence feeds is a future trend in intrusion detection. This involves leveraging up-to-date information on known threats to improve the detection capabilities of IDS and respond more effectively to emerging security risks.[3]

3.8 Benefits of Zero Trust in Network Traffic Analysis for Intrusion Detection:

The continuous monitoring and behavioral analysis inherent in Zero Trust NTA enable early detection of anomalies. By recognizing deviations from established baselines, organizations can identify potential security threats at their nascent stages, preventing them from escalating. Micro-segmentation, a key component of Zero Trust, significantly reduces the attack surface. By isolating segments and enforcing least privilege access, organizations limit the pathways that potential intruders can exploit, thereby enhancing overall security.[5]

Zero Trust NTA is well-suited for dynamic and evolving network environments, including cloud infrastructures and remote work scenarios. Its adaptive nature allows organizations to maintain a strong security posture despite changes in the network architecture or threat landscape. Zero Trust NTA provides comprehensive visibility into all network activities. By inspecting and logging all traffic, organizations gain a deeper understanding of user behaviors, device interactions, and application usage, empowering them to make informed security decisions. Automated incident response mechanisms in Zero Trust NTA enable organizations to respond swiftly and effectively to potential security incidents. By automating containment measures and predefined responses, the impact of intrusions can be minimized.

3.9 Future Trends in Zero Trust NTA for Intrusion Detection:

The integration of artificial intelligence (AI) and machine learning (ML) is a growing trend in Zero Trust NTA. These technologies enhance the system's ability to detect complex, evolving threats by analyzing patterns and anomalies in large datasets. The future of Zero Trust NTA involves the continued enhancement of User and Entity Behavior Analytics. Analyzing user activities and access patterns will become more sophisticated, enabling the system to identify subtle deviations indicative of potential threats.

Zero Trust principles will be extended to secure IoT devices as the Internet of Things (IoT) grows. To guarantee the security of connected devices, network traffic analysis for the Internet of Things will entail ongoing monitoring and behavior analysis. There will be a greater integration of Zero Trust NTA with cloud-native security strategies due to the ongoing shift towards cloud-native architectures. This will guarantee a unified and consistent security posture throughout.

IV. CONCLUSION

The purpose of an Intrusion Detection System (IDS) is to monitor network or system activities for malicious or unauthorized behavior and to detect and respond to security threats. IDS plays a crucial role in maintaining the security of a computer network by identifying and alerting administrators to potential security incidents. Here are some key purposes of an Intrusion Detection System. IDS is designed to detect various types of threats, including malware, unauthorized access attempts, suspicious network traffic, and other malicious activities. It helps in identifying potential security breaches before they can cause significant damage. IDS offers visibility into network traffic, allowing administrators to identify patterns and trends. This information is useful for understanding the overall security posture of the network and making informed decisions about security policies and measures. IDS logs and records information about detected incidents. [3]

This data is valuable for forensic analysis, helping security professionals investigate and understand the nature of security breaches, the extent of the damage, and the methods used by attackers. IDS provides real-time monitoring of network and system activities, enabling a quick response to security incidents. This is essential for minimizing the impact of security breaches and preventing further compromise. When an IDS detects a potential security incident, it generates alerts or notifications to notify system administrators or security personnel. This information is crucial for a timely and effective response to mitigate the impact of the incident.

V. REFERENCES

[1] Bahareh Abolhasanzadeh. 2015. Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features. In 2015 7th Conference on Information and Knowledge Technology (IKT). 1–5. https://doi.org/10.1109/IKT.2015.7288799

[2] Isra Al-Turaiki and Najwa Altwaijry. 2021. A Convolutional Neural Network for Improved Anomaly-Based Network Intrusion Detection. Big Data 9, 3 (2021), 233–252.
https://doi.org/10.1089/big.2020.0263

[3] Abeer Alalmaie, Priyadarsi Nanda, and Xiangjian He. unpublished. Zero Trust-NIDS: Extended Multi-View Approach for Network Trace Anonymization and Auto-Encoder Convolutional Neural Network for Network Intrusion Detection. TrustCom (unpublished).

[4] Mohammed M. Alani. 2022. Implementation-Oriented Feature Selection in UNSW-NB15 Intrusion Detection Dataset. In Intelligent Systems Design and Applica-tions, Ajith Abraham, Niketa Gandhi, Thomas Hanne, Tzung-Pei Hong, Tatiane Nogueira Rios, and Weiping Ding (Eds.). Springer International Publishing, Cham, 548–558.

[5] Muder Almiani, Alia AbuGhazleh, Amer Al-Rahayfeh, Saleh Atiewi, and Abdul Razaque. 2020. Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory 101 (2020), 102031. https://doi.org/10.1016/j.simpat.2019.102031 Modeling and Simulation of Fog Computing.

[6] Longy O. Anyanwu, Jared Keengwe, and Gladys A. Arome. 2010. Scalable In-trusion Detection with Recurrent Neural Networks. In 2010 Seventh Interna-tional Conference on Information Technology: New Generations. 919–923. https://doi.org/10.1109/ITNG.2010.45

[7] Bo Cao, Chenghai Li, Yafei Song, Yueyi Qin, and Chen Chen. 2022. Network Intrusion Detection Model Based on CNN and GRU. Applied Sciences 12, 9 (2022).https://doi.org/10.3390/app12094184

[8] D. E. Denning. 1987. An Intrusion-Detection Model. Ieee Transactions On Software Engineering 13, 2 (1987), 222–232.