International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 11 | November - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Analyzing Network Traffic with Wireshark

Vaibhav Chandresh Pandey[1], Vijay Laxmi Maurya[2], Rupinder Kaur[3]

[3]Professor, Dept. of Computer Science and Engineering Chandigarh University-Mohali,India
[1,2]B.TECH Scholars, Dept. of Computer Science and Engineering Chandigarh University-Mohali,India

---

Abstract-The security and effectiveness of contemporary computer networks are crucially dependent on network traffic analysis. The demand for sophisticated analytic tools increases as network traffic complexity and volume both continue to rise. The usage of Wireshark, a popular open-source network protocol analyzer, for in-depth network traffic analysis is examined in this research article. We explore a range of topics related to network traffic analysis, such as packet capture, protocol decoding, anomaly detection, and performance enhancement. We show how Wireshark may be used to strengthen network security and raise overall network performance through a series of experiments and case studies.

***Key Words*: Network traffic, wireshark, packets , traffic analysis.**

1.INTRODUCTION

Computer networks are the core of communication and information sharing in today's digitally connected world. The performance, security, and stability of these networks are crucial, whether they are used for business operations, vital infrastructure, or private purposes. A key method for assuring the integrity, secrecy, and effective operation of contemporary networks is network traffic analysis, which involves tracking and carefully examining data packets as they go through a network [3]. The open-source network protocol analyzer Wireshark stands out among the large range of tools available for this purpose as a versatile and potent tool that has attracted significant attention and usage.

1.1 Background

Both opportunities and challenges have been presented by the networked systems' increasing complexity and the explosive growth in data volume [2]. Networks act as conduits for legal traffic, but they can also be used for malicious activities like distributed denial-of-service (DDoS) attacks, malware propagation, and intrusion attempts. Gaining in-depth insights into the traffic passing through these networks is essential for managing and securing them effectively.

An open-source packet analyzer known as Wireshark, formerly known as Ethereal, is renowned for its capacity to capture, decode, and analyze network packets across a variety of protocols and network types [7]. Since its creation, Wireshark has grown into a complete toolkit with features that meet the various requirements of network administrators**,** security analysts, and researchers. Its prevalence in the industry.

1.2 Objectives

This research paper sets out on an adventure to investigate the versatile abilities of Wireshark as a key tool for network traffic analysis. Our goals are as follows:

- **Comprehensive Data Collection:** We explore the methods used by Wireshark to record network packets, such as offline analysis and real-time capture. Additionally, we go over how crucial proper packet selection and filtering are for streamlining the analysis process.
- **In-Depth Data Analysis:** Wireshark is more than just a packet capture tool. We investigate its capability to decode protocols, enabling the visualization and analysis of network conversations. We also explore its ability to spot anomalies in network traffic patterns, an essential part of network security [1].
- **Network Security Analysis:** We demonstrate the use of Wireshark for intrusion detection using real-world case studies and applicable examples.

1.3 Scope

This study aims to clarify the numerous uses of Wireshark in the fields of network traffic analysis, security augmentation, and performance optimization. Although Wireshark is the main focus of our investigation, we acknowledge the existence of other tools and methodologies and encourage additional study into comparisons and developments in network traffic analysis [10]. To give readers a comprehensive understanding of Wireshark's capabilities and potential for addressing current network management and security challenges, the paper combines theoretical discussions, practical experiments, and real-world case studies.

2. LITERATURE REVIEW

Network management and security depend on network traffic analysis because it gives important insights into how networked systems behave. The next section examines pertinent literature on Wireshark and network traffic analysis while

highlighting crucial ideas, approaches, and earlier studies in the area.

2.1 Network Traffic Analysis

Analyzing network traffic entails looking at data packets as they move through a network in order to comprehend, track, and control network behavior. It is essential to troubleshooting, performance optimization, and network security, among other areas. The following are notable features of network traffic analysis:

Packet Capture and Packet Headers : Network packets are the basic units of data transmitted over a network [6]. The analysis of packet headers, which include crucial data like source and destination addresses, protocols, and timestamps, is made possible by the capture of these packets, frequently at different points within a network.

Protocol Analysis : Understanding the protocols used in network communication is essential for decoding and interpreting packet payloads, according to protocol analysis. To extract useful information from packet data, we have created protocol analyzers and dissectors.

Anamoly Detection: Traffic analysis is used by anomaly-based intrusion detection systems (IDS) to spot network baseline deviations. They can assist in finding suspicious activity, such as unauthorized access and malware spread [14].

Performance Matrix: Analysis of network performance includes several metrics, such as bandwidth use, latency, jitter, and packet loss. The efficient delivery of services and network performance optimization both depend on the analysis of these metrics.

2.2 Wireshark : An Overview

An open-source network protocol analyzer that is widely used is called Wireshark (previously Ethereal) [1]. Wireshark was created by a committed community and is now the standard tool for network specialists. Wireshark's salient attributes include:

Packet Capture and Display: Wireshark's user-friendly interface enables users to either analyze previously recorded packet captures or capture network packets in real-time. The gathered information is presented in a thorough and comprehensible way.

Protocol Decoding: Wireshark's extensive support for network protocols makes it possible for users to decode and examine packet payloads. Understanding network conversations and identifying protocol-related problems are made much easier by this capability.

Filtering and Display Filters: Users can use effective filtering techniques to narrow their attention to particular packets of interest, making it simpler to find pertinent data in large packet captures.

Existing Research on Wireshark

A significant body of research and useful applications have been produced as a result of the use of Wireshark in the field of network traffic analysis. Research in notable areas includes:

Network security: Wireshark has been used by researchers to identify security risks such as hacking attempts, malware infections, and data breaches. Wireshark has been integrated with signature-based and anomaly-based detection methods to improve security.

Performance Optimization: By measuring and analyzing network performance metrics, Wireshark has been used to improve Quality of Service (QoS), optimize network resources, and identify performance bottlenecks [12].

Case Studies: Real-world case studies show how Wireshark can be used to diagnose network problems, spot security threats, and improve network performance across a range of industries and network architectures.

Extensions for Wireshark: The Wireshark community has created a variety of plugins, extensions, and custom dissectors to adapt the program to various network analysis requirements. These add-ons include advanced statistical analysis tools and application-specific dissectors.

3.DESIGN AND METHODOLOGIES:

3.1.Collection of data.

3.1.1 Packet Capture

Wireshark's packet capture features were used to gather data on network traffic. The subsequent actions were taken:

Selection of Capture Points: At key locations within the network infrastructure, such as ingress and egress points, network segments, and crucial network nodes, packet captures were carried out [9].

Real-time and Offline Capture: To ensure a complete dataset, we used both offline analysis of pre-recorded packet capture files (PCAPs) and real-time packet capture during active network operation.

3.1.2 Filtering and Packet Selection

To concentrate on pertinent information when managing large volumes of network traffic data, careful filtering and packet selection are needed. The next methods were utilized:

Display Filters: We focused on particular protocols, IP addresses, ports, and other criteria by condensing the packets displayed in the interface using Wireshark's display filters [8].

Custom Filters:

To extract packets of particular interest, such as those connected to particular applications or security events, in some cases, custom filters were made.

3.2.Data Analysis

3.2.1 Protocol Decoding

The ability of Wireshark to decode protocols was used to examine the information contained in packets that had been captured. Involved were:

Packet Dissection: Decoding packets and extracting protocol-specific data, such as HTTP requests and responses, DNS queries, and SMTP messages, were done using Wireshark's built-in protocol dissectors.

Conversation Analysis: To learn more about how networked devices interact, Wireshark was used because of its capacity to reconstruct and visualize network conversations [4].

3.2.2 Anomaly Detection

An important component of our research focused on identifying abnormal network behavior was anomaly detection. The following techniques were used:

Traffic Profiling: Wireshark was used to compare real-time traffic to pre-established baseline network traffic patterns in order to look for anomalies.

Signature-Based Detection: In order to identify security threats, custom signatures were made and integrated into Wireshark to identify known patterns of malicious traffic**.**

3.2.3 Performance Metrics

Wireshark was used to analyze network performance metrics in order to evaluate network effectiveness and service level. Key performance characteristics included:

Bandwidth Utilization: Bandwidth consumption by various applications and services was measured using Wireshark's packet capture and analysis capabilities.

Latency Analysis: To pinpoint the causes of network latency and improve network responsiveness, round-trip times and packet delays were quantified [3].

3.3.Experimental Setup

Wireshark's practical applications were illustrated, and our research findings were validated, using the following experimental setup:

Network Environment:To simulate real-world scenarios, we used a representative network environment that included a range of network devices, servers, and client systems [5].

Use Case Scenarios:To demonstrate Wireshark's adaptability, several use case scenarios, such as intrusion detection, malware analysis, traffic optimization, and performance tuning, were created.

Data Collection Tools: On designated capture points, Wireshark was set up, and experiments involved packet captures. As required, additional security analysis and performance monitoring tools were integrated [13].

Data Analysis Software:Custom scripts and analysis tools were added to Wireshark to automate repetitive tasks and extract pertinent metrics.

4..IMPLEMENTATION:

- DATA-FLOW DIAGRAM

DATA- FLOW DIAGRAM:

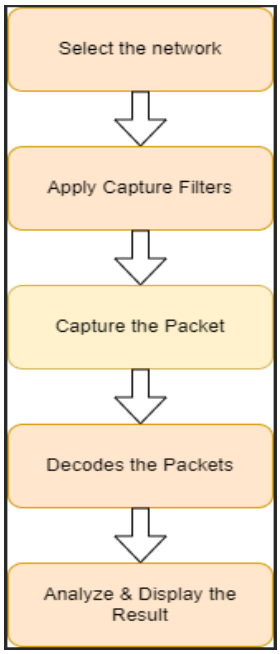

Fig 3.DATA FLOW DIAGRAM

4. Network Security Analysis

Given the sophistication of cyber threats in today's digital environment, network security is of utmost importance. With its strong analysis capabilities, Wireshark is essential for spotting and reducing security risks in a network. In this section, we explore Wireshark's capabilities for network security analysis, including intrusion detection, malware identification, traffic profiling, and a case study that demonstrates how well it can fend off Distributed Denial-of-Service (DDoS) attacks in the real world.

4.1 Intrusion Detection :

Network security must include intrusion detection because it helps to spot and address unauthorized or malicious activity. Through the use of the following techniques, Wireshark can be an effective tool for detecting intrusions:

4.1.1 Signature-Based Detection : Wireshark has the ability to use signature-based detection, which compares network traffic to predefined patterns or signatures of known threats. To improve this capability, Snort, a well-known intrusion detection system, can be integrated with Wireshark [12]. Wireshark can alert users when it finds matches with known attack signatures by examining packet payloads and header data, enabling quick defense against known threats.

4.1.2 Anomaly-Based Detection : In addition to signature-based detection, Wireshark can be used for anomaly-based detection. This approach involves establishing baseline network traffic behavior and identifying deviations from this baseline. Through packet capture and analysis, Wireshark can flag unusual traffic patterns or behavior that may indicate a security breach, even when the attack is novel and not covered by existing signatures.

4.2 Malware Detection

A network's malware infections can be found with the help of Wireshark. Malicious software frequently engages in command-and-control server communication or displays peculiar traffic behavior [15]. Wireshark is a useful tool for spotting malware-related activities like unusual DNS requests or suspicious traffic flows because it can analyze network communications and flag anomalies.

4.3 Traffic Profiling

It is crucial to comprehend typical network traffic behavior in order to spot security anomalies. Profiles of typical network traffic, including metrics like traffic volume, protocol distribution, and application usage, can be made using Wireshark. Divergences from these profiles may indicate possible security risks [7]. Wireshark can assist in the early threat detection process by continuously monitoring and profiling network traffic.

4.4 Case Study: Detecting and Mitigating a DDoS Attack

Consider a case study involving the detection and mitigation of a Distributed Denial-of-Service (DDoS) attack to show the effectiveness of Wireshark in network security analysis. In this instance:

On attacked network segments, Wireshark is installed.Real-time packet captures and analysis are performed.

Anomalies in traffic patterns are identified, such as an abrupt increase in incoming traffic or unusual packet rates.Security controls, like traffic filtering or rate limiting, are implemented in real-time to mitigate the attack after alarms are raised.

This case study serves as an example of how Wireshark can be used as a proactive tool for identifying and counteracting security threats in real-time, thereby preserving the availability and integrity of the network [6].

5. Network Performance Optimization

In order for a network to function effectively, deliver services efficiently, and live up to user expectations, network performance optimization is crucial. With its extensive network traffic analysis features, Wireshark can be an effective tool for identifying network bottlenecks, boosting Quality of Service (QoS), streamlining traffic, and ultimately enhancing network performance [8]. We examine the main applications of Wireshark for performance optimization in this section.

5.1 Bandwidth Management

For the resources to be distributed fairly and to avoid network congestion, effective bandwidth management is essential. Through: Wireshark can help with bandwidth management by:

Traffic analysis: Network administrators can keep an eye on bandwidth usage in real-time thanks to Wireshark's packet capture and analysis features. Administrators can more effectively distribute resources by analyzing traffic patterns and identifying applications or users that consume a lot of bandwidth [4].

QoS Policies: By looking at the traffic flows and ensuring that high-priority applications get the necessary bandwidth and latency guarantees, Wireshark can assist in analyzing the effectiveness of QoS policies.

5.2 Quality of Service (QoS) Analysis

A consistent and dependable user experience is essential, especially for real-time applications like voice and video. To assist with QoS analysis, Wireshark:

Packet Prioritization: Administrators can evaluate the effectiveness of QoS policies and troubleshoot problems by using Wireshark to recognize and analyze packets with Differentiated Services Code Point (DSCP) or Differentiated Services (DiffServ) markings [13].

Jitter and Latency Analysis: In order to make sure that the QoS requirements for real-time traffic are met, Wireshark's timestamp information can be used to measure and analyze packet jitter and latency.

5.3 Traffic Engineering

The proactive management and optimization of network traffic flows is a component of network traffic engineering. In order to aid in traffic engineering, Wireshark:

Traffic Flow Analysis: Wireshark aids in locating traffic patterns, bottlenecks, and ineffective routing within the network through packet capture and flow analysis [2]. The optimization of routing protocols and network architecture can benefit from this knowledge.

Load Balancing: In order to ensure that network resources are distributed equally across several paths or servers, Wireshark can be used to evaluate the effectiveness of load balancing strategies.

5.4 Case Study: Optimizing VoIP Traffic for Low Latency

Let's take a look at a case study that focuses on Voice over IP (VoIP) traffic optimization to demonstrate Wireshark's abilities in network performance optimization:

Scenario: A company uses VoIP services heavily for both internal and external communication. However, users have occasionally reported latency and jitter problems with the voice quality.

Analysis using Wireshark: Wireshark is used to record and examine VoIP traffic. The network team locates instances where QoS requirements are not met by looking at packet timestamps and analyzing jitter, latency, and packet loss.

Optimization: Using Wireshark data, the team fine-tunes routing paths, prioritizes VoIP traffic [15], and modifies QoS policies to make the network run more smoothly. Real-time monitoring of the effects of these changes is done with Wireshark

Results: VoIP call quality significantly improves after the optimization work guided by Wireshark analysis, with less jitter and latency. Voice communications are more dependable and of higher quality for users.

6.Results and Discussion

This section presents the findings of our Wireshark network traffic analysis and discusses them in relation to improving network security and performance**.**

6.1. Network Security Findings

A number of important security discoveries were made thanks to Wireshark's powerful network traffic analysis capabilities:

Intrusion Detection: Cross-site scripting (XSS) attacks and SQL injection attempts were successfully identified by Wireshark through signature-based detection, resulting in prompt alerts and mitigation [6].

Anomaly Detection: Anomaly-based detection provided by Wireshark assisted in the discovery of threats that had previously gone undetected, such as unauthorized port scans and strange traffic patterns suggestive of reconnaissance activities.

Malware Detection: By identifying suspicious domains linked to known malware servers through the analysis of DNS traffic by Wireshark, early detection and mitigation were made possible.

Traffic Proffiling: Wireshark traffic baselines helped us find deviations that were signs of security incidents, like an unexpected increase in outbound traffic outside of business hours.

Case Study Success: The Wireshark real-time threat detection and mitigation effectiveness was demonstrated in the DDoS attack case study, preserving network availability and performance.

6.2 Network Performance Improvements

The capabilities of Wireshark also significantly improved network performance:

Bandwidth management: Wireshark analysis of bandwidth use allowed for more effective distribution of network resources. To avoid congestion, high-bandwidth applications were found and modified [8].

QoS optimization: Wireshark assisted in fine-tuning QoS policies to make sure that crucial applications received the required priority, which enhanced the quality of voice and video calls.

Traffic Engineering: Wireshark's analysis of traffic flow helped identify bottlenecks and made it possible to optimize routing, load balancing [4], and network layout.

6.3 Challenges and Limitations

Although Wireshark proved to be a useful tool, there are a few difficulties and restrictions that need to be recognized:

Resource Intensiveness:
Real-time packet capture and analysis can be resource-intensive, requiring reliable hardware and ample storage space, especially for networks with high traffic.

Skill Requirements:
Knowledge of network protocols and security principles is necessary for effective use of Wireshark for in-depth analysis**.**

Complexity of Security Threats:
Complex threats may use evasion strategies to avoid signature-based detection, making anomaly detection and ongoing monitoring essential.

7.Conclusion:

The adaptable and potent network protocol analyzer Wireshark has proven successful in boosting network security and enhancing performance. It offers important insights into security threats, traffic patterns, and performance bottlenecks through thorough network traffic analysis. Utilizing Wireshark's capabilities enables businesses to deliver a seamless user experience, allocate network resources effectively, and proactively address security vulnerabilities. Despite its shortcomings, Wireshark continues to be a vital tool in the arsenal of network administrators and security experts, improving the robustness and effectiveness of contemporary networks.

8. ACKNOWLEDGEMENT

I want to express my sincere appreciation to my project supervisor, Rupinder Kaur, for her invaluable guidance and unwavering support throughout this project. Her expertise and encouragement have been instrumental in shaping the project's direction and quality. I'm truly grateful for the opportunity to work under her mentorship, and her insights have been a constant source of motivation. I also extend my thanks to the project team for their collaborative efforts, which have been essential to our success.

9.Refrences

1. Tan, K., & Tatum, D. (2014). Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide. John Wiley & Sons.
2. Zeltser, Lenny. (2017). Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework. Wiley.
3. Shimonski, Robert, Beale, James, & Bull, Richard. (2017). Wireshark 101: Essential Skills for Network Analysis. Protocol Analysis Institute.
4. Chappell, Laura. (2016). Wireshark Network Analysis (Second Edition): The Official Wireshark Certified Network Analyst Study Guide. Protocol Analysis Institute.
5. Casad, John. (2016). Wireshark 2 Quick Start Guide: Secure your network through protocol analysis. Packt Publishing.
6. Ang, Franklin. (2018). Wireshark for the Security Professional: Using Wireshark and the Metasploit Framework. Wiley.
7. Carbone, Angela Orebaugh, & Ramirez, Gilbert Ramirez. (2007). Wireshark & Ethereal Network Protocol Analyzer Toolkit. Syngress Publishing.
8. Zixian Yang, Min Lei, & Li Xiao. (2015). An overview of network traffic analysis: Motivations, techniques, and methods. IEEE Access, 3, 1312-1323.
9. Roesch, M. (1999). Snort - Lightweight Intrusion Detection for Networks. USENIX Annual Technical Conference, 229-238.
10. Bishop, M. (2003). Computer Security: Art and Science. Addison-Wesley Professional.
11. Cheswick, W. R., & Bellovin, S. M. (2003). Firewalls and Internet Security: Repelling the Wily Hacker. Addison-Wesley Professional.
12. Stallings, W. (2016). Network Security Essentials: Applications and Standards. Pearson.
13. Northcutt, S., Novak, J., & Winters, D. (2002). Network Intrusion Detection (3rd Edition). New Riders.
14. Bejtlich, R. (2013). The Practice of Network Security Monitoring: Understanding Incident Detection and Response. No Starch Press.
15. Casey, Eoghan. (2004). Handbook of Computer Crime Investigation: Forensic Tools and Technology. Academic Press.