****

**TECHNICAL MANUAL**

Real-time Monitor Software Setup

Linux 2 Web Server

**Prepared By :** *Mark Byrne* **Prepared on :** *28/6/2023*

**Table of Contents**

[Introduction 4](#_Toc139141922)

[Purpose and Scope 5](#_Toc139141923)

[Overview of Wireshark 5](#_Toc139141924)

[Key Features 5](#_Toc139141925)

[Prerequisites 5](#_Toc139141926)

[Document Structure 6](#_Toc139141927)

[Document Revision History 6](#_Toc139141928)

[Installation and Configuration 6](#_Toc139141929)

[User Management 8](#_Toc139141930)

[Data Ingestion 8](#_Toc139141931)

[Search and Analysis 9](#_Toc139141932)

[Data Management 9](#_Toc139141933)

[Security and Access Control 9](#_Toc139141934)

[Maintenance and Upgrades 10](#_Toc139141935)

[Best Practices 10](#_Toc139141936)

[Resources and References 10](#_Toc139141937)

|  |
| --- |
| Introduction |

## Purpose and Scope

The purpose of this document is to provide an introduction to Wireshark, an open-source protocol analyzer used for real time traffic monitoring. It aims to familiarise users with the features, capabilities, and benefits of Wireshark for providing detailed information on network packets to support the analysis of network issues, troubleshooting problems, and performing network security analysis.

## Overview of Wireshark

Wireshark is a powerful open-source network protocol analyzer that allows users to capture, inspect, and analyze network traffic. With support for a wide range of protocols, Wireshark provides detailed information about packet structure, headers, and payload, making it an essential tool for troubleshooting network issues, analyzing performance, and ensuring network security. Its intuitive interface, traffic statistics, filtering capabilities, and customization options make it a versatile tool for both wired and wireless networks, assisting network administrators, security professionals, and developers in gaining deep insights into network communications.

## Key Features

Below is a summary of the key features and capabilities of Wireshark:

Packet Capture: Wireshark captures packets from various network interfaces, such as Ethernet, Wi-Fi, and Bluetooth, allowing users to analyze both wired and wireless network traffic. It supports live packet capturing as well as reading packet capture files from other tools.

Protocol Support: Wireshark supports a vast array of network protocols, including popular ones like TCP/IP, HTTP, DNS, DHCP, FTP, SMTP, and many more. It can dissect and decode these protocols, providing detailed information about the packet structure, headers, and payload.

Packet Analysis: Wireshark provides an intuitive and powerful interface to analyze captured packets. Users can examine individual packets and drill down into protocol-specific details. It offers features like filtering, sorting, and color coding to focus on specific packets or types of traffic.

Traffic Statistics: Wireshark generates statistical summaries of network traffic, providing insights into packet counts, data volumes, packet rates, and other metrics. It can help identify patterns, anomalies, and potential performance issues.

Protocol Troubleshooting: Wireshark's deep inspection capabilities allow users to identify and troubleshoot network issues. By analyzing packet-level details, it helps pinpoint problems like packet loss, latency, misconfigurations, or protocol errors.

Traffic Capture Filters: Wireshark supports flexible capture filters to capture specific types of network traffic, based on criteria like source/destination IP addresses, ports, protocols, or even specific packet content. These filters help focus on relevant traffic and reduce the volume of captured data.

Protocol Decoding Customization: Users can extend Wireshark's protocol decoding capabilities by creating custom dissectors. This allows analyzing proprietary or non-standard protocols, making Wireshark a versatile tool for diverse network environments.

Export and Reporting: Wireshark enables exporting captured packets and analysis results to various file formats, such as PCAP, CSV, or plain text. This facilitates sharing findings with colleagues or integrating Wireshark into other network analysis workflows.

## Prerequisites

* Processor: A reasonably modern processor, such as an Intel Core i3 or equivalent.
* Memory: At least 2 GB of RAM. However, more RAM is beneficial if you plan to capture and analyze large network traffic.
* Storage: Wireshark itself does not require much disk space. A few hundred megabytes should be sufficient. However, if you intend to store captured network traffic, you will need additional disk space based on your requirements.
* Display: A monitor capable of a resolution of 1024x768 or higher is recommended.
* Network Interface: Wireshark relies on network interfaces to capture network traffic. Ensure that you have a compatible network interface installed on your system.
* Administrative Privileges: On some operating systems, administrative privileges (e.g., root access on Linux or elevated permissions on Windows) may be required to install and use Wireshark effectively, as it needs low-level access to network interfaces for packet capture.

## Document Structure

## Document Revision History

Version 1.0 - Mark Byrne - 28/6/2023

|  |
| --- |
| Installation and Configuration |
| Step 1: Update System Packages:  sudo apt update    Step 2: Install Wireshark  sudo apt install wireshark      Step 3: Set Up Capturing Privileges:  By default, Wireshark requires root privileges to capture network traffic. However, it is not recommended to run Wireshark as the root user for security reasons. To allow non-root users to capture packets, you can use the setcap command. Run the following command:  sudo setcap cap\_net\_raw,cap\_net\_admin=eip /usr/bin/dumpcap  Step 4: Launch Wireshark:  You can launch Wireshark by opening a terminal and running the following command:  sudo wireshark |

|  |
| --- |
| User Management |
| Wireshark, as a network protocol analyzer, primarily focuses on capturing and analyzing network traffic rather than providing built-in user management features. Therefore, it does not have extensive user management capabilities like user roles, permissions, or user account creation and management.  Wireshark does not provide its own authentication mechanisms. Instead, it relies on the authentication mechanisms provided by the underlying operating system or other tools.  Wireshark can leverage the underlying operating system's user management and authentication mechanisms. This means that the user management features available to you in Wireshark will depend on the user management capabilities provided by the operating system on which it is installed.  On Windows systems, Wireshark can utilize Windows user accounts and their associated permissions. If a user has administrative privileges on the system, they can typically run Wireshark with elevated permissions to capture network traffic. |

|  |
| --- |
| Data Ingestion |
| Wireshark itself does not directly support data ingestion from various sources like logs, files, network data, APIs, etc. Wireshark is primarily focused on capturing and analyzing network traffic in real-time or from saved capture files.  However, Wreshark can open packet captures from a large number of capture programs. |

|  |
| --- |
| Search and Analysis |
| Wireshark provides several features related to search and analysis, allowing users to search for specific data, analyze captured packets, and build search queries to filter results. Below are some examples  **Display Filter:** Wireshark allows users to create display filters to search for specific packets or types of traffic. Display filters use a syntax similar to pcap filters and can be based on various criteria such as source/destination IP addresses, ports, protocols, packet content, and more. By applying display filters, users can focus on specific packets or types of traffic they are interested in.  **Follow TCP Stream:** Wireshark offers a convenient feature called "Follow TCP Stream" that allows users to analyze the complete conversation between two endpoints in a TCP session. It presents the entire stream of data exchanged between the endpoints, making it easier to analyze application-layer protocols like HTTP, FTP, SMTP, etc.  **Conversations:** The "Conversations" feature in Wireshark provides a summary view of network conversations based on source and destination IP addresses and ports. It helps identify which endpoints are communicating and provides statistics on the number of packets, bytes, and conversation duration. This feature assists in analyzing network traffic at a higher level.  **Statistics and Protocol Hierarchy:** Wireshark provides various statistical summaries and protocol hierarchy views to analyze captured packets. The "Statistics" menu offers several options, such as packet length distribution, endpoint statistics, conversations, protocol hierarchy, and more. These statistics provide insights into packet counts, data volumes, protocol distribution, and other metrics.  Expert Information: Wireshark's "Expert" feature flags potential issues or anomalies in network traffic. It identifies errors, warnings, and informational messages related to protocols, retransmissions, duplicate packets, and other network-related events. The expert information can help identify and troubleshoot network problems.  **Customized Columns:** Wireshark allows users to customize the columns displayed in the packet list view. This feature enables users to choose specific packet attributes or fields they want to analyze or search for, making it easier to focus on relevant data during analysis |

|  |
| --- |
| Data Management |
| Wireshark does not have built-in features specifically related to data management, such as indexing configuration, data lifecycle management, or working with summary indexes and accelerated data models.  Wireshark does support exporting captured packets and analysis results to various file formats, such as PCAP, CSV, or plain text. This allows you to store and manage captured data using external tools and processes tailored to your specific data management requirements. |

|  |
| --- |
| Security and Access Control |
| Wireshark is primarily a network protocol analyzer and packet capture tool, it does not have built-in features specifically focused on security and access control.  Wireshark does not have native user authentication or authorization mechanisms. It does not require user login or manage access control directly. However, organizations can implement authentication and authorization measures at the system or network level to restrict access to the Wireshark application and captured packet data.  Access control and permissions for Wireshark are typically managed at the operating system level. By controlling user access to the system or network interfaces, administrators can limit who can use Wireshark and capture network traffic. |

|  |
| --- |
| Maintenance and Upgrades |
| Wireshark itself does not have built-in features for these maintenance and upgrade tasks, they are essential considerations for ensuring the smooth operation, security, and longevity of the Wireshark installation and the associated data.  **Routine Maintenance Tasks:** Routine maintenance tasks, such as log rotation and data grooming, are typically handled at the operating system or storage level rather than within the Wireshark application itself.  **Applying Updates and Patches:** Wireshark is actively maintained and regularly updated with bug fixes, security patches, and new features. To apply updates and patches to Wireshark, users need to download and install the latest version from the official website or use package management tools provided by their operating system. Keeping Wireshark up-to-date is important to benefit from bug fixes, security enhancements, and performance improvements.  **Backup and Disaster Recovery Procedures:** Backing up Wireshark configuration files, profiles, and capture files is crucial to prevent data loss and ensure continuity. It is recommended to regularly back up important data associated with Wireshark, including custom configuration settings, preferences, and any critical capture files. In the event of a system failure or data loss, having backups in place facilitates recovery and minimizes disruptions. |

|  |
| --- |
| Best Practices |
| Wireshark does not have built-in features specifically related to performance optimization, data onboarding and normalization, security and compliance recommendations, or scaling and capacity planning. |

|  |
| --- |
| Resources and References |
| * Official Documentation: The official documentation for Wireshark can be found on the Wireshark website. It includes user guides, tutorials, and reference manuals: <https://www.wireshark.org/docs/> * Wireshark Wiki: The Wireshark Wiki is a collaborative resource with a wealth of information, including tips, tricks, and troubleshooting guides: <https://wiki.wireshark.org/> * Wireshark Q&A: The Wireshark Q&A site is a question-and-answer platform where users can ask and find answers to Wireshark-related questions: <https://ask.wireshark.org/> * Wireshark Mailing Lists: The Wireshark project maintains various mailing lists for discussions related to development, user support, and announcements. You can find the list of available mailing lists here: <https://www.wireshark.org/lists/> * Wireshark University: Wireshark University offers online training courses on network analysis and using Wireshark effectively. They provide self-paced courses as well as instructor-led training: <https://www.wiresharktraining.com/> * Wireshark Certified Network Analyst (WCNA): The WCNA certification is offered by Wireshark University and validates the knowledge and skills in network analysis using Wireshark. More information can be found here: <https://www.wiresharktraining.com/certification> |