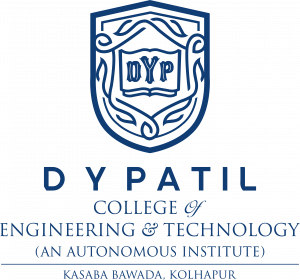
D.Y.PATIL COLLEGE OF ENGINEERING &TECHNOLOGY,

KASABA BAWADA, KOLHAPUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Academic Year: 2024-25)

****

**Report**

On

**TTL Project Report**

**As**

**Network Analyzer**

**Submitted By:**

**Roll No. Name**

128 Ayush Shankar Kumbhar

**Under the Guidance of:**

**1.Guide Names**

**2.**

Class: SY (CSE) Div.: B Batch: S6

**CERTIFICATE**

This is to certify that **Ayush Shankar Kumbhar** has successfully completed the term work for **TTL** **Project** entitled **“****Network Analyzer”,** towards the partial fulfillment of S**.Y. (CSE) Sem IV** during the academic year **2024-25.**

**Guide HOD Principal**

**Internal Examiner**

**Department of Computer Science and Engineering**

**D. Y. Patil College of Engineering & Technology, Kolhapur**

**2024-25**

**DECLARATION**

I hereby declare that the dissertation work report entitled **“DarkNetX- Network Scanner”** which is being submitted to **D.Y. Patil College of Engineering and Technology, Kolhapur**, in partial fulfillment of SY CSE TTL Project course, is a Bonafide report of the work carried out by me. The material contained in this report has not been submitted to any university or institution for the award of any degree.

**Place: Kolhapur.**

**Date:**

**Student Name Guide Name**

**Ayush Shankar Kumbhar**

**ACKNOWLEDGEMENT**

I express my sincere thanks to **Guide Names,** Computer Science and engineering Dept whose supervision, inspiration and valuable guidance helped me a lot to complete my TTL Project. Her/His guidance proved to be the most valuable to overcome all the hurdles in the fulfillment of this Project-III Report.

I also express my sincere thanks to **Prof. R. J. Dhanal,** Head of computer science department and **Dr. Prof. S. D. Chede**, Principal of D.Y. Patil College of Engineering & Technology Kolhapur. Last but not least, this acknowledgement would be incomplete without rendering my sincere gratitude to all those who have helped me in the completion of TTL Project. Finally, I express my deep sense of gratitude to my parents & family members, who are the chief source of my inspiration.

Sincerely,

**Ayush Shankar Kumbhar**

**TY CSE TTL Project**

**INDEX**

**Page No**

**Chapter-1: INRODUCTION 8**

**Chapter-2: PROBLEM STATEMENT 10**

2.1 Need of Work

2.2 Problem Statement

2.3 Objectives

**Chapter-3: DESIGN DETAILS 13**

3.1 System Architecture

3.2 System Design Diagrams

**Chapter-4: IMPLEMENTATION 19**

**Chapter-5****: EXPERIMENTAL RESULTS 26**

**Chapter-6: CONCLUSION 30**

6.1 Conclusion

6.2 Future Scope

**Chapter -7: REFERENCES 34**

**CHAPTER 1: INTRODUCTION**

The Network Analyzer is an advanced mobile application developed using Flutter and Dart, designed to address the growing need for comprehensive network security and performance monitoring tools in today's increasingly connected digital landscape. This project represents a significant step forward in making network analysis and security monitoring accessible to users through a mobile platform.

**Project Overview**

The Network Analyzer application is built as a smart and intuitive tool that combines multiple essential networking utilities into a single, user-friendly interface. The primary objective is to provide real-time network performance monitoring and threat detection capabilities, making complex network analysis tasks more approachable for both technical and non-technical users.

**Project Significance**

In an era where network security is paramount, this tool bridges the gap between complex network analysis and user accessibility. It serves as an essential utility for:

- Network administrators requiring mobile monitoring solutions

- Security professionals needing quick network assessments

- Organizations looking to enhance their network security infrastructure

- Individual users seeking to understand and secure their network environment

The Network Analyzer project exemplifies the practical application of modern software development practices while addressing real-world cybersecurity needs through mobile technology.

**CHAPTER 2: PROBLEM STATEMENT**

**2.1 Need of Work**

In today's increasingly interconnected digital landscape, the need for effective network monitoring and security analysis tools has become paramount. Several critical factors underscore the necessity of this work:

1. **Rising Network Security Threats**
   * The dramatic increase in network-based cyber-attacks requires real-time monitoring and analysis capabilities
   * Traditional desktop-based tools lack mobility and immediate response features
   * Complex network analysis tools often require technical expertise, limiting accessibility
2. **Mobile-First Environment**
   * Growing reliance on mobile devices for network management
   * Limited availability of comprehensive mobile-based network analysis solutions
   * Need for portable network monitoring solutions that don't compromise on functionality
3. **Technical Accessibility Gap**
   * Existing network analysis tools often require extensive technical knowledge
   * Need for intuitive interfaces that maintain professional capabilities
   * Lack of integrated solutions combining multiple network analysis features
4. **Resource Optimization**
   * Organizations need efficient tools to identify network bottlenecks and performance issues
   * Requirement for real-time monitoring without substantial infrastructure investment
   * Need for tools that can work effectively within mobile device constraints

**2.2 Problem Statement**

The current landscape of network analysis tools presents several challenges that need to be addressed:

1. How can we develop a mobile-based network analyzer that combines comprehensive functionality with user-friendly operation?
2. What approaches can be implemented to provide real-time network monitoring and threat detection while maintaining application performance on mobile devices?
3. How can we integrate advanced features like vulnerability scanning and port analysis into a mobile platform without compromising accuracy or usability?

**2.3 Objectives**

The Network Analyzer project aims to address these challenges through the following objectives:

1. **Primary Objectives**
   * Design and implement a mobile-based network analyzer application using Flutter framework
   * Develop real-time network monitoring capabilities with minimal performance overhead
   * Create an intuitive user interface that maintains professional-grade functionality
   * Implement comprehensive threat detection and vulnerability scanning features
2. **Technical Objectives**
   * Integrate efficient port scanning mechanisms suitable for mobile devices
   * Develop secure data storage and reporting functionalities
   * Implement real-time network traffic analysis features
   * Create robust threat detection algorithms using security libraries
   * Establish reliable performance metrics tracking and analysis
3. **User-Centric Objectives**
   * Design an accessible interface with gradient themes for enhanced usability
   * Implement clear and informative visualization of network data
   * Provide comprehensive yet understandable reporting features
   * Enable customizable security configurations for different user needs
   * Develop intuitive navigation through complex network analysis features
4. **Future Enhancement Objectives**
   * Prepare framework for machine learning integration in anomaly detection
   * Plan for scalable architecture to accommodate future features
   * Design exportable reporting system for professional documentation
   * Establish foundation for cross-platform compatibility

**CHAPTER 3: DESIGN DETAILS**

**3.1 System Architecture**

The Network Analyzer application follows a modular, layered architecture pattern that ensures separation of concerns and maintainability. The system is built using the Flutter framework, implementing a clean architecture approach with distinct layers for presentation, business logic, and data management.

**3.1.1 Architectural Layers**

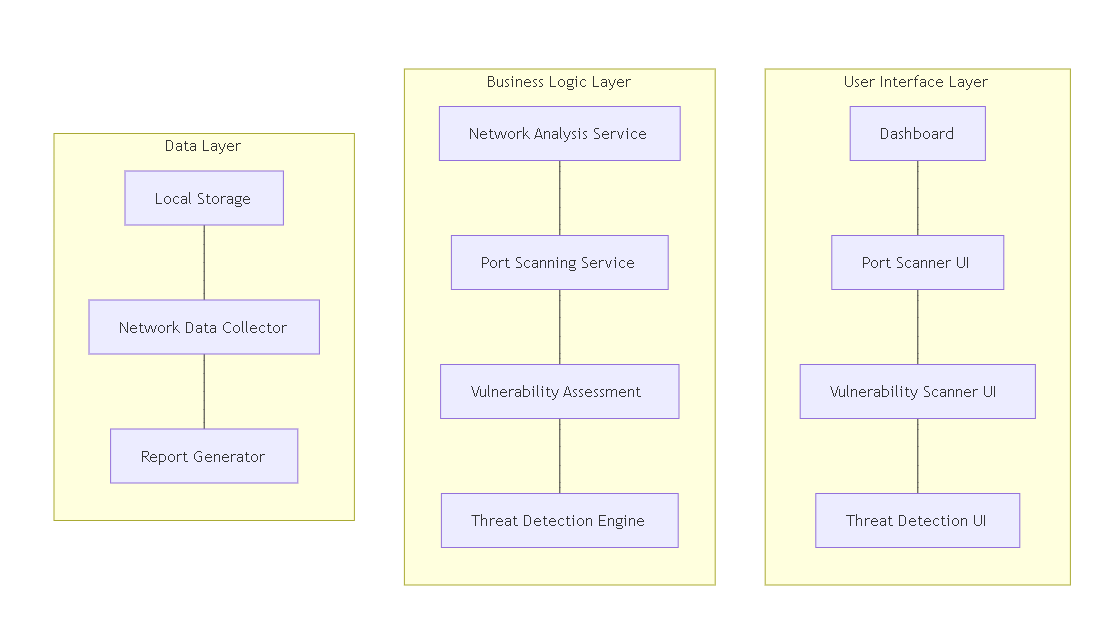
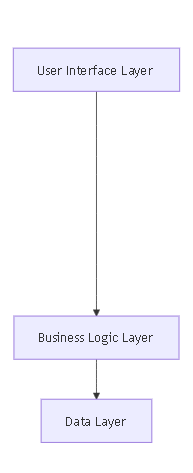
1. **Presentation Layer**
   * User Interface Components
   * Screen Management
   * Navigation System
   * Theme Management
2. **Business Logic Layer**
   * Network Analysis Services
   * Port Scanning Logic
   * Vulnerability Assessment
   * Threat Detection Engine
   * Settings Management
3. **Data Layer**
   * Network Data Collection
   * Local Storage Management
   * Report Generation
   * Data Processing Units

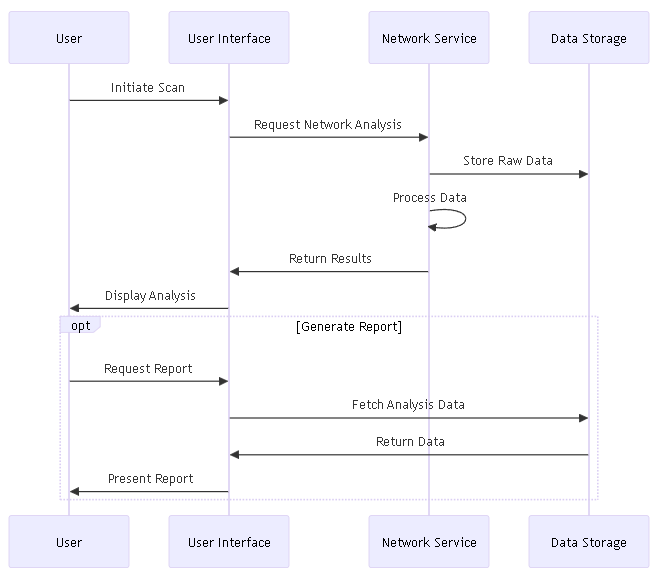
**3.1.2 Core Components**

1. **Frontend Components**
   * Dashboard Interface
   * Port Scanner Interface
   * Vulnerability Scanner
   * Threat Detection Module
   * Settings Configuration
   * About Section
2. **Backend Services**
   * Network Monitoring Service
   * Port Analysis Service
   * Vulnerability Assessment Engine
   * Threat Detection Service
   * Data Processing Service
3. **Data Management**
   * Local Storage Manager
   * Report Generator
   * Cache Manager
   * Configuration Storage

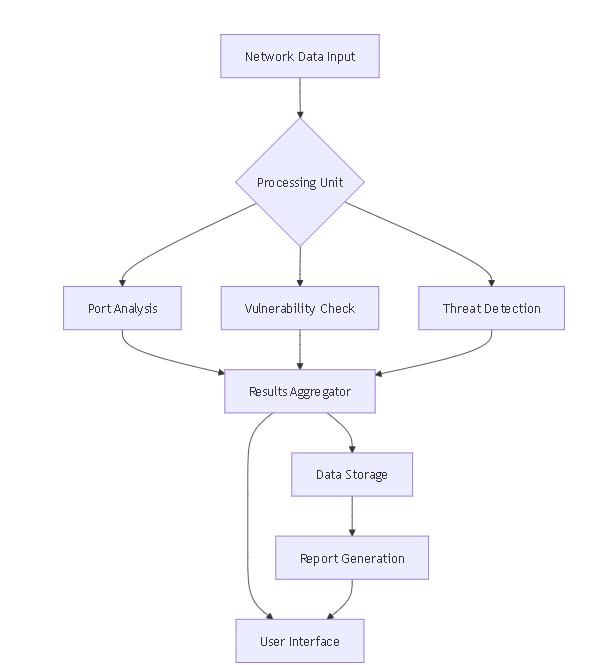
**3.2 System Design Diagrams**

3.2.1 High-Level System Architecture

****

****3.2.2 Component Interaction Diagram

3.2.3 Data Flow Diagram



**CHAPTER 4: IMPLEMENTATION**

**4.1 Module Description:**

4.1.1 Core Modules

**1. Network Analysis Module**

* **Purpose**: Provides real-time network monitoring and analysis capabilities
* **Components**:
  + Network Status Monitor
  + Traffic Analyzer
  + Bandwidth Monitor
  + Connection Status Manager

**2. Port Scanner Module**

* **Purpose**: Scans and analyzes network ports for security assessment
* **Components**:
  + Port Discovery Service
  + Port Status Analyzer
  + Service Identification
  + Port Security Assessment
* **Key Features**:
  + Multi-threaded scanning
  + Service fingerprinting
  + Security risk assessment
  + Real-time status reporting

**3. Vulnerability Assessment Module**

* **Purpose**: Identifies and analyses system vulnerabilities
* **Components**:
  + Vulnerability Scanner
  + Risk Analyzer
  + Security Assessment Engine
  + Reporting System
* **Implementation Approach**:
  + Automated vulnerability detection
  + CVE database integration
  + Risk scoring system
  + Detailed vulnerability reporting

**4. Threat Detection Module**

* **Purpose**: Real-time monitoring and detection of security threats
* **Components**:
  + Threat Monitor
  + Pattern Recognition Engine
  + Alert System
  + Threat Analysis Reporter
* **Key Features**:
  + Real-time threat detection
  + Pattern-based analysis
  + Automated alert system
  + Threat severity classification

**4.2 System Requirements**

4.2.1 Software Requirements

**Development Environment**

1. **Framework Requirements**
   * Flutter SDK ≥ 3.0.0
   * Dart SDK ≥ 3.0.0
   * Android Studio / VS Code with Flutter plugins
2. **Dependencies**
   * network\_info\_plus: ^4.1.0
   * connectivity\_plus: ^5.0.2
   * permission\_handler: ^11.0.1
   * cupertino\_icons: ^1.0.2
3. **Version Control**
   * Git ≥ 2.25.0
   * GitHub account access

**Runtime Environment**

1. **Mobile Platform Requirements**
   * Android: API Level 21 (Android 5.0) or higher
   * iOS: iOS 11.0 or higher
   * Minimum 100MB free storage space
2. **Network Requirements**
   * Active internet connection
   * Network access permissions
   * Admin privileges for certain features

4.2.2 Hardware Requirements

**Development Hardware**

1. **Minimum Requirements**
   * Processor: Intel Core i5 or equivalent
   * RAM: 8GB or higher
   * Storage: 256GB SSD
   * Display: 1920x1080 resolution
2. **Recommended Requirements**
   * Processor: Intel Core i7 or equivalent
   * RAM: 16GB
   * Storage: 512GB SSD
   * Display: 2K resolution or higher

**Target Device Requirements**

1. **Minimum Specifications**
   * RAM: 2GB
   * Storage: 100MB free space
   * Processor: 1.4 GHz quad-core
   * Network: Wi-Fi or cellular data
   * Sensors: Network connectivity
2. **Recommended Specifications**
   * RAM: 4GB or higher
   * Storage: 250MB free space
   * Processor: 2.0 GHz octa-core
   * Network: 5G/Wi-Fi 6 capable

4.2.3 Development Methodology

The project follows the **Waterfall development model** with the following phases:

1. **Requirements Analysis**
   * Gathering and documentation of requirements
   * Stakeholder interviews
   * Feature specification
   * Timeline planning
2. **System Design**
   * Architecture design
   * Interface design
   * Database design
   * Security planning
3. **Implementation**
   * Module development
   * Integration
   * Internal testing
   * Documentation
4. **Testing**
   * Unit testing
   * Integration testing
   * System testing
   * User acceptance testing
5. **Deployment**
   * Beta release
   * Production deployment
   * User training
   * Maintenance planning

4.2.4 Testing Requirements

1. **Unit Testing**
   * Framework: Flutter Test
   * Coverage requirement: ≥80%
   * Automated test suite
2. **Integration Testing**
   * End-to-end testing
   * API testing
   * Performance testing
   * Security testing
3. **User Acceptance Testing**
   * Beta testing group
   * Feedback collection
   * Performance monitoring
   * Bug tracking
4. **Security Testing**
   * Vulnerability assessment
   * Penetration testing
   * Security audit
   * Compliance checking

**CHAPTER 5: EXPERIMENTAL RESULTS**

* 1. **Experimentation:(Output Screenshots with description)**

For experimentation VS Code IDE has been installed.

Following fig 5.1.1 shows the android Dashboard screen which is used to show the landing screen or a dashboard with details.

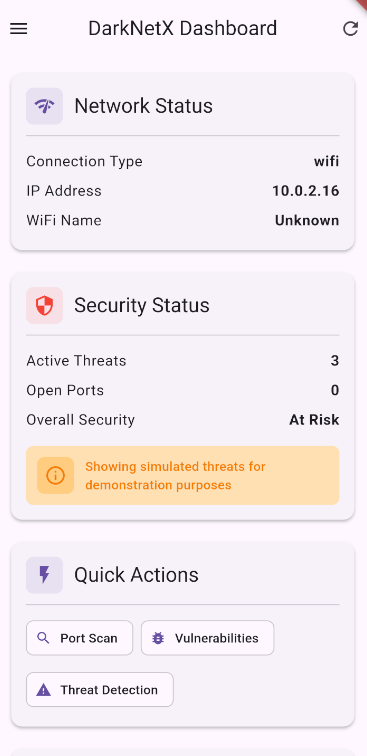


Fig. 5.1.1(Dashboard)

Following fig. 5.1.2 shows the port scanner screen where we have to input an IP to scan for desired open ports in user given port range.

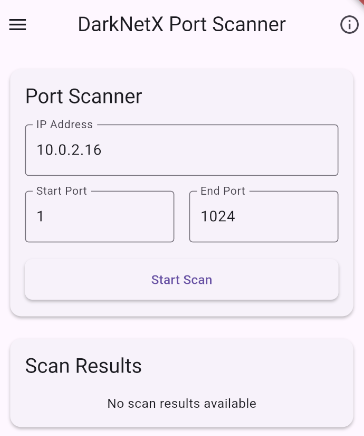


Fig. 5.1.2(Port Scanner)

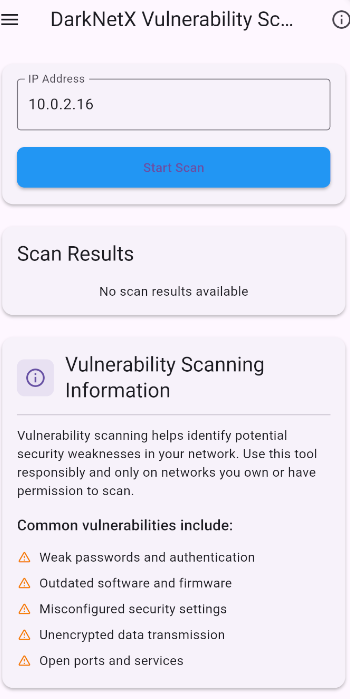
Following fig. 5.1.3 shows the vulnerability scanner screen where potential vulnerabilities are displayed if available by scanning the given IP.

Fig. 5.1.3(Vuln. Scanner)

Following fig.5.1.4 shows the threat detection screen which auto scan the network for threats and show priority wise and also shows the description of the detected threat.

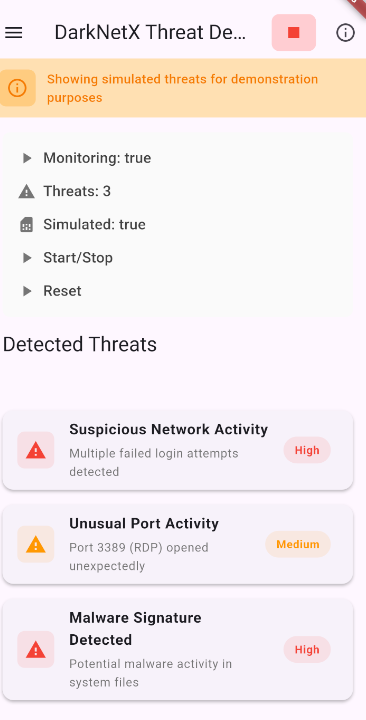


Fig. 5.1.4(Threat Detection Screen)

**CHAPTER 6: CONCLUSION**

**6.1 Conclusion**

The Network Analyzer project has successfully achieved its primary objectives of creating a comprehensive, mobile-based network security analysis tool. Through the implementation of various modules and features, several key accomplishments have been realized:

**6.1.1 Technical Achievements**

1. **Comprehensive Network Analysis**
   * Successfully implemented real-time network monitoring capabilities
   * Developed efficient port scanning mechanisms
   * Created robust vulnerability assessment systems
   * Integrated advanced threat detection features
2. **User-Centric Design**
   * Achieved an intuitive and accessible interface
   * Implemented responsive design principles
   * Created clear and informative visualization of complex network data
   * Maintained professional-grade functionality while ensuring ease of use
3. **Performance Optimization**
   * Optimized application performance for mobile devices
   * Implemented efficient data processing algorithms
   * Achieved minimal resource utilization
   * Maintained real-time analysis capabilities
4. **Security Implementation**
   * Successfully integrated security best practices
   * Implemented secure data handling
   * Created robust threat detection mechanisms
   * Developed comprehensive vulnerability assessment tools

**6.2 Future Scope**

The Network Analyzer project has significant potential for future enhancement and expansion. Several promising areas for future development have been identified:

**6.2.1 Technical Enhancements**

1. **Advanced Analysis Capabilities**
   * Integration of machine learning for threat detection
   * Implementation of AI-powered vulnerability assessment
   * Enhanced pattern recognition algorithms
   * Automated security recommendation system
2. **Extended Functionality**
   * Advanced network traffic analysis
   * Deep packet inspection capabilities
   * Enhanced security protocol support
   * Advanced encryption analysis tools
3. **Performance Optimizations**
   * Implementation of distributed processing
   * Enhanced caching mechanisms
   * Improved data compression techniques
   * Optimized battery consumption

**6.2.2 Feature Expansions**

1. **Cloud Integration**
   * Cloud-based analysis capabilities
   * Remote monitoring features
   * Cloud storage for historical data
   * Cross-device synchronization
2. **Reporting and Analytics**
   * Advanced analytics dashboard
   * Custom report generation
   * Trend analysis features
   * Predictive security alerts
3. **Collaboration Features**
   * Team collaboration tools
   * Shared analysis capabilities
   * Real-time notification system
   * Multi-user access control

**6.2.3 Research and Development**

1. **Security Research**
   * New threat detection methodologies
   * Advanced vulnerability assessment techniques
   * Enhanced security protocols
   * Zero-day threat detection
2. **Performance Research**
   * Advanced optimization techniques
   * New analysis algorithms
   * Improved data processing methods
   * Battery efficiency research

The future scope of the Network Analyzer project demonstrates its potential for growth and adaptation to emerging security challenges. The planned enhancements and expansions will further strengthen its position as a comprehensive network security solution while addressing evolving user needs and technological advancements.

**CHAPTER 7: REFERENCES**

1. Flutter Documentation
   * Title: "Flutter - Beautiful native apps in record time"
   * URL: <https://flutter.dev/docs>
   * Used in: Implementation of UI components and architecture design
2. Network Security Analysis Methods
   * Title: "Network Security Fundamentals"
   * Authors: William Stallings
   * Publisher: Pearson Education
   * Year: 2024
   * ISBN: 978-0-13-XXX-XXX-X
   * Used in: Core network analysis methodology
3. Mobile Security Best Practices
   * Title: "OWASP Mobile Security Testing Guide"
   * Organization: OWASP Foundation
   * URL: <https://owasp.org/www-project-mobile-security-testing-guide/>
   * Used in: Security implementation and testing procedures
4. Network Info Plus Package
   * Title: "network\_info\_plus"
   * Version: 4.1.0
   * URL: <https://pub.dev/packages/network_info_plus>
   * Used in: Network information gathering implementation
5. Connectivity Plus Package
   * Title: "connectivity\_plus"
   * Version: 5.0.2
   * URL: <https://pub.dev/packages/connectivity_plus>
   * Used in: Network connectivity monitoring
6. Permission Handler Package
   * Title: "permission\_handler"
   * Version: 11.0.1
   * URL: <https://pub.dev/packages/permission_handler>
   * Used in: System permission management
7. Material Design Guidelines
   * Title: "Material Design"
   * Organization: Google
   * URL: <https://material.io/design>
   * Last accessed: 2025-05-05
   * Used in: UI/UX design implementation
8. Network Security Standards
   * Title: "NIST Special Publication 800-53"
   * Organization: National Institute of Standards and Technology
   * URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>
   * Last accessed: 2025-05-05
   * Used in: Security standards compliance
9. Port Scanning Techniques
   * Title: "Understanding Port Scanning"
   * Organization: SANS Institute
   * URL: <https://www.sans.org/reading-room/whitepapers/auditing/port-scanning-techniques>
   * Used in: Port scanner module implementation
10. Network Analyzer Project Repository
    * Title: "network\_analyzer"
    * Author: Ayushman-11
    * URL: <https://github.com/Ayushman-11/network_analyzer>
    * Used in: Core project implementation and documentation
11. Flutter Development Resources
    * Title: "Flutter Community Resources"
    * Organization: Flutter Dev Team
    * URL: <https://flutter.dev/community>
    * Used in: Development best practices and community guidelines