IBM Innovation Center for Education

PHEMESOFT Software That Matters

TRUSTED DELIVERY PARTNER for IBM ICE

YENEPOYA
(DEEMED TO BE UNIVERSITY)

THE YENEPOYA INSTITUTE OF ARTS SCIENCE
COMMERCE AND MANAGEMENT
(a constituent unit of Yenepoya Deemed to be University)

INTRUSION DETECTION SYSTEM (IDS)

PROJECT SYNOPSIS

MASTER OF COMPUTER SCIENCE AND APPLICATIONS

SUBMITTED BY :

Ashwath A S	24MCA203
B Mohammad Nizamuddin	24MCA204
Mahammad Shammas	24MCA214
Sheik Mahammad Shahid V	24MCA219

GUIDED BY:

MR. SHASHANK

TITLE PAGE

1. Name of the student: Ashwath A S

2. Class Roll No. 24MCA203

3. Campus ID: 38401

4. Present official Address: YIASCM Blamatta, Mangalore 575002

5. Email: 38401@yenepoya.edu.in

6. Phone No. +91 9606635857

7. Branch: Computer Science

8. Batch: 2024-2026

9. Proposed Topic: Intrusion Detection System (IDS)

TABLE OF CONTENTS

1.1 Introduction

The **Automated Intrusion Detection System (IDS)** is designed to monitor and analyze network traffic for potential security threats. Using packet sniffing techniques and firewall integration, the system detects suspicious activities such as DDoS attacks, port scans, unauthorized access attempts**,** and malware propagation in real-time.

By leveraging advanced detection mechanisms, including signature-based and behavioural anomaly detection**,** the IDS ensures strong network security and mitigates threats before they cause damage.

1.2 Key Features

- ✓ **Real-time Traffic Monitoring** – Continuously analyzes incoming network packets. **Intrusion Detection & Alerts** – Detects high-volume requests, port scans, and unauthorized access attempts.
- ✓ **Firewall Integration** – Automatically blocks malicious IPs using Windows Firewall (netsh) or Linux (iptables**) .**
- ✓ **Custom Detection Rules** – Implements specific rules for DDoS, brute-force attacks, and suspicious activity**.**
- ✓ **Logging & Reporting** – Stores detected threats in structured logs for further analysis.
- ✓ **Machine Learning Integration** – Enhances detection accuracy by learning attack patterns over time.
- ✓ **User-Friendly GUI** – Designed with Tkinter for managing intrusion events interactively.

1.3 Technology Stack

Frontend:

- ✓ **Tkinter**: Allows users to view threats, manage logs, and configure detection settings.

Backend:

- ✓ **Scapy** – Handles packet sniffing and network analysis.
- ✓ **Windows Firewall (`netsh`) / Linux (`iptables`)** – Automates intruder blocking.
- ✓ **Nmap** – Assists in network scanning and threat identification**.**
- ✓ **SQLMap** – Detects and mitigates SQL injection vulnerabilities**.**
- ✓ **CVE Integration** – Fetches real-time vulnerability data to strengthen security measures.

1.4 Specialized Field: Cybersecurity and Ethical Hacking

This project falls under the domain of cybersecurity and ethical hacking, ensuring networks remain secure from external attacks, internal threats, and automated exploits. The IDS helps organizations mitigate security risks proactively, preventing unauthorized access and data breaches.

2.1 Methodology

The development of the Automated Intrusion Detection System (IDS) follows a structured and iterative approach to ensure effective threat detection and security enforcement.

2.2 Requirement Analysis & Tool Selection

- ✓ Define the functionalities, including packet sniffing, logging, and automated blocking.
- ✓ Research and integrate Scapy, Nmap, SQLMap, and CVE databases.
- ✓ Ensure compatibility across Windows and Linux systems.

2.3 System Architecture and Design

- ✓ Design a layered detection system, combining rule-based and anomaly-based detection.
- ✓ Define interaction between the front-end (GUI) and backend threat analysis tools.

2.4 Frontend Development (Tkinter)

- ✓ Develop an interactive dashboard displaying alerts and logs.
- ✓ Enable users to configure detection thresholds and view reports.

2.5 Backend Integration

- ✓ Implement network monitoring with Scapy.
- ✓ Integrate firewall automation to block suspicious IPs
- ✓ Develop CVE-based risk analysis to match detected threats with known vulnerabilities.

2.6 Final Testing and Documentation

- ✓ Validate detection accuracy across different network environments
- ✓ Ensure performance efficiency to avoid unnecessary system slowdowns
- ✓ Generate detailed security audit reports for analysis.

3.1 Facilities required for proposed work

The developing of this IDS project requires all of these software.

3.2 Development Environment

- ✓ **Python 3.12**: The primary programming language for building the front-end (Tkinter) and integrating the backend tools.
- ✓ **Tkinter**: Built-in Python library for developing the graphical user interface (GUI).
- ✓ **IDE VSCode**: To write, test, and debug Python code efficiently.

3.3 Detection & Mitigation Tools

- ✓ **Scapy**: It enables packet crafting, sniffing, and analysis in real-time, Used to capture incoming packets, extract source IPs, and monitor suspicious activity
- ✓ **Windows Firewall (netsh) / Linux iptables – Automated IP Blocking:** netsh advfirewall (Windows) and iptables (Linux) automatically block flagged malicious IPs detected by IDS.Prevents DoS attacks, brute force attempts, and unauthorized scans.
- ✓ **Nmap:** Open-source tool for mapping networks and detecting vulnerabilities.
- ✓ **SQLMap**: A tool to test for and exploit SQL injection vulnerabilities in web applications.

3.4 Testing and Deployment

- ✓ **VirtualBox:** for setting up virtual machines that simulate target environments for penetration testing.
- ✓ **Operating Systems**:
  - ✓ **Kali Linux**: Preferred for running security tools like Nmap, OpenVAS, and Metasploit.
  - ✓ **Windows**: For testing the compatibility of the tool across platforms and development of front-end.

3.5 Reporting Tools

- ✓ **PDF and HTML Libraries**: Python libraries like ReportLab and WeasyPrint for generating and exporting detailed reports in PDF and HTML formats.